DASD MANAGER PLUS for DB2®
User Guide

Supporting

Version 11.1 of DASD MANAGER PLUS for DB2
Version 10.2 of Database Performance for DB2

June 2013
Contacting BMC Software

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

**United States and Canada**

**Address**

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

**Telephone**  
1 713 918 8800  or  
1 800 841 2031

**Fax**  
1 713 918 8000

**Outside United States and Canada**

**Telephone**  
+01 713 918 8800

**Fax**  
+01 713 918 8000

© Copyright 2004-2013 BMC Software, Inc.

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

DB2, IBM, MVS, OS/390, RACE, System z, z/OS, and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

**Restricted rights legend**

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 UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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

Support website

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

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

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 1 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:<yourSupportContractID>, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC

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

- Product information
  - Product name
  - Product version (release number)
  - License number and password (trial or permanent)
- Operating system and environment information
  - Machine type
  - Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - System hardware configuration
  - Serial numbers
  - Related software (database, application, and communication) including type, version, and service pack or maintenance level
- Sequence of events leading to the issue
- Commands and options that you used
- Messages received (and the time and date that you received them)
  - Product error messages
  - Messages from the operating system, such as file system full
  - Messages from related software
License key and password information

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

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

About this book

Related publications ................................................................. 27
Conventions ........................................................................... 28
Summary of changes ................................................................. 28
  Version 11.1.00  June 2013 ...................................................... 28
  Version 10.1.00  April 2011 ...................................................... 37
  Version 9.3.00  December 2009 ............................................... 41

Chapter 1  Introduction .......................................................... 49

Overview .............................................................................. 49
  Solution integration .............................................................. 50
  Product benefits ................................................................ 52
  Product features ................................................................ 53
System setup .......................................................................... 60
  DB2 Support .................................................................... 60
  Features of DB2 that DASD MANAGER PLUS does not currently support . . . . 61
  System requirements ........................................................... 61
  Software requirements ......................................................... 61
Interaction with other products from BMC Software .................. 62
  Administrative products for DB2 .......................................... 62
  Backup and Recovery products for DB2 ............................... 62
  Utility products for DB2 ....................................................... 63
  Associated common components that DASD MANAGER PLUS uses .......... 63
  Release-level compatibility .................................................. 64

Chapter 2  Getting started with the ISPF interface .................... 67

Overview .............................................................................. 67
  Using the main menu .......................................................... 68
  Using additional panel types ............................................... 70
  Using ISPF commands ........................................................ 71
Navigating through the panels ................................................ 72
  Using Fast Path Navigation ................................................ 75
  Using space estimation ....................................................... 75
Getting help ......................................................................... 75
  Using wildcard characters ................................................ 76
  Long object name fields ...................................................... 76
Viewing current environment information ................................ 77
## Chapter 4  Maintaining and generating actions  
### Before you begin  
- Actions: 254  
- Services: 255  
- Action Generation: 260  
### Working with actions  
- Accessing the action maintenance and generation function: 270  
- Creating an action: 271  
- Creating an action by copying an existing action: 273  
- Editing actions: 274  
- Deleting actions: 278  
### Working with services  
- Editing action services: 279  
- Copying Services from one action into another: 285  
- Creating and editing services: 286  
- Example of creating a user-defined service: 288  
- Editing program details: 292  
- Previewing Generated Services: 297  
### Generating actions  
- Restarting or starting over an action: 303  
- Understanding return codes for batch programs: 306  
- Controlling access to actions, services, and service syntax: 307  
## Chapter 5  Using object sets  
### Before you begin  
- Overview of object sets: 310  
- Wildcards for object names: 310  
- Patterns for object sets: 311  
- Set Specification panel (for object sets): 316  
- Example of defining object sets: 318  
- Common tasks: 320  
### Using object sets from the main menu  
- Creating object sets by using the main menu: 322  
- Creating dynamic SQL object sets: 324  
- Editing object sets: 327  
- Viewing object set results: 329  
- Identifying actions associated with an object set: 330  
- Copying object sets: 331  
- Deleting object sets: 333  
- Renaming object sets: 334  
### Using object sets from action services  
- Creating an object set for an action service: 336  
- Associating an existing object set with an action: 337  
- Associating an object set with an action service by name: 338  
- Viewing object set results: 339  
### Controlling access to object sets: 340
Chapter 6 Collecting and managing statistics 341

Before you begin ........................................... 342
Overview of collection and update tasks ........................................... 343
Tips for collecting statistics ........................................... 344
Collecting statistics ........................................... 346
IBM RUNSTATS utility ........................................... 346
BMCSTATS utility ........................................... 346
DASD MANAGER PLUS historical database ........................................... 347
BMCSTATS options ........................................... 349
Sampling statistics ........................................... 358
Minimum size requirements ........................................... 358
Specifying sampling percentages ........................................... 359
Restricting sampling by object type ........................................... 360
Sampling table columns ........................................... 360
Accessing the statistics display features ........................................... 360
Displaying DB2 object lists ........................................... 361
Creating a DB2 object list ........................................... 361
Listing data sets used by a database ........................................... 364
Updating DB2 catalog statistics ........................................... 365
Browsing and updating statistics ........................................... 366
DB2 and BMCSTATS statistics ........................................... 366
DB2 table space statistics ........................................... 367
DB2 table statistics ........................................... 368
DB2 table space partition statistics ........................................... 369
DB2 column statistics ........................................... 371
DB2 index statistics ........................................... 373
DB2 index partition statistics ........................................... 374
Updating the DB2 catalog interactively ........................................... 375
Table space statistics update ........................................... 375
Table statistics update ........................................... 376
Table space partition table statistics update ........................................... 378
Column statistics update ........................................... 379
Index statistics update ........................................... 380
Index partition statistics update ........................................... 382
Managing DB2 and BMC Software statistics ........................................... 383
Having authority to collect statistics ........................................... 383
Updating DB2 columns (by DASD MANAGER PLUS) ........................................... 384
Updating DB2 real-time statistics ........................................... 389
Updating the DB2 catalog by using BMCSTATS ........................................... 390
Updating the DB2 catalog by using BMCUPRS ........................................... 398
Updating DASD MANAGER PLUS statistics by using BMCCPRS ........................................... 401
Altering, renaming, or dropping and re-creating objects ........................................... 404
Deleting BMC Software statistics ........................................... 405

Chapter 7 Analyzing statistical trends 407

Before you begin ........................................... 408
Analyzing BMC Software statistics ........................................... 408
BMC Software statistics for DB2 objects ........................................... 409
Setting up a BMCTRIG job ........................................ 504
Creating an action with BMCTRIG .......................... 504
Specifying BMCTRIG syntax options ....................... 507
Generating a BMCTRIG job .................................. 524
Using worklist format or standard JCL ....................... 526
Viewing exceptions that BMCTRIG produces ............. 545
Working with reorganization guidelines for thresholds . 549
Generating an automatic reorganization with BMCTRIG .. 553
Setting up a REORG corrective action ....................... 556
Working with restarts and batch JCL generation .......... 565
Using BMCTRIG-generated job restart ..................... 566
Working with restart and startover processing ............ 566
Using the restart program .................................. 567
Restarting jobs that BMCTRIG generates .................. 569
Using batch processing ................................... 571

Chapter 9 Changing the secondary quantity .... 575
Before you begin ............................................ 575
Understanding the ALTERSEC utility ....................... 576
Altering the secondary quantity (SQTY) ................. 579

Chapter 10 Producing reports ...................... 581
Before you begin ............................................ 582
Standard reports ............................................. 583
   Standard reports descriptions ......................... 585
   Organization of report options ....................... 587
   Average length of RID chain report ................. 587
   Event recording report ................................ 590
   Index analysis report ................................ 593
   Index clustering analysis report ..................... 596
   Index combination statistics report ................. 600
   Index keys analysis report .......................... 604
   Index leaf distribution report ....................... 606
   Indexspace space detail report ...................... 609
   Monthly detail by tablespace report .................. 611
   Page update analysis report ......................... 614
   Statistics exception report ......................... 617
   Table column detail report .......................... 623
   Table space extent summary report ................. 626
   Table space offset RIDs report ....................... 629
   Table space space detail report ..................... 631
   Volume analysis report .............................. 635
   Volume free space detail report ..................... 637
Model JCL for batch generation ....................... 640
QMF reports ............................................. 641
Customizable reports ................................ 644
   Understanding customizable reports ............... 644
   Creating customizable reports ..................... 650
Running customizable reports ........................................... 650
Running a customizable report in batch mode ......................... 653
Customizing the report programs ........................................... 655
Adding a report to the customizable reports menu ....................... 656
Understanding output data sets ........................................... 657
Location return code descriptions ......................................... 657
Customizable report samples .............................................. 657
Action report ................................................................... 657
Average monthly growth report ............................................. 660
Corrective action cross reference report .................................... 662
Data set list by database name report ....................................... 663
Database space detail report ................................................ 665
Database space trend report ................................................ 666
Database storage utilization report ......................................... 668
Exception report for BMCTRIG action ..................................... 671
Index clustering analysis report ............................................. 672
Modified table space pages report .......................................... 674
Space estimation report ...................................................... 675
Space estimation trend report ............................................... 677
Space utilization by table report ............................................ 679
Table space and index storage limits report .............................. 680
Tables with more than average growth report ......................... 681
Table space and index extents report ...................................... 682
Volume space trend report .................................................. 683

Chapter 11   Exporting and deploying product definitions 685
Overview of exporting object definitions ................................. 686
How the Export utility works ............................................... 686
Recommendations for setting up connections ............................. 690
Enterprise list and personal list of connections ......................... 690
Required authorizations for using Export .................................. 691
Task summary for exporting object definitions ........................... 693
Preparing your environment for exporting ............................... 694
Accessing the Export utility .................................................. 695
Setting up connections ....................................................... 696
Setting up enterprise connections ......................................... 696
Setting up destination connections ....................................... 698
Specifying definitions to export ............................................ 700
Specifying definitions to export ............................................ 701
Specifying action definitions ............................................... 701
Specifying service definitions .............................................. 702
Specifying object set definitions .......................................... 703
Specifying object-action priority definitions ............................. 703
Specifying exceptions definitions ........................................ 704
Verifying the list of definitions to export ................................ 706
Selecting the destinations for an export ................................... 706
Deploying the definitions ................................................... 707
Reviewing the deployment report ......................................... 708
### Appendix A Using the skeleton library compiler

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>709</td>
</tr>
<tr>
<td>Compiling SLIBs</td>
<td>710</td>
</tr>
<tr>
<td>Changing SLIBs</td>
<td>711</td>
</tr>
<tr>
<td>- Testing changes using ISPF file tailoring</td>
<td>712</td>
</tr>
<tr>
<td>- Compiling changed SLIBs</td>
<td>713</td>
</tr>
<tr>
<td>Processing SLIBs</td>
<td>713</td>
</tr>
<tr>
<td>- Generating the SLIB report</td>
<td>714</td>
</tr>
</tbody>
</table>

### Appendix B Customizable reports

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>718</td>
</tr>
<tr>
<td>Customizable report programs</td>
<td>719</td>
</tr>
<tr>
<td>- Report program structure</td>
<td>720</td>
</tr>
<tr>
<td>- Sample report program</td>
<td>720</td>
</tr>
<tr>
<td>- Sample report</td>
<td>721</td>
</tr>
<tr>
<td>Report program customization</td>
<td>722</td>
</tr>
<tr>
<td>- Report layout</td>
<td>722</td>
</tr>
<tr>
<td>- Variables</td>
<td>724</td>
</tr>
<tr>
<td>- SQL</td>
<td>727</td>
</tr>
<tr>
<td>- JCL</td>
<td>729</td>
</tr>
<tr>
<td>ASURXSQL external function</td>
<td>730</td>
</tr>
<tr>
<td>ASURXSQL return codes</td>
<td>731</td>
</tr>
<tr>
<td>SQL support</td>
<td>733</td>
</tr>
<tr>
<td>ASURXRPT external function</td>
<td>733</td>
</tr>
<tr>
<td>ASURXLSPI external function</td>
<td>734</td>
</tr>
<tr>
<td>ASURXSEI external function</td>
<td>735</td>
</tr>
<tr>
<td>ASURXLOCI external function</td>
<td>739</td>
</tr>
<tr>
<td>Notes on REXX</td>
<td>740</td>
</tr>
<tr>
<td>- Comments</td>
<td>740</td>
</tr>
<tr>
<td>- REXX delimiters</td>
<td>740</td>
</tr>
<tr>
<td>- Case</td>
<td>740</td>
</tr>
<tr>
<td>- Continuation characters</td>
<td>740</td>
</tr>
<tr>
<td>- Concatenation operator</td>
<td>741</td>
</tr>
<tr>
<td>Annotated customizable report</td>
<td>741</td>
</tr>
<tr>
<td>Sample program</td>
<td>741</td>
</tr>
<tr>
<td>JCL and execution</td>
<td>754</td>
</tr>
<tr>
<td>Sample report</td>
<td>755</td>
</tr>
<tr>
<td>Complete sample program</td>
<td>756</td>
</tr>
</tbody>
</table>

### Appendix C Example of utility automation

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>761</td>
</tr>
<tr>
<td>BMCSTATS utility</td>
<td>761</td>
</tr>
<tr>
<td>About this example</td>
<td>762</td>
</tr>
<tr>
<td>Modifying the sample files</td>
<td>762</td>
</tr>
<tr>
<td>Adding corrective action definitions</td>
<td>763</td>
</tr>
<tr>
<td>Modifying thresholds and corrective actions</td>
<td>763</td>
</tr>
<tr>
<td>Modifying the default corrective actions</td>
<td>764</td>
</tr>
<tr>
<td>Contents 13</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>Running the BMCSTATS utility</td>
<td>764</td>
</tr>
<tr>
<td>Modifying the BMCTRIG JCL</td>
<td>765</td>
</tr>
<tr>
<td>Executing or scheduling the BMCTRIG job</td>
<td>766</td>
</tr>
<tr>
<td>Scheduling the BMCTRIG generated jobs</td>
<td>766</td>
</tr>
<tr>
<td><strong>Appendix D  Using the UIM Server</strong></td>
<td>767</td>
</tr>
<tr>
<td>Overview</td>
<td>767</td>
</tr>
<tr>
<td>Starting and stopping the UIM server</td>
<td>767</td>
</tr>
<tr>
<td>Verifying that the UIM server is running</td>
<td>768</td>
</tr>
<tr>
<td>Accessing the BMC UIM server Commands web page</td>
<td>769</td>
</tr>
<tr>
<td>Viewing active users</td>
<td>771</td>
</tr>
<tr>
<td>Changing the security authorization timeout feature temporarily</td>
<td>771</td>
</tr>
<tr>
<td>Enabling or disabling the overall tracing option temporarily</td>
<td>772</td>
</tr>
<tr>
<td>Enabling or disabling specific tracing options temporarily</td>
<td>773</td>
</tr>
<tr>
<td>Refreshing the UIM server content</td>
<td>774</td>
</tr>
<tr>
<td>Resolving UIM Server problems</td>
<td>775</td>
</tr>
<tr>
<td><strong>Appendix E  Obtaining enhanced database performance evaluation using SQL Performance for DB2</strong></td>
<td>779</td>
</tr>
<tr>
<td>Overview</td>
<td>779</td>
</tr>
<tr>
<td>Reorganizations based on the candidate table only</td>
<td>780</td>
</tr>
<tr>
<td>Reorganizations based on the candidate table and BMCTRIG evaluations</td>
<td>781</td>
</tr>
<tr>
<td>How Performance Advisor works</td>
<td>781</td>
</tr>
<tr>
<td>Preparing to use REORG Advisor</td>
<td>782</td>
</tr>
<tr>
<td>Populating the REORG candidate table</td>
<td>783</td>
</tr>
<tr>
<td>Making BIND modifications</td>
<td>783</td>
</tr>
<tr>
<td>Obtaining sample JCL files</td>
<td>784</td>
</tr>
<tr>
<td>Setting up automated reorganizations</td>
<td>785</td>
</tr>
<tr>
<td><strong>Glossary</strong></td>
<td>793</td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>817</td>
</tr>
</tbody>
</table>
Figures

Components that DASD MANAGER PLUS uses ................................. 64
DASD MANAGER PLUS main menu description .......................... 68
Organization of DASD MANAGER PLUS panels ......................... 74
DASD MANAGER PLUS plans and synonyms ................................. 77
Functions of options ..................................................................... 87
General Options panel ................................................................. 94
JCL Generation Jobcard Options Update panel .......................... 95
JCL Generation STEPLIB Options Update panel ......................... 98
JCL Generation Static Data Set Options Update panel ............... 99
JCL Generation Tape Options Update panel ................................. 104
JCL Generation Options for Sort Files Update panel ................. 106
JCL Generation Data Set Options for SORTOUT Update panel .... 109
JCL Generation Generation Data Group Options Update panel .... 113
JCL Generation Debugging, Display and Execution Options Update panel ................................................................. 114
Execution JCL with comments ....................................................... 115
JCL Generation BMC Utility Option Module Names Update panel ... 117
Online Reorg Utility Options panel .............................................. 118
PROC and STEP Names panel ..................................................... 120
BMCTRIG Control-M Options panel ........................................... 123
User Defined Variables panel ..................................................... 124
JCL Generation Product Options File (POF) Functions panel ....... 125
Reusing an existing POF ............................................................. 130
Specifying the name of the existing POF .............................. 131
Specifying the prefix for a copy data set ................................... 132
Changing the SLIB variable for the copy data set in AJX#DSNS .... 132
JCL Generation Jobcard Options Update panel ......................... 134
Including the JCL library ............................................................ 134
JCL Generation Debugging, Display and Execution Options Update panel ................................................................. 135
Including JCL between steps .................................................... 135
Including JCL at the end of the jobs ......................................... 135
Expanded INCLUDE members ..................................................... 136
Using the &GDG symbolic variable ............................................. 141
Data set names resolved with the &GDG symbolic variable .... 141
Worklist with GDGs defined in user options ......................... 142
JCL When GDGs are defined in user options ......................... 142
BMCCOPY Copy Options panel ............................................... 144
Local primary copy options with GDG variable specified ......... 145
Recovery primary copy options with GDG variable specified .... 145
Recovery backup copy options with GDG variable specified .... 146
Worklist with GDGs defined in Copy DD options ..................... 146
Figures 17
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Partition Pagegroup Statistics panel</td>
<td>425</td>
</tr>
<tr>
<td>Storage Group Statistics panel</td>
<td>426</td>
</tr>
<tr>
<td>Volume Statistics panel</td>
<td>427</td>
</tr>
<tr>
<td>Statistics graph components</td>
<td>432</td>
</tr>
<tr>
<td>Graphics Options panel</td>
<td>433</td>
</tr>
<tr>
<td>Printer Graphics Options panel</td>
<td>435</td>
</tr>
<tr>
<td>Tablespace Estimation panel (nonpartitioned)</td>
<td>438</td>
</tr>
<tr>
<td>Tablespace Estimation (Partitioned) panel</td>
<td>439</td>
</tr>
<tr>
<td>Tablespace Estimation (Partitioned by Growth) panel</td>
<td>441</td>
</tr>
<tr>
<td>Index Space Estimation panel (nonpartitioned)</td>
<td>445</td>
</tr>
<tr>
<td>Index Space Estimation panel (partitioned)</td>
<td>446</td>
</tr>
<tr>
<td>Tablespace Estimation panel</td>
<td>448</td>
</tr>
<tr>
<td>Initiating a reorganization based on exception values</td>
<td>455</td>
</tr>
<tr>
<td>Evaluating exceptions and generating corrective actions</td>
<td>472</td>
</tr>
<tr>
<td>Evaluating exceptions</td>
<td>473</td>
</tr>
<tr>
<td>Generating jobs</td>
<td>474</td>
</tr>
<tr>
<td>Evaluating exceptions for automation</td>
<td>475</td>
</tr>
<tr>
<td>BMCTRIG main panel</td>
<td>487</td>
</tr>
<tr>
<td>Override Thresholds panel in BMCTRIG Override System Trigger panel</td>
<td>488</td>
</tr>
<tr>
<td>Exception thresholds on System Thresholds and Corrective Actions panel</td>
<td>489</td>
</tr>
<tr>
<td>Exception Thresholds panel</td>
<td>489</td>
</tr>
<tr>
<td>Edit Exception Threshold panel</td>
<td>491</td>
</tr>
<tr>
<td>Edit Exception panel</td>
<td>493</td>
</tr>
<tr>
<td>Exception definitions on System Thresholds and Corrective Actions panel</td>
<td>494</td>
</tr>
<tr>
<td>Exception Definitions List panel</td>
<td>495</td>
</tr>
<tr>
<td>Edit User-defined Exception panel</td>
<td>495</td>
</tr>
<tr>
<td>Corrective actions on System Thresholds and Corrective Actions panel</td>
<td>498</td>
</tr>
<tr>
<td>Corrective Actions panel</td>
<td>498</td>
</tr>
<tr>
<td>Edit Corrective Action panel</td>
<td>499</td>
</tr>
<tr>
<td>Object-action priorities on System Thresholds and Corrective Actions panel</td>
<td>501</td>
</tr>
<tr>
<td>Object-Action Priorities panel</td>
<td>501</td>
</tr>
<tr>
<td>Edit Object-Action panel</td>
<td>502</td>
</tr>
<tr>
<td>Example of Edit Action Services panel with BMCTRIG</td>
<td>505</td>
</tr>
<tr>
<td>BMCTRIG main panel for example job EXCEPTI</td>
<td>506</td>
</tr>
<tr>
<td>Example of options to eliminate duplicates and set index escalation</td>
<td>511</td>
</tr>
<tr>
<td>Rebind options</td>
<td>513</td>
</tr>
<tr>
<td>REORG with Resize option</td>
<td>516</td>
</tr>
<tr>
<td>Job and JCL Generation options</td>
<td>518</td>
</tr>
<tr>
<td>Reorg-Related Override Exceptions</td>
<td>522</td>
</tr>
<tr>
<td>Statistics-Related Exceptions options</td>
<td>523</td>
</tr>
<tr>
<td>Copy-Related Exceptions options</td>
<td>524</td>
</tr>
<tr>
<td>BMCTRIG worklist</td>
<td>525</td>
</tr>
<tr>
<td>Generating standard JCL</td>
<td>527</td>
</tr>
<tr>
<td>Interim worklist</td>
<td>529</td>
</tr>
<tr>
<td>Generated standard JCL</td>
<td>533</td>
</tr>
<tr>
<td>Generating worklist JCL</td>
<td>539</td>
</tr>
<tr>
<td>Worklist JCL Generation options panel</td>
<td>540</td>
</tr>
<tr>
<td>Generated worklist JCL</td>
<td>541</td>
</tr>
<tr>
<td>Exceptions reported in AEXPRINT</td>
<td>546</td>
</tr>
</tbody>
</table>
Active Exceptions panel ........................................ 548
Object Exception Detail panel .................................... 548
REORG panel ................................................... 558
Reorg options .................................................... 560
Copy DD options panel ........................................... 561
COPYDDN dname1 JCL panel ..................................... 562
Sort options panel ............................................... 562
Statistics options ................................................ 563
Conditional Reorg options ....................................... 564
Edit Action Services panel ....................................... 564
Online restart .................................................... 567
BMCTRIG Generated Job Restart panel .......................... 569
Sample AEXIN_DD input stream for BMCTRIG JCL restart 571
Sample JCL for batch restart processing .......................... 571
Altering the secondary quantity ................................. 576
ALTERSEC –SQL command ...................................... 577
Flow diagram for ALTERSEC ..................................... 578
Alter Secondary Quantity panel .................................. 579
Report Selection Menu panel .................................... 583
Standard DASD MANAGER PLUS reports ....................... 584
Average Length of RID Chain panel ............................. 588
Average Length of RID Chain report ............................ 589
Event Recording Request panel ................................ 590
Event List ......................................................... 592
Index Analysis panel ............................................. 593
Index Analysis report ............................................ 595
Index Clustering Analysis panel .................................. 597
Index Clustering Analysis report ................................. 599
Index Combination Statistics panel ............................... 601
Index Combination Statistics report .............................. 603
Index Keys Analysis panel ....................................... 604
Index Key Analysis report ...................................... 605
Index Leaf Distribution panel ..................................... 607
Index Leaf Distribution report .................................... 608
Indexspace Space Detail panel .................................... 609
Indexspace Space Detail report ................................... 610
Monthly Detail by Tablespace panel ............................... 612
Monthly Detail by Tablespace report ............................. 613
Page Update Analysis panel ..................................... 615
Page Update Analysis report ..................................... 616
Statistics Exception Selection panel ............................. 618
Statistics Exception Type Selection panel ....................... 619
Exceptions Selection panel ...................................... 619
Exceptions report for a single exception type .................. 621
Exceptions Selection panel ...................................... 622
Exceptions report for all exception types found for object 623
Table Column Detail panel ....................................... 624
Table Column Detail report ...................................... 625
Tablespace Extent Summary panel ................................. 626
Tablespace Extent Summary report .............................................. 628
Tablespace Offset RIDs panel .................................................. 629
Tablespace Offset RIDs report .................................................. 630
Tablespace Space Detail panel .................................................. 632
Tablespace Space Detail report .................................................. 633
Volume Analysis panel ............................................................ 635
Volume Analysis report ........................................................... 636
Volume Free Space Detail panel .................................................. 638
Volume Free Space Detail report ................................................. 639
JCL for batch generation ........................................................... 640
Parms input line and required input line ..................................... 646
Input parameters at the end of the report ..................................... 647
Input parameters in generated batch JCL ...................................... 647
Internal KB switch ................................................................. 650
Customizable Report Selection List .......................................... 651
Generated batch JCL from the online dialog (without long name) .... 652
Generated batch JCL from the online dialog (with long name) ......... 652
Batch JCL to run reports outside the online dialog ....................... 653
Input tables ............................................................................. 655
Set of report menu code ........................................................... 656
Action Report: Example 1 .......................................................... 658
Action Report: Example 2 .......................................................... 660
Average Monthly Growth report ................................................ 660
Corrective Action Cross Reference report .................................... 662
Data Set List by database name report ........................................ 664
Database Space Detail Report .................................................... 665
Database Space Trend Report ..................................................... 667
Database Storage Utilization Report .......................................... 669
Exception report for BMCTRIG Action ....................................... 671
Index Clustering Analysis Report .............................................. 672
Modified Table Space Pages Report .......................................... 674
Space Estimation Report ........................................................... 675
Space Estimation Trend Report .................................................. 677
Space Utilization by Table Report .............................................. 679
Table Space and Index Storage Limits Report ............................. 680
Tables with More than Average Growth Report ......................... 681
Table Space and Index Extents Report ........................................ 682
Volume Space Trend Report ...................................................... 683
Processing flow of the SLIB compiler ........................................ 710
Sample runtime report ............................................................. 714
Basic report program structure .................................................. 720
Sample REXX report program ................................................... 720
Volume free space report .......................................................... 721
Sample report layout ............................................................... 723
SQL as argument ..................................................................... 727
JCL to Run Customized REXX Reports ...................................... 729
Sample REXX program using ASURXLOC ................................. 739
Specifying the DB2 object and report title .................................. 742
Defining report formats ............................................................ 743
Connecting to the DB2 subsystem and opening the plan ......................... 746
Initializing the report ........................................................................ 747
Getting the input and testing for wildcards ........................................ 747
Fetching the data ............................................................................. 749
Preparing and writing the data ......................................................... 750
The report error procedure .............................................................. 751
Reporting CAF error code and message ............................................ 752
The SQL error procedure ................................................................. 752
Invoking the cleanup procedure ...................................................... 754
JCL for customized REXX report ...................................................... 754
A sample customizable report .......................................................... 755
Complete sample program .............................................................. 756
Tables

Product benefits ................................................................. 52
Problem and Solutions ..................................................... 53
Supported versions of DB2 UDB for OS/390 and z/OS .......... 60
Unsupported COPY PLUS options ........................................ 65
Unsupported LOADPLUS options ......................................... 65
Unsupported REORG PLUS parameters ............................... 65
Unsupported UNLOAD PLUS options ................................. 65
Common information and input fields ............................... 69
Common ISPF commands .................................................. 71
Navigation from the main menu options ........................... 72
Fast Path Navigation commands ...................................... 75
Supported wildcard characters ........................................ 76
Guidelines for data sets .................................................... 88
DASD MANAGER PLUS plans ............................................ 89
Execution plans ............................................................... 90
Setup procedures ............................................................ 92
Data set sizing options ..................................................... 100
Work data sets in the JCL cleanup job step ..................... 102
Work data sets used by utilities ...................................... 102
Values for 7-track tape drives ........................................ 105
POF keywords used to suppress the ddname .................... 110
Comment codes for data set sizing ................................. 115
User-defined variables .................................................... 138
Data set sizing values and sources ................................. 139
Information sources for BMCTRIG utility ......................... 230
Information sources for utilities other than BMCTRIG .... 231
Variable option notation .................................................. 232
DASD MANAGER PLUS utilities and commands ............... 256
Command descriptions .................................................... 257
JCL elements for generating services ............................... 258
Supported symbolic variables ......................................... 259
Tasks for generating actions .......................................... 262
Worklist commands ........................................................ 264
Column format of worklists ............................................. 266
Return codes for ASUSMAIN, ASUSCPRS, and ASUSUPRS ...... 306
Return codes for ASUSTRIG ............................................... 307
Wildcard characters ........................................................ 310
Object patterns .............................................................. 312
Resulting object sets for pattern and resolution types ....... 313
Object set tasks .............................................................. 320
Collecting and updating statistics ........................................... 343
Statistics collection and update tasks .................................... 344
DASD MANAGER PLUS historical database ............................. 347
BMCSTATS sample table space parameter options ...................... 353
BMCSTATS sample index parameter options ............................ 354
Source column .................................................................. 363
Status field ....................................................................... 363
DB2 Object list commands .................................................... 363
Authorization requirements .................................................... 383
Table columns that are updated in the DB2 catalog .................. 385
Partitioned Table Space/Index Space Columns Updated in the DB2 Catalog 388
Statistics reset in table SYSIBM.SYSTABLESPACESTATS ............. 389
Statistics reset in table SYSIBM.SYSINDEXSPACESTATS ............. 390
Statistics graphs summary ..................................................... 428
Example actions for eliminating duplicate actions ................. 463
Grouped services ................................................................ 466
BMCTRIG processing options ................................................. 471
Reorganization-related exceptions ......................................... 477
Statistics-related exceptions .................................................. 482
Copy-related exceptions ....................................................... 484
Applying system-level thresholds to objects ............................ 490
Applying object-action priority .............................................. 502
What to Analyze options ..................................................... 508
Saving and Reporting options ............................................... 509
Eliminate Duplicates and Index Escalation options .................. 511
Corrective Action Generation options .................................... 512
Triggered Rebind utilities ...................................................... 513
Rebind options .................................................................. 514
Resize options ................................................................... 515
BMCTRIG interpretation of Resize option values ..................... 516
Standard JCL tasks .............................................................. 526
Standard JCL worklist commands ........................................... 528
JOBSEQ variables .............................................................. 531
Worklist JCL benefits .......................................................... 538
Supported REORG mapping table variables ............................ 557
Shrlevel access during reorganization .................................... 559
Shrlevel options ................................................................. 559
Reorg options .................................................................. 560
Sort options ..................................................................... 563
JCL DD statement dispositions for startover ......................... 567
JCL DD statement dispositions for restarts ............................ 567
How ALTERSEC works ....................................................... 577
DASD MANAGER PLUS standard reports .............................. 585
Views for QMF reports .......................................................... 641
Descriptions of QMF reports ................................................ 642
DASD MANAGER PLUS customizable reports ...................... 645
Input parameters ................................................................. 648
Reasons for days to full warning .......................................... 668
Definition validation ............................................................ 687
Tables

- POF keywords for specifying the UIM host definition ............................................. 691
- Authorization to edit the ECL ............................................................................... 692
- Task summary for setting up and deploying definitions ........................................... 693
- Runtime report statistics ......................................................................................... 715
- External functions not available in REXX .............................................................. 719
- Format names ........................................................................................................ 723
- How to use variables .............................................................................................. 724
- Wildcards in report-dependent variables ............................................................... 726
- Information passed by the EXEC PARM ................................................................. 729
- DD Statements for Report Programs ..................................................................... 730
- ASURXSQL external function .............................................................................. 730
- ASURXSQL return codes ....................................................................................... 731
- ASURXRPT external function ................................................................................ 733
- ASURXRPT return codes ....................................................................................... 734
- ASURXLSL return codes ....................................................................................... 735
- ASURXSE external function .................................................................................. 735
- ASURXSE return codes ......................................................................................... 738
- ASURXLOC return codes ....................................................................................... 740
- Information passed by the EXEC PARM ................................................................. 743
- Detail line format definitions ................................................................................ 744
- Variables in the detail line format definitions ....................................................... 745
- Top of page format definitions (second line) ......................................................... 745
- ALLOW_NETCMD variable settings ...................................................................... 769
About this book

This book contains detailed information about the DASD MANAGER PLUS for DB2 product and is intended for system administrators, database administrators, or applications developers. To use this book, you should be familiar with the following items:

- IBM® DB2 Universal Database for z/OS® (DB2) DBMS
- z/OS operating system
- job control language (JCL)
- Interactive System Productivity Facility (ISPF)

For example, you should know how to respond to ISPF panels and perform common tasks in this environment. 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 (http://www.bmc.com/support), you can use either of the following methods to access related publications that support your product or solution:

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

This book uses the following special conventions:

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

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

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

Summary of changes

This section summarizes changes to the functionality of the product, listing the changes by product version and release date. The summary includes enhancements to the product and any major changes to the documentation.

This book uses change bars to mark significant changes.

Version 11.1.00 June 2013

This release fixes known problems in the product and includes the following enhancements:
End of support for DB2 Version 8

Starting with this release, DASD MANAGER PLUS does not support DB2 Version 8. Earlier releases will continue to support Version 8.

BMCSTATS enhancements

DASD MANAGER PLUS for DB2 supports the following BMCSTATS enhancements:

- LOB statistics collection
- Inline LOB statistics collection
- Native support for the COLGROUP option
- Collection of KEYCARD data automatically on a DB2 Version 10 or later subsystem

Because DASD MANAGER now collects KEYCARD statistics automatically, the KEYCARD N option is no longer available while executing in a DB2 Version 10 environment.

- SQL -911 error processing

The 911ACTION option allows you to specify whether to ignore or stop processing on SQL -911 errors.

- SQL -911 error reduction

The new OPTIMIZECOMMIT installation and syntax options allow you to limit the number of SQL commits executed during a BMCSTATS run.

- 31- or 64-bit address mode

The new ATBWORKAREA installation and syntax options allow you to request that frequency and cardinality statistics be gathered in 31-bit mode (below the bar) or 64-bit mode (above the bar).

- BMCSTATS large job storage consolidation

The new QUIESCEINTERVAL syntax option enables users with a very large number of DB2 objects per job to avoid potential storage depletion abends. This option allows you to specify a point at which BMCSTATS will slow processing in order to consolidate storage holdings, and then resume normal processing.
- Dropped support for IMAGECOPY

  BMCSTATS no longer supports the IMAGECOPY option, which was used to run against the most recent image copy.

- Dropped support for ALTDD

  BMCSTATS no longer supports the ALTDD option, which was used to specify an alternate data set for input.

**BMCTRIG enhancements**

DASD MANAGER PLUS for DB2 supports the following BMCTRIG enhancements:

- Removes dependency on BMCSTATS

  BMCTRIG no longer requires BMCSTATS data for exception analysis.

- Provides a new value for the DB2RTS option

  BMCTRIG now allows you to specify the value O for the DB2RTS option. This value tells BMCTRIG to ignore exceptions that require BMCSTATS data to be analyzed.

- Allows symbolic variables in utility and worklist data set names

  BMCTRIG now supports limited symbolic variables for UTILDSN and worklist DSN. BMCTRIG resolves these symbolic variables at execution time.

- Provides the following new DB2 RTS exceptions:

<table>
<thead>
<tr>
<th>Panel field</th>
<th>Keyword</th>
<th>Exception name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reorg on data space usage</td>
<td>REОРGDATASPACE</td>
<td>REORDSPC</td>
<td>Allows you to specify whether to reorganize a table space based on the percentage of data that occupies the data set</td>
</tr>
<tr>
<td>Copy on dirty pages</td>
<td>IXDIRTY</td>
<td>IXDIRTY</td>
<td>Allows you to specify the percentage of index pages modified since the last image copy</td>
</tr>
<tr>
<td>Reorg after deletes</td>
<td>REОРGDELETES</td>
<td>REORDELS</td>
<td>Allows you to specify whether to reorganize an object based on the percentage of rows deleted since the last reorganization</td>
</tr>
<tr>
<td>Reorg after inserts</td>
<td>REОРGINserts</td>
<td>REORINS</td>
<td>Allows you to specify whether to reorganize an object based on the percentage of rows inserted since the last reorganization</td>
</tr>
</tbody>
</table>
Suppresses exceptions from being analyzed during the trigger analysis phase based on category type:

— The TRIGNOREORG option suppresses all exceptions categorized with type REORG.

— The TRIGNOSTATS option suppresses all exceptions categorized with type STATS.

— The TRIGNOCOPY option suppresses all exceptions categorized with type COPY.

Drops support for the REALTIME STATS option

Current syntax using the REALTIME STATS option will automatically use the equivalent DB2RTS options.

**BMCCPRS and BMCUPRS enhancements**

BMCCPRS and BMCUPRS now allow you to copy COLGROUP statistics.

**User interface enhancements**

DASD MANAGER PLUS supports the following user interface enhancements:

- Preview syntax action

  When editing a selected action service step or selected syntax on the Service Syntax List panel, you can now preview that syntax via a new PREVIEW option.

- Browse action for syntax options
A new Browse action on the Service Syntax List panel allows you to browse syntax options.

- FIND command for syntax options

You can use the new command-line FIND command while editing or browsing syntax options to find text.

**BMC Common Repository for object sets**

DASD MANAGER object sets are stored in a common set of DB2 tables, enabling object sets to be shared across products. The following products can use an object set that has been defined in any of these products:

- COPY PLUS for DB2
- DASD MANAGER PLUS
- RECOVER PLUS for DB2
- RECOVERY MANAGER for DB2

**BMCCOPY option changes**

DASD MANAGER PLUS has added or changed support for the following COPY PLUS (BMCCOPY) options:

- AUX
- CLONE
- DATAMVR
- FULLRESET
- IXEXPAND
- INVCACHE
- IXSIZE
- MAXFULLDAYS
- NACTIVE
- QUIESCE BEFORE WRITE (removed)
- ZIIP
**BMCCOPYI option changes**

DASD MANAGER PLUS has added or changed support for the following COPY PLUS (BMCCOPYI) options:

- AUX
- DATAMVR
- DISPLOCK (removed)
- HISTRETN (removed)
- IXDSNUM (removed)
- IXSIZE (removed)
- MIGRSKIP (removed)
- MIGRVOL (removed)
- OUTSIZE (removed)
- READONLY (removed)
- SMARTSTACK (removed)
- ZIIP

**BMCMOD option changes**

DASD MANAGER PLUS has added or changed support for the COPY PLUS (BMCMOD) option NOCOPYPEND.

**BMCREORG option changes**

DASD MANAGER PLUS has added or changed support for the following REORG PLUS (BMCREORG) options:

- AUX
- CLONE
- CLUSTERRATIO (removed)
- DRAIN_WAIT (now allows an integer specification)
- DSNTYPE
- IDCACHE
- FORCE
- FORCE_DELAY
- FORCE_REPORT
- REPORT
- TOTALPAGEPCT
- TSSAMPLEPCT
- ZIIP
CHEK DA option changes

DASD MANAGER PLUS has added or changed support for the DB2 CHECK DATA (CHEK DA) option XMLSCHEMAONLY.

CHEK LOB option changes

DASD MANAGER PLUS has added or changed support for the following DB2 CHECK LOB (CHEK LOB) options:

- PUNCHDDN
- SHRLEVEL

COPY option changes

DASD MANAGER PLUS has added or changed support for the following DB2 COPY options:

- FCCOPYDDN
- FLASHCOPY

RECOVER option changes

DASD MANAGER PLUS has added or changed support for the following DB2 RECOVER options:

- BACKOUT
- ENFORCE
- LOGRANGES
- RESTOREBEFORE
- TAPEUNITS
- VERIFYSET
REORG option changes

DASD MANAGER PLUS has added or changed support for the following DB2 REORG options:

- AUTOESTSPACE
- AUX
- FLASHCOPY
- FCCOPYDDN
- FORCE
- KEYCARD (removed)
- SORTKEYS (removed)
- WORKDDN2 (removed)

REPAIR option changes

DASD MANAGER PLUS has added or changed support for the following DB2 REPAIR options:

- NOAREORPEND
- NOAREORPENDSTAR
- NOAUXCHKP (for table spaces)
- NOAUXWARN (for table spaces)
- NORBDPEND (for indexes)
- PSRBDPEND (for indexes)
- RBDPEND (for indexes)

JCL Generation and Execution changes

This release includes the following changes for JCL Generation and Execution, which is a component of DASD MANAGER PLUS.

DSNTIAD plan name

The JCL Generation component now lets you specify the name of the DB2 plan that will run the IBM DSNTIAD program. You can specify the name in either of the following locations:

- JCL Generation Debugging, Display and Execution Options panel
- DSNTIAD_PLAN keyword in the product options file (POF)
JCL members before job steps

The JCL Generation component now lets you include a JCL member before each job step in JCL. You can specify the name of the member in either of the following locations:

- JCL Generation Debugging, Display and Execution Options panel
- PRE_JOBSTEP_INCLUDE keyword in the POF

SYSPRIN2 output data set

The JCL Generation component offers a new SYSPRIN2 data set for viewing a BMC utility's SYSPRINT output while the utility is running or when the utility's execution is canceled. You can use either of the following locations to include the SYSPRIN2 DD in the JCL:

- JCL Generation Static Data Set Options panel
- INCLUDE_SYSPRIN2 keyword in the POF

Suppressed DD names

The JCL Generation component offers the following POF keywords to suppress adding DD names to prefixes for the unload (SYSREC) data sets:

- UNLD_FREF_SUPPR_SUFF
- UNLD1_SUPPRESS_SUFF
- UNLD2_SUPPRESS_SUFF
- UNLD3_SUPPRESS_SUFF
- UNLD4_SUPPRESS_SUFF

Omitted unit names for data sets

Your shop standards might require that you omit the UNIT parameter from your JCL. The JCL Generation component now lets you omit the parameter by specifying a value of NONE for the unit name.

New command for resetting POF variables

You can now enter one of the following commands on the Command line of the product to easily reset all of the ISPF variables in the ISPF profile with the variables in the POF:

- To reset the variables in the initial POF, enter TSO POFRESET.
- To reset the variables in a specified initial or user POF, enter TSO POFRESET POF(dataSetName(POFMember)).
To enable these commands, you must modify the POFRESET CLIST. The CLIST is located in the \HLQ\DBLIB library.

**Options for BMC Control-M**

For DASD MANAGER PLUS, JCL Generation now provides a panel for specifying options that the BMCTRIG function uses. From the BMCTRIG Control-M Options panel, you can specify whether to generate BMC Control-M job schedule entries for jobs that do not contain IEFBR14 steps. You can also use this panel to specify the names of the input and output data sets for the entries.

**Changes to minimum requirements**

This version of DASD MANAGER PLUS has the following changes to minimum requirements. For full requirements information, see the DASD MANAGER PLUS product documentation.

- BMC Common Statistics version 11.1.00
- DB2 Solution Common Code (SCC) version 11.1.00
- DB2 Utilities Common Code (D2U) version 10.2.00
- IBM z/OS® Version 1.10

**Documentation changes**

This release includes the following documentation changes:

- All messages are now available in the BMC Documentation Center, which is accessible from the BMC Support Central site (http://www.bmc.com/support). A separate messages manual is no longer available.

- Installation and configuration information is now located in the following books:
  
  — *Installation System User Guide*
  
  — *BMC Products and Solutions for DB2 Configuration Guide*

**Version 10.1.00  April 2011**

This release fixes known problems in the product and includes the following enhancements:

**DB2 Version 10 support**

DASD MANAGER PLUS for DB2 supports the following features of DB2 Version 10:
- online schema deferred ALTER

  BMCTRIG supports the new **ARERPEND** exception to detect table spaces and indexes placed in advisory REORG-pending (AREOR) status.

- compression dictionaries that were created during DB2 SQL INSERT processing

- DEFINE NO for XML columns

  XML objects that are defined with DEFINE NO are bypassed if the underlying data set has not materialized.

- index include columns

  BMCSTATS collects statistics on columns added to a unique index that are not part of the key and are not used to enforce uniqueness.

- temporal tables

  Statistics are now collected for the new temporal data types.

- timestamp columns that are defined with a precision other than 6

  The TIMESTMP data type now allows variable precision in the fractional seconds portion of the timestamp.

- timestamp columns that are defined as TIMESTAMP WITH TIME ZONE

- skip-level migration

  You can migrate DB2 catalogs directly from DB2 Version 8 new-function mode (NFM) to DB2 Version 10 conversion mode (CM).

- universal table spaces that are defined as MEMBER CLUSTER

  The Tablespace Statistics panels now indicate if the table space has a MEMBER CLUSTER page set structure.

- new system and database authorities

  BMCSTATS performs authorization checks against the new DB2 administrative authorities.
Additional enhancements

DASD MANAGER PLUS for DB2 supports the following additional enhancements:

- Simple Space Estimation (SSE)
  
  DASD MANAGER PLUS, CATALOG MANAGER, and CHANGE MANAGER have added a new stand-alone feature that allows you to estimate simple space for table spaces or index objects, giving you “what if” capability. Unlike DASD MANAGER PLUS statistics, you do not need to run BMCSTATS before using SSE. For more information, see Chapter 7, “Analyzing statistical trends.”

- index on expression
  
  DASD MANAGER PLUS supports collecting statistics on expressions. The following new statistics tables (and their associated synonyms) contain data for the first key target:
  
  — BMCATSnn.RS_KEYTGT_DIST (synonym BMCASU_SKTGDIST) contains one or more rows for the first key target of an extended index key.
  
  — BMCATSnn.RS_KEYTGT_DISTSTATS (synonym BMCASU_SKTGDISTSTA) contains zero or more rows for the first key target of a data-partitioned secondary index.

  For more information, see the DASD MANAGER PLUS for DB2 Reference Manual.

- AREO* pending (AREOPEND keyword and AREOPEND exception) shared exception
  
  AREO* pending allows you to specify whether an exception should be raised if the index or the table space is in advisory REORG-pending status. For more information, see Chapter 8, “Analyzing objects by using BMCTRIG.”

- new LOB exceptions:
  
  — Reorg disorg LOB (REORGDISORGLOB keyword and REORGLOB exception) allows you to specify whether to reorganize a table space based on the percentage of not-perfectly-chunked LOBs that have been inserted since the last REORG.
  
  — LOB Disorganize (LOBORGRATIO keyword and LOBORGR exception) allows you to specify whether to reorganize a table space based on the organization percentage of the table space.
  
  — LOB Freespace (LOBFREESPACE keyword and LOBFRSPC exception) allows you to specify whether to reorganize a table space based on the percentage of the LOB that is freespace.
For more information, see Chapter 8, “Analyzing objects by using BMCTRIG”.

- access path stability

BMCTRIG now supports the REBIND options PLANMGMT and PLANMGMTSCOPE. For more information, see the DASD MANAGER PLUS for DB2 Reference Manual.

- processing to an IBM System z® Integrated Information Processor (zIIP)

DASD MANAGER PLUS now provides the option to offload eligible processing to a zIIP. To enable and use zIIP processing, you must have an installed and authorized version of the EXTENDED BUFFER MANAGER (XBM) product or the SNAPSHOT UPGRADE FEATURE (SUF) technology.

For more information about the XBM component that enables the use of zIIPs, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

Documentation enhancements

Chapter 10 includes an updated customized reports table that indicates which reports require BMCSTATS to run.

Changes to minimum requirements

This version of DASD MANAGER PLUS has the following changes to minimum requirements. For full requirements information, see “Software requirements” on page 61.

- DB2 Solution Common Code (SCC) version 10.01.00
- DB2 Utilities Common Code (D2U) version 10.01.00
- BMC Common Statistics version 10.01.00
- z/OS Version 1.10
- XBM or SUF version 5.6.00

Changes to support

The Imagecopy and Altdd options in BMCSTATS will no longer be supported in a future release of this product.
This release fixes known problems in the product and includes the following enhancements:

- DASD MANAGER PLUS for DB2 supports the Export utility. After changing product definitions on a local DASD MANAGER PLUS repository, you can export the changes to DB2 subsystems residing within or across sysplexes. New product options file (POF) keywords support the Export utility.

**NOTE**

Export is available only to those who have a license and password for the Database Performance for DB2 solution.

For more information, see Chapter 11, “Exporting and deploying product definitions” and Appendix D, “Using the UIM Server.”

- DASD MANAGER PLUS integrates STATS API (the BMC Common Statistics component) in BMCSTATS. STATS API resides in its own database and contains the statistics repository tables for DASD MANAGER PLUS. This component provides the following benefits to BMCSTATS and to the other BMC utility products that use it:
  - improves consistency of DB2 statistics
  - improves product performance
  - improves serviceability
  - supports enhanced statistics in DB2 Version 9

- DASD MANAGER PLUS can now use the Performance Advisor feature of the SQL Performance for DB2 products to provide more robust database performance evaluations. Performance Advisor is available in version 6.2.00 or later of SQL Performance for DB2. For more information, see Appendix E, “Obtaining enhanced database performance evaluation using SQL Performance for DB2.”

- Space estimation is now available for partition-by-growth table spaces. The following reports now include partition-by-growth information:
  - Table Space and Index Storage Limits report
  - Space Estimation report
  - Database Space Trend report
  - Space Estimation Trend report
- A new worklist command, -SQLM, supports multiple ALTER statements within a single -SQL command. Including multiple statements can help reduce the total number of steps in the worklist and improve worklist performance. For more information, see the DASD MANAGER PLUS for DB2 Reference Manual.

- The -CHEK worklist command now invokes the DB2 CHECK LOB utility to support LOB table spaces. For more information, see the DASD MANAGER PLUS for DB2 Reference Manual.

- The following statistics tables have been renamed and no longer contain the version and release number Vnn (for example, V91) in their names. However, the version is still included in the name as represented by nn in the following table. (For example, BMCASU93 would be BMCAS93 for version 9.3.) In addition, all statistics table names have changed from BMCASU to BMCATS (for example, BMCATS11.RS_COLDIST). For more information, see the DASD MANAGER PLUS for DB2 Reference Manual.

**Updated tables**

<table>
<thead>
<tr>
<th>BMCATSnn.RS_COLDIST</th>
<th>BMCASU93.ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCATSnn.RS_COLDISTSTAT</td>
<td>BMCASU93.ACT_SEQ_SVC</td>
</tr>
<tr>
<td>BMCATSnn.RS_COLSTATS</td>
<td>BMCASU93.DO_WORKIDS</td>
</tr>
<tr>
<td>BMCATSnn.RS_COLUMNS</td>
<td>BMCASU93.EVENTS</td>
</tr>
<tr>
<td>BMCATSnn.RS_INDEXES</td>
<td>BMCASU93.SERVICE</td>
</tr>
<tr>
<td>BMCATSnn.RS_INDEXPART</td>
<td>BMCASU93.SVC_DD</td>
</tr>
<tr>
<td>BMCATSnn.RS_INDEXPART_DIST</td>
<td>BMCASU93.SVC_DD_DSN</td>
</tr>
<tr>
<td>BMCATSnn.RS_STOGROUP</td>
<td>BMCASU93.SVC_SNTX</td>
</tr>
<tr>
<td>BMCATSnn.RS_TABLEPART</td>
<td>BMCASU93.SVC_SNTXT</td>
</tr>
<tr>
<td>BMCATSnn.RS_TABLES</td>
<td>BMCASU93.WL_SYNC</td>
</tr>
<tr>
<td>BMCATSnn.RS_TABLESPACE</td>
<td>BMCASU93.CORR_ACTS</td>
</tr>
<tr>
<td>BMCATSnn.RS_TSPART_DIST</td>
<td>BMCASU93.EXCEPTIONS2</td>
</tr>
<tr>
<td>BMCATSnn.RS_VOLUMES</td>
<td>BMCASU93.EXCEPTION_DFN</td>
</tr>
<tr>
<td>BMCASU93.OBJ_SETS</td>
<td>BMCASU93.OBJ_ACT</td>
</tr>
<tr>
<td>BMCASU93.OBJSET_SPECS</td>
<td>BMCASU93.OBJ_ACT_JC</td>
</tr>
<tr>
<td>BMCASU93.OBJSET_SQL</td>
<td>BMCASU93.THRESHOLDS</td>
</tr>
</tbody>
</table>

- DASD MANAGER PLUS has added the following statistics tables:
  - BMCATSnn.RS_KEYTARGETSTATS
  - BMCATSnn.RS_KEYTARGETS

- DASD MANAGER PLUS has added following synonyms:
  - BMCASU_KEYTARGETS
  - BMCASU_KEYTARGETSTATS
DASD MANAGER PLUS now includes new and updated installation options:

— Using the FREQVAL installation option, you can indicate whether BMCSTATS should collect frequency statistics for columns when using the TABLE option.

— Using the BMCSYNC option, you can specify whether to use the BMCSYNC table for synchronizing access to DB2 spaces that are concurrently executing BMC utility products.

— When allocating a work unit, the dynamic bind feature now checks the WU option for a work unit. If no work unit is specified, dynamic bind uses SYSDA as the default.

— DASD MANAGER PLUS no longer includes the SSID installation option. The product now automatically obtains the SSID information.

For more information, see “Installation options” on page 232.

You can now view the installation options module from the Current Environment Information panel by specifying DOPTS on the COMMAND line. For batch processing, you can also produce an installation options module listing if you specify MSGLEVEL 1 in the SYSIN. For more information, see “Viewing current environment information” on page 77.

DASD MANAGER PLUS has added or updated the following parameters and keywords for BMCSTATS:

— **Freqval** (FREQVAL) indicates whether BMCSTATS should collect frequency statistics for columns when using the TABLE option.

— The product now supports space estimations or calculations involving the REORGSPACE or REORGSPACE_KB statistic for partition-by-growth table spaces. Statistics reports display space estimations for these fields.

DASD MANAGER PLUS now supports compressed indexes.

DASD MANAGER PLUS now supports DECFLOAT columns.
You can now specify CLONE as a keyword to process cloned objects for the following DB2 utilities:

— CHECK DATA
— COPY
— MERGECOPY
— MODIFY RECOVERY
— QUIESCE
— REORG
— REPAIR
— RECOVER/REBUILD
— UNLOAD

You can now specify CLONE as a keyword to process cloned objects for the following BMC utilities:

— BMC COPY
— BMC COPY IMAGECOPY
— BMC MODIFY

DASD MANAGER PLUS has added LOB support for the following DB2 utility features:

— CHECK DATA: The LOBERROR keyword has been added. CHECK DATA is not allowed for LOB table spaces and is ignored.

— DSN1COPY: The LOB keyword will appear on LOB table spaces to prevent errors on non-LOB table spaces.

— REORG: The SHRLEVEL NONE, LOG YES, and UNLOAD CONTINUE keywords are required for LOB table spaces. The product ignores the Partition, Ending part, Offposlimit, Indreflimit, Rebalance, and Statistics options for LOB table spaces.

— RUNSTATS: The SHRLEVEL REFERENCE, SHRLEVEL CHANGE, REPORT YES/NO, UPDATE ALL/NONE, HISTORY, and INDEX keywords are the only keywords allowed for LOB table spaces.

— UNLOAD: This service is not allowed for LOB table spaces and is ignored.

DASD MANAGER PLUS now uses the CHEK LOB service to run the DB2 CHECK LOB utility.

Standard and custom reports now include support for XML objects.
DASD MANAGER PLUS has added XML support for the following DB2 utility features:

— CHECK DATA: The XMLERROR keyword has been added. CHECK DATA is not allowed for XML objects and is ignored.

— COPY: XML table spaces can be copied through the ONLINE generation function.

— LOAD: The STATISTICS and KEEPDICTIONARY keywords are not allowed on XML objects and are ignored.

— RECOVER: The SHRLEVEL CHANGE keyword is not allowed on XML indexes for index rebuilds and will be changed to SHRLEVEL REFERENCE.

— REORG: The REBALANCE keyword is not allowed on XML objects or partitioned base objects and is ignored.

DASD MANAGER PLUS has added Unicode support for entering hexadecimal strings for the following BMC utilities:

— LOADPLUS
— REORG PLUS
— UNLOAD PLUS

DASD MANAGER PLUS displays all non-translatable data as hexadecimal characters, enclosed in braces.

DASD MANAGER PLUS has added Unicode support for the following DB2 utilities:

— LOAD: DASD MANAGER PLUS supports entering hexadecimal strings for the CONTINUEIF clause.

— REORG: DASD MANAGER PLUS supports entering hexadecimal strings for the WHEN clause.

DASD MANAGER PLUS has added or changed support for the following COPY PLUS options:

— AUXILIARY
— CLONE
— ENCIIPHER
— REFERENTIAL
— STACK (can now accept the keyword CABINET)
— DSN1COPY (removed)
- DASD MANAGER PLUS has added or changed support for the following REORG PLUS options:
  - MAXNEWPARTS
  - LAST
  - SELECT/UPDATE/DELETE (not supported for LOB objects)

- DASD MANAGER PLUS has added or changed support for the following LOADPLUS options:
  - CHAR and VARCHAR file reference specifications (BLOBF, DBCLOBF, and CLOBF)
  - DISCARDS
  - DRAIN_WAIT
  - dynamic allocation options for LOB/XML copy data sets (DSNPAT, SMS, SMSUNIT, DATACLAS, MGMTCLAS, STORCLAS, MAXEXTSZ, UNIT, UNITCNT, VOLCNT, AVGVOLSP, SIZEPCT, THRESHLD, EXPD, RETPD, and GDGLIMIT)
  - dynamic allocation for LOB copy data sets (LOBLOCPFCPY, LOBLOCBFCPY, LOBREMPFCPY, and LOBREMBFCPY)
  - dynamic allocation for XML copy data sets (XMLLOCPFCPY, XMLLOCBFCPY, XMLREMPFCPY, and XMLREMBFCPY)
  - ENFORCE
  - ENUMROWS
  - field specification options (POSITION, CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, SMALLINT, INTEGER, BIGINT, DECIMAL, FLOAT, BINARY, VARBINARY, DATE, TIMESTAMP, TIME, ROWID, XML, EXIT, MIXED, ROUND, EXTERNAL, TRIM, TRUNCATE, and CENTURY)
— FORMAT BMCUNLOAD
— IDCDDN
— MAPDDN
— RETRY
— RETRY_DELAY
— SORTKEYS
— TOTALPAGEPCT
— UNICODE
— DELETEFILES (removed)
— DSNUTILB (removed because the product obtains the value from the installation options module)

DASD MANAGER PLUS has added or changed support for the following UNLOAD PLUS options:

— BIGINT
— BINARY
— CHAR and VARCHAR file reference specifications (BLOBF, DBCLOBF, and CLOBF)
— DDLDDN
— dynamic allocation options (OUTPUT, UNIT, DSNNAME, LRECL, VOLCNT, GDGLIMIT, STORCLAS, DATACLAS, MGMTCLAS, UNITCNT, SPACE, PCTPRIM, MAXPRIM, MAXSECD, NBRSECD, FILESZPCT, VOLUMES, DISKRETN, DISKEXPD, TRTCH, RETPD, and EXPDT)
— INLINE
— NULLCHAR
— NULLTYPE
— RECFM
— ROWSETSZ
— RTRIM
— SELECT_ELEMENT
— SNAPCOPY
— USERRECL
— VARBINARY
— XML
— XML and LOB options (DSNTYPE, DIR, and SUBSETS)
— ZONEDDECOVP
— SYSPLEX, HOME UOW, MAX UOW, Repeatable, and Number rows (removed)

DASD MANAGER PLUS now has the following changes to its minimum requirements:

— z/OS 1.7 or later
— DB2 Solution Common Code (SCC) version 1.6.00 with PTF BPJ0295
— DB2 Utilities Common Code (D2U) version 9.3.00
Introduction

This chapter contains the following topics:

Overview .............................. 49
Solution integration .......................... 50
Product benefits .............................. 52
Product features .............................. 53
System setup .................................. 60
  DB2 Support ............................... 60
  Features of DB2 that DASD MANAGER PLUS does not currently support .......................... 61
  System requirements .......................... 61
  Software requirements .......................... 61
Interaction with other products from BMC Software .............................. 62
  Administrative products for DB2 .............................. 62
  Backup and Recovery products for DB2 .............................. 62
  Utility products for DB2 .............................. 63
  Associated common components that DASD MANAGER PLUS uses .............................. 63
Release-level compatibility .............................. 64

Overview

The DASD MANAGER PLUS product integrates all of the statistics collection and analysis tools that you need to tune a DB2 system for optimal performance and storage usage. By using DASD MANAGER PLUS, you can gather and store statistics for physical objects in a DB2 production environment, analyze trends, estimate space requirements, monitor changes in the database, automate utility generation, and export object definitions to other DB2 subsystems.
DASD MANAGER PLUS provides the following functionality:

- generates BMC Software and IBM utilities and commands
- collects and manages statistics, including improved BMCSTATS capabilities
- analyzes statistical trends using data displays and graphs
- sets thresholds for statistics and reporting exceptions
- sets thresholds and corrective actions for generating utilities automatically
- reports on events, statistics, and exceptions
- estimates space requirements for new and existing objects from statistics
- facilitates deploying object definitions

Solution integration

DASD MANAGER PLUS for DB2 is also a component of the Database Performance for DB2 solution. Customers who acquire this solution benefit from the features of the following individual products and technologies, as well as from additional features that are available when one Database Performance for DB2 solution component can rely on the presence of the other components.

Product components

- DASD MANAGER PLUS for DB2
- REORG PLUS for DB2
- SNAPSHOT UPGRADE FEATURE (SUF) for DB2

Technology components

- BMCSORT
- JCL Generation and Execution
- User Interface Middleware (UIM) server
- BMC Mainframe DNA Host Services (DHS) and BMC Mainframe DNA common console, a graphical user interface
- DB2 Solution Common Code (SCC)
- DB2 Utilities Common Component (D2U)
- BMC Common Statistics (ATS)
Features

This solution provides the following features:

- helps you to determine the maintenance tasks that are required for your DB2 objects and then automate the execution of those tasks
- perform tasks from an easy-to-use graphical user interface
- enable users of all experience levels to execute and automate database maintenance tasks by using a common interface
- improve DB2 performance by using thresholds and statistics to generate reorganizations of the data only when needed

You can then perform reorganizations without disrupting application access to the data.

- enable users to use the Export utility to copy object definitions residing on a local controlling DASD MANAGER PLUS repository to destination DASD MANAGER PLUS repositories on other DB2 subsystems
- determine when an exception condition warrants a corrective action
- instruct REORG PLUS to use the DASD MANAGER PLUS exceptions table to determine whether an object should be reorganized (by using the value BMC on the CONDEXEC command option)

Using the exceptions table gives REORG PLUS an expanded set of conditions to determine whether the reorganization is needed. For information about the CONDEXEC command option, see the REORG PLUS for DB2 Reference Manual.

- enhance the performance of REORG PLUS (by using the value BMCSTATS on the ANALYZE command option)

In the ANALYZE phase, REORG PLUS uses the statistics already gathered by BMCSTATS instead of gathering the statistics itself. For information about the ANALYZE command option, see the REORG PLUS for DB2 Reference Manual.
Product benefits

DASD MANAGER PLUS provides the benefits as shown in Table 1:

Table 1  Product benefits

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>monitoring and managing of database performance</td>
<td>With BMCSTATS, you can perform the following tasks:</td>
</tr>
<tr>
<td></td>
<td>■ update history tables, for example, to show trends</td>
</tr>
<tr>
<td></td>
<td>■ update DB2 real-time statistics tables</td>
</tr>
<tr>
<td></td>
<td>■ update statistics by their types</td>
</tr>
<tr>
<td></td>
<td>■ use DB2 security exits</td>
</tr>
<tr>
<td></td>
<td>■ integrate with other BMC utilities</td>
</tr>
<tr>
<td>flexible control of database performance</td>
<td>DASD MANAGER PLUS provides object sets, which are enhanced filters. Object sets enable you to perform one action on a group of objects, for instance, by using patterns for including and excluding objects from the object set specification and specifying SQL text. For more information, see Chapter 5, “Using object sets.”</td>
</tr>
<tr>
<td>improved problem detection and maintenance activities</td>
<td>DASD MANAGER PLUS and the BMCTRIG utility provide exceptions to monitor for threshold violations. You can also define your own exceptions and thresholds.</td>
</tr>
<tr>
<td></td>
<td>You can assign priorities so that certain problems are addressed more quickly than others. You can group objects so that a single action runs on all of them. You can balance workloads across jobs. For more information, see Chapter 8, “Analyzing objects by using BMCTRIG.” For a list of the exceptions that DASD MANAGER PLUS provides, see the DASD MANAGER PLUS for DB2 Reference Manual.</td>
</tr>
<tr>
<td>increased productivity</td>
<td>BMCTRIG supports system-wide thresholds. You can also define a service (user-defined program or utility) so that you no longer have to use a limited number of utilities or programs, but can customize them to suit your needs. Instead of having separate jobs to perform different kinds of analysis, you can now run one BMCTRIG job that detects multiple types of problems.</td>
</tr>
<tr>
<td>deploying object definitions</td>
<td>With the Export utility, you can export changes to definitions residing on a local DASD MANAGER PLUS repository to one or more DB2 subsystems residing within or across sysplexes. For more information, see Chapter 11, “Exporting and deploying product definitions.”</td>
</tr>
</tbody>
</table>
Also, you can use DASD MANAGER PLUS to solve some common problems in a DB2 production environment as shown in Table 2:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>You want to track growth and changes to an application on a weekly basis.</td>
<td>Set up a BMCSTATS utility for all objects in the application and run the job weekly. You can use wildcard characters in the object names. You can also update the DB2 catalog with the new statistics by specifying Update DB2 Catalog = Yes.</td>
</tr>
<tr>
<td>You want to project DASD requirements based on application growth estimates of 30 percent.</td>
<td>Set up a BMCSTATS utility to gather statistics on all objects in the application. Run the DASD MANAGER PLUS space estimation function by using these current BMCSTATS statistics.</td>
</tr>
<tr>
<td>You want to find all table spaces and indexes that are in more than 50 extents.</td>
<td>Set up a BMCTRIG utility job and specify all table spaces (%.%), Index = Yes, and Extents = 50. To ensure that object statistics exist or that the statistics are current, add BMCSTATS before BMCTRIG in this utility job.</td>
</tr>
<tr>
<td>You want to improve application response times that are suffering due to lack of access to indexes.</td>
<td>Set up a BMCTRIG utility job to check all indexes in the application. Set the PctClus parameter to 80 percent. Then, set up a corrective action (with no object names specified) that includes a reorganization step. The BMCTRIG job automatically generates a reorganization of all table spaces with indexes that have a clustering percentage equal to or less than 80.</td>
</tr>
</tbody>
</table>

**Product features**

DASD MANAGER PLUS offers the features that this section describes.

**Generating utilities**

The DASD MANAGER PLUS online utility generation function has the following capabilities:

- generates multiple utilities in worklist format for convenient execution
- generates maintenance utility jobs that give DB2 utilities, BMC Software utilities, and user-defined programs the efficiency and power of worklist commands
- provides interactive RESTART capabilities
- runs utilities for a group of related objects by using wildcard characters
- stores utility events in a date-stamped and timestamped log so that you can track all objects and the utilities that are run against them
- allows you to request dynamic allocation for most utility data sets
Collecting and managing statistics

DASD MANAGER PLUS provides a comprehensive statistics database and the BMCSTATS utility to collect the data necessary for complete trend analysis. With DASD MANAGER PLUS and BMCSTATS, you can accomplish the following tasks:

- Update DB2 history tables, update DB2 real-time statistics tables, and perform updates based on the type of statistics.
- Collect statistics on individual or all partitions by using BMCSTATS, which provides more statistics collection than RUNSTATS or STOSPACE.
- Store statistics for all BMCSTATS runs in the DASD MANAGER PLUS historical database.
- Create batch reports to help make decisions about space allocation, reorganization, image copies, table spaces, indexes, tables, and storage groups.
- Display statistics for the first, last, and previous BMCSTATS runs.
- Display DASD MANAGER PLUS statistics with DB2 statistics for comparison.
- Optionally update fields in the DB2 catalog by using BMCSTATS and update all DB2 catalog columns that RUNSTATS updates.
- Display DASD MANAGER PLUS statistics graphs.
- Assist BMCTRIG in monitoring changes in the database.
- Use the DB2 catalog update function (BMCUPRS) to update the DB2 catalog and thereby assist in selecting access paths.
- Use the copy statistics function (BMCCPRS) to copy statistics from the DB2 catalog to DASD MANAGER PLUS tables.
- Collect the same statistics as DB2 Version 9 or later of RUNSTATS for all supported versions of DB2, including history and real-time statistics.
- Bypass updating the DB2 catalog for objects in which BMCSTATS finds a zero cardinality.
- Specify the number of frequent values to collect (for table columns, first key columns, concatenated table columns, and concatenated key columns).
- Collect the most frequently occurring values, the least frequently occurring values, or both types of values.
Provide further information about the cardinality and distribution of a column or group of columns through histogram statistics (through an invocation of DSNUTILB).

Specify whether to count the number of unique values for partial keys of concatenated key columns.

Evaluate the frequent values of partial keys.

Specify the percentage of pages to sample in collecting statistics for table spaces or indexes.

Optionally roll up the partition level statistics to the object level in cases in which not all partition statistics are available.

Specify the number of temporary data sets that DFSORT can use for sorting, and specify the device type for the sort work files when collecting column group distribution statistics.

NOTE
Column group statistics on tables do not require this specification. However, index HISTOGRAM and IXNUMQUANTILES do require it.

Collect statistics for one or more partitions without collecting statistics on the entire table space or index.

Determine whether you have the authority to collect statistics by invoking the STATAUTH security parameter.

By invoking the OPNDB2ID parameter, collect statistics even if your logon ID does not have Resource Access Control Facility (RACF®) authority to read the data set.

Setting and analyzing thresholds

The DASD MANAGER PLUS provides a threshold feature (BMCTRIG) to monitor changes in your database and automatically create maintenance jobs that you specify. BMCTRIG has the following capabilities:

- sets thresholds that monitor changes in a series of statistical collections over time
- provides direct integration with the DB2 real-time statistics catalog tables allowing you to set thresholds and initiate actions based on these statistics
- creates customized reports that show any exceptions for analysis
Product features

- Includes user-defined programs (user-defined services) or your own job steps as part of the utility job stream
- Provides some pre-defined exceptions that it detects against DB2 table spaces and indexes and supports user-defined exceptions
- Allows system-level triggers to set different threshold values for different objects in your subsystem

You can use system-level triggers in different combinations with exceptions and BMCTRIG command syntax for fine-tuned control.

- Provides the ability to assign a priority to an exception threshold

For example, you can choose to generate JCL only for exceptions above a certain priority. Also, you can choose to age the priority of exceptions to increase work priority over time.

- Provides the ability to analyze exception thresholds by category

For example, you can analyze reorganization-related exceptions and ignore copy-related exceptions.

- Supports actions, also called corrective actions, that BMCTRIG performs automatically when BMCTRIG encounters a specified exception

For example, you can create a corrective action that runs a BMCSTATS service when the product encounters a NOSTATS exception. In a single run of BMCTRIG, you can generate various corrective actions for different types of exceptions.

- Supports enhanced database performance evaluation when used with the BMC SQL Performance for DB2

- Supports object-action priorities that allow you to establish work priority for a combination of objects and actions

- Includes workload balancing intelligence in BMCTRIG to optionally balance the generated jobs based on the sizes of the objects that are included in the generated jobs

- Allows you to process groups of objects for a service during a single run instead of processing one object at a time

Combining objects also reduces the number of job steps to process when using standard JCL. This feature is also available with online generation.

- Eliminates duplicate work in generated jobs
Product features

- supports running BMCTRIG on one or more individual partitions

BMCTRIG provides the following processing options:

- detects exception conditions only
- detects exception conditions and generates utilities (corrective actions)
- generates utility actions only
- detects exception conditions and registers work (candidates)
- generates utilities (as corrective actions) for exception conditions that have been detected but not addressed
- limits corrective actions to exception conditions that exceed a priority level that you have specified
- generates only the worklist and not the JCL to save you processing time
- allows you to view and modify exceptions before generation

You can instruct BMCTRIG to stop after detecting exceptions. You can then view the exception table online, update priorities, and change the corrective action. Finally, you can instruct BMCTRIG to resume the generation process and use the modified exception data.

- excludes objects in bad status, which increases the reliability of generated work
- generates ALTER commands for related partitions and indexes before REORG commands
- provides functionality for space, performance, and copy-related exceptions

**Standard JCL option**

You can choose whether to generate jobs in worklist format or standard job control language (JCL). With standard JCL, you can generate multiple-step jobs and take advantage of JCL’s familiarity, flexibility, and easy restart capabilities.
When requesting standard JCL, you can specify the following characteristics for generated jobs:

- automatic workload balancing
- maximum number of objects per job
- grouping of objects into a single invocation of a utility
- number of jobs to be generated
- job name and other job card parameters, using the job card user exit

At the end of each service, you can delete work data sets.

**Other threshold features**

The DASD MANAGER PLUS threshold function offers the following additional features:

- allows you to choose whether to detect thresholds on archived objects
- generates REBIND commands for packages and plans that are related to table spaces or indexes for which DB2 catalog statistics are updated during threshold job execution
- evaluates space, status, and DB2 RTS-related exceptions, based on current statistics, without running BMCSTATS
- allows resizing of table space or index objects (or both) during generated REORG utility jobs
- allows you to specify that DB2 objects not be made smaller when requesting automatic resizing
- allows you to exclude table spaces that have no rows
- excludes table spaces and indexes from BMCTRIG utility generation, based on a minimum size, maximum size, or both
- skips utility generation when you specify an index object and utilities that do not run against indexes, or allows you to choose to run the utilities against the corresponding table space by using the IXESCALATE option
- allows you to benefit from restart JCL comments that describe utility restart parameters, data set dispositions, and volume refer-backs
- eliminates duplicate work
Customizing reports

DASD MANAGER PLUS provides fully documented, customizable report programs that you can run online or in batch mode. By using the report programs, you can produce reports about data in the DASD MANAGER PLUS statistical database, the IBM DB2 for OS/390® and z/OS catalog, or your own databases. You can run the reports as they are written; customize them by changing the SQL, report layout, user-defined variables, and processing logic; or you can use the external functions to write your own report programs. You can also add your own reports to the customizable report menu.

DASD MANAGER PLUS writes reports in Restructured Extended Executor (REXX), a general-purpose, interpreted language that exists on all OS/390 and z/OS systems. Because REXX is an interpreted language, you edit and run jobs instead of edit, compile, link, bind, and run jobs. For more information and examples of using REXX to create customized reports, see Appendix B, “Customizable reports.”

Analyzing trends

DASD MANAGER PLUS can assist you in performing trend analysis when tuning the DB2 environment, as follows:

- prevents performance problems by responding quickly to deteriorating trends
- determines frequency of utility execution to relieve stress on the batch window
- uses statistics graphs to compare DB2 catalog and BMCSTATS statistics
- determines from a single statistic when to reallocate a data set during reorganization and how large to make the data set
- recomputes space requirements for existing database objects when the workload changes
- assists in capacity planning of existing and future applications
- develops programs and queries against the historical database for trend analysis to suit particular installation needs
Estimating space

The DASD MANAGER PLUS space estimation feature uses statistics to estimate space requirements for existing or DB2 objects (table spaces and index spaces). You can perform what-if simulations on existing objects to recalculate space requirements when the workload changes. You can also invoke Simple Space Estimation (SSE) as a stand-alone tool from the COMMAND line, allowing you to estimate space without the pre-requisite object.

Deploying object definitions

For those who have a license and password for Database Performance for DB2 solution, you can use the Export utility to deploy object definitions.

With the Export utility, you can ensure that your DASD MANAGER PLUS object definitions match on all DB2 subsystems where DASD MANAGER PLUS resides. You can copy definitions from a local “controlling” DASD MANAGER PLUS repository to “destination” DASD MANAGER PLUS repositories on other DB2 subsystems. The subsystems can reside in the same sysplex or across sysplexes.

System setup

Review this section for recommendations and requirements before you use DASD MANAGER PLUS.

DB2 Support

Table 3 shows the versions of DB2 Universal Database (UDB) for OS/390 and z/OS that DASD MANAGER PLUS supports.

<table>
<thead>
<tr>
<th>DASD MANAGER PLUS version</th>
<th>DB2 7.1</th>
<th>DB2 8</th>
<th>DB2 9</th>
<th>DB2 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1.x</td>
<td>supported</td>
<td>supported</td>
<td>supported</td>
<td>not supported</td>
</tr>
<tr>
<td>9.3.00</td>
<td>not supported</td>
<td>supported</td>
<td>supported</td>
<td>not supported</td>
</tr>
<tr>
<td>10.1.00</td>
<td>not supported</td>
<td>supported</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td>11.1.00</td>
<td>not supported</td>
<td>not supported</td>
<td>supported</td>
<td>supported</td>
</tr>
</tbody>
</table>
Features of DB2 that DASD MANAGER PLUS does not currently support

The current release of DASD MANAGER PLUS supports the features listed in “Summary of changes” on page 28 for the generally available (GA) release of Versions 9 and 10 of DB2.

DASD MANAGER PLUS version 11.1.00 does not support the following features:

- hash objects
- space estimation for XML objects
- Unicode data in SYSIN streams
- object names in Unicode that do not translate to EBCDIC
- delimited object name entries in ISPF dialogs and batch programs
- RECOVER INDEX (but does support REBUILD INDEX)
- PARALLEL option of RECOVER
- INCURSOR, IGNOREFIELDS, DISCARDQN, and SHRLEVEL options of LOAD
- LISTDEF (but DASD MANAGER PLUS provides its own wildcard functions and object sets)
- FIELDPROCs, EDITPROCs, and VALIDPROCs for XML columns
- indexes that contain keys with random ordering

System requirements

This version of DASD MANAGER PLUS requires z/OS 1.10 or later.

Software requirements

This version of DASD MANAGER PLUS has the following requirements for additional IBM or BMC software:

- version 2.3.01 of BMCSORT
- version 11.1.00 of the DB2 Utilities Common Code (D2U)
- version 11.1.00 of the DB2 Solution Common Code (SCC)
- version 11.1.00 of the BMC Common Statistics Code
- version 5.6.00 of XBM or SUF
Interaction with other products from BMC Software

In addition to functioning as a component of the Database Performance for DB2 solution, you can use DASD MANAGER PLUS with the Administrative Products for DB2 and the Utility Products for DB2 from BMC Software.

Administrative products for DB2

BMC Software offers the following Administrative products:

- ALTER and CHANGE MANAGER for DB2
- CATALOG MANAGER for DB2

DASD MANAGER PLUS and CATALOG MANAGER share some functions with each other and with ALTER and CHANGE MANAGER, as follows:

- ALTER and CHANGE MANAGER can access the DASD MANAGER PLUS BMCSTATS and space estimation functions. They can also access the EVENTS table.

- CATALOG MANAGER has commands for accessing the DASD MANAGER PLUS BMCSTATS, BMCTRIG, and space estimation functions, as well as the EVENTS table and statistics displays.

- ALTER and CHANGE MANAGER can access the CATALOG MANAGER audit function and drop recovery logs.

For a discussion on the interaction between these products, see Chapter 2 of the DASD MANAGER PLUS for DB2 Reference Manual.

Backup and Recovery products for DB2

BMC Software offers the following Backup and Recovery products for DB2:

- COPY PLUS
- RECOVER PLUS for DB2
- RECOVERY MANAGER for DB2
Utility products for DB2

DASD MANAGER object sets are stored in a common set of DB2 tables, enabling object sets to be shared across products. The following products can use an object set that has been defined in any of these products:

- COPY PLUS
- DASD MANAGER PLUS
- RECOVER PLUS
- RECOVERY MANAGER

Utility products for DB2

BMC Software offers the following Utility products for DB2:

- LOADPLUS
- UNLOAD PLUS
- REORG PLUS

DASD MANAGER PLUS can run the Utility products for DB2 in utility jobs that you run manually or automatically with BMCTRIG. You can integrate DASD MANAGER PLUS with the utilities, as follows:

- LOADPLUS and REORG PLUS can produce statistics and update the DASD MANAGER PLUS tables. However, LOADPLUS and REORG PLUS do not update all catalog columns. For more information, see the LOADPLUS and REORG PLUS documentation.

- UNLOAD PLUS, REORG PLUS, and COPY PLUS for DB2 can use the snapshot feature of the BMC Software EXTENDED BUFFER MANAGER (XBM) product.

Associated common components that DASD MANAGER PLUS uses

In addition to its own processing components, DASD MANAGER PLUS uses the common BMC components described in Figure 1. For more information, see the Installation System User Guide.
Release-level compatibility

DASD MANAGER PLUS is compatible with the following BMC Software products:

- LOADPLUS
- UNLOAD PLUS
- REORG PLUS
- COPY PLUS

DASD MANAGER PLUS supports all LOADPLUS, UNLOAD PLUS, REORG PLUS, and COPY PLUS command syntax that is compatible with a worklist environment.

DASD MANAGER PLUS does not support the following items:

- wildcard table space names in COPY PLUS
- dynamic allocation of output for UNLOAD PLUS
- product-specific options listed in Table 4 through Table 7

---

**Figure 1  Components that DASD MANAGER PLUS uses**

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Solution Common Code</td>
<td>BMC DB2 Solution Common Code (SCC) is a set of technologies that provide common processes for several BMC products for DB2. DASD MANAGER PLUS uses SCC technologies for such processes as setting object statuses.</td>
</tr>
<tr>
<td></td>
<td>This component is installed during DASD MANAGER PLUS installation but is maintained separately from DASD MANAGER PLUS.</td>
</tr>
<tr>
<td>DB2 Utilities Common Code (D2U)</td>
<td>D2U is a set of technologies that provides common processes for the BMC Utility products for DB2 and the DASD MANAGER PLUS product. The D2U code is automatically installed with the products that use it.</td>
</tr>
<tr>
<td></td>
<td>This component is installed during DASD MANAGER PLUS installation but is maintained separately from DASD MANAGER PLUS.</td>
</tr>
<tr>
<td>BMC Common Statistics component</td>
<td>The BMC Common Statistics component (ATS) is a technology that collects statistics and updates repository tables for DASD MANAGER PLUS.</td>
</tr>
<tr>
<td></td>
<td>This component is installed during DASD MANAGER PLUS installation but is maintained separately from DASD MANAGER PLUS.</td>
</tr>
<tr>
<td>User Interface Middleware (UIM) Server</td>
<td>The UIM server is a TCP/IP application that facilitates communication between mainframe components. The UIM Server component is installed automatically with the Database Performance for DB2 solution.</td>
</tr>
</tbody>
</table>
### Table 4 Unsupported COPY PLUS options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPYDSN</td>
<td>overrides the default names of copy data sets for the local site primary or backup or both</td>
</tr>
<tr>
<td>DSNAME</td>
<td>overrides the default names of copy data sets for both the local site and the recovery site</td>
</tr>
<tr>
<td>DSNUM PART</td>
<td>copies a partitioned table space and makes and registers copies by partition instead of by table space</td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>removes names from a wildcard expansion</td>
</tr>
<tr>
<td>RECOVERYDSN</td>
<td>overrides the default names of copy data sets for the recovery site primary or the backup or both</td>
</tr>
</tbody>
</table>

### Table 5 Unsupported LOADPLUS options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETEFILES</td>
<td>tells LOADPLUS whether to delete the SORTOUT, SORTWK, SYSUT1, and SYSERR files after the load completes successfully</td>
</tr>
<tr>
<td>DSNUEXIT</td>
<td>allows you to specify a user-written exit routine that creates user-defined variables</td>
</tr>
<tr>
<td>SKIPIX</td>
<td>tells LOADPLUS whether to skip building or updating the secondary indexes that are associated with the partitioned table space that LOADPLUS is loading</td>
</tr>
</tbody>
</table>

### Table 6 Unsupported REORG PLUS parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCHDDN</td>
<td>overrides the default ddname of the archive data set that contains discarded rows from SELECT and DELETE operations</td>
</tr>
<tr>
<td>DDLDDN</td>
<td>allows you to override the default ddname of the DDLIN data set</td>
</tr>
<tr>
<td>IDCDDN</td>
<td>overrides the default ddname of the input data set that contains the IDCAMS command statements that REORG PLUS uses to redefine the VSAM data sets</td>
</tr>
<tr>
<td>WORKDDN</td>
<td>overrides the default ddname or ddname prefix of the work data sets</td>
</tr>
</tbody>
</table>

### Table 7 Unsupported UNLOAD PLUS options (part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>specifies whether you want UNLOAD PLUS to allocate your primary and secondary unload data sets dynamically</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>tells UNLOAD PLUS to use statistics to estimate the number of rows for a specific table or partition</td>
</tr>
</tbody>
</table>
Table 7  Unsupported UNLOAD PLUS options (part 2 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMRATIO</td>
<td>Tells UNLOAD PLUS to use a compression ratio to calculate the estimated number of rows for a table space in any of the following circumstances:</td>
</tr>
<tr>
<td></td>
<td>- when HURBA is specified</td>
</tr>
<tr>
<td></td>
<td>- as a default because there are no DB2 statistics</td>
</tr>
<tr>
<td></td>
<td>- when the statistics are out of date</td>
</tr>
<tr>
<td>DELETEFILES</td>
<td>Tells UNLOAD PLUS whether to delete the SYSREC, SYSRED, SYSCNTL, and SORTWK files when the unload is unsuccessful and ON FAILURE phase TERMINATE is specified for the phase that is unsuccessful</td>
</tr>
<tr>
<td>INFILE option</td>
<td>Specifies a data set other than the DB2 table space data set as the source of the input data</td>
</tr>
<tr>
<td>VSAMDD</td>
<td></td>
</tr>
</tbody>
</table>
Getting started with the ISPF interface

This chapter contains the following topics:

Overview ................................................................. 67
  Using the main menu .................................................. 68
  Using additional panel types ....................................... 70
Using ISPF commands .................................................. 71
Navigating through the panels ...................................... 72
Using Fast Path Navigation .......................................... 75
Getting help ............................................................. 75
Using wildcard characters .......................................... 76
Long object name fields .............................................. 76
Viewing current environment information ....................... 77

Overview

The user interface for the DASD MANAGER PLUS product consists of ISPF dialog panels that comply with Common User Access (CUA) conventions. When you select DASD MANAGER PLUS from the ISPF menu, the logo panel appears. Press Enter to display the DASD MANAGER PLUS main menu (Figure 2).
Using the main menu

The main menu contains a list of options from which you can make a selection. To select an option from the menu, type the option number or a question mark (?) for information about this release in the option entry field to the left of the option list.

From the DASD MANAGER PLUS main menu, you can select

- options for generating services and managing statistics
- DB2 object sets
- support options for reporting, setting user and graphic options, and viewing the current environment
- BMCTRIG-generated job restart
- a list that describes newly supported features for the current version of the product, and other information about the product
- options for exporting object definitions

**NOTE**

Figure 3 provides a panel map, showing the primary panels that appear after you select a particular option from the main menu.
The main menu and other DASD MANAGER PLUS panels include the following types of fields:

- **Information-only fields** are static fields that you cannot edit.
- **Input fields** are dynamic fields that you can edit. For example, if these fields contain information, you can write over them. As explained in Chapter 3, “Setting up DASD MANAGER PLUS,” you can customize highlighting to distinguish input fields from information fields.

Most DASD MANAGER PLUS panels contain at least one input field. In the main menu, the **COMMAND line** and **Current SQLID** are examples of input fields. Use the **Tab** key to move between fields. To exit the DASD MANAGER PLUS main menu, press **END**.

Table 8 describes the common fields that appear on the main menu and on other DASD MANAGER PLUS panels.

### Table 8  Common information and input fields  (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Field type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 subsystem ID</td>
<td>information</td>
<td>active DB2 subsystem (SSID)</td>
</tr>
<tr>
<td>panel title</td>
<td>information</td>
<td>descriptive title for the panel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The title line of the main menu includes the product name, version, and maintenance level of DASD MANAGER PLUS that is currently installed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: If you call DASD MANAGER PLUS Customer Support for assistance, you need the version and maintenance level from the main menu panel.</td>
</tr>
<tr>
<td>panel number</td>
<td>information</td>
<td>sequence number of the panel (used when a single function spans multiple panels)</td>
</tr>
<tr>
<td>error message</td>
<td>information</td>
<td>short text for an error message</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An error message overwrites any panel number. Other types of messages, such as warnings, also appear in this space.</td>
</tr>
<tr>
<td>COMMAND line</td>
<td>input</td>
<td>line that you use to enter TSO commands</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field appears on all panels.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: Typically, you will run DASD MANAGER PLUS options from the main menu. Consequently, you will use the option entry field and input fields more frequently than the command line.</td>
</tr>
</tbody>
</table>
Using additional panel types

In addition to menus, DASD MANAGER PLUS uses the following basic types of ISPF panels:

- **A selection list** contains one or more items on which you can perform specific actions. DASD MANAGER PLUS typically displays valid actions above or below the list of items.

  To make a selection, type the abbreviation for an option in the action (Act) field beside one or more objects in the list. For example, to select an item in the list, type **S** in the Act field.

  If you select more than one item, only the first item appears on the screen. When you press END, the next item appears, and so on.

- **An edit list** is a type of selection list that contains one or more edit fields for one or more items. You can tab through the fields and edit their values. You can also use the actions on the item list.

- **A confirmation panel** lets you confirm that you want to run the requested action. Press **Enter** to confirm the action or **END** to return to the previous panel without performing the action.

Every panel for DASD MANAGER PLUS has the following common elements:

- panel title at the top center of the panel
  - error messages at the top right corner
  - subsystem ID (SSID) for the current active IBM DB2 for OS/390 subsystem in the upper left corner

---

### Table 8  Common information and input fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Field type</th>
<th>Description</th>
</tr>
</thead>
</table>
| option input field | input | field that specifies an option from the list  
  Because this is, for the most part, a numbered list, you select the option by entering its associated number or a question mark (?) to see what’s new for this release. |
| Current SQLID | input | field that sets the current SQLID  
  You can use this field if you have authorization. The default value is your TSO logon ID. |
Using ISPF commands

Most ISPF commands in DASD MANAGER PLUS work the same as they do in other ISPF applications. Table 9 describes the most commonly used ISPF commands.

Table 9  Common ISPF commands (part 1 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCZOOM (F4)</td>
<td>zooms into the contents of a long object name field</td>
</tr>
<tr>
<td></td>
<td>Many panels truncate names that are too long for the displayed fields. You</td>
</tr>
<tr>
<td></td>
<td>can use the F4 key on most of the panels to see the full names.</td>
</tr>
<tr>
<td>CANCEL</td>
<td>returns to the previous panel without saving any change that you made on</td>
</tr>
<tr>
<td></td>
<td>the current panel</td>
</tr>
<tr>
<td>DOWN (or F8)</td>
<td>scrolls the panel down</td>
</tr>
<tr>
<td></td>
<td>More: + on a panel indicates that more information is available below the</td>
</tr>
<tr>
<td></td>
<td>current line.</td>
</tr>
<tr>
<td></td>
<td>Scrolling is available on a Model 2 3270 mainframe terminal, which uses a</td>
</tr>
<tr>
<td></td>
<td>24-line by 80-column display.</td>
</tr>
<tr>
<td>END (or F3)</td>
<td>validates and processes information (the same as the Enter key)</td>
</tr>
<tr>
<td></td>
<td>In most 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</td>
</tr>
<tr>
<td></td>
<td>commands</td>
</tr>
<tr>
<td></td>
<td>For a sequence of related panels, pressing Enter validates the information</td>
</tr>
<tr>
<td></td>
<td>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 (F10 or F22)</td>
<td>scrolls the panel to the left</td>
</tr>
<tr>
<td></td>
<td>More: &lt; or MORE DATA &lt;= on a panel indicates that more information</td>
</tr>
<tr>
<td></td>
<td>is available to the left.</td>
</tr>
<tr>
<td>NEXT</td>
<td>displays the next panel in a sequence</td>
</tr>
<tr>
<td>PFSHOW</td>
<td>displays the active function keys</td>
</tr>
<tr>
<td></td>
<td>Some panels use every available line to display input variables. To display</td>
</tr>
<tr>
<td></td>
<td>all variables, enter PFSHOW OFF on the COMMAND line.</td>
</tr>
<tr>
<td>PREV (or F12)</td>
<td>displays the previous panel in a sequence</td>
</tr>
<tr>
<td>RIGHT (F11 or F23)</td>
<td>scrolls the panel to the right</td>
</tr>
<tr>
<td></td>
<td>More: &gt; or MORE DATA ==&gt; on a panel indicates that more information</td>
</tr>
<tr>
<td></td>
<td>is available to the right.</td>
</tr>
</tbody>
</table>
Navigating through the panels

In DASD MANAGER PLUS, you can navigate through the panels to perform various tasks. You can access all product functionality from the main menu, as summarized in Table 10.

Table 9  Common ISPF commands (part 2 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLIT (or F2)</td>
<td>divides the panel and displays the ISPF Primary Option Menu in the new panel</td>
</tr>
<tr>
<td></td>
<td>If you start the product on both panels, ensure that each product is at</td>
</tr>
<tr>
<td></td>
<td>the same version, release, and maintenance level.</td>
</tr>
<tr>
<td>SSE</td>
<td>starts the BMC Simple Space Estimation (SSE) feature to estimate</td>
</tr>
<tr>
<td></td>
<td>space requirements for table space or index objects</td>
</tr>
<tr>
<td>SWAP (or F9)</td>
<td>switches from one split panel to another</td>
</tr>
<tr>
<td>UP (or F7)</td>
<td>scrolls the panel up</td>
</tr>
<tr>
<td></td>
<td><strong>More:</strong> - on a panel indicates that more information is available above</td>
</tr>
<tr>
<td></td>
<td>the current line.</td>
</tr>
<tr>
<td></td>
<td>Scrolling is available on a Model 2 3270 mainframe terminal, which</td>
</tr>
<tr>
<td></td>
<td>uses a 24-line by 80-column display.</td>
</tr>
</tbody>
</table>

Table 10  Navigation from the main menu options (part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility Actions (WORKIDs)</td>
<td>Make a selection and enter an action name or search criteria. Type an</td>
</tr>
<tr>
<td></td>
<td>action in the action name field and press Enter. A list of actions</td>
</tr>
<tr>
<td></td>
<td>matching your entry appear, and you can edit them. You can also create</td>
</tr>
<tr>
<td></td>
<td>and edit action services.</td>
</tr>
<tr>
<td>Services (Utilities)</td>
<td>View a list of available services from which to select and edit, or</td>
</tr>
<tr>
<td></td>
<td>create a new service.</td>
</tr>
<tr>
<td>Object Sets (Filters)</td>
<td>Make a selection and enter an object set name or search criteria. If</td>
</tr>
<tr>
<td></td>
<td>you set up an action that you want to apply only to certain objects or</td>
</tr>
<tr>
<td></td>
<td>object types, you can specify an object set.</td>
</tr>
<tr>
<td>Thresholds, Corrective Actions, and</td>
<td>Define exception condition thresholds and the corrective actions for</td>
</tr>
<tr>
<td>Priorities</td>
<td>system-level triggers. You can view and modify active exceptions and</td>
</tr>
<tr>
<td></td>
<td>edit object action priorities.</td>
</tr>
<tr>
<td>Statistics/Catalog Update/Space</td>
<td>Specify the qualified name of the object or pattern for the object and</td>
</tr>
<tr>
<td>Estimation List</td>
<td>press Enter. You can select DB2 objects and display statistics from</td>
</tr>
<tr>
<td></td>
<td>the DB2 catalog and BMC statistics tables, or exception tables. After</td>
</tr>
<tr>
<td></td>
<td>running BMCSTATS, this option can also be used to perform space</td>
</tr>
<tr>
<td></td>
<td>estimation against existing objects.</td>
</tr>
</tbody>
</table>
Figure 3 shows how to access the primary panels (numbered options) for DASD MANAGER PLUS by using the main menu. You can also type the question mark (?) to list the newly supported features for the current version of the product.

<table>
<thead>
<tr>
<th>Option</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reports</td>
<td>Type a report option and press Enter. You can generate and view reports on the data</td>
</tr>
<tr>
<td></td>
<td>accumulated in the DASD MANAGER PLUS database and the DB2 catalog. You can also</td>
</tr>
<tr>
<td></td>
<td>customize many of these reports.</td>
</tr>
<tr>
<td>User Options</td>
<td>Select any of the following options for customization:</td>
</tr>
<tr>
<td></td>
<td>- General Options</td>
</tr>
<tr>
<td></td>
<td>- JCL generation options</td>
</tr>
<tr>
<td></td>
<td>- Graphic Display options</td>
</tr>
<tr>
<td></td>
<td>- Current environment information</td>
</tr>
<tr>
<td>BMCTRIG Generated Job</td>
<td>Specify RESTART or STARTOVER parameters, generate JCL, and optionally edit and submit</td>
</tr>
<tr>
<td>Restart</td>
<td>JCL. (This option is for worklist JCL only.)</td>
</tr>
<tr>
<td>Export Definitions</td>
<td>Select definitions for actions, services, object sets, object-action priorities,</td>
</tr>
<tr>
<td></td>
<td>and exceptions for deployment to other DB2 subsystems. You can also set up and</td>
</tr>
<tr>
<td></td>
<td>manage destination connection information.</td>
</tr>
<tr>
<td>About This Release</td>
<td>Review newly supported features for the current version of the product.</td>
</tr>
</tbody>
</table>
Navigating through the panels

Figure 3   Organization of DASD MANAGER PLUS panels
Using Fast Path Navigation

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

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>CATALOG MANAGER</td>
<td>BMCCAT</td>
</tr>
<tr>
<td>CHANGE MANAGER</td>
<td>BMCCCHG</td>
</tr>
<tr>
<td>DASD MANAGER PLUS</td>
<td>BMCDASD</td>
</tr>
</tbody>
</table>

For example, if you are currently using DASD MANAGER PLUS and want to view an object description in CATALOG MANAGER, enter **BMCCAT** on the DASD MANAGER PLUS **COMMAND line**. The main menu for the requested product is displayed. In this case, the DASD MANAGER PLUS session is temporarily suspended and then resumed when you exit CATALOG MANAGER.

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

Using space estimation

Space estimation enables you to determine the amount of space that a table space or index will require based on data structure definitions and their estimated usages. You can obtain space estimates by using menus within DASD MANAGER PLUS, or by using the Simple Space Estimation (SSE) stand-alone feature from the **COMMAND line**. To initiate space estimation, enter **SSE** on the **COMMAND line** of DASD MANAGER PLUS, CATALOG MANAGER, or CHANGE MANAGER. For more information, see Chapter 7, “Analyzing statistical trends.”

Getting help

For information about how to use a panel, press **F1**.
Using wildcard characters

You can use wildcard characters to search for and to list DB2 and DASD MANAGER PLUS objects. Table 12 describes the wildcard characters that DASD MANAGER PLUS supports.

Table 12  Supported wildcard characters

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

NOTE

Using wildcards in an object name search can potentially fail to match any characters unless you end the string with a wildcard. The search can fail if the name is stored in DB2 with trailing blanks. For example, when trying to match database name QZUD16, specifying *ZUD16 might not return any data, but specifying *ZUD16* will produce a match.

Long object name fields

Many panels truncate long object names due to the restricted space on the ISPF panel. You can control what part of the long name displays on the panel through the Truncation location field on the General Options panel (for more information, see “Setting general user options” on page 93). To view (and where applicable, edit) the contents of a long object name field, many of the panels also include a zoom feature. To use the zoom feature, specify Z in the Act field (where indicated) or position the cursor over the desired field and press F4 (BMCZOOM) on the ISPF panel.
Viewing current environment information

To display the information about the current DASD MANAGER PLUS environment, select User Options from the main menu, then Current environment information. The DASD MANAGER PLUS Plans & Synonyms panel displays the information (Figure 4).

**Figure 4  DASD MANAGER PLUS plans and synonyms (part 1 of 2)**

<table>
<thead>
<tr>
<th>Plan Names</th>
<th>Table or View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility Generation: ASUQA.ASUADAQDJ</td>
<td>Execution (Public): ASUQA.ASUADAQXM</td>
</tr>
<tr>
<td>Statistics . . . : ASUQA.ASUADAQDS</td>
<td>Execution (Auth) : ASUQA.ASUADAQXA</td>
</tr>
<tr>
<td>Statistics Display: ASUQA.ASUADAQDZ</td>
<td>Execution (Install): TISINSTL</td>
</tr>
<tr>
<td>Report . . . . . : ASUQA.ASUADAQDR</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Synonym</th>
<th>Table or View</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_BMCSYNC</td>
<td>BMCUTIL.CMN_BMCSYNC</td>
</tr>
<tr>
<td>BMC_BMCUTIL</td>
<td>BMCUTIL.CMN_BMCUTIL</td>
</tr>
<tr>
<td>BMCASU_ACT_SEQ_SVC</td>
<td>ASU102.ACT_SEQ_SVC</td>
</tr>
<tr>
<td>BMCASU_ACTION</td>
<td>ASU102.ACTION</td>
</tr>
<tr>
<td>BMCASU_CORR_ACTS</td>
<td>ASU102.CORR_ACTS</td>
</tr>
<tr>
<td>BMCASU_EXCEPT_DFN</td>
<td>ASU102.EXCEPTION_DFN</td>
</tr>
<tr>
<td>BMCASU_KEYTARGETS</td>
<td>ATS102.RS_KEYTARGETS</td>
</tr>
<tr>
<td>BMCASU_KEYTARGETST</td>
<td>ATS102.RS_KEYTARGETSTATS</td>
</tr>
<tr>
<td>BMCASU_LOBSTATS</td>
<td>ATS102.RS_LOBSTATS</td>
</tr>
<tr>
<td>BMCASU_OBJ_ACT</td>
<td>ASU102.OBJ_ACT</td>
</tr>
<tr>
<td>BMCASU_OBJ_ACT_JC</td>
<td>ASU102.OBJ_ACT_JC</td>
</tr>
<tr>
<td>BMCASU_SCOLDIST</td>
<td>ATS102.RS_SCOLDIST</td>
</tr>
<tr>
<td>BMCASU_SCOLDISTSTA</td>
<td>ATS102.RS_SCOLDISTSTAT</td>
</tr>
<tr>
<td>BMCASU_SCOLSTATS</td>
<td>ATS102.RS_SCOLSTATS</td>
</tr>
<tr>
<td>BMCASU_SCOLSTATS</td>
<td>ATS102.RS_SCOLSTATS</td>
</tr>
<tr>
<td>BMCASU_SINDEXES</td>
<td>ATS102.RS_SINDEXES</td>
</tr>
<tr>
<td>BMCASU_SINDEXES</td>
<td>ATS102.RS_SINDEXES</td>
</tr>
<tr>
<td>BMCASU_SINDEXPART</td>
<td>ATS102.RS_SINDEXPART</td>
</tr>
<tr>
<td>BMCASU_SIXPART_DIS</td>
<td>ATS102.RS_SIXPART_DIS</td>
</tr>
<tr>
<td>BMCASU_SKTGDIST</td>
<td>ATS102.RS_SKTGDIST</td>
</tr>
<tr>
<td>BMCASU_SKTGDISTSTA</td>
<td>ATS102.RS_SKTGDISTSTA</td>
</tr>
</tbody>
</table>
### Figure 4  DASD MANAGER PLUS plans and synonyms  (part 2 of 2)

<table>
<thead>
<tr>
<th>Plan/Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCASU_STOGROUP</td>
<td>ATS102.RS_STOGROUP</td>
</tr>
<tr>
<td>BMCASU_STABLEPART</td>
<td>ATS102.RS_TABLEPART</td>
</tr>
<tr>
<td>BMCASU_STABLES</td>
<td>ATS102.RS_TABLES</td>
</tr>
<tr>
<td>BMCASU_TABLESPACE</td>
<td>ATS102.RS_TABLESPACE</td>
</tr>
<tr>
<td>BMCASU_TABLEPART_DIS</td>
<td>ATS102.RS_TABLEPART_DIST</td>
</tr>
<tr>
<td>BMCASU_SVC_DD</td>
<td>ASU102.SVC_DD</td>
</tr>
<tr>
<td>BMCASU_SVC_DD_DSN</td>
<td>ASU102.SVC_DD_DSN</td>
</tr>
<tr>
<td>BMCASU_SVC_SNTX</td>
<td>ASU102.SVC_SNTX</td>
</tr>
<tr>
<td>BMCASU_SVC_SNTXT</td>
<td>ASU102.SVC_SNTXT</td>
</tr>
<tr>
<td>BMCASU_SVC_TABLESPACE</td>
<td>ASU102.SVC_TABLESPACE</td>
</tr>
<tr>
<td>BMCASU_SVC_TABLES</td>
<td>ASU102.SVC_TABLES</td>
</tr>
<tr>
<td>BMCASU_SVC_TABLESPACE</td>
<td>ASU102.SVC_TABLESPACE</td>
</tr>
<tr>
<td>BMCASU_SMSPART_DIS</td>
<td>ASU102.SM_SMPART_DIST</td>
</tr>
<tr>
<td>BMCASU_SMSPART_DSN</td>
<td>ASU102.SM_SMPART_DSN</td>
</tr>
<tr>
<td>BMCASU_SMSPART_SNTX</td>
<td>ASU102.SM_SMPART_SNTX</td>
</tr>
<tr>
<td>BMCASU_SVC_THRESOLDS</td>
<td>ASU102.THRESOLDS</td>
</tr>
<tr>
<td>BMCASU_VENDOR</td>
<td>ASU102.VENDOR</td>
</tr>
<tr>
<td>BMSCC_OBJSET_DEF</td>
<td>BMCUTIL.CMN_OSC_DEF</td>
</tr>
<tr>
<td>BMSCC_OBJSET_SQL</td>
<td>BMCUTIL.CMN_OSC_SQL</td>
</tr>
<tr>
<td>BMSCC_OBJSETS</td>
<td>BMCUTIL.CMN_OSC</td>
</tr>
<tr>
<td>CATAUXRELS</td>
<td>SYSIBM.SYSAUXRELS</td>
</tr>
<tr>
<td>CATCOLDIST</td>
<td>SYSIBM.SYSCOLDIST</td>
</tr>
<tr>
<td>CATCOLDISTSTATS</td>
<td>SYSIBM.SYSCOLDISTSTATS</td>
</tr>
<tr>
<td>CATCOLSTATS</td>
<td>SYSIBM.SYSCOLSTATS</td>
</tr>
<tr>
<td>CATCOLUMNS</td>
<td>SYSIBM.SYSCOLUMNS</td>
</tr>
<tr>
<td>CATCOPY</td>
<td>SYSIBM.SYSCOPY</td>
</tr>
<tr>
<td>CATDATABASE</td>
<td>SYSIBM.SYSDATABASE</td>
</tr>
<tr>
<td>CATFIELDS</td>
<td>SYSIBM.SYSFIELDS</td>
</tr>
<tr>
<td>CATINDEXES</td>
<td>SYSIBM.SYSINDEXES</td>
</tr>
<tr>
<td>CATINDEXPART</td>
<td>SYSIBM.SYSINDEXPART</td>
</tr>
<tr>
<td>CATINDEXSTATS</td>
<td>SYSIBM.SYSINDEXSTATS</td>
</tr>
<tr>
<td>CATKEYS</td>
<td>SYSIBM.SYSKEYS</td>
</tr>
<tr>
<td>CATKEYTARGETS</td>
<td>SYSIBM.SYSKEYTARGETS</td>
</tr>
<tr>
<td>CATKEYTARGETS_HIST</td>
<td>SYSIBM.SYSKEYTARGETS_HIST</td>
</tr>
<tr>
<td>CATKEYTARGETSTATS</td>
<td>SYSIBM.SYSKEYTARGETSTATS</td>
</tr>
<tr>
<td>CATKEYTGTDIST</td>
<td>SYSIBM.SYSKEYTGTDIST</td>
</tr>
<tr>
<td>CATKEYTGTDIST_HIST</td>
<td>SYSIBM.SYSKEYTGTDIST_HIST</td>
</tr>
<tr>
<td>CATKEYTGTDISTSTATS</td>
<td>SYSIBM.SYSKEYTGTDISTSTATS</td>
</tr>
<tr>
<td>CATLOBSTATS</td>
<td>SYSIBM.SYSLOBSTATS</td>
</tr>
<tr>
<td>CATLOBSTATS_HIST</td>
<td>SYSIBM.SYSLOBSTATS_HIST</td>
</tr>
<tr>
<td>CATPACKAGE</td>
<td>SYSIBM.SYSPACKAGE</td>
</tr>
<tr>
<td>CATPACKDEP</td>
<td>SYSIBM.SYSPACKDEP</td>
</tr>
<tr>
<td>CATPACKLIST</td>
<td>SYSIBM.SYSPACKLIST</td>
</tr>
<tr>
<td>CATPLAN</td>
<td>SYSIBM.SYSPLAN</td>
</tr>
<tr>
<td>CATPLANDEPE</td>
<td>SYSIBM.SYSPLANDEP</td>
</tr>
<tr>
<td>CATRELS</td>
<td>SYSIBM.SYSCRELS</td>
</tr>
<tr>
<td>CATSTOGROUP</td>
<td>SYSIBM.SYSTOGROUP</td>
</tr>
<tr>
<td>CATSYNONYMS</td>
<td>SYSIBM.SYSSYNONYMS</td>
</tr>
<tr>
<td>CATTABLEPART</td>
<td>SYSIBM.SYSTABLEPART</td>
</tr>
<tr>
<td>CATTABLES</td>
<td>SYSIBM.SYSTABLES</td>
</tr>
<tr>
<td>CATTABLESPACE</td>
<td>SYSIBM.SYSTABLESPACE</td>
</tr>
<tr>
<td>CATTABSTATS</td>
<td>SYSIBM.SYSTABSTATS</td>
</tr>
<tr>
<td>CATUSERAUTH</td>
<td>SYSIBM.SYSUSERAUTH</td>
</tr>
<tr>
<td>CATVOLUMES</td>
<td>SYSIBM.SYSVOLUMES</td>
</tr>
</tbody>
</table>

---

Figure contains a list of database objects and their descriptions. The text indicates the use of DB2 and includes terms such as `plan`, `name`, `description`, and references to specific DB2 objects like `SYSDATABASE`, `SYSCOLUMNS`, `SYSCOPY`, `SYSTEMS`, etc.
This panel shows the date of maintenance, options module, plan names, synonyms, table names, and other useful information about the product.

To display the product fixes (PTFs) for your release of DASD MANAGER PLUS and JCL Generation (JCLGEN), enter the MAINT command on the COMMAND line.

To display a list of the keywords and values in the installation options module, enter the DOPTS command on the COMMAND line. This list does not include the installation option keywords that have been replaced by keywords in the product options file (POF). You can assemble the installation options in the session to create an additional installation options module.
Chapter 3  Setting up DASD MANAGER PLUS

This chapter contains the following topics:

Before you begin ................................................................. 82
Using options to control your environment ........................................ 82
   Using the installation options .............................................. 83
   Using the user options .................................................... 84
   Using the product options ................................................. 85
   Using local keywords to override options .................................. 86
   Putting it all together ...................................................... 87
Allocating data sets .................................................................. 88
Naming objects ....................................................................... 88
Setting up execution and security .................................................. 89
   Access to DASD MANAGER PLUS plans .................................. 89
   Execution component plans ............................................... 90
   Access to the DB2 catalog .................................................. 91
   Access to the DASD MANAGER PLUS database ...................... 92
   Access to other databases ............................................... 92
Setting options and authorizations ................................................. 92
   Setting general user options .............................................. 93
   Setting the JCL options for job cards ................................... 95
   Setting the JCL options for STEPLIBs .................................. 97
   Setting the JCL options for static data sets ............................. 99
   Setting the JCL options for tapes ...................................... 104
   Setting the JCL options for temporary work data sets ................. 105
   Setting the JCL options for permanent data sets ...................... 108
   Setting the JCL Generation GDG options .............................. 112
   Setting the JCL debugging, display, and Execution options ......... 114
   Setting the JCL utility installation options module name options .... 117
   Setting the online reorg options ........................................ 118
   Setting the non-worklist JCL options ................................. 119
   Setting the BMCTRIG Control-M options ............................ 122
   Setting user variables .................................................... 124
   Creating a user POF ...................................................... 125
   Updating a user POF ...................................................... 126
   Using multiple POFs ..................................................... 127
   Refreshing the initial POF ............................................... 128
Before you begin

Before you set up the DASD MANAGER PLUS product for the Interactive System Productivity Facility (ISPF) interface, ensure that you have completed the installation process. For more information, see the Administrative Products for DB2 Installation Guide.

Using options to control your environment

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

DASD MANAGER PLUS 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
DASD MANAGER PLUS generates JCL, SQL, IBM DB2 commands, and utilities to manage DB2 objects. You can specify the default data set names that DASD MANAGER PLUS generates into job streams and you can specify other default values to appear on various utility option panels. Your ISPF profile stores these values, so you do not have to enter them each time you run DASD MANAGER PLUS.

Typically, the person who installs DASD MANAGER PLUS sets default values for user options. For information about establishing installation option values at installation, see the installation guide. For more information about installation options, see “Descriptions of the installation options” on page 234 of this guide.

---

**NOTE**

For more information about user options for graphic display, see “Graphing statistics” on page 428. For information about the option for displaying the maintenance and current environment, see “Long object name fields” on page 76.

You can also set the Graphic Display Options (from the User Options panel) to determine how statistics graphs will look. Chapter 7, “Analyzing statistical trends,” describes this task fully.

---

### 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 DASD MANAGER PLUS use the global values that are stored in the installation options module to determine how to process information. The OS/390 and z/OS Installation System generates the installation options module when you install DASD MANAGER PLUS. The module contains an assembly-language program with an options macro.

You can customize the installation of DASD MANAGER PLUS for all users of the installation by editing the default values in the installation options module. The default name of the DASD MANAGER PLUS installation options module is ASUDOPTS, but you can change this name during installation. The source of the installation option modules is located in the HLQ.UDBCNTL data set. HLQ identifies the high-level qualifier that you specify when you install the products. See “Installation options” on page 232 for a sample installation options module with keyword descriptions.

---

**NOTE**

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

Many options that were previously stored in the ASUDOPTS module now come from the product options file (POF). For more information, see “Using the product options” on page 85.
Using the user options

The first time that you run DASD MANAGER PLUS, the product copies the values from the installation options module into your ISPF profile. The values in this profile are local values 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 products use the following input streams:

- AEXIN is used by the Execution component.
- AJXIN and AJXPOFIN are used by the Batch JCL Generation component.

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, use the JCL Generation Options panels (under User Options on the DASD MANAGER PLUS main menu). You can use literal characters or symbolic variables to specify the values on the options panels. (For more information about symbolic variables, see Table 31.)

To save your user options in your profile, you must exit the product. If your ISPF session abnormally ends, you lose your modifications to the user options.

Refreshing the user options

You can refresh the values in your user options by editing and reassembling the installation options module. To refresh an option value in all existing ISPF and user profiles, type a comma and an R after the option value and then enclose the value in parentheses. The following example specifies a subsystem ID:

```
DB2CAT=('DB2CAT',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 information about overriding options, see “Overriding options” on page 147.

For more information about refreshing user options, see the installation guide.
Using the product options

The POFDS parameter in the installation options module, specifies an 80-character sequential file. This file, the product options file (POF), contains parameters and values for the JCL Generation options. The POF is built during the installation of the products. The file is located in the HLQ.UCBCNTL 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. With DASD MANAGER PLUS, you can create action POFs (also called user POFs) and associate their values with specific actions. For more information about action POFs, see “Using action POFs to reset JCL options” on page 149.

In addition, the installation system will use the same application ID (or profile) for the products in the BMCDB2 CLIST. This single application ID enables the JCL Generation options to be shared with other products, such as the BMC CATALOG MANAGER for DB2 product. Thus, when you specify an option for generating JCL in one product, your selection applies to all of the products. Although BMC recommends that you use a single application ID, you can choose individual product application IDs on the BMCDB2PR panel.

**NOTE**
If the value of 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 version of the product that has recently been installed. For example, if the value of the BMC_COPY_LOAD keyword in a user POF is BMC8100.ACM.D81.LOAD, when you install version 9.1.01 of the product, update the value of the keyword to a version 9.1.01 LINK library (BMC9101.ACM.D91.DBLINK).

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

- The first time that you run a product, it sets all of the values that are in the ISPF profile to the values in the initial POF. If you have not specified a POF, the product uses default values from the variables in the profile.

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

- If you specify a new initial POF in the POFDS installation option, the values in the POF that are marked with refresh ,(R) are used to reset the ISPF variables. The value of the POFDATE keyword in the new initial POF is saved in the ISPF profile.
For more information about POFs, see “Using action POFs to reset JCL options” on page 149.

Using local keywords to override options

You can override DASD MANAGER PLUS user options for an action by using the override options from the JCL generation panels or by specifying a JCL options DSN in the action. If you use the Override Options panels when you generate JCL, the product uses the options that you specify only for the current JCL generation. If you specify a JCL options DSN in the action, the product always uses the options that you specify when you generate JCL for that action. For more information about using override options or an action POF, see “Overriding options” on page 147.

NOTE

If you are creating an action and you do not specify a data set name in the JCL Options DSN field, the product uses the following files and variables when generating JCL online or in batch:

- (batch generation) the default POF
- (online generation) the data set name and JCLGEN variables that you specified in your user profile
Putting it all together

Figure 5 illustrates how DASD MANAGER PLUS uses the installation options module, product options (POF), user options, and override options.

**Figure 5  Functions of options**

- **Installation options module**: Generated at installation.
- **Product options file (POF)**: Generated at installation.
- **User POF**: Created by user.
- **ISPF profile**: Initialized by installation options module and POF when the product is invoked.
- **Modified by product user options**: Moved by product user options.
- **Modified copy of ISPF profile**: Modified by component override options.
- **Input streams**: Edited in JCL or edited directly.
Allocating data sets

When you use DASD MANAGER PLUS, you specify data sets for JCL, diagnostic output, worklist files, and report files. When you specify these data sets, use the guidelines in Table 13.

**NOTE**

You must predefined partitioned data sets (PDSs). If a member of a PDS does not exist, the product dynamically allocates it.

---

### Table 13 Guidelines for data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>Type</th>
<th>Record format</th>
<th>Record length</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCL</td>
<td>sequential file</td>
<td>fixed length (FB)</td>
<td>80 characters</td>
</tr>
<tr>
<td>PDS member</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>diagnostic output</td>
<td>sequential file</td>
<td>variable length (VB)</td>
<td>137 characters</td>
</tr>
<tr>
<td>worklist</td>
<td>sequential file</td>
<td>fixed length (FB)</td>
<td>80 characters</td>
</tr>
<tr>
<td>PDS member</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>report</td>
<td>sequential file</td>
<td>fixed length (FBA)</td>
<td>80 characters</td>
</tr>
<tr>
<td>PDS member</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Naming objects

You must name a DASD MANAGER PLUS object when you create it. DASD MANAGER PLUS actions, services, and service syntax have one-part names. Actions and service syntax have a limit of 20 characters. Services have a limit of eight characters. None of them can use embedded spaces. For more information about rules for setting the owner ID, see “Controlling access to actions, services, and service syntax” on page 307.

Object sets have a two-part name or owner ID (owner.identifier). The owner can be from one to eight characters. The identifier can be from one to 18 characters. The two parts have the same restrictions as nondelimited DB2 identifiers, so you cannot use special or double-byte character set (DBCS) characters. If you do not explicitly specify the owner, the owner defaults to your logon ID. For more information about rules for setting the owner ID, see “Controlling access to object sets” on page 340.
Setting up execution and security

The DASD MANAGER PLUS Execution program and its Execution Monitor program manipulate your system’s catalog data because they run SQL. Use caution when you integrate these programs into your security environment. This section provides information for setting up proper safeguards to use the Execution and the Execution Monitor programs.

Access to DASD MANAGER PLUS plans

You can grant access to some features of DASD MANAGER PLUS by controlling authorization to various DASD MANAGER PLUS plans. Table 14 lists the plans that the functions in DASD MANAGER PLUS use. In the plan names, \( vr \) represents the version and release.

<table>
<thead>
<tr>
<th>Plan name</th>
<th>Function name</th>
<th>Plan description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASU( vr )DJ</td>
<td>BMCTRIG Utility Job Generation</td>
<td>controls access to utility-job generation from BMCTRIG. Any user who needs to perform online or BMCTRIG JCL generation should be authorized to use this plan.</td>
</tr>
<tr>
<td>ASU( vr )DR</td>
<td>Report Display</td>
<td>controls access to displaying reports. Any user who needs to report events and exceptions online should be authorized to use this plan.</td>
</tr>
<tr>
<td>ASU( vr )DS</td>
<td>Statistics Collection DB2 Catalog Update Exception, Corrective Action, and Analysis</td>
<td>controls access to statistics collection and to the operations that update the catalog. Any user who needs to run BMCSTATS or who needs to run BMCTRIG to evaluate objects should be authorized to use this plan.</td>
</tr>
<tr>
<td>ASU( vr )DZ</td>
<td>Browse DASD MANAGER PLUS Database Statistics</td>
<td>controls access to the Browse function, which displays statistics from the DASD MANAGER PLUS databases. Any user who needs to display statistics online should be authorized to use this plan. You can restrict the online programs to limit a user to defining objects and specifying and analyzing changes. Consider placing this minimum restriction on the online programs but restricting authorization to run the Execution plans. Doing so allows you to control which users can run changes.</td>
</tr>
</tbody>
</table>
You can restrict access to functions of the Execution component by using PLAN authorizations. Table 15 lists the plans that the Execution component uses in ALTER, CHANGE MANAGER, and DASD MANAGER PLUS.

<table>
<thead>
<tr>
<th>Plan name</th>
<th>Function name</th>
<th>Plan description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEXvHA (ALTER and CHANGE MANAGER)</td>
<td>Execution Monitor Entry (Authorization)</td>
<td>enables users to execute a worklist when EXECUTE authority is granted. You should carefully consider who receives authorization to use this plan.</td>
</tr>
<tr>
<td>AEXvDA (DASD MANAGER PLUS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEXvHM (ALTER and CHANGE MANAGER)</td>
<td>Execution Monitor</td>
<td>provides users with the ability to attach to DB2 with alternate authorization IDs for the -AUTH commands. This plan does not control who has authorization to execute a worklist. Because this plan does not affect who can run Execution, you can grant PUBLIC authority to this plan. The Execution plan contains some packages that use dynamic SQL. Some of these packages cause long-running SQL and might need to be added to your resource limit specification table (RLST). The packages are as follows:</td>
</tr>
<tr>
<td>AEXvDM (DASD MANAGER PLUS)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- AEXAUNLD unloads data from tables.
- AEXSQLIO performs all worklist -SQL commands, including deletions before a data-only migration -LOAD or -BMCL command.
- AEXESTDL performs some of the restart logic before restarting a -LOAD command, including deleting previously loaded rows.

By restricting authorization to run the Execution plans, you can control what change and migrate functions users can perform. For example, by granting unlimited access to Specification and Analysis while controlling access to the Execution Monitor Entry, you can allow your users to run ALTER for training purposes or use it as a system dictionary.

The Execution Security Exit provides further control over the Execution component’s authorization switching function.
Access to the DB2 catalog

If the authorization for the subsystem is controlled through security exits, the user ID that is running the product needs SELECT authority on the following DB2 tables:

- SYSIBM.SYSAUXRELS
- SYSIBM.SYSCOLDIST
- SYSIBM.SYSCOLDISTSTATS
- SYSIBM.SYSCOLSTATS
- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSCOPY
- SYSIBM.SYSDATABASE
- SYSIBM.SYSFIELDS
- SYSIBM.SYSINDEXES
- SYSIBM.SYSINDEXPART
- SYSIBM.SYSINDEXSTATS
- SYSIBM.SYSKEYS
- SYSIBM.SYSKEYTARGETS
- SYSIBM.SYSKEYTARGETS_HIST
- SYSIBM.SYSKEYTARGETSTATS
- SYSIBM.SYSKEYTGTDIST
- SYSIBM.SYSKEYTGTDIST_HIST
- SYSIBM.SYSKEYTGTDISTSTATS
- SYSIBM.SYSLOBSTATS
- SYSIBM.SYSLOBSTATS_HIST
- SYSIBM.SYSPACKAGE
- SYSIBM.SYSPACKDEP
- SYSIBM.SYSPACKLIST
- SYSIBM.SYSPLAN
- SYSIBM.SYSPLANDEP
- SYSIBM.SYSSRELS
- SYSIBM.SYSSTOGROUP
- SYSIBM.SYSSYNONYMS
- SYSIBM.SYSTABLEPART
- SYSIBM.SYSTABLES
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSTABSTATS
- SYSIBM.SYSUSERAUTH
- SYSIBM.SYSVOLUMES
Access to the DASD MANAGER PLUS database

Give DASD MANAGER PLUS database access only to users who must perform diagnostic functions on DASD MANAGER PLUS. You give access to the database by using DASD MANAGER PLUS functions and USE authority for plans.

Access to other databases

DASD MANAGER PLUS does not affect the requirement for a user to have proper authorization to data and structures that the product is manipulating. A user must have read access to the DB2 catalog and appropriate access to other structures as required for the execution.

NOTE
You need only read access on the Virtual Storage Access Method (VSAM) data sets that contain DB2 objects against which you run the BMCSTATS utilities.

Setting options and authorizations

Typically, the options and authorizations in this section are set during the installation process. However, to manually set your options and authorizations in DASD MANAGER PLUS, use the procedures in Table 16.

Table 16 Setup procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Setting general user options”</td>
<td>page 93</td>
</tr>
<tr>
<td>“Setting the JCL options for job cards”</td>
<td>page 95</td>
</tr>
<tr>
<td>“Setting the JCL options for STEPLIBs”</td>
<td>page 97</td>
</tr>
<tr>
<td>“Setting the JCL options for static data sets”</td>
<td>page 99</td>
</tr>
<tr>
<td>“Setting the JCL options for tapes”</td>
<td>page 104</td>
</tr>
<tr>
<td>“Setting the JCL options for temporary work data sets”</td>
<td>page 105</td>
</tr>
<tr>
<td>“Setting the JCL options for permanent data sets”</td>
<td>page 108</td>
</tr>
<tr>
<td>“Setting the JCL Generation GDG options”</td>
<td>page 112</td>
</tr>
<tr>
<td>“Setting the JCL debugging, display, and Execution options”</td>
<td>page 114</td>
</tr>
<tr>
<td>“Setting the JCL utility installation options module name options”</td>
<td>page 117</td>
</tr>
</tbody>
</table>
Setting general user options

Use this procedure to display DASD MANAGER PLUS options and to specify the default values for the DB2 SSID and the DB2 catalog. You can also specify default values for the data set names and processing options that DASD MANAGER PLUS uses when generating utility jobs.

To set general user options

1. On the DASD MANAGER PLUS main menu, select User Options and press Enter.

2. On the User Options panel, select General Options and press Enter.

NOTE
DASD MANAGER PLUS uses symbolic variables in the names of data sets, in job cards, and elsewhere. For a list of the symbolic variables, see Table 31.
The General Options panel (Figure 6) shows options that are specific to the DB2 subsystem and to generating actions.

Figure 6    General Options panel

3 Review the options and edit the default values as needed by tabbing to the relevant input fields and typing over the current values.

You can use symbolic variables to specify the data set name (DSN) options. Use HELP for option entry information.

4 Press END to save the changes and return to the DASD MANAGER PLUS main menu.
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 7):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation</td>
<td>Jobcard Options</td>
</tr>
</tbody>
</table>

![AJXOJOBP JCL Generation Jobcard Options Update panel]

Type data and press Enter.

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

Enter Jobcards below:

```plaintext
//&USERID.&JOBCHAR JOB (&ZACCTNUM),`&PGMR`,
//  CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//  NOTIFY=&USERID
/*ROUTE XEQ BMCPLX1
/*JOBPARM SYSAFF=&ZSYSID
Jcllib . . . . . .
Sysexec. . . . . .
Region size . . OM (See JCL Reference for valid options)
Memlimt . . . . . (See JCL Reference for valid options)
Time parameter . . (See JCL Reference for valid options)
System MLIB. . . . ISP.SISPMENU
Runtime HLQ. . . . AEX.QA0920
LLQ. . . . . . . . BMC
ULLQ . . . . . . (Leave blank if not using runtime enablement)
```

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

3. In the Enter Jobcards below field, type the job statement information that you want to add to the JCL.
4 In the Jcllib field, enter the name of one of the following types of PDSs:
   - PDS that contains customized JCL to be included in a job
   - PDS that specifies the cataloged procedures (PROCs) that are used for non-<br>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 execute.

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 sets the value to an HLQ for Execution.

This value supports the following symbolic variables:

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

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

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

During installation, if you chose to use the runtime enablement feature, the Installation System set this value to BMC. If you chose not to use the feature, the Installation System set the value to DB.
12 In the ULLQ field, type the low-level qualifier for user-defined data sets for the installation environment.

During installation, if you chose to use the runtime enablement feature, the Installation System set this value to blank, and you should not change the value. If you chose not to use the feature, the Installation System set the value to UDB.

13 Press END to save your changes, and to return to the DASD MANAGER PLUS main menu.

**Setting the JCL options for STEPLIBs**

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

**To set the JCL options for STEPLIBs**

1 Use the following menu selections to display the JCL Generation STEPLIB Options Update panel (Figure 8):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation</td>
<td>Steplib Options</td>
</tr>
</tbody>
</table>
Setting the JCL options for STEPLIBs

Figure 8   JCL Generation STEPLIB Options Update panel

| AJXOSTP ------------ JCL GENERATION STEPLIB OPTIONS UPDATE------------------- |
| COMMAND ===>
| Type data and press Enter.
| DSNEXIT . . . . . . . . SYS3.DBDC.DSNEXIT
| DB2 DSNLOAD . . . . . . SYS2.DB2V81M.DSNLOAD
| Override lib. . . . .
| CATALOG MANAGER . .
| ALTER/CHANGE MANAGER .
| DASD MANAGER PLUS . .
| EXECUTION . . . . . . 'BMCADMN.V920.D91.BMCLINK'
| COPY PLUS . . . . . .
| REORG PLUS . . . . .
| LOADPLUS . . . . . .
| UNLOAD PLUS . . . .
| RECOVER PLUS . . .
| CHECK PLUS . . . . .
| SQL EXPLORER . . . .
| DATA PACKER . . . .
| Additional lib . . .
| IOA LOAD 1 . . . . .
| IOA LOAD 2 . . . . .

2 Specify the data set names for the libraries:

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

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

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

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

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

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

E In the Additional lib field, type the data set name for the additional LINK library that should appear last in the STEPLIB statement.
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 9):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options panel</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Static Data Set Options</td>
</tr>
</tbody>
</table>

   **Figure 9  JCL Generation Static Data Set Options Update panel**

   AJXOSDO ------- JCL GENERATION STATIC DATA SET OPTIONS UPDATE -----------------
   COMMAND ===> Type data and press Enter.

   Data set sizing option... N   (N-No Sizing,B-Bmcstats,
   C-DB2 Catalog,O-Object Sampling)
   Data set sizing device... 3390 (3380/3390)
   Max cylinders............. 99999 (Do not exceed this primary value in JCL.)
   If max cylinders are exceeded, use the following for DASD data sets
   Max primary quantity... 10   (Cylinders, 1 - 9999)
   Max secondary quantity... 2   (Cylinders, 1 - 9999)
   Max unit count............. (Blank or 1 - 59 volumes)
   Include data set cleanup step.. N   (Y/N)
   Return code for cleanup step .. 4   (04)
   Temporary unit.............. SYSDA   (SYSDA, SYSALLDA, etc.)
   Include SYSPRIN2 DD........... N   (Y/N)

   Note See Debugging, Display and Execution Options to display sizing options in the JCL.
In the **Data set sizing option** field, type **N**, **B**, **C**, or **O** to specify the sizing method, as shown in **Table 17**:

---

**NOTE**

Whether or not data set sizing is performed, DB2 catalog access is required to resolve any symbolic variables. For more information about data set sizing, see “Sizing the data sets” on page 139.

---

### Table 17  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 5 on page 107.</td>
</tr>
<tr>
<td>sizing by using statistics from the DASD MANAGER PLUS tables</td>
<td>B</td>
<td>- The DASD MANAGER PLUS product must be installed and interacting with ALTER or CHANGE MANAGER to use this option.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If a column is defined as a LONG VARCHAR, this option averages the row size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you select this option for Batch JCL Generation, the DATASETSIZING B keyword is inserted into the AJXIN input stream.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Current BMCSTATS statistics should be available for objects in the worklist before you select this option.</td>
</tr>
<tr>
<td>sizing by using statistics from the DB2 catalog</td>
<td>C</td>
<td>- If a column is defined as a LONG VARCHAR, this option does not average the row size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you select this option for Batch JCL Generation, the DATASETSIZING C keyword is inserted into the AJXIN input stream.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Current IBM RUNSTATS catalog statistics should be available for objects in the worklist before you select this option.</td>
</tr>
<tr>
<td>estimating sizes based on physical, random sampling of VSAM data sets</td>
<td>O</td>
<td>- If a column is defined as a LONG VARCHAR, this option averages the row size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you select this option for batch JCL generation, the DATASETSIZING O keyword is inserted into the AJXIN input stream.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 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.</td>
</tr>
</tbody>
</table>
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 114.

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.

---

**NOTE**

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

---

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

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

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

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

If you want to use multiple data sets on DASD, specify a value greater than 1 for **Max unit count**. On the JCL Generation Individual Data Set Options panels (see page 108), 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.

The product deletes the data sets at the end of Execution. For Batch Execution JCL Generation, selecting this option inserts the JCLCLEANUP YES keyword into the AJXIN input stream.

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 as follows: 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 18 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 18  Work data sets in the JCL cleanup job step

<table>
<thead>
<tr>
<th>Work data set</th>
<th>ddname</th>
<th>Used in JCL cleanup by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discard (SYSDISC)a</td>
<td>SYSDnnnn</td>
<td>BMC LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Error</td>
<td>SYSERnnn</td>
<td>BMC CHECK PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BMC 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>Puncha</td>
<td>SYSPUNCH</td>
<td>BMC REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM REORG</td>
</tr>
<tr>
<td>Unload (SYSREC)b</td>
<td>SYSRnnnn or Runnnyyyyy</td>
<td>BMC REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM REORG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The unload data sets that are used by REORG PLUS and IBM REORG are deleted automatically.</td>
</tr>
<tr>
<td>Work</td>
<td>not applicable</td>
<td>utilities that are listed in Table 19</td>
</tr>
</tbody>
</table>

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

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

Table 19 lists the work data sets that are used by the corresponding utilities.

Table 19  Work data sets used by utilities

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

7 In the Return code for cleanup step field, specify the value that should be compared against the condition code that is returned from any previous job step.
You can specify any two-digit value for the return code; however, BMC recommends that you specify a value of 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 DASD MANAGER PLUS 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 10):

   **From this menu** | **Select this item and press Enter**
   --- | ---
   DASD MANAGER PLUS main menu | User Options
   User Options | JCL generation options
   JCL Generation main menu | Tape Options

   **Figure 10  JCL Generation Tape Options Update panel**

   
   AJXOTAPE ------------ JCL GENERATION TAPE OPTIONS UPDATE ---------------------
   COMMAND ===> 
   Type data and press Enter.
   
   Tape Unit 1 . . . . . . . . . CART (blank or TAPE, CART, etc)
   Tape Unit 2 . . . . . . . . . TAPE (blank or TAPE, CART, etc)
   Tape Unit 3 . . . . . . . . . TAPE (blank or TAPE, CART, etc)
   Tape Volume count . . . . . . 99 (1 - 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.
In the **Tape TRTCH** field, type the parity, data conversion, translation, and compression value for 7-track tape drives as shown in **Table 20**.

**Table 20  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></td>
</tr>
<tr>
<td>odd parity, conversion, and no translation</td>
<td>C</td>
</tr>
<tr>
<td>even parity, no conversion, and no translation</td>
<td>E</td>
</tr>
<tr>
<td>odd parity, no conversion, and translation</td>
<td>T</td>
</tr>
<tr>
<td>even parity, no conversion, and translation</td>
<td>ET</td>
</tr>
<tr>
<td>data compression</td>
<td>COMP</td>
</tr>
<tr>
<td>no data compression</td>
<td>NOCOMP</td>
</tr>
</tbody>
</table>

**NOTE**

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

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.

Press **END** to save your changes and to return to the DASD MANAGER PLUS 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**
- **DATAWORK**
- **LOGSORT**

DASD MANAGER PLUS generates the temporary work data sets (such as **SORTWORK**) by using **DISP=(,PASS)** in the JCL.
To set the JCL options for SORTWORK data sets

1. Use the following menu selections to display the JCL Generation Options for Sort Files Update panel (Figure 11):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Individual Data Set Options</td>
</tr>
<tr>
<td>JCL Generation Individual Data Set Options</td>
<td>Sortwork</td>
</tr>
</tbody>
</table>

![AJXOSWO -------- JCL GENERATION OPTIONS FOR SORT FILES UPDATE ----------------- COMMAND ===>](image)

Type data and press Enter.

<table>
<thead>
<tr>
<th>AJXOSWO -------- JCL GENERATION OPTIONS FOR SORT FILES UPDATE ----------------- COMMAND ===&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type data and press Enter.</td>
</tr>
<tr>
<td>Number of SORTWORK Data Sets.. 5        (1 - 32)</td>
</tr>
<tr>
<td>SORTWORK unit name. . . . . SYSDA     (SYSDA, 3380, etc)</td>
</tr>
<tr>
<td>Number of DATAWORK Data Sets.. 5        (1 - 32)</td>
</tr>
<tr>
<td>DATAWORK unit name. . . . . SYSDA     (SYSDA, 3380, etc)</td>
</tr>
<tr>
<td>Number of LOGSORT Data Sets.. 1        (1 - 32)</td>
</tr>
<tr>
<td>LOGSORT unit name . . . . SYSDA     (SYSDA, 3380, etc)</td>
</tr>
<tr>
<td>Default Primary Quantity . . . 20     (Cylinders)</td>
</tr>
<tr>
<td>Default Secondary Quantity . . . 2      (Cylinders)</td>
</tr>
<tr>
<td>SMS Data Class. . . . . . . . . . .     (Blank or Data Class Name)</td>
</tr>
<tr>
<td>SMS Storage Class . . . . . . . .       (Blank or Storage Class Name)</td>
</tr>
<tr>
<td>SMS Management Class . . . . . .       (Blank or Management Class Name)</td>
</tr>
<tr>
<td>$ORTPARM data set name (below)</td>
</tr>
</tbody>
</table>

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.
Setting the JCL options for temporary work 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 quantities:

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

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

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

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

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

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

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

8 Press END to save your changes and to return to the DASD MANAGER PLUS 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 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 UNLOADDDN)
- Archive (ARCHDDN)
- Cntl file (CNTLDDN)
- Discard (DISCARDDDN)
- Error (ERRDDN)
- Map (MAPDDN)
- Report
- Punch (PUNCHDDN)
- Filter (FILTERDDN)

The permanent work data sets contain data that allow restarts. The product defines work data sets 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) and copy (SYSCOPY).

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. These data sets 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 JCL Generation Data Set Options for SORTOUT Update panel (Figure 12):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Individual Data Set Options</td>
</tr>
<tr>
<td>JCL Generation Individual Data Set Options</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), and unload (SYSREC) data set options in the JCL Generation override panels do not take effect.

**Figure 12** JCL Generation Data Set Options for SORTOUT Update panel

AJXODSPL --- 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 21).

### Table 21  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) 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 (see Figure 32).

When you use the &GDG variable, JCL Generation resolves the data set name by using the symbolic variable, and the name includes the GDG number (see Figure 33).

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 opening them.

- If you have very large data sets and you want to avoid extents or multiple data sets on DASD, specify a tape, virtual tape, or cartridge unit.

- For the Error data set (SYSERnnm) 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).

- As an alternative to specifying tape for the unit, you can specify a DASD unit that spans multiple data sets. Return to the JCL Generation Static Data Set Options panel (see Figure 9), and specify a value for **Max cylinders** and a value greater than 1 for **Max unit count**.
BMC recommends that you specify **SYSDA** as 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 run in parallel.

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.

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 IBM Storage Management Subsystem (SMS) definitions for the data set classes:

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

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

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

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

   If this value is exceeded, the product uses the alternate unit and the alternate SMS parameters. Zero indicates that a threshold is not specified for the unit. If you specify zero, an alternate unit and the alternate SMS parameters will not be used.

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

---

**NOTE**

The alternate unit name must be a valid tape unit name. For more information, see “Setting the JCL options for tapes” on page 104.
10 Specify the IBM Storage Management Subsystem (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 DASD MANAGER PLUS main menu.

Setting the JCL Generation GDG 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 Generation GDG options

1 Use the following menu selections to display the JCL Generation Generation Data Group Options Update panel (Figure 13):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Generation Data Group Options</td>
</tr>
</tbody>
</table>
Setting the JCL Generation GDG options

Chapter 3 Setting up DASD MANAGER PLUS 113

Figure 13  JCL Generation Generation Data Group Options Update panel

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 when IBM COPY is used 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 the data set is uncataloged.
   - NSCR indicates to uncatalog the generation data set upon reaching the maximum number of generation data sets to keep.

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 In the Type GDG Model data set below field, type the name of the GDG model data set.
7 Press END to save your changes and return to the DASD MANAGER PLUS 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 debugging, display, and execution options

1 Use the following menu selections to display the JCL Generation Debugging, Display and Execution Options Update panel (Figure 14):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Debugging, Display and Execution Options</td>
</tr>
</tbody>
</table>

Figure 14  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. . . . . . .
Pre Job Step JCL INCLUDE member name . . . . . .
Post Step JCL INCLUDE member name . . . . . . .
Post Job JCL INCLUDE member name . . . . . . .
Include in AEXIN parameters:
SYNCEDELETE . . N (Y/N) BINDFAIL . . N (Y/N)
HASHFAIL . . . N (Y/N) HASHWARNRC . . (NUMERIC)
REBINDFAIL . . N (Y/N) REBINDRC . . (NUMERIC)
2MEGSQL . . . N (Y/N) NOFAILNOIMAGECPY N (Y/N)
STOPWAIT . . . 3 (NUMERIC)
STOPWAIT SECS . 10 (NUMERIC)
2 Specify whether to include debugging comments in the generated JCL:

---

**NOTE**

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

---

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 \textit{dsso/cc}, where \textit{dsso} is the data set sizing option and \textit{cc} is a comment code. Table 22 lists the comment codes that Execution generates in the JCL.

**Table 22** Comment codes for data set sizing

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

Figure 15 shows example comments in the Execution JCL.

**Figure 15** Execution JCL with comments (part 1 of 2)

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

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

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

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

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

**NOTE**
The alternate program is only used for nonworklist JCL.

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

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

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

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

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

For more information about the keywords, see the list of keywords in the *DASD MANAGER PLUS for DB2 Reference Manual*. 

```
//* S = SORTWK SIZE IS (WORK SPACE * 2 / #SORTWKS)
//SORTWK01 DD UNI=T SYSDA,
// SPACE=(CYL,(1,1)), ESTIMATE=C/S
// DISP=(NEW,DELETE)
```
10 Press END to save your changes and return to the DASD MANAGER PLUS 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 utility module name options**

1 Use the following menu selections to display the JCL Generation BMC Utility Option Module Names Update panel (Figure 16):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Utility Options</td>
</tr>
<tr>
<td>Utility Options</td>
<td>BMC Utility Option Module Names</td>
</tr>
</tbody>
</table>

**Figure 16  JCL Generation BMC Utility Option Module Names Update panel**

AJXOUOP --- 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 return to the DASD MANAGER PLUS main menu.
Setting the online reorg options

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Online Reorg Utility Options panel to specify the options for reorganizing table spaces.

NOTE

CATALOG MANAGER 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 17):

<table>
<thead>
<tr>
<th>From this panel</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Utility Options</td>
</tr>
<tr>
<td>Utility Options</td>
<td>Online Reorg Options</td>
</tr>
</tbody>
</table>

2. In the BMCREORG XBMID field, specify the BMC EXTENDED BUFFER MANAGER (XBM) subsystem (SSID) that the BMC REORG PLUS utility accesses.

REORG PLUS uses XBM or its XBM SNAPSHOT UPGRADE FEATURE (SUF) technology to create a snapshot of the data sets to be reorganized. ALTER and CHANGE MANAGER use this value when reorganizing a table space by using an online reorganization (SHRLEVEL CHANGE).
Setting the non-worklist JCL options

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

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

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

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

**NOTE**

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

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

4 Press END to save your changes, and to return to the DASD MANAGER PLUS 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 18):

<table>
<thead>
<tr>
<th>From this panel</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Utility Options</td>
</tr>
<tr>
<td>Utility Options</td>
<td>Non worklist JCL PROC Options</td>
</tr>
</tbody>
</table>
2 In the Use JCL Procedures (PROCS) for standard JCL? field, type Y or N to specify whether to generate a PROC name instead of the EXEC PGM= statement in non-worklist JCL.

If you specify Y, you must provide a PROC name and a STEP name for the utilities for which you want to generate JCL. 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 95). To add JCLLIBs to the concatenation, modify the AJXJCLU SLIB. Alternatively, you can include the JCLLIB or PROCLIB in your job card 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.
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_BMCLOAD_NAME** =
- **PROC_BMCCOPY_NAME** =
- **PROC_BMCUCUNLOAD_NAME** =
- **PROC_BMCUPRS_NAME** =
- **PROC_DSNUTILB_NAME** =
- **PROC_DSN1COPY_NAME** =
- **PROC_IDCAMS_NAME** =
- **PROC_IEFBR14_NAME** =
- **PROC_BMCLOAD_NAME** =
- **PROC_DSN1COPY_NAME** =
- **PROC_IEFBR14_NAME** =
- **PROC_TSO_NAME** =
- **PROC_USER_DEFINED** =
- **PROC_BMCCHECK_STEP** =
- **PROC_BMCCOPY_STEP** =
- **PROC_BMCCPRS_STEP** =
- **PROC_BMCLOAD_STEP** =
- **PROC_BMCCOPY_STEP** =
- **PROC_BMCUCUNLOAD_STEP** =
- **PROC_BMCUPRS_STEP** =
- **PROC_DSNUTILB_STEP** =
- **PROC_DSN1COPY_STEP** =
- **PROC_IDCAMS_STEP** =
- **PROC_IEFBR14_STEP** =
- **PROC_TSO_STEP** =

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

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

- **PROC_BMCCHECK_STEP** =
- **PROC_BMCCOPY_STEP** =
- **PROC_BMCCPRS_STEP** =
- **PROC_BMCLOAD_STEP** =
- **PROC_BMCCOPY_STEP** =
- **PROC_BMCUCUNLOAD_STEP** =
- **PROC_BMCUPRS_STEP** =
- **PROC_DSNUTILB_STEP** =
- **PROC_DSN1COPY_STEP** =
- **PROC_IDCAMS_STEP** =
- **PROC_IEFBR14_STEP** =
- **PROC_TSO_STEP** =
Setting the BMCTRIG Control-M 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 BMCTRIG Control-M Options panel to specify whether to generate BMC Control-M job schedule entries for jobs that do not contain IEFBR14 steps. You can also use this panel to specify the names of the input and output data sets for the entries.

To set the Control-M options for BMCTRIG

1. Use the following menu selections to display the BMCTRIG Control-M Options panel (Figure 19):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Utility Options</td>
</tr>
<tr>
<td>Utility Options</td>
<td>BMCTRIG Control-M Options</td>
</tr>
</tbody>
</table>

6. Press END to save your changes, and to return to the DASD MANAGER PLUS main menu.
2 At **Schedule only non-BR14 jobs in Control-M**, type Y or N to specify whether to generate BMC Control-M job schedule entries for jobs that do not contain IEFBR14 steps.

When the value is Y, JCL Generation generates two data sets in the JCL for the BMCTRIG utility:

- the input data set CNTLMSCH, which contains the name of the Control-M job schedule
- the output data set CNTLMOUT, which contains the job schedule name that is specified in CNTLMSCH and the z/OS job name of the jobs that BMCTRIG generates that do not contain IEFBR14 steps

3 At **Job schedule name Input data set**, type the name of the input data set (DDNAME CNTLMSCH) that contains the Control-M job schedule. The data set contains the following record:

```
JOBSCHEDULE=jobScheduleTag
```

4 At **Job schedule name Output data set**, type the name of the output data set (DDNAME CNTLMOUT) that contains the job schedule name and the job sequence number of the non-IEFBR14 jobs.

5 Press END to save your changes.
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 Update 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 Update panel (Figure 20):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>User Defined Variable Values</td>
</tr>
</tbody>
</table>

Figure 20  User Defined Variables panel

AJXOUSRV ------------ USER DEFINED VARIABLES UPDATE ------------
COMMAND ===>

Type data and press Enter.

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

NOTE: Symbolic variables cannot be input to these values

2 Specify the values for the variables.

The maximum length of a variable name is eight characters.

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

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

1 Use the following menu selections to display the JCL Generation Product Options File (POF) Functions panel (Figure 21):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Product Option File (POF) Functions</td>
</tr>
</tbody>
</table>

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

   The name can be either an existing sequential, 80-column data set or a member of a PDS.

3 In the **WRITE User POF data set from Profile Variables** field, type **Y** to write the ISPF variable values (located in the ISPF profile) to the user POF.
Updating a user POF

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

To update the user POF in the JCL Generation options panels

1. In the various options panels, specify your changes to the JCL Generation options.
2. Use the following menu selections to display the JCL Generation Product Options File (POF) Functions panel:

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Product Option 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**.

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.CLIST` 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.
Using multiple POFs

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

**Scenario 1**

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

**Scenario 2**

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

**Scenario 3**

You have different requirements for test and production data. For test data, you want to store the copies on DASD. For production data, you want to store the copies on tape. You can use separate POFs for test data and production data.

To create an additional user POF and reset the profile variables, perform the following steps:

5 Press Enter.
Refreshing the initial POF

1. Follow the steps in “Creating a user POF” on page 125 to create an additional POF.

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

To specify the use of a different user POF, from the JCL Generation Product Options File (POF) Functions panel, in the Type User POF Name below field, type the name of the data set for your customized POF.

Refreshing the initial POF

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

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

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

To refresh the initial POF, perform the following steps:

1. Edit the initial POF outside of the product.

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

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

4. Save the POF.

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

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

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

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

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

To generate the reports

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

**NOTE**

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

When you invoke the product or reset the POF, the reports are listed on the output for your TSO session.

Reusing a POF in a subsequent installation

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

1. Run the Installation System.

2. From the Install System JCL Generation File Information panel, in the **Use Existing POF to Populate the New Product Options File** field, type **Y** (Figure 22).

**Figure 22  Reusing an existing POF**

3. Enter the names of the data set and member for the existing POF (Figure 23).

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

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

For example, if you specify the data set prefix for local primary copies (Figure 24), the name of the primary copy data set resolves to the following name without modifications to the SLIB:

<SSIDname>.IC.T.ICPY.<databaseName>.<tableSpaceName>.<ddname>

For data sets that are not dynamically allocated, JCL Generation appends the ddname to the prefix to create the name of the data set.
Now, assume that your site’s DBA decides that users should not have the ability to change the data set prefix from the options panels. The DBA can override the value in the SLIB. By specifying the value for the copy data set in the AJX#DSNS SLIB (Figure 25), the DBA can uphold your site’s naming standards.

Figure 25 Changing the SLIB variable for the copy data set in AJX#DSNS (part 1 of 2)
After changing an SLIB variable, the DBA should use JCL Generation to test the changes. If the SLIB is coded correctly, the DBA must then recompile the SLIB. The DBA can use the SLIB compiler tool that is supplied with the Administrative products to compile the SLIB. For more information about testing the changes or using the SLIB compiler, see the product documentation.

**Adding steps to the JCL**

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, perform the following steps:

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

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS main menu</td>
<td>User Options</td>
</tr>
<tr>
<td>User Options</td>
<td>JCL generation options</td>
</tr>
<tr>
<td>JCL Generation main menu</td>
<td>Jobcard Options</td>
</tr>
</tbody>
</table>

2. From the JCL Generation Jobcard Options Update panel, in the JCLIB field, type the name of the partitioned data set and press END (Figure 26).
When you use the JCLLIB option, the JCL Generation component generates the following statement in the JCL (Figure 27):

**Figure 27 Including the JCL library**

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

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

4. From the JCL Generation Debugging, Display and Execution Options Update panel, specify the name of the member that contains the JCL that you want to run, and press END (Figure 28):
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 29).

Figure 29 Including JCL between steps

```/*------------------------*/
/* END OF JOBSTEP */
/*------------------------*/
```  

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 30).

Figure 30 Including JCL at the end of the jobs (part 1 of 2)

```/*------------------------*/
/* END OF JOB INCLUDE MEMBER */
/*------------------------*/
```
Specifying additional job generation options in DASD MANAGER PLUS

In addition to the hundreds of options that the POF provides, DASD MANAGER PLUS provides the option of specifying a data set to hold JCL that the BMCTRIG component generates.

Customizing DASD MANAGER PLUS (for experienced users)

DASD MANAGER PLUS has several options for customization, which only experienced users should attempt. This section discusses the following features that you can use to customize the product:

- JCL Generation component
- Generation Data Groups
- overriding options
Generating JCL

When you generate JCL for Execution, the DASD MANAGER PLUS product uses symbolic variables to resolve all data set names that appear on the interface panels of the components. (For a list of the symbolic variables for DASD MANAGER PLUS, see Table 31.) Parameters (from the product options file (POF) and Execution panels) are passed to Batch JCL Generation using the AJXIN and AJXPOFIN input streams. These parameters include the names of input files, JCL files, and the diagnostic output files. (For more information about the AJXIN and AJXPOFIN input streams, see the *DASD MANAGER PLUS for DB2 Reference Manual.*)

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:
  
  — sizing the data sets
  
  — using the values specified for using DASD or tape units, as well as the values for tape-related options
  
  — using the values specified for the SMS and alternate SMS data sets, data set thresholds, and alternate units
  
  — processing the options for image copy GDGs
- resolves the names of work data sets that are passed from the JCL Generation option panels, such as SORTWORK, and the permanent data sets that Execution uses
- merges DD statements that more than one command uses (for example, SYSUT1 and SORTWORK) so that the component does not generate duplicates and uses the highest estimated space

You can add debugging comments in the generated JCL by specifying `Gen JCL debug Yes` on the DASD MANAGER PLUS General Options panel (see “Setting general user options” on page 93). You can then see information such as the variables that the product uses for creating the JCL and their assigned values.
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. These symbolic variables are listed in the *DASD MANAGER PLUS for DB2 Reference Manual*.

The generated JCL includes DD statements for all data sets that the job or the Execution component needs, as well as the EXEC statement for the program and any necessary control parameters. For many of the work data sets that Execution uses, you can create the JCL for a cleanup job step that automatically deletes the work data sets at the end of the run. Other data sets are commented out in the JCL, providing you with the option to delete these as well.

**Modifying the ISPF skeletons and variables**

The JCL Generation component generates JCL by using a method that improves upon standard ISPF file-tailoring services. The method used is based on ISPF skeletons. The $AJXDOC member in the HLQ.SLIB data set lists and briefly describes each non-DD and DD statement skeleton that JCL Generation uses. The member also lists the variables that JCL Generation uses to construct the default names for permanent data sets.

Each skeleton name in the $AJXDOC member has a corresponding member in the HLQ.DBSLIB data set. Some of the members that contain skeletons also provide comments, tips, and suggestions for using the skeletons.

You can specify up to five user-defined variables in the POF. The JCL Generation User Defined Variables panel allows you to specify variable names (up to eight characters). Each variable has a corresponding symbolic variable, as shown in Table 23. You can use the symbolic variables in your job cards or data set prefixes. For information about specifying the variables, see “Setting user variables” on page 124.

<table>
<thead>
<tr>
<th>POF variable</th>
<th>Symbolic variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER_VAR1_CHAR</td>
<td>&amp;UVR1</td>
</tr>
<tr>
<td>USER_VAR2_CHAR</td>
<td>&amp;UVR2</td>
</tr>
<tr>
<td>USER_VAR3_CHAR</td>
<td>&amp;UVR3</td>
</tr>
<tr>
<td>USER_VAR4_CHAR</td>
<td>&amp;UVR4</td>
</tr>
<tr>
<td>USER_VAR5_CHAR</td>
<td>&amp;UVR5</td>
</tr>
</tbody>
</table>

The *DASD MANAGER PLUS for DB2 Reference Manual* lists and describes the symbolic and SLIB variables that JCL Generation uses and specifies the length of each variable.
To improve the performance of the JCL construction phase of JCL Generation, BMC uses an SLIB compiler. Consequently, if you edit SLIBs after installation, you must recompile them. For information, see Appendix A, “Using the skeleton library compiler.”

**Sizing the data sets**

You can use the JCL Generation data set sizing function to tailor the data set sizes when the JCL is built. The function gathers information from one of the following sources (shown in general order of accuracy, from most accurate to least accurate):

1. the BMC DASD MANAGER PLUS product statistics database (statistics that the BMCSTATS utility gathers)
2. the DB2 system catalog (statistics that the IBM RUNSTATS utility gathers)
3. the results of VSAM object sampling
4. the default data set allocation parameters that are set from the JCL Generation Individual Data Set Options panel

Whether it uses the BMCSTATS historical database or object sampling, or even if it does not use sizing, JCL Generation obtains some information from the DB2 system catalog. You determine the source to use on the JCL Generation Static Data Set Options panel. (For information, see “Setting the JCL options for static data sets” on page 99.)

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 24 shows the statistics that JCL Generation uses for space estimation and the source of the statistics.

<table>
<thead>
<tr>
<th>Value</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BMCSTATS</td>
</tr>
<tr>
<td>number of active pages</td>
<td>X</td>
</tr>
<tr>
<td>number of modified pages</td>
<td>X</td>
</tr>
<tr>
<td>page size</td>
<td></td>
</tr>
<tr>
<td>maximum row length</td>
<td></td>
</tr>
<tr>
<td>average row length</td>
<td>X</td>
</tr>
<tr>
<td>number of rows</td>
<td>X</td>
</tr>
<tr>
<td>number of nonclustering indexes</td>
<td></td>
</tr>
<tr>
<td>longest key</td>
<td>X</td>
</tr>
<tr>
<td>number of foreign keys</td>
<td></td>
</tr>
</tbody>
</table>

Table 24 Data set sizing values and sources (part 1 of 2)
Specifying generation data groups

You can specify GDGs for local and recovery image-copy data sets, as follows:

- specify the &GDG symbolic variable in User Options, the installation options module, or the POF

- specify the &GDG symbolic variable on the COPYDDN options within individual utilities

**Using user options to specify GDGs**

A quick method of specifying GDGs is to add the &GDG symbolic variable to the local (SYSCnnnn) and recovery (RECVnnnn) image-copy data set prefixes as needed. When you use an &GDG variable (see Figure 32), the product resolves the data set name by using the symbolic variables and includes the GDG number. You can add this symbolic variable to the data set prefixes on the JCL Generation Data Set Options for Copies panels for local primary and backup copies and recovery primary and backup copies. (Access the JCL Generation Data Set Options for Copies from the JCL Generation main menu, similar to the steps in “To set the JCL options for job cards” on page 95.)

---

**Table 24 Data set sizing values and sources (part 2 of 2)**

<table>
<thead>
<tr>
<th>Value</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BMCSTATS</td>
</tr>
<tr>
<td>number of indexes</td>
<td>X</td>
</tr>
<tr>
<td>longest foreign key</td>
<td></td>
</tr>
</tbody>
</table>

---

**TIP**

To specify the sizing method, see “Setting the JCL options for static data sets” on page 99.
Specifying generation data groups

Figure 32  Using the &GDG symbolic variable

When you add the &GDG variable to the end of the field, the product resolves the local image-copy data sets as Figure 33 shows.

Figure 33  Data set names resolved with the &GDG symbolic variable

When you add the &GDG variable to the end of the field, the product resolves the local image-copy data sets as Figure 33 shows.
Figure 34 shows a sample worklist that the product generates when you specify GDG variables in the User Options panels which you can access from the DASD MANAGER PLUS main menu.

### Figure 34  Worklist with GDGs defined in user options

```
***************************** Top of Data *****************************
- TIME 000001 '2003-02-28-14.07.41.976612'                               91895345
- SSID 000001 DEAE                                                       29519769
- WKID 000002 JENTEST.UTILITY                                            40692400
- SYNC 000003                                                            08617204
   93823484
- BMCC 000004
  COPY
  TABLESPACE JEND30.C791278
  SHRLEVEL REFERENCE
  FULL YES
  COPYDDN (C0001,D0001)
  RECOVERYDDN (R0001,V0001)
   42978305
- BMCC 000005
  COPY
  TABLESPACE JEND30.JENS0130
  SHRLEVEL REFERENCE
  FULL YES
  COPYDDN (C0002,D0002)
  RECOVERYDDN (R0002,V0002)

***************************** Bottom of Data *****************************
```

Figure 35 shows the JCL that the product generates when you define GDGs in the User Options panels which you can access from the DASD MANAGER PLUS main menu.

### Figure 35  JCL When GDGs are defined in user options (part 1 of 2)

```c
//*--------------------------------------------------------------------
//*  UTILITY COPY DD STATEMENTS
//*--------------------------------------------------------------------
//* DATA SET SIZE REQUIRED FOR DD C0001 IS AN ESTIMATE
//C0001 DD DSN=RDAJXN.CPY.DEAE.JEND30.C791278(+1),
//         DCB=(SYS1.MODEL),
//         DISP=(NEW,CATLG,CATLG),
//         SPACE=(CYL,(1,1),RLSE), ESTIMATE-C/H
//         UNIT=SYSDA
//* DATA SET SIZE REQUIRED FOR DD C0002 IS AN ESTIMATE
//C0002 DD DSN=RDAJXN.CPY.DEAE.JEND30.JENS0130(+1),
//         DCB=(SYS1.MODEL),
//         DISP=(NEW,CATLG,CATLG),
//         SPACE=(CYL,(1,1),RLSE), ESTIMATE-C/H
//         UNIT=SYSDA
//*--------------------------------------------------------------------
//*  UTILITY DUAL COPY DD STATEMENTS
//*--------------------------------------------------------------------
//* DATA SET SIZE REQUIRED FOR DD D0001 IS AN ESTIMATE
//D0001 DD DSN=RDAJXN.CPY.DEAE.JEND30.C791278(+2),
//         DCB=(SYS1.MODEL),
//         DISP=(NEW,CATLG,CATLG),
//         SPACE=(CYL,(1,1),RLSE), ESTIMATE-C/H
```
Using Copy DD options to specify GDGs

For finer control than the User Options panels (that you access from the DASD MANAGER PLUS main menu) provide, specify GDGs on COPYDDN options panels. Specifications on the COPYDDN options panels override data set prefixes on the User Options panel (Figure 36).

Figure 35  JCL When GDGs are defined in user options  (part 2 of 2)
You can specify GDGs on the COPYDDN options panels of the following utilities:

- **BMCCOPY**
- **FULLCOPY**
- **INRCOPY**
- **MERGCOPY**
- **BMCREORG**
- **REORG**
- **LOAD**
- **BMCLOAD**

**Figure 36** shows the BMCCOPY Copy Options panel.

**Figure 36  BMCCOPY Copy Options panel**

If you specify J on the first two COPYDDN options and the two RECOVERYDDN options and then press END, the product displays a series of panels. Using these panels, you can specify all GDG data sets, all non-GDG data sets, or a combination of GDG and non-GDG data sets. Specify a GDG data set for each primary copy (COPYDDN dd1 and RECOVERYDDN dd1).

**Figure 37** shows the DD options for COPYDDN dd1. The *Dsname Prefix* specifies a GDG variable.
Specifying generation data groups

Chapter 3 Setting up DASD MANAGER PLUS

Press END to display the DD options for RECOVERYDDN dd1, the recovery primary data set (Figure 38). The Dsname Prefix specifies a GDG primary recovery data set.

Press END again to display the DD options for RECOVERYDDN dd2, the recovery backup data set (Figure 39). The Dsname Prefix specifies a GDG data set.
Figure 39  Recovery backup copy options with GDG variable specified

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASUDUM</td>
<td>JCL</td>
</tr>
<tr>
<td>RECOVERYDDN ddname2 JCL TABLESPACE QZUD41..%</td>
<td>Row 1 to 10 of 10</td>
</tr>
<tr>
<td>Service Syntax: BMCCOPY.DEMO_BMCCOPY</td>
<td></td>
</tr>
<tr>
<td>Enter data, then press end. More:</td>
<td></td>
</tr>
<tr>
<td>Blank fields default to the Workid JCL Options.</td>
<td></td>
</tr>
<tr>
<td>UNIT . . . . . . SYSDA (Device type / Unit)</td>
<td></td>
</tr>
<tr>
<td>Dname Prefix . . ASU.RCV2.&amp;SSID..&amp;OBNOD(&amp;GDG)</td>
<td></td>
</tr>
<tr>
<td>(Prefix for output data set name)</td>
<td></td>
</tr>
<tr>
<td>RETPD . . . . . . (1-9999 Retention period)</td>
<td></td>
</tr>
<tr>
<td>EXPDT . . . . . . (YYYY/DDD or YYDDD Expiration date)</td>
<td></td>
</tr>
<tr>
<td>****************************************************** Bottom of data ******************************************************</td>
<td></td>
</tr>
</tbody>
</table>

The product resolves the image-copy data set names in the JCL in Figure 40.

Figure 40  Worklist with GDGs defined in Copy DD options

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASUDUM</td>
<td>JCL</td>
</tr>
<tr>
<td>RECOVERYDDN ddname2 JCL TABLESPACE QZUD41..%</td>
<td>Row 1 to 10 of 10</td>
</tr>
<tr>
<td>Service Syntax: BMCCOPY.DEMO_BMCCOPY</td>
<td></td>
</tr>
<tr>
<td>Enter data, then press end. More:</td>
<td></td>
</tr>
<tr>
<td>Blank fields default to the Workid JCL Options.</td>
<td></td>
</tr>
<tr>
<td>UNIT . . . . . . SYSDA (Device type / Unit)</td>
<td></td>
</tr>
<tr>
<td>Dname Prefix . . ASU.RCV2.&amp;SSID..&amp;OBNOD(&amp;GDG)</td>
<td></td>
</tr>
<tr>
<td>(Prefix for output data set name)</td>
<td></td>
</tr>
<tr>
<td>RETPD . . . . . . (1-9999 Retention period)</td>
<td></td>
</tr>
<tr>
<td>EXPDT . . . . . . (YYYY/DDD or YYDDD Expiration date)</td>
<td></td>
</tr>
<tr>
<td>****************************************************** Bottom of data ******************************************************</td>
<td></td>
</tr>
</tbody>
</table>

In the JCL, the product resolves the image-copy data sets in Figure 41.
Figure 41  Resolution of &GDG and &GDGDU variables

The JCL lists the image-copy data sets under the following headings:

- Utility Copy DD Statements
- Utility Dual Copy DD Statements
- Utility Recovery Site Image Copy DD Statements
- Utility Dual Recovery Site Image Copy DD Statements

**Overriding options**

You specify DASD MANAGER PLUS installation options in the installation options module and the product options file (POF), which the installation process generates. You can assemble the installation options module to change the defaults, or each user can override the values by customizing the user options or the utility options. JCL generation options come from the POF, rather than the installation options module, so you must edit the options there.
To override the default values, use the DASD MANAGER PLUS User Options panel, which you can access from the main menu. Then you can select **General Options** (for the installation options module) or **JCL generation options** (for the Product Option File Functions panel).

**Figure 42** describes installation options for most DASD MANAGER PLUS utility functions.

Each user’s Time Sharing Option (TSO) ISPF profile contains the values for all options except plan names. Storing these values in the user profile allows you to reassemble the installation options module without affecting the options for an existing user unless you use the Refresh option. If you refresh the installation options, you override any difference in the user options but you also populate an existing utility option.

You can reset JCL generation ISPF variables from a POF by using the **Product Option File (POF) Functions** option of the **JCL Generation Options** menu.

When you edit user options, they replace corresponding values in the ISPF profile. You can use the JCL generation options POF function to write ISPF variables to an alternate POF, which you can then use with an action or to reset the values in future sessions.
Using action POFs to reset JCL options

You can use an options file for an action (also called an action POF or user POF) to reset all of the options for creating JCL. To create an action options file, select the Product Option File (POF) Functions option of the JCL Generation Options menu. You can also use an action POF to set options for different sets of applications, particularly if the applications have different naming standards.

Overview of product options

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

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

Product options

Figure 43 provides an example of a product options file.

Figure 43 Product options file (part 1 of 12)

POFDATE = 2012/11/13 09:20:37
*-------------------------------------------------------------
* POF WRITTEN FROM VERSION: V11.01.00
* FORMAT:
* KEYWORD=PARM COLUMNS 1-80.
* PARM SYNTAX:
* VALUE - EVERYTHING AFTER THE = IS CONSIDERED THE VALUE.
* LEADING AND TRAILING BLANKS ARE REMOVED.
* VALUE.(R) TO INDICATE REFRESH OPTION.
* NO SPLITTING OF VALUE ACROSS LINES. IF IT WON'T FIT ON
* LINE WITH KEYWORD, ENTER '>' AFTER = AND PUT THE PARM
* ON NEXT LINE.
* BLANK LINES ARE IGNORED.
* ASTERISK IN COLUMN 1 INDICATES THAT LINE IS A COMMENT.
*-------------------------------------------------------------
ACM_AMS = Y
ACM_BASDIAG = SYSOUT
ACM_BRPTDIAG = SYSOUT
ACM_BRPTDSN = '&PREFIX..BASELINE.REPORT'
ACM_CDLDSN = '&PREFIX..&SSID..CDL(CDL)'
ACM_CDLPS = 15
ACM_CDLSS = 5
ACM_CDLU = SYSDA
ACM_CMPDIAG = SYSOUT
ACM_CPLCDLO = '&PREFIX..&SSID..CDL(CDL)'
ACM_CPLDIAG = SYSOUT
ACM_CPLWDSN = '&PREFIX..&SSID..&TASKID'
ACM_CPLWDSNO = '&PREFIX..&SSID..&WORKID'
ACM_DBRM1 =
ACM_DBRM2 =
ACM_DBRM3 =
ACM_DYNSORTW_NUM = 32
ACM_DYNSORTW_UNIT = SYSDA
ACM_GLID =
ACM_IBMR_MAP_REQ = Y
ACM_IMPDIAG = SYSOUT
ACM_JDSN = '&PREFIX..ANALYSIS(&WORKID)'
ACM_JDSNB = '&PREFIX..BASELINE(&WORKID)'
ACM_JDSNBG = '&PREFIX..JCLGEN(&WORKID)'
ACM_JDSNBR = '&PREFIX..BASELINE(BLRPTJCL)'
ACM_JDSNC = '&PREFIX..COMPARE(CMPJCL)'
ACM_JDSNCPL = '&PREFIX..TASKID(&TASKID)'
ACM_JDSNCPL0 = '&PREFIX..EXEC(&WORKID)'
ACM_JDSNE = '&PREFIX..EXEC(&WORKID)'
ACM_JDSNI = '&PREFIX..IMPORT(&WORKID)'
ACM_PARALLEL_MAXINIT = 3
ACM_PARALLEL_MININIT = 2
ACM_PARALLEL_WORKLST = N
ACM_PARALLEL_XIMGRP = XIMACM
ACM_PARALLEL_XIMPROC = XIMACM
ACM_PARALLEL_XIMSTRT = N
ACM_PARALLEL_XIMTRCE = N
ACM_PIC = N
ACM_SDSN = SYSOUT
ACM_SDSNE = SYSOUT
ACM_WDSN = '&PREFIX..&SSID..&WORKID'
ACM_WLORDER =
ACM_WLORDERMSG = Y
ACM_WLPS = 15
ACM_WLSS = 5
ACM_WLU = SYSDA
ADDLOAD1 = BMCRMD.V810INST.NONSMPE.LOAD,(R)
ADDLOAD2 = BMCRMD.V810INST.UDBLINK,(R)
ARCH_DATACLASS =
ARCH_DATACLASS_ALT =
ARCH_EXPDT =
ARCH_MGMTCLASS =
ARCH_MGMTCLASS_ALT =
ARCH_PREFIX = &PREFIX..&WKID
ARCH_PRIQTY = 10
Figure 43  Product options file (part 3 of 12)

```plaintext
ARCH_RETPD =
ARCH_SECQTY = 2
ARCH_STACK = N
ARCH_STORCLASS =
ARCH_STORCLASS_ALT =
ARCH_THRESH = 0
ARCH_UNIT = SYSDA
ARCH_UNIT_ALT =
ASU_XP_LOGD_DATAC=
ASU_XP_LOGD_MGMTC=
ASU_XP_LOGD_PRIQTY=10
ASU_XP_LOGD_SECQTY=2
ASU_XP_LOGD_STORC=
ASU_XP_LOGD_UNIT=SYSDA
ASU_XP_LOGDSN=&PREFIX..XPORT.LOG
ASU_XP_UIMSRVHOST=
ASU_XP_UIMSRVPORT=1
ASU_XP_UIMSRVTIMEOUT=300
BINDFAIL = N
BLRP_DATACLASS =
BLRP_DATACLASS_ALT =
BLRP_EXPDT =
BLRP_MGMTCLASS =
BLRP_MGMTCLASS_ALT =
BLRP_PREFIX = &PREFIX..&OBNOD
BLRP_PRIQTY = 10
BLRP_RETPD =
BLRP_SECQTY = 2
BLRP_STACK = N
BLRP_STORCLASS =
BLRP_STORCLASS_ALT =
BLRP_THRESH = 0
BLRP_UNIT = SYSDA
BLRP_UNIT_ALT =
BMC_CHECK_LOAD =
BMC_CHECK_OPTS = ACK$OPTS
BMC_COPY_LOAD = &PREFIX.RNTM.BMCLINK,(R)
BMC_COPY_OPTS = ACP$OPTS
BMC_LOAD_LOAD = &PREFIX.RNTM.BMCLINK,(R)
BMC_LOAD_OPTS = AMU$OPTS
BMC_RECOVER_LOAD = &PREFIX.RNTM.BMCLINK,(R)
BMC_RECOVER_OPTS = AFR$OPTS
BMC_REORG_LOAD = &PREFIX.RNTM.BMCLINK,(R)
BMC_REORG_OPTS = ARU$OPTS
BMC_REORG_XBMD =
BMC_UNLOAD_LOAD = &PREFIX.RNTM.BMCLINK,(R)
BMC_UNLOAD_OPTS =
CAT_LOAD = &PREFIX.RNTM.BMCLINK,(R)
CHGMAN_LOAD = &PREFIX.RNTM.BMCLINK,(R)
CLEANUP_RC = 4
CNTL_DATACLASS =
```
Overriding options

Figure 43  Product options file (part 4 of 12)

```
CNTL_EXPDT =
CNTL_MGMTCLASS =
CNTL_PREFIX = &PREFIX..&WKID..&SSID
CNTL_PRIQTY = 1
CNTL_RETPD =
CNTL_SECQTY = 1
CNTL_STORCLASS =
CNTL_UNIT = SYSDA
CNTLMOUT_DSN=&PREFIX..&SSID..CNTLMOUT(&JOBNAME)
CNTLMSCH_DSN=&PREFIX..&SSID..CNTLMSCH(&JOBNAME)
CPYEXP_DATACLASS =
CPYEXP_EXPDT =
CPYEXP_MGMTCLASS =
CPYEXP_PREFIX = &PREFIX..&WKID
CPYEXP_RETPD =
CPYEXP_STORCLASS =
CPYEXP_SUPPRESS_SUFF = N
CPYEXP_UNIT = SYSDA
DASD_LOAD = &PREFIX.DBLINK,(R)
DASDDOPT = RASUBFD,(R)
DATA_PACKER_LOAD = &PREFIX.RNTM.BMCLINK,(R)
DATASETSIZING = N
DATAWK_NBR = 4
DATAWK_UNIT = SYSDA
DB2EXIT = SYS3.DEBF.DSNEXIT,(R)
DB2LOAD = CSGI.DB2V81M.DSNLOAD,(R)
DEF_GDG_BASE = N
DEF_GDG_LIMIT = 10
DEF_GDG_NOSCR = N
DEF_GDG2_LIMIT = 10
DIAG_MSGCLASS =
DISC_DATACLASS =
DISC_DATACLASS_ALT =
DISC_EXPDT =
DISC_MGMTCLASS =
DISC_MGMTCLASS_ALT =
DISC_PREFIX = &PREFIX..&WORKID..&OBNO
DISC_PRIQTY = 10
DISC_RETPD =
DISC_SECQTY = 2
DISC_STORCLASS =
DISC_STORCLASS_ALT =
DISC_THRESH = 0
DISC_UNIT = SYSDA
DISC_UNIT_ALT =
DISP_ALLOW_POPUP = Y
DISP_AUTO_TAB = +
DISP_LOCATION = M
DISP_OMIT_CHAR = <>
DISP_STATS = N
DISP_VAR_DBG = N
```
Figure 43  Product options file (part 5 of 12)

```plaintext
DROPR_NOIC = N
DSNCHECK44 = N
DSNTIAD_PLAN =
ERR_DATACLASS =
ERR_DATACLASS_ALT =
ERR_EXPDT =
ERR_MGMTCLASS =
ERR_MGMTCLASS_ALT =
ERR_PREFIX = &PREFIX..&WKID..&STEPN
ERR_PRIQTY = 10
ERR_RETPD =
ERR_SECQTY = 2
ERR_STORCLASS =
ERR_STORCLASS_ALT =
ERR_THRESH = 0
ERR_UNIT = SYSDA
ERR_UNIT_ALT =
EXEC_LOAD = &PREFIX.RNTM.BMCLINK,(R)
FCPY_DATACLASS =
FCPY_EXPDT =
FCPY_MGMTCLASS =
FCPY_PREFIX = &PREFIX..&OBNO..&PART
FCPY_PRIQTY = 10
FCPY_RETPD =
FCPY_SECQTY = 2
FCPY_STORCLASS =
FCPY_SUPPRESS_SUFF = N
FCPY_UNIT = SYSDA
FILT_DATACLASS =
FILT_EXPDT =
FILT_MGMTCLASS =
FILT_PREFIX = &PREFIX..&WKID..&STEPN
FILT_PRIQTY = 10
FILT_RETPD =
FILT_SECQTY = 2
FILT_STORCLASS =
FILT_UNIT = SYSDA
GDG_MODEL = SYS1.MODEL
HASHFAIL = N
HIFESTRNC =
INCLUDE_SYSPRIN2 = N
IOALOAD1 = DFD.V6218.IOAI.LOAD
IOALOAD2 = DFD.V6218.IOAI.CTRANS
JCLCLEANUP = N
JCLLIB =
JES3 = N
JOB_INCLUDE_MEMBER =
JOBCARD1 = //&USERID.&JOBCHAR JOB (&ZACCTNUM),&PGMR,
JOBCARD2 = //  CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
JOBCARD3 = //  NOTIFY=&USERID
JOBCARD4 = /*ROUTE XEQ BMCPLX1
```
Figure 43  Product options file (part 6 of 12)

```plaintext
JOBCARD5 = /*JOBPARM SYSAFF=&ZSYSID
LISTDEF_DSN =
LL_CLIB = CLIB
LL_CLIB2 =
LL_CLIB3 =
LL_CLIB4 =
LL_CLIB5 =
LL_LINK = LINK
LL_LINK2 =
LL_LINK3 =
LL_LINK4 =
LL_LINK5 =
LL_MLIB = MLIB
LL_MLIB2 =
LL_MLIB3 =
LL_MLIB4 =
LL_MLIB5 =
LL_PLIB = PLIB
LL_PLIB2 =
LL_PLIB3 =
LL_PLIB4 =
LL_PLIB5 =
LL_SLIB = SLIB
LL_SLIB2 =
LL_SLIB3 =
LL_SLIB4 =
LL_SLIB5 =
LL_TLIB = TLIB
LL_TLIB2 =
LL_TLIB3 =
LL_TLIB4 =
LL_TLIB5 =
LL_XML = XML
LL_XML2 =
LL_XML3 =
LL_XML4 =
LL_XML5 =
LLQ = BMC,(R)
LOGWK_NBR = 4
LOGWK_UNIT = SYSDA
MAP_DATACLASS =
MAP_DATACLASS_ALT =
MAP_EXPDT =
MAP_MGMTCLASS =
MAP_MGMTCLASS_ALT =
MAP_PREFIX = &PREFIX..&WKID..&SSID
MAP_PRIQTY = 10
MAP_RETPD =
MAP_SECQTY = 2
MAP_STORCLASS =
MAP_STORCLASS_ALT =
```
MAP_THRESH = 0
MAP_UNIT = SYSDA
MAP_UNIT_ALT =
MAX_CYL = 99999
MAX_PRIQTY = 2000
MAX_SECQTY = 200
MAX_UNITCNT =
MEMLIMIT =
ORTPARM_DSN =
PCPY1_DATACLASS =
PCPY1_DATACLASS_ALT =
PCPY1_EXPDT =
PCPY1_MGMTCLASS =
PCPY1_MGMTCLASS_ALT =
PCPY1_PREFIX = &PREFIX..&OBNOD..P&PART
PCPY1_PRIQTY = 10
PCPY1_RETPD =
PCPY1_SECQTY = 2
PCPY1_STACK = N
PCPY1_STORCLASS =
PCPY1_STORCLASS_ALT =
PCPY1_SUPPRESS_SUFF = N
PCPY1_THRESH = 0
PCPY1_UNIT = SYSDA
PCPY1_UNIT_ALT =
PCPY2_DATACLASS =
PCPY2_DATACLASS_ALT =
PCPY2_EXPDT =
PCPY2_MGMTCLASS =
PCPY2_MGMTCLASS_ALT =
PCPY2_PREFIX = &PREFIX..&OBNOD..P&PART
PCPY2_PRIQTY = 10
PCPY2_RETPD =
PCPY2_SECQTY = 2
PCPY2_STACK = N
PCPY2_STORCLASS =
PCPY2_STORCLASS_ALT =
PCPY2_SUPPRESS_SUFF = N
PCPY2_THRESH = 0
PCPY2_UNIT = SYSDA
PCPY2_UNIT_ALT =
PRE_JOBSTEP_INCLUDE =
PROC_BMCHECK_NAME =
PROC_BMCHECK_STEP =
PROC_BMCCOPY_NAME =
PROC_BMCCOPY_STEP =
PROC_BMCCPRS_NAME =
PROC_BMCCPRS_STEP =
PROC_BMCLOAD_NAME =
PROC_BMCLOAD_STEP =
PROC_BMCRECOVER_NAME =
Figure 43  Product options file (part 8 of 12)

```plaintext
PROC_BMCRECOVER_STEP =
PROC_BMCREORG_NAME =
PROC_BMCREORG_STEP =
PROC_BMCSTATS_NAME =
PROC_BMCSTATS_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 =
PUNCH_MGMTCLASS =
PUNCHPREFIX = &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
```
Figure 43  Product options file (part 9 of 12)

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 = OM
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 =
SRTOUT_EXPDT =
SRTOUT_MGMTCLASS =
SRTOUT_MGMTCLASS_ALT =
SRTOUT_PREFIX = &PREFIX..&WKID..&STEPN
SRTOUT_PRIQTY = 10
SRTOUT_RETPD =
SRTOUT_SECQTY = 2
Figure 43  Product options file (part 10 of 12)

```plaintext
SRTOUT_STORCLASS =
SRTOUT_STORCLASS_ALT =
SRTOUT_THRESH = 0
SRTOUT_UNIT = SYSDA
SRTOUT_UNIT_ALT =
STEP_INCLUDE_MEMBER =
STOPWAIT = 3
STOPWTSECS = 10
SUPPRESS_COMMENTS = N
SYNCDELETE = N
SYSEXEC = &PREFIX.RNTM.BMCREXX,(R)
SYSTEM_MLIB = SYS1.PROD.ISPMENU,(R)
SYSUT_DATACLASS =
SYSUT_DATACLASS_ALT =
SYSUT_EXPDT =
SYSUT_MGMTCLASS =
SYSUT_MGMTCLASS_ALT =
SYSUT_PREFIX = &PREFIX..&WKID..&STEPN
SYSUT_PRIQTY = 10
SYSUT_RETPD =
SYSUT_SECQTY = 2
SYSUT_STORCLASS =
SYSUT_STORCLASS_ALT =
SYSUT_THRESH = 0
SYSUT_UNIT = SYSDA
SYSUT_UNIT_ALT =
SZDEVT = 3390
TAPE_EXPDT =
TAPE_RETPD =
TAPE_VOLCNT = 99
TAPE1 = CART
TAPE2 = TAPE
TAPE3 = TAPE
TEMPLATE_DSN =
TEMPUNIT = SYSDA
TRTCH =
TRTCH =
TSOPROGRAM =
TSOSUBEXIT = N
ULLQ = ,,(R)
UNLD_FREF_DATACLASS =
UNLD_FREF_DIRBLOCK = 250
UNLD_FREF_MGMTCLASS =
UNLD_FREF_PREFIX = &PREFIX..&MSSID..&WORKID8
UNLD_FREF_PRIQTY = 10
UNLD_FREF_SECQTY = 2
UNLD_FREF_STORCLASS =
UNLD_FREF_SUPPR_SUFF = N
UNLD_FREF_UNIT = SYSDA
UNLD1_DATACLASS =
UNLD1_DATACLASS_ALT =
```
Figure 43 Product options file (part 11 of 12)

```plaintext
UNLD1_EXPDT =
UNLD1_MGMTCLASS =
UNLD1_MGMTCLASS_ALT =
UNLD1_PREFIX = &PREFIX..&MSSID..&WORKID8
UNLD1_PRIQTY = 10
UNLD1_RETPD =
UNLD1_SECQTY = 2
UNLD1_STACK = N
UNLD1_STORCLASS =
UNLD1_STORCLASS_ALT =
UNLD1_SUPPRESS_SUFF = N
UNLD1_THRESH = 0
UNLD1_UNIT = SYSDA
UNLD1_UNIT_ALT =
UNLD2_DATACLASS =
UNLD2_DATACLASS_ALT =
UNLD2_EXPDT =
UNLD2_MGMTCLASS =
UNLD2_MGMTCLASS_ALT =
UNLD2_PREFIX = &PREFIX..&MSSID..&WORKID8
UNLD2_PRIQTY = 10
UNLD2_RETPD =
UNLD2_SECQTY = 2
UNLD2_STACK = N
UNLD2_STORCLASS =
UNLD2_STORCLASS_ALT =
UNLD2_SUPPRESS_SUFF = N
UNLD2_THRESH = 0
UNLD2_UNIT = SYSDA
UNLD2_UNIT_ALT =
UNLD3_DATACLASS =
UNLD3_EXPDT =
UNLD3_MGMTCLASS =
UNLD3_MGMTCLASS_ALT =
UNLD3_PREFIX = &PREFIX..&MSSID..&WORKID8
UNLD3_RETPD =
UNLD3_STORCLASS =
UNLD3_SUPPRESS_SUFF = N
UNLD3_UNIT = SYSDA
UNLD4_DATACLASS =
UNLD4_MGMTCLASS =
UNLD4_MGMTCLASS_ALT =
UNLD4_PREFIX = &PREFIX..&MSSID..&WORKID8
UNLD4_STORCLASS =
UNLD4_SUPPRESS_SUFF = N
UNLD4_UNIT = SYSDA
USER_VAR1_CHAR =
USER_VAR2_CHAR =
USER_VAR3_CHAR =
USER_VAR4_CHAR =
USER_VAR5_CHAR =
WORK_DATACLASS =
WORK_MGMTCLASS =
```
Overriding options

Figure 43  Product options file (part 12 of 12)

WORK_STORCLASS =
2MEGSQL = N

--- 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.

Descriptions of the keywords that are listed in Figure 43 follow.

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_AMS=Y

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

- Y  generates AMS statements (IDCAMS DELETE and DEFINE) in a worklist
- N  generates a -STOP worklist command that enables you to complete the DELETE and DEFINE commands before the DB2 object CREATE commands that are located later in the worklist

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

This keyword is not included in the AJXPOFIN input stream.

ACM_BASDIAG=SYSOUT

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

This keyword is not included in the AJXPOFIN input stream.

ACM_BRPTDIAG=SYSOUT

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

This keyword is not included in the AJXPOFIN input stream.
Overriding options

**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.
Overriding options

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

For CHANGE MANAGER, this keyword defines the default data set name that is used for Baseline JCL. This data set can be either a sequential or a partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. CHANGE MANAGER automatically uses the work ID as the member name.

This keyword is not included in the AJXPOFIN input stream.

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

For ALTER and CHANGE MANAGER, this keyword defines the default data set name that is used for batch JCL Generation. This data set can be either a sequential or a partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. The products automatically use the work ID as the member name.

This keyword is not included in the AJXPOFIN input stream.

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

For CHANGE MANAGER, this keyword specifies the default data set name where the product places the generated Baseline Report JCL. This data set can be either a sequential or partitioned data set.

This keyword is not included in the AJXPOFIN input stream.

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

For CHANGE MANAGER, this keyword defines the default data set name that is used for Compare JCL. This data set can be either a sequential or partitioned data set.

This keyword is not included in the AJXPOFIN input stream.

**ACM_JDSNCPL=’&PREFIX..TASKID(&TASKID)’**

For CHANGE MANAGER, this keyword specifies the default data set name where the CM/PILOT component places the generated Execution JCL. This data set can be either a sequential or partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. CM/PILOT automatically uses the task ID as the member name.

This keyword is not included in the AJXPOFIN input stream.
ACM_JDSNCPLO='&PREFIX..EXEC(&WORKID)'

For the CM/PILOT component of CHANGE MANAGER, this keyword defines the default data set name that is used for Execution JCL. This data set can be either a sequential or partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. The product automatically uses the work ID as the member name.

This keyword is not included in the AJXPOFIN input stream.

ACM_JDSNE='&PREFIX..EXEC(&WORKID)'

For ALTER and CHANGE MANAGER, this keyword defines the default data set name that is used for Execution JCL. This data set can be either a sequential or partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. The products automatically use the work ID as the member name.

This keyword is not included in the AJXPOFIN input stream.

ACM_JDSNI='&PREFIX..IMPORT(&WORKID)'

For ALTER and CHANGE MANAGER, this keyword defines the default data set name that is used for Import JCL. This data set can be either a sequential or partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. The products automatically use the work ID as the member name.

This keyword is not included in the AJXPOFIN input stream.

ACM_PARALLEL_MAXINIT=3

For the Database Administration solution, this keyword specifies the maximum number of the BMC Cross-System Image Manager (XIM) initiators to use when executing a worklist in parallel. This value controls the number of permanent work data sets that are allocated in the execution JCL. The valid range of values is 1 through 32. The maximum number of initiators should not exceed the number of objects in a worklist.

ACM_PARALLEL_MININIT=2

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

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

Y  Execute the worklist in parallel.
   However, if adequate XIM resources are not available, the Execution function fails. In addition, if the required parallelism worklist commands (such as -BEGG and -ENDDG) are not included in the worklist, the worklist is not executed in parallel.

N  Execute the worklist sequentially, even if the required parallelism worklist commands are included in the worklist.

ACM_PARALLEL_XIMGRP=XIMACM

For the Database Administration solution, this keyword specifies the group name for the XIM technology. The group name for XIM must be unique for each instance of XIM that is running on an OS/390 or z/OS image.

ACM_PARALLEL_XIMPROC=XIMACM

For the Database Administration solution, this keyword specifies the name of the procedure that the solution uses to start the XIM technology automatically. BMC recommends that the name of the XIM started task procedure be unique for each instance of XIM that is running on an OS/390 or z/OS image.

ACM_PARALLEL_XIMSTRT=N

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

ACM_PARALLEL_XIMTRCE=N

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

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

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

ACM_PIC=N

For ALTER and CHANGE MANAGER, this keyword indicates whether an image copy should be taken of each table space before a database is dropped, a table is dropped, or the table space is dropped or reorganized.
Overriding options

Chapter 3 Setting up DASD MANAGER PLUS

This keyword is not included in the AJXPOFIN input stream.

**ACM_SDSN=SYSOUT**

For ALTER and CHANGE MANAGER, this keyword specifies the default data set for diagnostic messages for Analysis. The value can be a sequential file, the keyword SYSOUT, or TERM (terminal). If you use SYSOUT, the diagnostic messages are written to the JES SPOOL. If you use TERM, the diagnostic messages are written to your terminal.

This keyword is not included in the AJXPOFIN input stream.

**ACM_SDSNE=SYSOUT**

For ALTER and CHANGE MANAGER, this keyword specifies the default data set for diagnostic messages for Execution. The value can be a sequential file or the keyword SYSOUT. If you use SYSOUT, the diagnostic messages are written to the JES SPOOL.

This keyword is not included in the AJXPOFIN input stream.

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

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

This keyword is not included in the AJXPOFIN input stream.

**ACM_WLORDER**

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

- **C** Sorts the worklist by each table’s cardinality in descending sequence.
- **N** Sorts the worklist by table order in ascending sequence, according to the table owner and table name.
- **A** Sorts the worklist by either table cardinality or by table order, depending on whether the worklist is processed in parallel.

  - 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.
- **Blank** Generates the objects in the worklist in an unsorted, random order.

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

**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.UDBLINK**

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

---

**TIP**

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

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

**ARCH_DATACLASS_ALT**

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

**ARCH_EXPDT**

This keyword specifies the expiration date of the archive data set on tape. A data set cannot have an expiration date and a retention period. The valid values are *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 99999.
**ARCH_STACK=N**

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

**ARCH_STORCLASS**

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

**ARCH_STORCLASS_ALT**

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

**ARCH_THRESH=0**

This keyword specifies the maximum anticipated size for the archive data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

**ARCH_UNIT=SYSDA**

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

**ARCH_UNIT_ALT**

This keyword specifies the alternate unit name for the archive data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keyword. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**ASU_XP_LOGD_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 values are `yyddd` or `yyyy/ddd`.

**BLRP_MGMTCLASS**

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

**BLRP_MGMTCLASS_ALT**

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

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

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

**BLRP_PRIQTY=10**

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

**BLRP_RETPD**

For CHANGE MANAGER, this keyword specifies the retention period for the baseline recovery point data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.
**BLRP_SECQTY=2**

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

**BLRP_STACK=N**

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

**BLRP_STORCLASS**

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

**BLRP_STORCLASS_ALT**

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

**BLRP_THRESH=0**

For CHANGE MANAGER, this keyword specifies the maximum anticipated size for the baseline recovery point data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

**BLRP_UNIT=SYSDA**

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

**BLRP_UNIT_ALT**

For CHANGE MANAGER, this keyword specifies the alternate unit name for the baseline recovery point data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
### Overriding options

#### BMC_CHECK_LOAD

This keyword specifies the name of the LINK library for the BMC CHECK PLUS utility. This keyword replaces the CHECK+_LOAD AJXPOFIN keyword. If both BMC_CHECK_LOAD and CHECK+_LOAD are included in the POF, the components use the value that is specified for CHECK+_LOAD.

---

**TIP**

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

#### BMC_CHECK_OPTS=ACKSOPTS

This keyword specifies the name of the BMC CHECK PLUS utility installation options module. This keyword replaces the CHECKDOPT AJXPOFIN keyword. If both BMC_CHECK_OPTS and CHECKDOPT are included in the POF, the components use the value that is specified for CHECKDOPT.

#### BMC_COPY_LOAD

This keyword specifies the name of the LINK library for the BMC COPY PLUS utility. This keyword replaces the COPY+_LOAD AJXPOFIN keyword. If both BMC_COPY_LOAD and COPY+_LOAD are included in the POF, the components use the value that is specified for COPY+_LOAD.

---

**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.
Overriding options

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

**BMC_LOAD_OPTS=AMUSOPTS**

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=ARUSOPTS**

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=ADUSOPTS**

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_LOADER**

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.
Overriding options

Chapter 3 Setting up DASD MANAGER PLUS

CHECK+_LOAD

This keyword specifies the name of the LINK library for the BMC CHECK PLUS utility. The BMC_CHECK_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_CHECK_LOAD and CHECK+_LOAD are included in the POF, the components use the value that is specified for CHECK+_LOAD.

CHECKDOPT=ACK$MMS

This keyword specifies the name of the installation options module for the BMC CHECK PLUS utility. The BMC_CHECK_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_CHECK_OPTS and CHECKDOPT are included in the POF, the components use the value that is specified for CHECKDOPT.

CHGMAN_LOAD

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

TIP

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

CLEANUP_RC=4

This keyword specifies the value of the return code from the JCL cleanup job step. The cleanup job step, which deletes permanent work data sets, is only performed if the condition code that is returned from any previous job step is less than or equal to the code specified in CLEANUP_RC.

CNTL_DATACLASS

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

CNTL_EXPDT

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

CNTL_MGMTCLASS

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

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 values are yyddd or yyyy/ddd.

CPYEXP_MGMTCLASS

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

CPYEXP_PREFIX=&PREFIX..&WKID

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

CPYEXP_RETPD

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

CPYEXP_STORCLASS

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

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

**CPYEXP_UNIT=SYSDA**

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

**DASD_LOAD**

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

_TIP_

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

**DASDDOPT**

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

**DATA_PACKER_LOAD**

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

_TIP_

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

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

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

**DATAWK_NBR=4**

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

**DATAWK_UNIT=SYSDA**

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

**DB2EXIT**

This keyword specifies the name of the DB2 EXIT library.

--- **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.
Overriding options

**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 values are *yydd* or *yyyy/ddd*. 

---

182  DASD MANAGER PLUS for DB2 User Guide
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.

- **Y** indicates to display the object name in a dialog
  - If the name is too long to be displayed in a dialog, the product displays the name in a panel.
- **N** indicates to display the object name in a panel

**DISP_AUTO_TAB=+**

For ALTER and CHANGE MANAGER, this keyword specifies whether to display an autotab character in front of an object name that is too long to be displayed.

- **N** indicates not to display an autotab character
- any character other than **N** displays as the autotab character

**DISP_LOCATION=M**

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

The following values are valid:

- **B** replaces characters at the left end (beginning) of the name
- **M** replaces characters in the middle of the name
- **E** replaces characters at the right end (end) of the name
**DISP_OMIT_CHAR=<>**

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

**DISP_STATS=N**

This keyword specifies whether to include comments that show the statistics that the components use to determine the sizes of the data sets in the generated JCL (Y or N).

**DISP_VAR_DBUG=N**

This keyword specifies whether to include the SLIB variables that JCL Generation uses to create the JCL, as well as their assigned values, in the generated JCL (Y or N).

**DROPR_NOIC=N**

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

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

**DSNCHECK44=N**

This keyword specifies whether to verify that the prefix of a data set name contains 44 characters (Y or N). Typically, JCL Generation verifies whether a prefix of a data set name contains 35 characters.

Change the value of this keyword to Y for the following reasons:

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

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

**DSNTIAD_PLAN**

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

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

ERR_DATACLASS_ALT

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

ERR_EXPDT

This keyword specifies the expiration date of the error data set on tape. A data set cannot have an expiration date and a retention period. The valid values are \texttt{yyddd} or \texttt{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).
Overriding options

**FCPY\_EXPDT**

For CATALOG MANAGER, this keyword specifies the expiration date of the flashcopy copy data set on tape. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume). A data set cannot have an expiration date and a retention period. The valid values are `yyddd` or `yyyy/ddd`.

**FCPY\_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..&OBNO\_P&PART**

For CATALOG MANAGER, this keyword specifies the prefix for the name of the flashcopy data set. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume).

**FCPY\_PRIQTY=10**

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

**FCPY\_RETPD**

For CATALOG MANAGER, this keyword specifies the retention period for the flashcopy data set on tape. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume). A data set cannot have an expiration date and a retention period. Valid values are 1 to 9999.

**FCPY\_SECQTY=2**

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

**FCPY\_STORCLASS**

For CATALOG MANAGER, this keyword specifies the SMS definition for the storage class associated with the flashcopy data set. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume).
**FCPY_SUPPRESS_SUFF=N**

For CATALOG MANAGER, this keyword specifies whether to suppress adding the DD name to the end of the name of the flashcopy data set (Y or N). If you specify Y, you must ensure that the data set name is unique. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume).

**FCPY_UNIT=SYSDA**

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

**FILT_DATACLASS**

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

**FILT_EXPDT**

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

**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_RETPO**

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.

**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.

---

**IOLOAD1**

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.

---

**IOLOAD2**

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

---

**TIP**

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

---

**JCLCLEANUP=N**

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

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

**JES3=N**

JCL Generation no longer uses this keyword.

**JOB_INCLUDE_MEMBER**

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

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

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

**LISTDEF_DSN**

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

**LL_CLIB=CLIB**

**LL_CLIB2**

**LL_CLIB3**

**LL_CLIB4**

**LL_CLIB5**

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

**LL_LINK=LINK**

**LL_LINK2**

**LL_LINK3**

**LL_LINK4**

**LL_LINK5**

These keywords specify the LLQ for the LOAD library data sets for the batch ISPF environment.
These keywords specify the LLQ for the message data sets for the batch ISPF environment.

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.
Overriding options

LLQ

This keyword specifies the LLQ for ISPF data sets for the batch ISPF environment. During installation, if you chose to use the runtime enablement feature, the Installation System set this value to BMC. If you chose not to use the feature, the Installation System set the value to DB.

**NOTE**

Changing the qualifier of the ISPF data sets might cause unpredictable results. Do not change the LLQ for the ISPF data sets.

LOAD+_LOAD

This keyword specifies the name of the LINK library for the BMC LOADPLUS utility. The BMC_LOAD_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_LOAD_LOAD and LOAD+_LOAD are included in the POF, the components use the value that is specified for LOAD+_LOAD.

LOADDOPT

This keyword specifies the name of the installation options module for the BMC LOADPLUS utility. The BMC_LOAD_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_LOAD_OPTS and LOADDOPT are included in the POF, the components use the value that is specified for LOADDOPT.

LOGWK_NBR=4

This keyword specifies the number of LOGSORT data sets. Valid values are 1 through 32.

LOGWK_UNIT=SYSDA

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

MAP_DATACLASS

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

MAP_DATACLASS_ALT

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

This keyword specifies the expiration date of the map data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyyddd 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).
Overriding options

MAP_THRESH=0

This keyword specifies the maximum anticipated size for the map data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

MAP_UNIT=SYSDA

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

MAP_UNIT_ALT

This keyword specifies the alternate unit name for the map data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

MAX_CYL=99999

This keyword specifies the maximum number of cylinders in the data set. If this value is exceeded for a data set, MAX_PRIQTY and MAX_SECQTY are used for any type of data set that does not have a specific threshold limit specified. Valid values are 1 through 99999.

MAX_PRIQTY=2000

This keyword specifies the primary quantity in cylinders that will be used when the value of MAX_CYL is reached. Valid values are 1 through 9999.

MAX_SECQTY=200

This keyword specifies the secondary quantity in cylinders that will be used when the value of MAX_CYL is reached. Valid values are 1 through 9999.

MAX_UNITCNT

This keyword specifies the value for the DASD unit count. Valid values are 1 through 59.

MEMLIMIT

This keyword specifies the limit on above-the-bar memory for an address space.
ORTPARM_DSN

This keyword specifies the name of the data set for the SyncSort parameters.

PCPY1_DATACLASS

This keyword specifies the SMS definition for the data class associated with the local primary copy data set.

PCPY1_DATACLASS_ALT

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

PCPY1_EXPDT

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

PCPY1_MGMTCLASS

This keyword specifies the SMS definition for the storage class associated with the local primary copy data set.

PCPY1_MGMTCLASS_ALT

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

PCPY1_PREFIX=&PREFIX..&OBNOD..P&PART

This keyword specifies the prefix for the name of the local primary copy data set.

PCPY1_PRIQTY=10

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

PCPY1_RETPD

This keyword specifies the retention period for the local primary copy data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 to 9999.
PCPY1_SECQTY=2

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

PCPY1_STACK=N

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

PCPY1_STORCLASS

This keyword specifies the SMS definition for the storage class associated with the local primary copy data set.

PCPY1_STORCLASS_ALT

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

PCPY1_SUPPRESS_SUFF=N

This keyword specifies whether to suppress adding the DD name to the end of the name of the local primary copy data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

PCPY1_THRESH=0

This keyword specifies the maximum anticipated size for the local primary copy data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

PCPY1_UNIT=SYSDA

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

PCPY1_UNIT_ALT

This keyword specifies the alternate unit name for the local primary copy data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
**PCPY2_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the local backup copy data set.

**PCPY2_DATACLASS_ALT**

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

**PCPY2_EXPDT**

This keyword specifies the expiration date of the local backup copy data set on tape. A data set cannot have an expiration date and a retention period. The valid values are `yyyy/ddd` or `yyyy/yyyyy/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 BMC CHECK PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCCHECK_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC CHECK PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCCOPY_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC COPY PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCCOPY_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC COPY PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCCPRS_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility. BMCCPRS references the utility that copies statistics from the DB2 catalog to the BMCSTATS tables.

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

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility. BMCCPRS references the utility that copies statistics from the DB2 catalog to the BMCSTATS tables.

This keyword is not included in the AJXPOFIN input stream.

PROC_BMLOAD_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_BMLOAD_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.
Overriding options

**PROC_BMCTRIG_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMCTRIG component of the BMC DASD MANAGER PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCTRIG_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMCTRIG component of the BMC DASD MANAGER PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCUNLOAD_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC UNLOAD PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCUNLOAD_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC UNLOAD PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCUPRS_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility. BMCUPRS refers to the part of the utility that takes the statistics from the tables in the DASD MANAGER PLUS database (with the BMCSTATS utility) and updates the DB2 catalog with the statistics.

This keyword is not included in the AJXPOFIN input stream.
Overriding options

**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).
Overriding options

This keyword is not included in the AJXPOFIN input stream.

**PROC_IDCAMS_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM IDCAMS program.

This keyword is not included in the AJXPOFIN input stream.

**PROC_IDCAMS_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM IDCAMS program.

This keyword is not included in the AJXPOFIN input stream.

**PROC_IEFBR14_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM IEFBR14 job.

This keyword is not included in the AJXPOFIN input stream.

**PROC_IEFBR14_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM IEFBR14 job.

This keyword is not included in the AJXPOFIN input stream.

**PROC_TSO_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for IBM TSO.

This keyword is not included in the AJXPOFIN input stream.

**PROC_TSO_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for IBM TSO.
This keyword is not included in the AJXPOFIN input stream.

**PROC_USE=N**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies whether to generate a cataloged procedure (PROC) name instead of the EXEC PGM= statement for non-worklist JCL.

This keyword is not included in the AJXPOFIN input stream.

**PROC_USER_DEF_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for a user-defined program.

This keyword is not included in the AJXPOFIN input stream.

**PROC_USER_DEFINED**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for a user-defined program.

This keyword is not included in the AJXPOFIN input stream.

**PUNCH_DATACLASS**

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

**PUNCH_EXPDT**

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

**PUNCH_MGMTCLASS**

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

**PUNCH_PREFIX=&PREFIX..&WKID..&STEPN**

When a limit key is changed, this keyword specifies the prefix of the data set that contains discard rows from the last partition of a table space after the table space is reorganized.
PUNCH_PRIQTY=1

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

PUNCH_RETPD

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

PUNCH_SECQTY=1

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

PUNCH_STORCLASS

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

PUNCH_UNIT=SYSDA

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

RCPY1_DATACLASS

This keyword specifies the SMS definition for the data class associated with the recovery primary copy data set.

RCPY1_DATACLASS_ALT

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

RCPY1_EXPDT

This keyword specifies the expiration date of the recovery primary copy data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.
**RCPY1_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the recovery primary copy data set.

**RCPY1_MGMTCLASS_ALT**

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

**RCPY1_PREFIX=&PREFIX..&OBNOD..P&PART**

This keyword specifies the prefix for the name of the recovery primary copy data set.

**RCPY1_PRIQTY=10**

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

**RCPY1_RETPD**

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

**RCPY1_SECQTY=2**

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

**RCPY1_STACK=N**

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

**RCPY1_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the recovery primary copy data set.

**RCPY1_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the recovery primary copy data set (used if the threshold is exceeded).
**RCPY1_SUPPRESS_SUFF=N**

This keyword specifies whether to suppress adding the DD name to the end of the name of the recovery primary copy data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

**RCPY1_THRESH=0**

This keyword specifies the maximum anticipated size for the recovery primary copy data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

**RCPY1_UNIT=SYSDA**

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

**RCPY1_UNIT_ALT**

This keyword specifies the alternate unit name for the recovery primary copy data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**RCPY2_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the recovery backup copy data set.

**RCPY2_DATACLASS_ALT**

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

**RCPY2_EXPDT**

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

**RCPY2_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the recovery backup copy data set.
RCPY2_MGMTCLASS_ALT

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

RCPY2_PREFIX=&PREFIX..&OBNOD..P&PART

This keyword specifies the prefix for the name of the recovery backup copy data set.

RCPY2_PRIQTY=10

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

RCPY2_RETDP

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

RCPY2_SECQTY=2

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

RCPY2_STACK=N

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

RCPY2_STORCLASS

This keyword specifies the SMS definition for the storage class associated with the recovery backup copy data set.

RCPY2_STORCLASS_ALT

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

RCPY2_SUPPRESS_SUFF=N

This keyword specifies whether to suppress adding the DD name to the end of the name of the recovery backup copy data set (Y or N). If you specify Y, you must ensure that the data set name is unique.
**RCPY2_THRESH=0**

This keyword specifies the maximum anticipated size for the recovery backup copy data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

**RCPY2_UNIT=SYSDA**

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

**RCPY2_UNIT_ALT**

This keyword specifies the alternate unit name for the recovery backup copy data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**REBINDFAIL=N**

This keyword specifies whether worklist execution continues if a rebind fails. If **REBINDFAIL=Y**, worklist execution stops with a return code of 8. The stop is noted in the sync tables, and an Execution restart continues with the command that caused the failure. If **REBINDFAIL=N**, worklist execution continues.

**REBINDRC**

This keyword allows worklist execution to continue if a rebind fails, but returns a value for a final condition code instead of 4, the default value. Execution writes warning messages to AEXPRINT but does not post entries in the sync tables.

**RECOVER+_LOAD**

This keyword specifies the name of the LINK library for the BMC RECOVER PLUS utility. The BMC_RECOVER_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_RECOVER_LOAD and RECOVER+_LOAD are included in the POF, the components use the value that is specified for RECOVER+_LOAD.

**RECOVERDOPT**

This keyword specifies the name of the installation options module for the BMC RECOVER PLUS utility. The BMC_RECOVER_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_RECOVER_OPTS and RECOVERDOPT are included in the POF, the components use the value that is specified for RECOVERDOPT.
REGION=0M

This keyword defines the REGION parameter in the EXEC statement.

REORG_MAPTAB

This keyword specifies the name of the mapping table that the IBM REORG or BMC REORG PLUS utility uses to map the row IDs (RIDs) in the source table to the RIDs in the target table. ALTER and CHANGE MANAGER use this value when reorganizing a table space by using an online reorganization (SHRLEVEL CHANGE). CATALOG MANAGER and DASD MANAGER PLUS do not use this value. The name can be from 1 to 72 characters long, and can contain the &ZUSER or &USERID symbolic variable.

NOTE

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

REORG+_LOAD

This keyword specifies the name of the LINK library for the BMC REORG PLUS utility. The BMC_REORG_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_REORG_LOAD and REORG+_LOAD are included in the POF, the components use the value that is specified for REORG+_LOAD.

REORGDOPT

This keyword specifies the name of the installation options module for the BMC REORG PLUS utility. The BMC_REORG_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_REORG_OPTS and REORGDOPT are included in the POF, the components use the value that is specified for REORGDOPT.

REPT_DATACLASS

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

REPT_DATACLASS_ALT

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

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

**REPT_MGMTCLASS**

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

**REPT_MGMTCLASS_ALT**

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

**REPT_PREFIX=&PREFIX..&WKID**

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

**REPT_PRIQTY=10**

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

**REPT_RETPD**

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

**REPT_SECQTY=2**

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

**REPT_STORCLASS**

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

**REPT_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the report data set (used if the threshold is exceeded).
REPT_THRESH=0

This keyword specifies the maximum anticipated size for the report data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

REPT_UNIT=SYSDA

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

REPT_UNIT_ALT

This keyword specifies the alternate unit name for the report data set, if the threshold value is exceeded. The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

RUNTIME_HLQ

This keyword specifies a high-level qualifier (HLQ) for ISPF data sets for the batch ISPF environment. During installation, if you chose to use the runtime enablement feature, the Installation System set this value to an HLQ for user runtime libraries. If you chose not to use the feature, the Installation System set the value to an HLQ for Execution.

SCHED_TRIG_CTLM_JOBS=N

This keyword specifies whether JCL Generation should generate BMC Control-M job schedule entries for jobs that do not contain IEFBR14 steps (Y or N). If the value is Y, JCL Generation generates an input data set (CNTLMSCH) and an output data set (CNTLMOUT) in the JCL for the BMCTRIG utility.

SORTWK_NBR=4

This keyword specifies the number of SORTWORK data sets. Valid values are 1 through 32.

SORTWK_PRIQTY=10

This keyword specifies the primary allocation (in cylinders) for the DATAWORK, LOGSORT, or SORTWORK data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.
**SORTWK_SECQTY=2**

This keyword specifies the secondary quantity (in cylinders) for the DATAWORK, LOGSORT, or SORTWORK data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**SORTWK_UNIT=SYSDA**

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

**SQLEXP_LOAD**

This keyword specifies the name of the LINK library for the BMC SQL Explorer for DB2 product.

---

**TIP**

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

---

**SORTOUT_DATACLASS**

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

**SORTOUT_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).

**SORTOUT_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.

**SORTOUT_MGMTCLASS**

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

**SORTOUT_MGMTCLASS_ALT**

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

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

**SRTOUT_PRIQTY=10**

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

**SRTOUT_RETPD=**

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

**SRTOUT_SECQTY=2**

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

**SRTOUT_STORCLASS**

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

**SRTOUT_STORCLASS_ALT**

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

**SRTOUT_THRESH=0**

This keyword specifies the maximum anticipated size for the SORTOUT data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

**SRTOUT_UNIT=SYSDA**

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

This keyword specifies the alternate unit name for the SORTOUT data set, if the threshold value is exceeded. The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**STEP_INCLUDE_MEMBER**

This keyword specifies the name of a JCL member to be included after each step in the JCL.

**STOPWAIT=3**

This keyword specifies the number of intervals to wait for a DB2 STOP command to stop a database or table space. 0 indicates that if the object does not stop, the worklist stops without waiting. The maximum value allowed for this keyword is 10 (which is 280 seconds).

**STOPWTSECS=10**

This keyword specifies the number of seconds to wait (during the first interval of the STOPWAIT keyword) for a DB2 STOP command to stop a database or table space.

**SUPPRESS_COMMENTS=N**

This keyword specifies whether to suppress the comments in the generated JCL (Y or N).

**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.
Chapter 3  Setting up DASD MANAGER PLUS  219

**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.

TIMEPARM

This keyword indicates the time limit in minutes for each step in a batch job stream.

TRTCH

This keyword specifies the parity, data conversion, translation, and compression for 7-track drives. The following values are valid:

- **B** does not use a 7-track drive
- **C** uses odd parity, conversion on, and translation off
- **E** uses even parity, conversion off, and translation off
- **T** uses odd parity, conversion off, and translation on
- **ET** uses odd parity, conversion off, and translation on
- **COMP** uses data compression on
- **NOCOMP** uses data compression off
TSOPROGRAM

This keyword specifies an alternate TSO monitor program for standard JCL. TSOPROGRAM is available for nonworklist JCL.

TSOSUBEXIT=N

This keyword specifies whether to use a TSO submit exit to generate job cards. If TSOSUBEXIT=Y, no job cards are put in the JCL (Y or N).

ULLQ

This keyword specifies the LLQ for user-defined data sets for the installation environment. During installation, if you chose to use the runtime enablement feature, the Installation System set this value to blank, and you should not change the value. If you chose not to use the feature, the Installation System set the value to UDB.

NOTE
Changing the qualifier of the data sets might cause unpredictable results. Do not change the LLQ for the data sets.

UNLD_FREF_DATACLASS

For ALTER and CHANGE MANAGER, this keyword specifies the SMS definition for the data class associated with the file reference (SYSREC) data set.

UNLD_FREF_DIRBLOCK = 250

For ALTER and CHANGE MANAGER, this keyword specifies the number of directory blocks for the file reference (SYSREC) data set.

UNLD_FREF_MGMTCLASS

For ALTER and CHANGE MANAGER, this keyword specifies the SMS definition for the storage class associated with the file reference (SYSREC) data set.

UNLD_FREF_PREFIX = &PREFIX..&MSSID..&WORKID8

For ALTER and CHANGE MANAGER, this keyword specifies the prefix for the name of the file reference (SYSREC) data set.

UNLD_FREF_PRIQTY = 10

For ALTER and CHANGE MANAGER, this keyword specifies the primary allocation (in cylinders) for the file reference (SYSREC) data set. Valid values are 1 through 99999.
UNLD_FREF_SECQTY = 2

For ALTER and CHANGE MANAGER, this keyword specifies the secondary allocation (in cylinders) for the file reference (SYSREC) data set. Valid values are 1 through 99999.

UNLD_FREF_STORCLASS

For ALTER and CHANGE MANAGER, this keyword specifies the SMS definition for the storage class associated with the file reference (SYSREC) data set.

UNLD_FREF_SUPPR_SUFF=N

For ALTER and CHANGE MANAGER, this keyword specifies whether to suppress adding the DD name to the end of the name of the file reference (SYSREC) data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

UNLD_FREF_UNIT = SYSDA

For ALTER and CHANGE MANAGER, this keyword specifies the unit for the file reference (SYSREC) data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

UNLD1_DATACLASS

This keyword specifies the SMS definition for the data class associated with the primary unload (SYSREC) data set.

UNLD1_DATACLASS_ALT

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

UNLD1_EXPDT

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

UNLD1_MGMTCLASS

This keyword specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set.
**UNLD1_MGMTCLASS_ALT**

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

**UNLD1_PREFIX=&USERID..&MSSID..&WORKID8**

This keyword specifies the prefix for the name of the primary unload (SYSREC) data set.

**UNLD1_PRIQTY=10**

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

**UNLD1_RETPD**

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

**UNLD1_SECQTY=2**

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

**UNLD1_STACK=N**

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

**UNLD1_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set.

**UNLD1_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set (used if the threshold is exceeded).
**UNLD1_SUPPRESS_SUFF=N**

This keyword specifies whether to suppress adding the DD name to the end of the name of the primary unload (SYSREC) data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

**UNLD1_THRESH=0**

This keyword specifies the maximum anticipated size for the primary unload (SYSREC) data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

**UNLD1_UNIT=SYSDA**

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

**UNLD1_UNIT_ALT**

This keyword specifies the alternate unit name for the primary unload (SYSREC) data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**UNLD2_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the backup unload (SYSREC) data set.

**UNLD2_DATACLASS_ALT**

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

**UNLD2_EXPDT**

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

**UNLD2_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set.
**UNLD2_MGMTCLASS_ALT**

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

**UNLD2_PREFIX=&USERID..&MSSID..&WORKID8**

This keyword specifies the prefix for the name of the backup unload (SYSREC) data set.

**UNLD2_PRIQTY=10**

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

**UNLD2_RETPD**

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

**UNLD2_SECQTY=2**

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

**UNLD2_STACK=N**

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

**UNLD2_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set.

**UNLD2_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set (used if the threshold is exceeded).
**UNLD2_SUPPRESS_SUFF=N**

This keyword specifies whether to suppress adding the DD name to the end of the name of the backup unload (SYSREC) data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

**UNLD2_THRESH=0**

This keyword specifies the maximum anticipated size for the backup unload (SYSREC) data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

**UNLD2_UNIT=SYSDA**

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

**UNLD2_UNIT_ALT**

This keyword specifies the alternate unit name for the backup unload (SYSREC) data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**UNLD3_DATACLASS**

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the SMS definition for the data class associated with the ROWID SYSREC data set.

**UNLD3_EXPDT**

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the expiration date of the ROWID SYSREC data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyyyddd 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.
Overriding options

**UNLD4_STORCLASS**

For the LOB DATA MOVER program in the Database Administration solution, this keyword specifies the SMS definition for the storage class associated with the large object (LOB) SYSREC data set.

**UNLD4.Suppress_Suff=N**

For the LOB DATA MOVER program in the Database Administration solution, this keyword specifies whether to suppress adding the DD name to the end of the name of the large object (LOB) SYSREC data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

**UNLD4.Unit=SYSDA**

For the LOB DATA MOVER program in the Database Administration solution, this keyword specifies the unit for the large object (LOB) SYSREC data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**UNLOAD+_LOAD**

This keyword specifies the name of the LINK library for the BMC UNLOAD PLUS utility. The BMC_UNLOAD_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_UNLOAD_LOAD and UNLOAD+_LOAD are included in the POF, the components use the value that is specified for UNLOAD+_LOAD.

**UNLOADDOPT**

This keyword specifies the name of the installation options module for the BMC UNLOAD PLUS utility. The BMC_UNLOAD_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_UNLOAD_OPTS and UNLOADDOPT are included in the POF, the components use the value that is specified for UNLOADDOPT.

**USER_VAR1_CHAR**
**USER_VAR2_CHAR**
**USER_VAR3_CHAR**
**USER_VAR4_CHAR**
**USER_VAR5_CHAR**

These keywords specify user-defined character variables. Each variable has a corresponding symbolic variable that you can use in job cards or data set prefixes. The maximum length of a variable name is eight characters.
**WORK_DATACLASS**

This keyword defines the Data Facility Storage Management Subsystem (DFSMS or SMS) data class name that the components use at data-set allocation time to define allocation attributes of the data set. A data class name is not required, even for SMS data sets. This option appears as "DATACLAS=" in the JCL for workfiles.

**WORK_MGMTCLASS**

This keyword defines the DFSMS or SMS management class name that the components use at data-set allocation time to define the migration, retention, and backup requirements of the data set. This option appears as "MGMTCLAS=" in the JCL for workfiles.

**WORK_STORCLASS**

This keyword defines the DFSMS or SMS storage class name that the components use at data-set allocation time to define processing requirements of the data set. This option appears as "STORCLAS=" in the JCL for nontape work files.

**Overriding installation options**

Utilities use installation options only in the absence of any other definitions. You can override installation options in several places, such as in the utility parameters, job generation parameters, TSO ISPF profile, or SLIB.

If BMCTRIG generates the utility, overriding installation options is a similar process. BMCTRIG uses information from the sources in Table 25.

**Table 25  Information sources for BMCTRIG utility**

<table>
<thead>
<tr>
<th>Order of use</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Installation options and default POF</td>
</tr>
<tr>
<td>2</td>
<td>POF of default action</td>
</tr>
<tr>
<td>3</td>
<td>utility parameters in the default action</td>
</tr>
<tr>
<td>4</td>
<td>BMCTRIG parameters in the BMCTRIG action</td>
</tr>
<tr>
<td>5</td>
<td>SLIB</td>
</tr>
</tbody>
</table>

If you want BMCTRIG to generate any utility with a separate POF, you must edit the corrective action option JCL Options DSN as shown in Figure 44, not the BMCTRIG action.
Utilities other than BMCTRIG use information from the sources in Table 26.

### Table 26  Information sources for utilities other than BMCTRIG

<table>
<thead>
<tr>
<th>Order of use</th>
<th>Source</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>installation options and default POF</td>
<td>everyone</td>
</tr>
<tr>
<td>2</td>
<td>ISPF/User Options</td>
<td>any job that you generate</td>
</tr>
<tr>
<td>3</td>
<td>action POF</td>
<td>anyone using the action</td>
</tr>
<tr>
<td>4</td>
<td>job generation</td>
<td>anyone running the job</td>
</tr>
<tr>
<td>5</td>
<td>override options</td>
<td>current generation only</td>
</tr>
<tr>
<td>6</td>
<td>utility parameters</td>
<td>anyone using the action</td>
</tr>
<tr>
<td>7</td>
<td>SLIB</td>
<td>everyone</td>
</tr>
</tbody>
</table>

**NOTE**

SLIB definitions are operating-system-specific or site-specific and represent an overall policy rather than the choices of a single user. If a conflict occurs between installation options or POF and SLIB definitions, the SLIB definition prevails.

BMCTRIG parameters, utility parameters, and installation options resolve into AJX variables that SLIB stores and that JCL generation uses. For more information about AJX variables, see Table 24.
Variables

DASD MANAGER PLUS variables begin with one or two ampersands (&), depending upon their location, as Table 27 shows.

Table 27  Variable option notation

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>TSO ISPF Profile, user options, and DASD MANAGER PLUS panels</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>JCL for installation options module&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

<sup>a</sup> The assembly code requires the installation options module to contain double ampersands to correctly convert the options into symbolic variables.

In the JCL for the installation options module, the default job card shows two ampersands before each installation option and a period delimiter between consecutive installation options, as follows:

```
JCL='//&&USERID.&JOBSEQ JOB (ACCT),''&&PGMR'',,'  
```

On DASD MANAGER PLUS panels, installation options use a symbolic parameter format with a single ampersand. For example, in the job card at the end of the BMCTRIG dialog (Figure 45), the &USERID installation option has one ampersand and a period delimiter between consecutive options.

Figure 45  Installation options format

```
//&USERID.&JOBSEQ JOB (5712),''HENDERSON'',  
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),  
// NOTIFY=RDAMLH  
//*  
//*
```

Therefore, if the logon ID were RND and the &JOBSEQ parameter started at 001, the variable in Figure 45 resolves to RND001.

Installation options

Figure 46 contains the installation options listing that the installation process creates for DASD MANAGER PLUS. The installation options module that the installation process creates resides in $xmnDOPT and also in HLQ.CNTL (where HLQ is the high-level qualifier) with the same member name as the installation options module. For descriptions of individual installation options, see “Descriptions of the installation options” on page 234.
Figure 46  DASD MANAGER PLUS installation options

```
*                   *                                           *
* MODULE : ASUOPTS  * FUNCTION : DASD DEFAULT OPTIONS         *
* COPYRIGHT : COPYRIGHT BMC SOFTWARE INC., 2013             *
* LEVEL : RELEASE 11.1 June 2013                           *
* FUNCTIONS : DEFINE THE DEFAULT PROFILE VARIABLES        *
*                                                            *
ASUOPTS $ALUOPT PRODUCT='DASD MANAGER',                     *
  DATE=&SYSDATC,                                             *
  JDSN='''ASU.DEDK.V111QA.JCLLIB(&&WKID)''',                 *
  SYSTRIGS=N,                                               *
  DB2NTRY=1,                                                 *
  DB2WAIT=1,                                                 *
  WDSN='''ASU.DEDK.V111QA.WORKLIST(&&WKID)''',               *
  LOCATION=,                                                *
  EPP=ASU111XM,                                             *
  EAP=ASU111XA,                                             *
  EIP=TISINSTL,                                             *
  POFDS=('AUS.DOPSEC.CNTL(PFASUB1)',R),                     *
  OPNDB2ID=Y,                                               *
  OSAUTHCHK=N,                                              *
  STATAUTH=(Y,R),                                           *
  DPNAM=,                                                   *
  ISPTLIB='ASU.QATEST.TLIB',                                *
  UPDCATIXS=N,                                              *
  UPDNUCAT=Y,                                               *
  CPLAN= ASU111DC,                                          *
  JPLAN=ASU111DJ,                                           *
  SPLAN=ASU111DS,                                           *
  ZPLAN=ASU111DZ,                                           *
  RPLAN=ASU111DR,                                           *
  WU=XXXXXX,                                                *
  SWU=SYSDA,                                                *
  WPS=10,                                                   *
  WSS=2,                                                    *
  BMCSYNC=Y,                                                *
  FREQVAL=Y,                                                *
  ATBWORKAREA=N,                                            *
  QUIESCETIMEVAL=000000,                                    *
  OPTIMIZECOMMIT=Y,                                         *
  PERFADVLOC=                                              *
END
```

**NOTE**

*R* in the variable syntax indicates that the specified value will refresh the variable’s existing value 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.
Overriding options

Descriptions of the installation options

This section describes the installation options and plan names for DASD MANAGER PLUS. For information about using the DASD MANAGER PLUS User Options panels to customize some of these parameters, see “Using options to control your environment” on page 82.

### NOTE
The POF defines many user options that the installation options module defined. For additional information about POFs and to see a sample, see the *Administrative Products for DB2 Installation Guide*. For information about the keywords and options, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

The following installation options and plan names are available to DASD MANAGER PLUS:

**ATBWORKAREA=N**

This option specifies whether the cardinality and frequency work area will be above the 64 bit bar, and if the collection routine will run in AMODE 64. Valid values are Y (yes) and N (no).

**BMCSYNC=Y**

This option specifies whether to use the BMCSYNC table. The BMCSYNC table synchronizes access to DB2 spaces that are concurrently executing BMC utility products. By default, the BMCSYNC table synchronizes access (Y). If you set this option to N, the BMCSYNC table will not synchronize access; setting the option to N also bypasses BMCUTIL table access, UTILID enqueue logic, and object name enqueue logic that is used for BMC utility concurrency control. Turning this feature off can lead to VSAM data set access failure in BMCSTATS or other utilities due to utility conflicts that are no longer detected. Valid values are Y (yes) and N (no).

**CPLAN=ASUvrDC**

DASD MANAGER PLUS does not use this plan.

**DATACLAS=N**

This option indicates whether support for the DATACLAS parameter is required for VCAT-defined DB2 objects. Valid values are Y (yes) and N (no).

The POF keywords that end with “_DATACLASS” replace this installation option. For information about the POF keywords, see “Product options” on page 149.
DATE=&SYSDATC

Use this option only if your assembler is ASMA90.

DB2CAT='DBDBCAT'

This option specifies the VSAM CATALOG alias that contains the data sets of the DB2 catalog.

DB2NTRY=100

For BMCSTATS and BMCTRIG, this option specifies the maximum number of times to retry gaining control of the relevant tables after an initial attempt fails. For BMCSTATS, DB2NTRY applies to BMCSYNC, and SYSCOPY tables. For BMCTRIG, DB2NTRY applies only to the SYSCOPY table. This option is also used in BMCSTATS table insert and delete logic. Valid values are 1 through 255. Use this option with DB2WAIT.

DB2WAIT=3

For BMCSTATS and BMCTRIG, this option specifies how long to wait (in seconds) between attempts to use the relevant table. For BMCSTATS, DB2WAIT applies to BMCSYNC, and SYSCOPY tables. For BMCTRIG, DB2WAIT applies only to the SYSCOPY table. This option is also used in BMCSTATS table insert and delete logic.

When another process is controlling any of these tables (making them unavailable), BMCSTATS or BMCTRIG waits for the number of seconds that DB2WAIT specifies and then tries again to use the table. BMCSTATS or BMCTRIG repeats the attempt up to the number of times that DB2NTRY specifies. Valid values are 1 through 655.

NOTE

The DB2WAIT waiting time is in addition to the DB2 resource timeout and utility values IRLMRWT and UTIMOUT which are set in DSNZPARM.

DPNAM=DPDELOAD

This option specifies the load module name for the BMC Software DATA PACKER product.

EAP=AEXvrDA

This option specifies the Execution Authorization plan, which determines whether a user is authorized to run Execution.
Overriding options

**EIP=DCIINSTL**

This option specifies the installation plan.

**EPP=AEXvrDM**

This option specifies the Execution primary plan.

**FREQVAL=Y**

This option indicates whether BMCSTATS should collect frequency statistics for columns when using the TABLE option. Valid values are Y (yes) and N (no).

**GDGDEF=N**

This option indicates whether JCL Generation creates the base of the generation data group (GDG) data set. Valid values are Y (yes) and N (no).

The DEF_GDG_BASE POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**GDGLIM=0**

This parameter specifies the number of GDG data sets to allow. GDGLIM contains the value of the LIMIT parameter. Valid values are 0 through 255. If you specify GDGLIM = 0, GDGDEF is automatically set to N.

The DEF_GDG_LIMIT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**ISPTLIB='HLQ.TLIB'**

This option is the fully qualified data set name for the ISPF TLIB. BMCTRIG uses this data set name.

**JC1=’//&&USERID.&JOBCHAR JOB (ACCT),’&PGMR’,**
**JC2=’// CLASS=A,MSGCLASS=X, MSGLEVEL=(1,1),’**
**JC3=’// NOTIFY=&&USERID’**
**JC4=’/****
**JC5=’/****

Options JC1 through JC5 are the default JOB statement that DASD MANAGER PLUS uses when generating JCL. You can use symbolic variables for the option values. For more information, see Table 31.
The JOBCARD1, JOBCARD2, JOBCARD3, JOBCARD4, and JOBCARD5 keywords replace these installation options. For information about the POF keywords, see “Product options” on page 149.

**JCLCLEAN=N**

This option specifies whether to automatically delete many of the permanent work data sets that the Execution component creates during worklist processing. Valid values are Y (yes) and N (no). If you specify Y, BMCTRIG deletes the work data sets that have a disposition (NEW, CATLG, CATLG). The product performs the automatic delete step only if the condition code that any previous service returns is 4 or less.

The JCLCLEANUP POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**JDSNE=''&&PREFIX..EXEC(&&WKID8)'''**

This option specifies the default data set name that is used for Execution JCL. This data set can be either a sequential file or a partitioned data set (PDS), but the data set must already exist because DASD MANAGER PLUS does not preallocate it. If the data set is partitioned, you must specify a member name or a variable that resolves to a member name.

**JPLAN=ASUvrDJ**

This option specifies the name of the DASD MANAGER PLUS plan that the BMCTRIG action generation uses.

**LOCATION**

This option is no longer used.

**MGMTCLAS=N**

This option indicates whether support for the MGMTCLAS parameter is required for VCAT-defined DB2 objects. Valid values are Y (yes) and N (no).

The POF keywords that end with “_MGMTCLASS” replace this installation option. For information about the POF keywords, see “Product options” on page 149.

**OPNDB2ID=Y**

This option grants the DB2 authorization ID to update BMCSTATS tables. This installation option allows users with STATS authority to collect stats even if their logon IDs do not have RACF authority to read the data set. Valid values are Y (yes) and N (no).
Overriding options

**OPTIMIZECOMMIT=Y**

This option specifies whether to do an SQL commit after every insert to the stats tables. Specifying N indicates that the user wants BMCSTATS to do the commit after every insert. Valid values are Y (yes) and N (no).

---

**NOTE**

This option can minimize SQL -911 errors, but is costly in terms of DB2 processing time.

---

**OSAUTHCHK=N**

This option specifies whether to check authorization for updates to elements defined within DASD MANAGER PLUS, specifically object sets, services, service syntax, and actions. Valid values are Y (yes) and N (no). If you specify OSAUTHCHK=Y, only the owner of an element can edit or delete it. For more information, see “Controlling access to actions, services, and service syntax” on page 307 and “Controlling access to object sets” on page 340.

---

**POFDS='&&HLQ..CNTL(&POFNAME)'**

This option specifies the name of the JCL Generation product options file (POF).

---

**PRODUCT='PRODUCT NAME'**

This option specifies the product name, such as PRODUCT='DASD MANAGER'.

---

**QUIESCEINTERVAL=0000000**

This option specifies the number of partitions that BMCSTATS processes, after which BMCSTATS consolidates and releases excess storage.

The option is intended for limited use to relieve main storage constraints imposed by very large object sets and should not be used unless storage occupancy is a problem.

The range is 10000 through 1000000 partitions.

---

**NOTE**

Specifying a value that is less than the lower range limit might result in high CPU processing consumption.
RECVMAX=100

This option is the Recovery DDN threshold, in cylinders. If the size of a data set exceeds the threshold, the utility uses the secondary allocation unit. To avoid using the secondary unit, specify 0.

The RCPY1_UNIT_ALT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

RECVMAXU=TAPE

This option specifies the Recovery DDN secondary, or alternate, unit if the required size exceeds the threshold.

The RCPY1_UNIT_ALT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

RECVPREF='&&PREFIX..&&OBNODE..P&&PART'

This option specifies the default prefix (high-level qualifier) for the RECVnnn recovery data sets. The &&OBNODE symbolic variable resolves to database.&SPNAME. The &SPNAME variable resolves to a table space name or to an index space name, depending on the type of object that is being copied.

The RCPY1_PREFIX POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

RECVPS=10

This option specifies the default primary space allocation, in cylinders, for RECVnnn recovery data sets.

The RCPY1_PRIQTY POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

RECVSS=2

This option specifies the default secondary space allocation, in cylinders, for RECVnnn recovery data sets.

The RCPY1_SECQTY POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

RECVUNIT=SYSDA

This option is the default unit that is used for creating RECVnnn recovery data sets.
Overriding options

The RCPY1_UNIT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

RPLAN=ASUvrDR

This option specifies the DASD MANAGER PLUS plan for displaying online reports.

SL1

This option indicates the STEPLIB library that contains the DASD MANAGER PLUS load modules. Keywords SL1, SL2, and SL3 indicate the libraries from which DASD MANAGER PLUS should load DB2 load modules.

The DB2EXIT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

SL2=('''SYS1.DSNEXIT''')

This option specifies the optional first STEPLIB library for DB2 load modules. This library is concatenated to the library that keyword SL1 specifies.

The DB2LOAD POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

SL3=('''SYS1.DSNLOAD''')

This option specifies the optional second STEPLIB library for DB2 load modules. This library is concatenated to the library that keywords SL1 and SL2 specify.

The ADDLOAD2 POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

SL4=('''SYS1.OTHER.LOADLIB1''')

This option specifies optional additional STEPLIB libraries.

The EXEC_LOAD POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

SL5=('''SYS1.OTHER.LOADLIB2''')

This option specifies optional additional STEPLIB libraries.

The ADDLOAD1 POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.
SPLAN=ASUvrDS

This option specifies the DASD MANAGER PLUS plan for statistics collection.

STATAUTH=Y

This option specifies whether DASD MANAGER PLUS will check users’ authorization to run BMCSTATS. Valid values are Y (yes) and N (no). If you use Y, the product requires the same authorization as for RUNSTATS.

STORCLAS=N

This option indicates whether support for the STORCLAS parameter is required for VCAT-defined DB2 objects. Valid values are Y (yes) and N (no).

The POF keywords that end with "_STORCLASS" replace this installation option. For information about the POF keywords, see “Product options” on page 149.

SWPS=10

This option specifies the primary space allocation, in cylinders, for sort work data sets.

The SORTWK_PRIQTY POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

SWSS=2

This option specifies the secondary space allocation, in cylinders, for sort work data sets.

The SORTWK_SECQTY POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

SWU=SYSDA

This option specifies the sort work unit.

The SORTWK_UNIT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.
**SYSCMAX=100**

This option specifies the SYSCOPY threshold, in cylinders. If the size of a data set exceeds the threshold, the utility uses the secondary unit. To avoid using the secondary allocation unit, specify 0. The SYSCMAX parameter generates SYSCOPY DD statements; the product does not use SYSCMAX when you choose COPY PLUS Dynamic Allocation.

The PCPY1_THRESH POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSCMAXU=TAPE**

This option specifies the SYSCOPY secondary, or alternate, unit if the required size exceeds the threshold.

The PCPY1_UNIT_ALT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSCPREF='&&PREFIX..&&OBNOD..P&&PART'**

This option specifies the default prefix (high-level qualifier) for the SYSCOOnnn data sets. The &&OBNOD symbolic variable resolves to database.&SPNAME. The &SPNAME variable resolves to a table space name or to an index space name, depending on the type of object that is being copied.

The PCPY1_PREFIX POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSCPS=10**

This option specifies the primary space allocation, in cylinders, for SYSCOOnnn data sets.

The PCPY1_PRIQTY POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSCSS=2**

This option specifies the secondary space allocation, in cylinders, for SYSRCOnnn data sets.

The PCPY1_SECQTY POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSCUNIT=SYSDA**

This option specifies the default UNIT for creating SYSCOOnnn data sets.
The PCPY1_UNIT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSRMAX=100**

This option specifies the SYSREC threshold, in cylinders, above which the utility will use the secondary unit for allocation. If the size of a data set exceeds the threshold, the utility uses the secondary unit. To avoid using the secondary unit, specify 0.

The UNLD1_THRESH POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSRMAXU=TAPE**

This option specifies the SYSREC secondary, or alternate, unit if the required size exceeds the threshold.

The UNLD1_UNIT_ALT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSRPREF='&&PREFIX..&&OBNOD'**

This option specifies the default prefix (high-level qualifier) for the SYSREnnn data sets.

The UNLD1_PREFIX POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSRPS=10**

This option specifies the primary space allocation, in cylinders, for SYSREnnn data sets.

The UNLD1_PRIQTY POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSRSS=2**

This option specifies the secondary space allocation, in cylinders, for SYSREnnn data sets.

The UNLD1_SECQTY POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSRUNIT=SYSDA**

This option specifies the default UNIT for creating SYSREnnn data sets.
The UNLD1_UNIT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**SYSTRIGS=N**

This option indicates whether BMCTRIG should apply the system exception thresholds, corrective actions, and object-action priorities by default.

- Y uses the new system definitions without requiring changes to existing BMCTRIG jobs and treats the syntax options as overrides
- N allows BMCTRIG jobs to continue running as is using the syntax options
- F causes any instream exceptions to be ignored and uses only the system exception thresholds and corrective actions

**SYSTYPE=S**

This option indicates whether DB2 subsystem character strings can contain a mixture of SBCS and DBCS data, as follows:

- M allows mixed data
- S only single-byte data

**SZDEVT=3380**

This option specifies the device type for data set sizing for JCL Generation. Valid values are 3380 and 3390.

The SZDEVT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**TAPE1=CART**

**TAPE2=TAPE**

**TAPE3=TAPE**

Options TAPE1 through TAPE3 specify the valid installation tape unit names for your site.

The TAPE1, TAPE2, and TAPE3 POF keywords replace these installation options. For information about the POF keyword, see “Product options” on page 149.

**UPDCATIXS=N**

This option indicates whether to update DB2 catalog statistics columns with index sampled statistics. Valid values are Y (yes) and N (no).
**WARNING**

Index sampling might produce statistics that cause SQL optimizer access selection problems. BMC recommends that you review the index sampled statistics for your indexes before deciding to use them for updating the catalog.

**UPDNUCAT=Y**

This option indicates whether to update nonupdatable DB2 catalog statistics columns. Valid values are Y (yes) and N (no). Nonupdatable statistics are catalog statistics that you cannot update by using SQL.

**NOTE**

If you are migrating installation option values from a version earlier than 9.1.00 of DASD MANAGER PLUS, the Installation System sets the UPDNUCAT installation option to Y, regardless of its previous setting. If you want to set this option to N, you can do so after installing this product.

**WDC=DATACLASS**

This option specifies the Data Facility Storage Management Subsystem (DFSMS or SMS) data class name (used at data set allocation time) to define the allocation attributes of the data set. A data class name is not required, even for SMS data sets. WDC appears as "DATACLASS= " in the JCL for workfiles.

The WORK_DATACLASS POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

**WDSN=’’&PREFIX..&SSID..&WKID8’’**

This option specifies the default worklist data set name for a new action. This data set can be either a sequential file or a partitioned data set (PDS), but the data set must already exist because DASD MANAGER does not preallocate it.

**WMC=MGMTCLASS**

This option specifies the IBM Storage Management Subsystem (SMS) management class name (used at data set allocation time) to define the migration, retention, and backup requirements of the data set. WMC appears as "MGMTCLASS= " in the JCL for workfiles.

The WORK_MGMTCLASS POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.
Post-installation considerations

WPS=10

This option specifies the default work primary space allocation, in cylinders, for work data sets that include Analysis and Execution diagnostics.

The WPS POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

WSC=STORCLASS

This option specifies the SMS storage class name (used at data set allocation time) to define processing requirements of the data set. WSC appears as "STORCLASS=" in the JCL for nontape work files.

The WORK_STORCLASS POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

WSS=2

This option specifies the default work secondary space allocation, in cylinders, for work data sets that include Analysis and Execution diagnostics.

The WSS POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

WU=SYSDA

This option specifies the default work unit and is used during the dynamic bind process. Work data sets include Analysis and Execution diagnostics data sets.

The TEMPUNIT POF keyword replaces this installation option. For information about the POF keyword, see “Product options” on page 149.

ZPLAN=ASUvrDZ

This option specifies the plan that displays DASD MANAGER PLUS statistics and provides online maintenance of DASD MANAGER objects, including object sets, actions, services, and thresholds.

Post-installation considerations

Review this section for recommendations and requirements before you use DASD MANAGER PLUS.
Recommendations for reducing system resource requirements

The following features require additional memory, space, and CPU cycles:

- maximum lengths for the following 2000 byte statistics columns: HIGHKEY, HIGH2KEY, LOWKEY, and LOW2KEY
- partial statistics rollup for COLCARDDATA and COLDISTDATA, resulting in increased data (up to 3400 bytes)
- maximum length of 2000 bytes for the COLVALUE column for column data collection
- name lengths increased from 30 to 128 characters and from 8 to 128 for creator names

These features affect the amount of time required to process columns (based on column length and the number of variable-length columns used), indexes, and partitioned table spaces. To minimize the system requirements when running DASD MANAGER PLUS jobs, BMC offers the following recommendations:

- Carefully consider the values you plan to set for each of the following fields, based on which columns are needed for SQL optimization. Specifying certain values for these columns can be expensive in terms of CPU processing time and main storage occupancy.
  - **FREQTYPE**
    Consider specifying a value other than L (least frequent values) or B (both least and most frequent values).
  - **KEYCARD**
    Consider specifying N, which indicates not to count the cardinality of concatenated keys.

**NOTE**

DASD MANAGER PLUS version 11.1 or later executing on a DB2 Version 10 or later subsystem automatically collects keycard information. **KEYCARD N** is ignored in this scenario.

**KEYCARD Y** is forced when collecting statistics on data-partitioned secondary indexes (DPSIs).

- **NUMCOLS**
  Consider specifying 1 to collect frequent values only for the first key column.
— COUNT
   Consider using a low value when specifying the maximum number of frequent values to collect.

— TABLE ALL
   Consider specifying N and gathering statistics only on tables for which you really need to gather statistics.

- Consider collecting column statistics only when an object has changed significantly. To help determine which objects are candidates for new statistics collection, use BMCTRIG.

- Although BMC has increased the default DDL allocation size, you should review your current allocations and ensure that the new tables are large enough to accommodate the increased data.

- Review the limit that you set for the REGION parameter (the amount of virtual storage that the utility uses). For more information about this parameter, see the DASD MANAGER PLUS for DB2 Reference Manual.

- BMCSTATS processes all table spaces and indexes in the SYSIN stream during initialization. Doing so optimizes the collection of related object column statistics and enables multitasking but requires additional memory for processing. Consider using multiple jobs if you are processing large objects or analyzing large amounts of index or column data. You can use BMCTRIG to split the objects across multiple jobs, separating larger objects from smaller ones by using exclude rules. You can run BMCSTATS against the larger objects with a smaller number of tasks, and run BMCSTATS against the smaller objects with a larger number of tasks.

- If you are processing a large number of partitions, you might need to increase the size of the BMCSYNC table space from the standard size that was allocated during installation. Estimate this allocation based on all of the following factors:
  
  — number of utilities that you are running concurrently
  — number of partitions that you are processing concurrently
  — number of files that you are dynamically allocating

- If you are experiencing SQL -911 errors followed by BMCSTATS job termination, you might need to specify 911ACTION IGNORE so that BMCSTATS will ignore SQL -911 errors and continue processing.

   Objects that receive SQL -911 errors will still need to be reprocessed.
Contention issues

If you are experiencing SQL-911 errors or DB2 locking problems during the BMCSTATS stats table update phase, you can specify OPTIMIZECOMMIT NO to guarantee that commits occur after every SQL DELETE, INSERT, or UPDATE statement.

Beware that specifying OPTIMIZECOMMIT NO will increase BMCSTATS processing overhead.

If you are processing a very large number of objects and have experienced main storage depletion problems, you can specify QUIESCEINTERVAL (page 238).

Specifying a value in the range of 10000 through 1000000 for QUIESCEINTERVAL allows BMCSTATS to consolidate and release main storage after the specified number of partitions are processed. Do not specify the keyword if you have not encountered main storage depletion errors with BMCSTATS.

Contention issues

Contention problems among the DASD MANAGER tables that are related to page-level locking can occur when processing jobs in parallel. The installation process defines DASD MANAGER tables as LOCKSIZE PAGE. However, consider changing the LOCKSIZE value to ROW on the following tables:

- BMCASUnn.ACT_SEQ_SVC
- BMCASUnn.ACTION
- BMCASUnn.DO_WORKID
- BMCASUnn.EXCEPTIONS2
- BMCASUnn.SERVICE
- BMCASUnn.SVC_DD
- BMCASUnn.SVC_DD_DSN
- BMCASUnn.SVC_SNTX
- BMCASUnn.SVC_SNTXT
- BMCASUnn.WL_SYNC

The variable \textit{nn} is the two-digit version number of DASD MANAGER. Likewise, altering the PCTFREE or FREEPAGE values for the DASD MANAGER tables helps to control where inserts occur, eliminating contention problems when running several BMCSTATS jobs in parallel.
MEMLIMIT system parameter

Be aware that when specifying LOCKSIZE ROW, the Internal Resource Lock Manager (IRLM) must work harder, which can slow down response time on the subsystem when processing against large table spaces. Specifying larger PCTFREE and FREEPAGE values means that objects require more DASD space. Before you alter the DASD MANAGER PLUS tables, carefully evaluate the method that you choose to reduce contention.

For information about performing other post-installation tasks for DASD MANAGER PLUS, see the Administrative Products for DB2 Installation Guide.

MEMLIMIT system parameter

DASD MANAGER PLUS requires above-the-bar memory and might abend if it attempts to access storage and this memory is not available.

The default value for the System Management Facility (SMF) MEMLIMIT parameter is 2 GB. This value is set in member SMFPRMxx in SYS1.PARMLIB. Use any of the following methods if you need to override the default value:

- Specify REGION=0M, which gives you unlimited below-the-bar and above-the-bar memory.

- If REGION=0M is not specified, specify MEMLIMIT=NOLIMIT to allow unlimited above-the-bar memory; in this case, your z/OS system administrator can still control above-the-bar memory allocations by using the System Management Facility (SMF) user exit facilities.

- If the above-the-bar memory limits that are enforced through system defaults are inadequate, contact your z/OS systems administrator.

- If you specify ATBWORKAREA YES, above-the-bar memory requirements are significantly increased. ATBWORKAREA NO is the default.
Now that you have set up DASD MANAGER PLUS for your environment, you can proceed to the following tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>create and generate actions</td>
<td>chapter 4</td>
</tr>
<tr>
<td>define an object set</td>
<td>chapter 5</td>
</tr>
<tr>
<td>collect statistics</td>
<td>chapter 6</td>
</tr>
<tr>
<td>analyze statistical trends</td>
<td>chapter 7</td>
</tr>
<tr>
<td>use BMCTRIG to analyze objects</td>
<td>chapter 8</td>
</tr>
<tr>
<td>change the secondary quantity</td>
<td>chapter 9</td>
</tr>
<tr>
<td>produce reports</td>
<td>chapter 10</td>
</tr>
<tr>
<td>export object definitions</td>
<td>chapter 11</td>
</tr>
</tbody>
</table>
This chapter explains how to create, modify, and generate actions and related services for the DASD MANAGER PLUS product. An action identifies one or more services (utilities) that you want to generate. You can create, edit, and maintain actions. To use actions, select Utility Actions (WORKIDs) from the main menu.
Action maintenance and generation involves the following tasks:

- creating an action
- selecting the services
- selecting or specifying syntax options and parameters
- (optional) creating a user-defined service
- generating the action and specifying processing options
- verifying successful execution or restarting the job

**NOTE**

You can run actions online as you need them. You can also set up actions to run at scheduled times to automate maintenance. For example, you can create an action that copies and reorganizes all table spaces in a database, and then run that job weekly by using a job scheduler.

You can also use action definitions to automate maintenance by using BMCTRIG. For more information, see Chapter 8, “Analyzing objects by using BMCTRIG.”

**Actions**

An action is an ordered set of services with associated syntax. A service is a program (or utility). Service syntax is a named set of syntax options for a service.

In Figure 47, the action called WEEKLYREORG has three services:

- BMCREORG
- BMCSTATS
- BMCCOPY

Each of these services refers to a named service syntax. The action called TESTSYSREORG has two services, BMCCOPY and BMCREORG. Each of these services also references a named service syntax. BMCCOPY, however, shares service syntax with the BMCCOPY that is associated with the action WEEKLYREORG.
**Services**

Services are programs that DASD MANAGER PLUS runs as part of an action. Services are either of the following types:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC Software required services</td>
<td>DASD MANAGER PLUS provides these service definitions, allows you to edit the option-level syntax, and provides guidance. DASD MANAGER PLUS can easily run these services, which include most BMC Software and IBM utilities. Although you cannot delete these services, you can modify them somewhat.</td>
</tr>
<tr>
<td>user-defined services</td>
<td>You can define the attributes and syntax to include the service in an action.</td>
</tr>
</tbody>
</table>
## BMC Software required services

Table 28 lists the services that BMC Software provides. You can run the services in DASD MANAGER PLUS actions.

<table>
<thead>
<tr>
<th>Service type</th>
<th>Service name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics utilities</td>
<td>BMCSTATS  BMCTRIG  BMCCPRS  BMCUPRS  RUNSTATS  STOSPACE</td>
</tr>
<tr>
<td>Image Copy utilities</td>
<td>BMCCOPY  BMCCOPYI  BMCMOD  FULLCOPY  INCRCOPY  DSN1COPY  MERGECOPY  MODICOPY</td>
</tr>
<tr>
<td>Reorganize utilities</td>
<td>BMCREORG  REORG  RESIZE*  ALTERSEC</td>
</tr>
<tr>
<td>UNLOAD and LOAD utilities</td>
<td>BMCUNLOAD  UNLOAD  BMCLOAD  LOAD</td>
</tr>
<tr>
<td>DB commands and utilities</td>
<td>QUIESCE  REPAIR  START DB  STOP DB</td>
</tr>
<tr>
<td>Check and Report utilities</td>
<td>CHECK DATA  CHECK LOB  REPORT with RECOVERY parameter  REPORT with TABLESPACESET parameter</td>
</tr>
<tr>
<td>Recovery utilities</td>
<td>RECOVER</td>
</tr>
<tr>
<td>Worklist commands</td>
<td>STOP (BMC Execution Monitor)  SYNC (BMC Execution Monitor)</td>
</tr>
</tbody>
</table>
Table 29 describes the utilities by their command names and provides the names that are generated for the utilities in the worklists.

### Table 29 Command descriptions (part 1 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Worklist form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTERSEC</td>
<td>SQL</td>
<td>sets a threshold for the maximum number of extents in one or more table spaces or indexes, and increases the secondary space allocation by a specified percentage if the threshold is exceeded</td>
</tr>
<tr>
<td>BMCCOPY</td>
<td>BMCC</td>
<td>runs the BMC Software COPY PLUS product to produce image copies of DB2 for OS/390 physical objects. You must have installed COPY PLUS to use this function. See the COPY PLUS for DB2 Reference Manual.</td>
</tr>
<tr>
<td>BMCCOPYI</td>
<td>BMCC</td>
<td>runs the BMC Software COPY PLUS product to produce copies of image copies. You must have installed COPY PLUS to use this function. See the COPY PLUS for DB2 Reference Manual.</td>
</tr>
<tr>
<td>BMCCPRS</td>
<td>BMCU</td>
<td>copies statistics from the DB2 catalog into the DASD MANAGER PLUS database</td>
</tr>
<tr>
<td>BMCLOAD</td>
<td>BMCL</td>
<td>runs the BMC Software LOADPLUS utility. You must have installed LOADPLUS to use this function. See the LOADPLUS for DB2 Reference Manual.</td>
</tr>
<tr>
<td>BMCMOD</td>
<td>BMCC</td>
<td>runs the BMC Software C+/MODIFY product to perform maintenance on SYSCOPY. You must have installed C+/MODIFY to use this function. See the COPY PLUS for DB2 Reference Manual.</td>
</tr>
<tr>
<td>BMCREORG</td>
<td>BMCR</td>
<td>runs the BMC Software REORG PLUS product to reorganize DB2 physical objects. You must have installed REORG PLUS to use this function. See the REORG PLUS for DB2 Reference Manual.</td>
</tr>
<tr>
<td>BMCSTATS</td>
<td>BMCU</td>
<td>runs the DASD MANAGER PLUS statistics-gathering utility</td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>BMCU</td>
<td>runs the DASD MANAGER PLUS utility automation function</td>
</tr>
<tr>
<td>BMCUNLOD</td>
<td>BMCD</td>
<td>runs the BMC Software UNLOAD PLUS product. You must have installed UNLOAD PLUS to use this function. See the UNLOAD PLUS for DB2 Reference Manual.</td>
</tr>
<tr>
<td>BMCUPRS</td>
<td>BMCU</td>
<td>updates the DB2 catalog statistics from the statistics in the DASD MANAGER PLUS database by using the SQL UPDATE statement</td>
</tr>
<tr>
<td>CHEK DA</td>
<td>CHEK</td>
<td>runs the DB2 CHECK DATA utility</td>
</tr>
<tr>
<td>CHEK LOB</td>
<td>CHEK</td>
<td>runs the DB2 CHECK LOB utility</td>
</tr>
<tr>
<td>DSN1COPY</td>
<td>DSN1</td>
<td>runs the DB2 DSN1COPY utility</td>
</tr>
<tr>
<td>FULLCOPY</td>
<td>COPY</td>
<td>makes a full image copy by using the DB2 COPY utility</td>
</tr>
<tr>
<td>INCRCOPY</td>
<td>COPY</td>
<td>makes an incremental copy by using the DB2 COPY utility.</td>
</tr>
</tbody>
</table>
**User-defined programs (also called user-defined services)**

DASD MANAGER PLUS supports user-defined programs to accommodate including your own services (previously called job steps) as part of an action. When you specify the JCL elements for generating the service as shown in Table 30, DASD MANAGER PLUS substitutes symbolic variables before generating control cards for JCL Generation:

**Table 30  JCL elements for generating services**

<table>
<thead>
<tr>
<th>JCL element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC statement</td>
<td>■ program name</td>
</tr>
<tr>
<td>information</td>
<td>■ PARM list that includes symbolic variables</td>
</tr>
<tr>
<td></td>
<td>■ APF authorization indicator</td>
</tr>
<tr>
<td></td>
<td>■ attach key indicator</td>
</tr>
<tr>
<td>DD statement</td>
<td>■ ddnames including symbolic variables (such as sequence number)</td>
</tr>
<tr>
<td>information (including</td>
<td>■ DD specifications including symbolic variables</td>
</tr>
<tr>
<td>STEPLIBS)</td>
<td>■ DD type of INPUT, OUTPUT, PGMLIB, or REPORT</td>
</tr>
<tr>
<td>syntax information</td>
<td>template syntax including symbolic variables</td>
</tr>
</tbody>
</table>
Table 31 lists the symbolic variables that DASD MANAGER PLUS supports. Use && (two ampersands) to indicate that DASD MANAGER PLUS should not substitute the variable and that it should remove one ampersand and generate the rest in the syntax. If the product encounters an ampersand with a symbol that it does not recognize, it generates the symbolic variable in the text. See the DASD MANAGER PLUS for DB2 Reference Manual for description of JCLGEN symbolic variables.

### Table 31  Supported symbolic variables

<table>
<thead>
<tr>
<th>Symbolic variables</th>
<th>Symbolic variables</th>
<th>Symbolic variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;ALID (same as &amp;WKID)</td>
<td>&amp;JID (YDDD)</td>
<td>&amp;RUNTYPa</td>
</tr>
<tr>
<td>&amp;CR</td>
<td>&amp;JDATE (YYYYDDD)</td>
<td>&amp;SC (seconds)</td>
</tr>
<tr>
<td>&amp;CREATOR</td>
<td>&amp;JDAY (Julian day)</td>
<td>&amp;SPNAME</td>
</tr>
<tr>
<td>&amp;DA (same as DD)</td>
<td>&amp;JOBTYPb</td>
<td>&amp;SS (seconds)</td>
</tr>
<tr>
<td>&amp;DATE (YMMDD)</td>
<td>&amp;JQID (same as &amp;WKID)</td>
<td>&amp;SSID</td>
</tr>
<tr>
<td>&amp;DATEJ (YYYYDDD)</td>
<td>&amp;JUL4Y (YYYYDDD)</td>
<td>&amp;TIME (HHMMSS)</td>
</tr>
<tr>
<td>&amp;DB</td>
<td>&amp;JULIAN (YDDDD)</td>
<td>&amp;TS</td>
</tr>
<tr>
<td>&amp;DBNAME</td>
<td>&amp;MEMBERb</td>
<td>&amp;TSNAME</td>
</tr>
<tr>
<td>&amp;DDD (day of month)</td>
<td>&amp;MI (minutes)</td>
<td>&amp;UIIDb (same as &amp;USER)</td>
</tr>
<tr>
<td>&amp;DD (Julian day)</td>
<td>&amp;MM (month)</td>
<td>&amp;USERb</td>
</tr>
<tr>
<td>&amp;DDSEQ</td>
<td>&amp;MO (month)</td>
<td>&amp;USERIDb</td>
</tr>
<tr>
<td>&amp;DSQ</td>
<td>&amp;OBJKW</td>
<td>&amp;VCD</td>
</tr>
<tr>
<td>&amp;DOPTSa</td>
<td>&amp;OBJNAME</td>
<td>&amp;WCID</td>
</tr>
<tr>
<td>&amp;DSNUM (same as &amp;PART)</td>
<td>&amp;OBJT</td>
<td>&amp;WKID</td>
</tr>
<tr>
<td>&amp;DT (YMMDD)</td>
<td>&amp;OBJTYPE</td>
<td>&amp;WKID8 (first 8 characters of WORKID or action)</td>
</tr>
<tr>
<td>&amp;HH (hour)</td>
<td>&amp;OBNAM</td>
<td>&amp;WORKID (same as &amp;WKID)</td>
</tr>
<tr>
<td>&amp;HMS (same as &amp;TIME)</td>
<td>&amp;PART</td>
<td>&amp;YEAR (YYYY)</td>
</tr>
<tr>
<td>&amp;HO (hour)</td>
<td>&amp;PART2 (always 2 digit)</td>
<td>&amp;YY</td>
</tr>
<tr>
<td>&amp;IX</td>
<td>&amp;PART3 (always 3 digit)</td>
<td>&amp;YYY</td>
</tr>
<tr>
<td>&amp;IXCR (index creator)</td>
<td>&amp;PART4 (always 4 digit)</td>
<td>&amp;ZPREFIXb</td>
</tr>
<tr>
<td>&amp;IXNAME</td>
<td>&amp;PGMRb</td>
<td>&amp;ZUSERb</td>
</tr>
<tr>
<td>&amp;PREFIXb</td>
<td>&amp;UTILId</td>
<td>&amp;UTIDa</td>
</tr>
</tbody>
</table>

*a for full automation mode only

*b for online generation only

For an example of how to create a user-defined service, see “Example of creating a user-defined service” on page 288.
The Action Generation function helps you build routine utility jobs that you can run repeatedly. The production environment requires the execution of maintenance utility job streams. Figure 48 shows the relationship between an action and the other two major components, the worklist and worklist JCL.

**Figure 48  Action, worklist, and worklist JCL**

<table>
<thead>
<tr>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>An action specifies:</td>
</tr>
<tr>
<td>• action profile</td>
</tr>
<tr>
<td>• action services, such as</td>
</tr>
<tr>
<td>• COPY</td>
</tr>
<tr>
<td>• REORG</td>
</tr>
<tr>
<td>• COPY</td>
</tr>
<tr>
<td>• DB2 objects (usually)</td>
</tr>
<tr>
<td>• syntax</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WORKLIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>The worklist contains control statements that are input into JCL. See Figure 51.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WORKLIST JCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>The worklist JCL contains the following statements:</td>
</tr>
<tr>
<td>• DD statements</td>
</tr>
<tr>
<td>• EXEC statements</td>
</tr>
<tr>
<td>• control statements</td>
</tr>
<tr>
<td>See page 264.</td>
</tr>
</tbody>
</table>
Figure 49 illustrates the process of generating an action.

**Figure 49  Generating an action for the first time**

1. **START**
2. Does the action that you want to use already exist? **NO**
   - Is the action that you want to create similar to an existing action? **NO**
     - **Create the action (page 271).**
   - **YES**
     - **Select the action.**
3. **YES**
   - **Copy the action (page 273).**
4. **Edit the action services (page 279).**
5. **Generate the action. (page 299).**
6. Does the syntax exist? **NO**
   - **Create service syntax.**
   - **YES**
     - **Modify or accept the options.**
7. Was the generation successful? **NO**
   - **START**
   - **YES**
     - **TASK COMPLETE**
Figure 50 illustrates the process of regenerating an action.

![Figure 50 Regenerating an action](image)

Table 32 lists common tasks for setting up and generating actions.

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an action (job name) to identify a unit of work (job) or list actions if you have already created one.</td>
<td>“Creating an action” on page 271</td>
</tr>
<tr>
<td>Create and edit services for the action or copy existing services.</td>
<td>“Creating an action by copying an existing action” on page 273 or “Copying Services from one action into another” on page 285</td>
</tr>
<tr>
<td>Copy an existing action.</td>
<td>“Creating an action by copying an existing action” on page 273</td>
</tr>
<tr>
<td>Delete an action.</td>
<td>“Deleting actions” on page 278</td>
</tr>
</tbody>
</table>
Table 32  Tasks for generating actions (part 2 of 2)

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document the job’s purpose and scheduling information (job profile creation).</td>
<td>“Creating an action” on page 271</td>
</tr>
<tr>
<td>Open the Action Services panel.</td>
<td>“Creating an action by copying an existing action” on page 273.</td>
</tr>
<tr>
<td>Edit the service syntax.</td>
<td>“Editing program details” on page 292</td>
</tr>
<tr>
<td>Specify the sequence of services (functions and commands) to run against DB2 objects.</td>
<td>“Creating an action” on page 271</td>
</tr>
<tr>
<td>Run multiple services from the same action.</td>
<td>“Creating an action” on page 271</td>
</tr>
<tr>
<td>Generate a job for a new or existing action.</td>
<td>“Generating actions” on page 299</td>
</tr>
<tr>
<td>Specify the worklist processing options.</td>
<td></td>
</tr>
<tr>
<td>Specify the JCL processing options.</td>
<td></td>
</tr>
<tr>
<td>Identify the data set to contain diagnostic messages.</td>
<td></td>
</tr>
<tr>
<td>Specify whether to record event entries in the DASD MANAGER PLUS EVENTS table.</td>
<td>“Understanding return codes for batch programs” on page 306</td>
</tr>
<tr>
<td>Restart a utility job from the point at which the job failed.</td>
<td></td>
</tr>
<tr>
<td>Restart a utility job from the beginning.</td>
<td></td>
</tr>
<tr>
<td>Submit the job for execution.</td>
<td>“Generating actions” on page 299</td>
</tr>
</tbody>
</table>

**Utility worklist**

A utility worklist consists of control cards that an action generates. DASD MANAGER PLUS uses the information supplied in the action services and syntax to build the worklist. DASD MANAGER PLUS places the generated worklist in one data set and the generated JCL in another.

DASD MANAGER PLUS also lets you edit worklists and use the same worklist repeatedly. If you build a worklist with the **Edit Worklist** option enabled, you can review the worklist to ensure that it meets your requirements.
Figure 51 illustrates a simple generated worklist.

**Figure 51** Simple generated worklist

```
  * TESTCASE BSP00028 26961273
-SSID 000001 DEAE 63931300
-WKID 000002 BSP00028.UTILITY 06240917
-SYNC 000003 74205673
-BMCU 000004 ASUSMAIN 40588047
  BMCSTATS TABLESPACE QZUD4.%
  EVENTS N
  KEYCARD Y
  NUMCOLS 8
  COUNT 6
  INDEX N
```

**Table 33** Worklist commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–TIME</td>
<td>time of worklist creation</td>
</tr>
<tr>
<td>*</td>
<td>job frequency comment</td>
</tr>
<tr>
<td>–SSID</td>
<td>DB2 subsystem where the worklist runs</td>
</tr>
<tr>
<td>–WKID</td>
<td>action for generating the worklist</td>
</tr>
<tr>
<td>–SYNC</td>
<td>sync point command that creates a SYNC table entry and runs a DB2 COMMIT WORK command</td>
</tr>
</tbody>
</table>

**Note:** DASD MANAGER PLUS bypasses generating a SYNC command in the worklist if the previous command in the worklist was also a SYNC command.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–BMCU</td>
<td>command that runs BMCSTATS</td>
</tr>
</tbody>
</table>

For more information about worklist commands, see the *DASD MANAGER PLUS for DB2 Reference Manual*. 
Worklist commands

Worklist commands begin with a dash (–), such as –SYNC, –COPY, and –BMCR. The BMC Software Execution program uses the commands to set up and run the following items:

- DB2 SQL statements
- DB2 commands and utilities
- BMC Software programs
- user-defined programs

DASD MANAGER PLUS programs use some worklist commands. The BMC Software ALTER and CHANGE MANAGER products use others. The *DASD MANAGER PLUS for DB2 Reference Manual* lists all of the worklist commands that DASD MANAGER PLUS uses.

Wildcard expansion

In Figure 51, the TABLESPACE object name is a wildcard pattern (QZUD4%.%). DASD MANAGER PLUS utilities, such as BMCTRIG, expand wildcard patterns at execution, not when generating the worklist. Therefore, when you reuse worklists that contain only these utilities, you do not have to rebuild the worklists to include new database objects or to remove objects that have been deleted from the database.

Another worklist example

Figure 52 shows a generated worklist for the COPY utility. The product reserves columns 73 to 80 for hash verification numbers.

Figure 52  Generated worklist for COPY utility (part 1 of 2)

| -TIME 000000 '2004-05-31-12.40.57.203157' | 23616282 |
| * | 94972003 |
| -SSID 000001 DEAE | 64586920 |
| -WKID 000002 UBCAC005.UTILITY | 46486447 |
| -SYNC 000003 | 73550053 |
| -BMCC 000004 | 41243667 |
| COPY | | |
| TABLESPACE QZUD11.0ZUS0111 | | |
| GROUP NO | | |
| SHRLEVEL REFERENCE | | |
| FULL YES | | |
| INDEXES NO | | |
| COPYDDN (C0001) | 48170458 |
| -JCLP 000004 BMCC DDNAME C0001 DSNPREF | | |
| &PREFIX..&SSID..&DBNAME..&TSNAME | | |
Worklist file structure

Looking at the worklist file structure can clarify what happens during execution. The worklist consists of a sequence of commands that occupy single or multiple lines. The file structure is sequential and fixed length, with 80-character records. Table 34 describes the column format of the worklist.

**WARNING**

Modifying the hash number can terminate worklist execution if you enable hash checking (HASHFAIL=Y).

---

Table 34  Column format of worklists (part 1 of 2)

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | the line identifier  

A dash (–) indicates the beginning of a command. An asterisk (*) indicates a comment. If there is no indicator, the line is a continuation of the previous COMMAND line.

| 2–5    | the command identifier, a four-character code that identifies the command  
|
|        | The command identifier appears only on the first line of the command. |
| 6      | required blank |

| 7–12   | the command sequence number  

A six-digit number, right-justified and zero-filled. Command sequence numbers are in ascending order. Do not change the order.  

You can add and delete commands; however, you must keep the command sequence numbers in ascending order. In files that DASD MANAGER PLUS generates, command sequence numbers are in increments of 10 (in some instances 1) to allow you to insert commands.  

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     | required blank |
Components of a multiple-line command

In a generated worklist, a multiple-line command has the following elements:

- dash, 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

Worklist comment lines

Comment lines consist of an asterisk (*) for the line identifier and comment text for the remainder of the line. They do not use sequence numbers or hash verification numbers. You cannot embed comment lines within a command because a comment signals the end of a command. If the product encounters a blank line where it expected a command, the product treats the line as a comment.

Worklist JCL Generation

The worklist provides the BMC Software JCL Generation function (JCLGEN) with all of the necessary information to generate JCL for execution. The worklist and JCLGEN relieve you of all JCL generation tasks. For the standard, nonworklist JCL generation that BMCTRIG provides, see Chapter 8, “Analyzing objects by using BMCTRIG.”
Figure 53 provides a high-level overview of the JCL generation process.

For more information about JCL generation, see “Generating JCL” on page 137.

If you specify Edit JCL = Y when generating the action, the product displays the JCL (Figure 54).

Figure 54  Generated JCL for an action (part 1 of 3)

```bash
//RDAJXN2U JOB (5213), 'UTILITY-JENTEST',
//  CLASS=A, MSGCLASS=X
//*
/* JOBPARM SYSAFF=DB2A
//*

############################################################################
* CREATED BY :  RDAJXN2
* TIMESTAMP  :  01/30/2006.15.24.31
* ENVIRONMENT:  ISPF 5.2MVS     TSO
* RELEASE    :  V08.02.00 01/30/2006
* DB2 VERSION: 815
############################################################################

*-----------------------------
* DASD MANAGER WORKLIST EXECUTION
*-----------------------------
//STEP1 EXEC PGM=AEXEMAIN,REGION=0M,
// PARM='DS815ECA'
//STEPLIB DD DSN=TIS.IVPDEC.DECA2.RNTM.BMCLINK,
// DISP=SHR
// DD DSN='SYS3.DECA.DSNEXIT',
// DISP=SHR
// DD DSN='CSGI.DB2V81M.DSNLOAD',
// DISP=SHR
```
Figure 54  Generated JCL for an action  (part 2 of 3)

```/* * *-----------------------------------------------------------------* */
/* * ABNLIGNR DD DUMMY */
/* * DSSPRINT DD SYSOUT=* */
/* * SYSDUMP DD SYSOUT=* */
/* * SYSTEM DD SYSOUT=* */
/* * UTPRINT DD SYSOUT=* */
/* * SYSOUT DD SYSOUT=* */
/* * AEXIN DD * */
ASU
SSID DEAE
WORKID JENTEST.UTILITY
EVENTS LINES 54 STATS
VCAT DEAECAT
UTILITYID JENTEST.ASUUTIL
COPYOPT  ACP$OPTS
UNLOADOPT  ADU$OPTS
LOADOPT  AMU$OPTS
REORGPT ARU$OPTS
CHECKOPT ACK$OPTS
RECOVEROPT AFR$OPTS
/* * //SYSIN DD SPACE=(CYL,(15,15)),UNIT=SYSDA,DISP=(NEW,PASS), */
/* * DSORG=PS,LRECL=80,BLKSIZE=3200,RECFM=FB */
/* * //SYSPRINT DD SPACE=(CYL,(15,15)),UNIT=SYSDA,DISP=(NEW,PASS) */
/* * //SYSTSIN DD SPACE=(TRK,(1,1)),UNIT=SYSDA,DISP=(NEW,PASS), */
/* * DSORG=PS,LRECL=80,BLKSIZE=3200,RECFM=FB */
/* * //SYSTSPRT DD DISP=(NEW,PASS), */
/* * DSN=&TSPRT1, */
/* * // SPACE=(CYL,(5,5)),UNIT=SYSDA, */
/* * DSORG=PS,LRECL=137,BLKSIZE=3155,RECFM=VBA */
/* * //AEXPRNT DD SYSOUT=* */
/* * //ACPDDGLP DD * */
/* * DEFINE GDG (NAME(&BASE) LIMIT(010) SCR) */
/* * //ACPDDGLB DD * */
/* * DEFINE GDG (NAME(&BASE) LIMIT(010) SCR) */
/* * //ACPDDGRP DD * */
/* * DEFINE GDG (NAME(&BASE) LIMIT(001) SCR) */
/* * //ACPDDGRB DD * */
/* * DEFINE GDG (NAME(&BASE) LIMIT(001) SCR) */
/* * //WORKLIST DD DISP=SHR, */
/* * DSN=RDAJXN3.TEST9.JCL(TESTW) */
/* * *----------------------------------------------------------------*/
/* * * UTILITY COPY DD STATEMENTS */
/* * *----------------------------------------------------------------*/
/* * * DATA SET SIZE REQUIRED FOR DD C0001 IS AN ESTIMATE */
/* * C0001 DD DSN=RDAJXN.CPY.DEAE.JEND30.C791278.C0001, */
/* * DISP=(NEW,CATLG,CATLG),*/
/* * SPACE=(CYL,(1,1),RLSE), ESTIMATE-C/H */
/* * UNIT=SYSDA */```
Working with actions

This section describes common tasks that you perform with actions.

Accessing the action maintenance and generation function

Use this procedure to access the DASD MANAGER PLUS action maintenance and generation function.

To access the Actions menu

1. On the DASD MANAGER PLUS main menu, select Utility Actions (WORKIDs) and press Enter.

2. On the main menu, change the SQLID if you are using it to control access to actions, services, and syntax, to the owner ID of the object that you want to access.

   For more information, see “Controlling access to actions, services, and service syntax” on page 307.

3. Make a selection on the Actions menu.
Creating an action

Enter an action name or pattern and an owner name, as follows:

<table>
<thead>
<tr>
<th>If you are</th>
<th>Specify this action or owner name</th>
</tr>
</thead>
<tbody>
<tr>
<td>creating an action</td>
<td>new, unique action name</td>
</tr>
<tr>
<td>displaying an existing action</td>
<td>name of the action that you want to see</td>
</tr>
<tr>
<td>displaying a group of related actions</td>
<td>wildcard pattern</td>
</tr>
<tr>
<td>displaying a selection list of all actions</td>
<td>% or *</td>
</tr>
<tr>
<td>copying an action</td>
<td>name of action (no wildcard)</td>
</tr>
<tr>
<td>renaming an action</td>
<td>name of action (no wildcard)</td>
</tr>
<tr>
<td>deleting an action</td>
<td>name of action (no wildcard)</td>
</tr>
</tbody>
</table>

Creating an action

Use this procedure to create an action and its associated profile. An action profile documents the action by allowing you to add a title, a comment, and frequency information. The profile also contains worklist, JCL, and POF data set names. After you generate the action, DASD MANAGER PLUS updates the profile information fields with status information about the generation.

NOTE

The action profile is informational only and is not part of an automated scheduling system. For more information about automating maintenance, see the Automation for DB2 User Guide.

To create an action

1 On the DASD MANAGER PLUS main menu, select Utility Actions (WORKIDs) and press Enter.

2 On the Actions panel, select Create Action, type the following information, and press Enter:

- In the Action Name field, enter a unique action name. The name cannot contain percent (%), asterisk (*), or embedded blank characters.
Creating an action

**NOTE**
The Action Name that you enter is used to generate a UTILITYID as follows:

- For online generation of the action, JCLGEN creates a UTILITYID using the first eight characters of the action name followed by a period and ASUUTIL (for example, UTILITYID JENTEST.ASUUTIL).
- For BMCTRIG generated worklist JCL format, the UTILITYID is a “T” followed by the first eight characters of the workid parameter passed to ASUSTRIG, followed by a period, an “S”, and a three-digit sequence number which is generated from the UTILSEQ syntax option to BMCTRIG (for example, UTILITYID TJENTEST.S001).
- For BMCTRIG generated standard JCL, the same information is passed to JCLGEN, but JCLGEN creates the UTILITYID for utilities as the job name followed by a period, followed by BMCTR (for example, for job name TBMC001, the UTILITYID would be shown as TBMC001.BMCTR).

- In the **Owner Name** field, enter an owner name or use the default.

**NOTE**
To create an action that is similar to an existing action, use the Copy command on the Action List panel. See “Copying Services from one action into another” on page 285.

The Create Action panel appears.

3 (optional) Enter the following information, and press Enter.

- In the **Action Name** field, modify the action name.
- In the **Owner Name** field, edit the owner name.
- In the **Action Description** field, enter a description to help identify the action.

The Edit Action Services panel appears.

4 Proceed to “To edit action services” on page 279.

For information about action services, see “Working with services” on page 279.
Creating an action by copying an existing action

Use this procedure to create a new utility action that is similar to an existing action.

To create an action by copying another action

1. On the DASD MANAGER PLUS main menu, select Utility Actions (WORKIDs) and press Enter.

2. On the Actions panel, select Copy Action, type the name of the action (without wildcards) that you want to copy, and press Enter.

NOTE

Alternately, you can open the Actions panel by selecting List Actions. You can specify a wildcard for the action name. Then type C next to the action that you want to copy, and press Enter.

The Copy Action panel appears (Figure 55).

Figure 55  Copy Action panel

3. Enter a unique action name in the Action Name field.

The action name cannot contain a percent sign (%), asterisk (*), or embedded blank characters. If you plan to use &WKID or &WKID8 during JCL generation, the action must be a valid PDS member name or data set node. Optionally, you can set the owner and enter a description. (For more information, see “Controlling access to actions, services, and service syntax” on page 307.)

4. Press Enter to copy the action.

The Action List panel shows the action that you copied and the new action that you created.
**Editing actions**

Use this procedure to modify an existing action.

**To edit an action**

1. Access the Action List panel as follows:
   
   **A** On the DASD MANAGER PLUS main menu, select Utility Actions (WORKIDs) and press Enter.
   
   **B** On the Actions panel, select List Actions, type the following information, and press Enter:
   
   - In the Action Name field, enter an action name or wildcard.
   - In the Owner Name field, enter an owner name or wildcard.

2. Review the fields on the Action List panel (Figure 56).

**Figure 56  Action List panel**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Act</td>
<td>the action that you want to perform on a line item</td>
</tr>
<tr>
<td>Action</td>
<td>action name</td>
</tr>
<tr>
<td>Owner</td>
<td>authorization ID of the person who owns the action</td>
</tr>
</tbody>
</table>
3 (optional) Edit the action profile, as follows:

**NOTE**
You do not have to enter information for the action profile. Action profile data does not affect how DASD MANAGER PLUS uses the action. The profile contains the JCL Gen POF, which provides default values for the JCL options. If you do not specify a data set for the action, the product uses the data set that the installation options module specifies in the POFDS option.

On the Action List panel, type P in the Act field next to the new action and press Enter.

The Edit Action panel appears (Figure 57). The **Action name** field shows the action that you selected. When generating the worklist, the product automatically updates the **Worklist generated by** and revision information.
Figure 57  Edit Action panel

B (optional) Enter the following information on the Edit Action panel:

- In the Description field, enter a description of up to 100 characters.
- In the JCL Options field, enter a data set name (POF).
- In the Title field, enter a title of up to 20 characters.

C Add scheduling information, as follows:

- In the Frequency field, specify how often to run the utility job by typing one of the following abbreviations:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>daily</td>
</tr>
<tr>
<td>W</td>
<td>weekly</td>
</tr>
<tr>
<td>B</td>
<td>biweekly</td>
</tr>
<tr>
<td>M</td>
<td>monthly</td>
</tr>
<tr>
<td>U</td>
<td>user-defined</td>
</tr>
</tbody>
</table>
In the **Day** field, specify the day of the month to run the utility job by using one of the following values:

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Day values</th>
</tr>
</thead>
<tbody>
<tr>
<td>weekly</td>
<td>1 through 7</td>
</tr>
<tr>
<td>biweekly</td>
<td>1 through 31</td>
</tr>
<tr>
<td>monthly</td>
<td>1 through 31</td>
</tr>
</tbody>
</table>

In the **Time of Day** field, specify the time of day to run the utility job.

Choose a value from 0000 through 2359, which indicates a 24-hour clock format.

**NOTE**
The Action Profile is informational only and is not part of an automated scheduling system. For more information about automating maintenance, see the *Automation for DB2 User Guide*.

The product sets the remaining fields only after you have generated the worklist:

- **Worklist generated by** shows your AUTHID after you have generated the worklist.

- **Worklist generated on** is the date when you last generated the worklist.

- **with RC** indicates the return code that resulted when you last generated the worklist, as follows:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The product generated the worklist successfully.</td>
</tr>
<tr>
<td>4</td>
<td>The product issued a warning message.</td>
</tr>
<tr>
<td>8</td>
<td>A generation error occurred.</td>
</tr>
</tbody>
</table>

- **Data Sets** is the data sets into which the product placed the generated worklist and JCL. You specify the data sets on the Action Job Generation panel. For more information, see “Creating an action” on page 271.
Deleting actions

Current Job Status is the current status of the action. This field displays one of the following statuses from the DO_WORKIDS table. The Execution Monitor (AEXEMAIN) modifies the status.

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>blank</td>
<td>The product did not generate the utility job.</td>
</tr>
<tr>
<td>N</td>
<td>The product generated the utility job but did not run it.</td>
</tr>
<tr>
<td>R</td>
<td>The utility job completed successfully.</td>
</tr>
<tr>
<td>S</td>
<td>The utility job started but had an error.</td>
</tr>
</tbody>
</table>

Press END to save the action profile.

4 Edit the services for the action as described in “To edit action services” on page 279.

5 Press END to save your changes.

Deleting actions

Use this procedure to delete an action when you no longer need it.

To delete an action

1 On the DASD MANAGER PLUS main menu, select Utility Actions (WORKIDs) and press Enter.

2 On the Actions panel, select Delete Action, specify the name of the action that you want to delete, and press Enter.

You cannot use wildcards.

NOTE

Alternately, you can open the Actions panel by selecting List Actions. You can specify a wildcard for the action name. Then, type D next to the action that you want to delete and press Enter.

The Delete Action panel appears (Figure 58).
Working with services

This section describes common tasks that you can perform with services (also called action services).

Editing action services

Use this procedure to set up services and syntax that DASD MANAGER PLUS can use to build the worklist and generate JCL for a new or existing action.

Before you begin

Before you edit the action services, you must already have created an action as instructed in “To create an action” on page 271.

To edit action services

1. Access the Edit Action Services panel as follows:

   A. On the DASD MANAGER PLUS main menu, select Utility Actions (WORKIDs) and press Enter.

   B. On the Actions panel, select List Actions, type the following information, and press Enter:
      - In the Action Name field, enter an action name or a wildcard.
      - In the Owner Name field, enter an owner name or a wildcard.

   The Action List panel appears.

3. Type Y to confirm, and press Enter to delete the action.

Figure 58  Delete Action panel

<table>
<thead>
<tr>
<th>Figure 58 Delete Action panel</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEAE ----------------------- Delete Action ----------------------</td>
</tr>
<tr>
<td>COMMAND ====&gt;</td>
</tr>
<tr>
<td>Are you sure you want to delete this action? N (Y = OK to delete)</td>
</tr>
<tr>
<td>Action Name . . . NEW</td>
</tr>
<tr>
<td>Action Owner . . . RDAEXX</td>
</tr>
<tr>
<td>Action Description:</td>
</tr>
<tr>
<td>&gt; NEW ACTION</td>
</tr>
</tbody>
</table>

3. Type Y to confirm, and press Enter to delete the action.
On the Action List panel, select the action by typing **E** (or **S**) in the **Act** field next to the action and pressing **Enter**.

The Edit Action Services panel shows any existing services that you have specified for the action. A newly created action contains a blank line with no services (Figure 59).

**Figure 59  Edit Action Services panel**

On the Edit Action Services panel, you can perform the following actions:

- access a service selection list
- create, edit, duplicate, or delete a service in the action
- access the service syntax list to edit or create service syntax
- copy services from an action
- manage object sets
- update the service profile for use within the action
- preview the generated syntax
- zoom into the contents of a long field

2 *(optional)* In the **JCL Options DSN** field, type a data set name in the field to associate a different POF with this action.

If this field is blank, the product uses the data set name that is specified in the installation options module in the POFDS option. For information about creating an action POF, see “Creating a user POF” on page 125.
3 Create or edit the action services, as follows:

A In the **Act** field, enter **?** to view a list of services.

The product displays a selection list of services and their descriptions. **Figure 60** shows the complete selection list. You can select an existing service. For more information about creating user-defined services, see “Example of creating a user-defined service” on page 288.

**Figure 60  Service List panel**

<table>
<thead>
<tr>
<th>Service</th>
<th>Vendor</th>
<th>BMC-Req</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTERSEC</td>
<td>BMC</td>
<td>Y</td>
<td>BMC DASD MGR ALTER SECONDARY ALLOCATION</td>
</tr>
<tr>
<td>BMCCOPY</td>
<td>BMC</td>
<td>Y</td>
<td>BMC COPY PLUS</td>
</tr>
<tr>
<td>BMCCOPYI</td>
<td>BMC</td>
<td>Y</td>
<td>BMC COPY PLUS COPY IMMAGECOPY</td>
</tr>
<tr>
<td>BMCCPRS</td>
<td>BMC</td>
<td>Y</td>
<td>BMC DASD MGR COPY CATALOG STATS</td>
</tr>
<tr>
<td>BМСLOAD</td>
<td>BMC</td>
<td>Y</td>
<td>BMC LOAD PLUS</td>
</tr>
<tr>
<td>BМСMOD</td>
<td>BMC</td>
<td>Y</td>
<td>BMC COPY PLUS MODIFY</td>
</tr>
<tr>
<td>BМСREG</td>
<td>BMC</td>
<td>Y</td>
<td>BMC REORG PLUS</td>
</tr>
<tr>
<td>BМССТАТ</td>
<td>BMC</td>
<td>Y</td>
<td>BMC DASD MGR BМССТАТ</td>
</tr>
<tr>
<td>BМСТРИГ</td>
<td>BMC</td>
<td>Y</td>
<td>DASD MGR DETECT EXCEPTIONS AND TRIGGER UTILITIES</td>
</tr>
<tr>
<td>BМСУНЛОД</td>
<td>BMC</td>
<td>Y</td>
<td>BMC UNLOAD PLUS</td>
</tr>
<tr>
<td>BМСУПР</td>
<td>BMC</td>
<td>Y</td>
<td>BMC DASD MGR UPDATE CATALOG STATS</td>
</tr>
<tr>
<td>CHEK DA</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 CHECK DATA</td>
</tr>
<tr>
<td>CHEK LOB</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 CHECK LOB</td>
</tr>
<tr>
<td>DSNICOPY</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 DSNICOPY</td>
</tr>
<tr>
<td>FULLCOPY</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 FULL IMAGE COPY</td>
</tr>
<tr>
<td>INCRCOPY</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 INCREMENTAL IMAGE COPY</td>
</tr>
<tr>
<td>LOAD</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 LOAD</td>
</tr>
<tr>
<td>MERСCOPY</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 MERGE COPY</td>
</tr>
<tr>
<td>MODICOPY</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 MODIFY RECOVERY</td>
</tr>
<tr>
<td>QUIESCE</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 QUIESCE</td>
</tr>
<tr>
<td>RECOVER</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 RECOVER TABLESPACE AND REBUILD INDEX</td>
</tr>
<tr>
<td>REORG</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 REORG</td>
</tr>
<tr>
<td>REP REC</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 REPORT RECOVERY</td>
</tr>
<tr>
<td>REP SET</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 REPORT TABLESPACESET</td>
</tr>
<tr>
<td>REPAIR</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 REPAIR OFF PENDING FLAGS</td>
</tr>
<tr>
<td>RESIZE</td>
<td>IBM</td>
<td>Y</td>
<td>BMC DASD MGR RESIZE STOGROUP OBJECTS (TRIG OR AUTO</td>
</tr>
<tr>
<td>RUNSTATS</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 RUNSTATS</td>
</tr>
<tr>
<td>START DB</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 START DATABASE COMMAND</td>
</tr>
<tr>
<td>STOP</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 STOP DATABASE COMMAND</td>
</tr>
<tr>
<td>STОPSPACE</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 STОPSPACE</td>
</tr>
<tr>
<td>SYNC</td>
<td>IBM</td>
<td>Y</td>
<td>BMC WORKLIST SYNC</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>IBM</td>
<td>Y</td>
<td>DB2 UNLOAD</td>
</tr>
</tbody>
</table>

B Type **S** in the **Act** field beside a service and press **Enter**.

The product inserts the services into the action in the order that you select them. You can select services more than once.
C Press END to save your changes and to return to the Edit Action Services panel.

**NOTE**
Steps D and E apply only if you are using online generation and only in some situations. Online generation means that you generate services independent of BMCTRIG. For more information about BMCTRIG, see Chapter 8, “Analyzing objects by using BMCTRIG.”

D In the **Object Name** field, type the object name.

You can use wildcards. For example, type QZUD40.% to specify all table spaces in the QZUD40 database. You cannot use delimited names such as TZU.“abc.”

**NOTE**
If you specify an object name or pattern when editing utility options, you can use that name or pattern to select processing options by table name, index name, or column name in some utilities. For example, you can select specific columns for BMCSTATS collection, or you can specify discard criteria for REORG. Object selections are stored with the service syntax. You can change the object pattern to provide different selection lists based on the object name or pattern expansion.

E In the **Type** field, type the two-letter abbreviation for the object type and press Enter.

If you specify an object set, you can leave the type field blank. DASD MANAGER PLUS retrieves all table space and index types that qualify. Use the following abbreviations for object types:

- TS (table space)
- IS (index space)
- IX (index)
- TT (table space set)
- SG (storage group)
- VL (volume)

**NOTE**
Not all object types are valid for all services.

If you specify IS for index space, the generated worklist includes the index names that match the IS name, pattern, or object set expansion. An exception is if the pattern is not expanded but is generated into the utility syntax (BMCSTATS, BMCTRIG, BMCUPRS, BMCCPRS). Type IS is allowed for services that allow IX.

F *(optional)* Enter a partition number.

Note that not all services support partition numbers.
G (optional) Enter an object set owner and name.

See “Creating an object set for an action service” on page 336 for information about using the M command to create, edit, and manage object sets.

4 (optional) On the Edit Action Services panel, create, select, or edit the service syntax by typing E in the Act field.

**NOTE**

If you do not select syntax, the action receives the service syntax at generation time from the repository that is marked as the default syntax for the service. For more information, see “Editing program details” on page 292.

The Service Syntax List panel appears showing the current syntax (Figure 61).

**Figure 61  Service Syntax List panel**

If you enter I, the Edit Service Syntax panel appears (Figure 62).
On this panel, you can edit syntax options. Continue to press Enter for more options.

If you make any changes, press END to save them before returning to the Edit Action Services panel (Figure 59).

5 (optional) On the Edit Action Services panel, type U in the Act field to update or view service information within an action.

The Edit Action Service panel appears (Figure 63).

You can indicate whether to group objects together for the service. Grouped services allow you to process multiple table spaces or indexes or partitions (only for certain utilities) at one time when you run a utility. For more information about grouped services, see “Grouped services” on page 466.
Copying Services from one action into another

Use this procedure to copy services from an existing action into the current action. This procedure is useful when you want to combine functions from two actions.

To copy services from an action

1 Create an action selection list containing the action into which to copy, as follows:

A On the DASD MANAGER PLUS main menu, select Utility Actions (WORKIDs) and press Enter.

B Select List Actions and press Enter.
C Specify the Action Name and press Enter.

You can use wildcard characters in the name.

2 On the Action List panel, type S in the Act field beside the action to copy and press Enter.

The Edit Action Services panel appears (Figure 59).

3 Determine the location of the copied services in relation to the current services.

4 Type A to place the copied services after the current service or B to place the copied services before the current service, and then press Enter to return to the Action List panel.

5 On the Action List panel, type S in the Act field beside the action name that you want to copy and press Enter.

NOTE
Alternately, you can type COPY followed by an action name at the COMMAND line to specify the action explicitly. You can also copy action services into the current action as follows:

1. On the Edit Action Services panel, type A to place the copied action after the current service or B to place the copied action before the current service.

2. Type COPY actionName on the COMMAND line, where actionName is a valid action and then press Enter. For example, type COPY action, where action is a valid action.

Creating and editing services

Use the following steps to create and edit a service.

To create and edit services

1 On the DASD MANAGER PLUS main menu, select Services and press Enter.

The Service List panel appears (Figure 64). This panel contains an entry for every BMC Software required services and user-defined services that DASD MANAGER PLUS supports.
Figure 64  Service List panel

On this panel, you can perform the following tasks:

- Enter C to create a user-defined service.
- Enter E to edit a service.
- Enter D to delete a service.
- Enter A next to a service to see the list of actions that use a service.
- Enter L next to a service to view a list of the service syntax entries for it.

To edit a service, enter E.

The Edit Service panel appears (Figure 65).

Figure 65  Edit Service panel

On this panel, you can perform the following tasks:

- For user-defined services, you can change all fields except Vendor.
Example of creating a user-defined service

For BMC Software required services, this panel is informational only. You cannot edit it.

For more information about the owner, see “Controlling access to actions, services, and service syntax” on page 307.

You can use Service Type to group services into categories such as COPY, REORG, STATS, or REPORT. Enter a type that is useful for your environment.

Selecting an option from Edit Program Details allows you to edit the service syntax as described in “Editing program details” on page 292.

Example of creating a user-defined service

Following is an example of creating a user-defined service that will be used with a worklist format. In this example, you will create a service named IKJEFT01. This service executes the DISPLAY DATABASE command, which shows the status of your database and is called from an action named DISPLYDB.

To create program details

1 On the DASD MANAGER PLUS main menu, select Services and press Enter.

The Service List panel appears (Figure 66). This panel contains an entry for every BMC required service and user-defined service that DASD MANAGER PLUS supports.

![Figure 66 Service List panel]

DECA --------------------- Service List -------------------- Row 1 to 31 of 46
COMMAND ===> SCROLL ===> PAGE
Please select an option and press Enter.
C = Create a new service E = Edit D = Delete
A = Actions using L = List syntaxes
Act Service Vendor BMC-Req Description
ALTERSEC BMC Y BMC DASD MGR ALTER SECONDARY ALLOCATION
BMCCOPY BMC Y BMC COPY PLUS
BMCCOPYI BMC Y BMC COPY PLUS COPY IMAGECOPY
To create a service, enter C.

The Create New Service panel appears (Figure 67).

**Figure 67  Create New Service panel**

| DECA ------------------------ Create New Service ------------------------------ |
| COMMAND ====> |
| Service name: Updated by: |
| Owner: RDAJXS Last Updated: |
| Vendor: USR |
| Service Type: |
| Description > |
| Edit Program Details: |
| 1. EXEC statement |
| 2. DD statements |
| 3. Model syntax |

In the **Service name** field, enter the Service name IKJEFT01.

In the **Edit Program Details** field, type 1 and press Enter to provide information to generate the EXEC statement for the service.

The Edit Program Details pop-up panel appears.

A Enter the following information:

- program name as it appears in the JCL in the PGM= keyword (IKJEFT01 in this example)

- APF Y for APF authorization (in this example, IKJEFT01 requires authorization)

The panel should then appear as shown in Figure 68:

**Figure 68  Edit Program Details panel**

| DECA -------------- Edit Program Details ---------------------- |
| COMMAND ====> |
| Program: IKJEFT01 (Program for EXEC PGM=) |
| PARM: |
| APF: Y (Y/N APF authorized) |
| Attach key: 8 (1-8 Attach key) |

B Press END to save your changes and return to the Edit Service panel.
Example of creating a user-defined service

5 In the Edit Program Details field, select 2 and press Enter.

The DD Statement List panel opens.

6 Enter the required DD statements:

A In the DD Statement List panel, enter I to insert each statement.

The DD Statement panel opens (Figure 69).

**Figure 69 DD Statement panel**

<table>
<thead>
<tr>
<th>DD name:</th>
<th>Sequence:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: (INFILE, OUTFILE, REPORT, PGMLIB)</td>
<td></td>
</tr>
<tr>
<td>DSN:</td>
<td></td>
</tr>
<tr>
<td>Description &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>DD Statement text (free-form)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Restart DD Statement text (free-form)</td>
<td></td>
</tr>
</tbody>
</table>

B In the DD name field, enter SYSTSPRT.

C In the Type field, enter REPORT.

This information tells the JCL Generation function where to send the output.

D Press END to save your changes and return to the DD Statement list panel, which should now look like Figure 70.
Example of creating a user-defined service

Chapter 4 Maintaining and generating actions

Figure 70    DD Statement List panel

For user-defined services, enter all DDs needed. For BMC and DB2 utilities, the standard DDs are defined in the JCL Options file and in the Service Syntax and this list only contains special case DD additions.

Service name: IKJEFT01

E = Edit, D = Delete, I = Insert, C = Copy

<table>
<thead>
<tr>
<th>Act</th>
<th>DDNAME</th>
<th>Type</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SYSTSPRT</td>
<td>REPORT</td>
<td></td>
</tr>
</tbody>
</table>

****************** Bottom of data ******************

If you needed to add more statements, you would repeat steps 6A through 6D. For this example, you have no additional statements to add.

**NOTE**

For BMC Software required services, this list contains only additional DD statements that are not in the JCL options file or syntax options. Do not add DD statements for BMC required services.

Do not add a DD statement for the input (such as SYSIN). The syntax DD is named as part of the service syntax definition. The syntax DD will be allocated to a job entry subsystem (JES) input stream for standard JCL and the automation component. The syntax DD will be allocated to a temporary data set if you run it in worklist format under AEXEMAIN.

E Press END to save your changes and return to the Edit Service panel.

7 To tell the JCL Generation function where the input is coming from, create Model Syntax.

A To create Model service syntax, select 3 in the Edit Program Details field and press Enter.

B In the Edit Service Syntax panel, enter the following information:

- Type DISPLAYDATABASE in the Syntax Name field.
- Type SYSTSIN in the DD Name field.
Editing program details

The panel should appear as shown in Figure 71:

**Figure 71  Edit Program Details panel**

C Press **Enter** to continue updating options and to include the syntax.

D In the Editing panel that appears, *starting in column 2*, enter the following syntax:

```
DSN SYSTEM(&SSID)
-DISPLAY DATABASE(&DBNAME) SPACE(&SPNAME) LIMIT(*)
```

**NOTE**

JCL Execution does not execute SQL commands in worklist format. SQL commands must be generated by BMCTRIG in standard JCL format.

E Press **END** to save your changes.

8 Associate this service with the action named DISPLYDB. For more information, see “Editing action services” on page 279.

Editing program details

Use this procedure to edit the program details, including editing EXEC, DD, and model syntax program details.

**To edit program details**

1 Access the Edit Service panel as follows:

A On the DASD MANAGER PLUS main menu, select **Services**, and press **Enter**.

B On the Service List panel, type **E** to edit a service, and press **Enter**.

The Edit Service panel appears.
2 Edit user-defined services as follows:

A In the **Edit Program Details** field, select **EXEC statement** and press **Enter** to provide information to generate the EXEC statement for the service.

The Edit Program Details pop-up panel appears **(Figure 72)**. (For BMC Software required services, you cannot edit this panel.)

**Figure 72 Edit Program Details panel**

```
DEAE -------------- Edit Program Details ----------------------
COMMAND ===> 
Program: (Program for EXEC PGM=) 
PARM: 'MYPARMS.&SSID'
APF: N (Y/N APF authorized)
Attach key: 8 (1-8 Attach key)
```

B Enter the following information:

- program name as it appears in the JCL in the PGM= keyword
- parameter information as it appears in the JCL in the PARM= keyword

The parameter list can contain symbolic variables. See **Table 31**.

C Press **END** to save your changes and return to the Edit Service panel.

3 Edit DD statements as follows:

A In the **Edit Program Details** field, select **DD statements** and press **Enter**.

The DD Statement List panel opens **(Figure 73)**.
Figure 73  DD Statement List panel

For user-defined services, enter all DDs needed. For BMC and DB2 utilities, the standard DDs are defined in the JCL Options file and in the Service Syntax, and this list only contains special case DD additions.

Service name: CCBPROG1

E = Edit, D = Delete, I = Insert, C = Copy

<table>
<thead>
<tr>
<th>Act DDNAME</th>
<th>Type</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCBTEST</td>
<td>INFILE</td>
<td>RDACZB.TEST.INFILE</td>
</tr>
</tbody>
</table>

B  Enter all DD statements that you require.

NOTE

For BMC Software required services, this list contains only additional DD statements that are not in the JCL options file or syntax options. Do not add DD statements for BMC Software required services.

Do not add a DD statement for the input (such as SYSIN). The syntax DD is named as part of the service syntax definition. The syntax DD will be allocated to a job entry subsystem (JES) input stream for standard JCL and the automation component. The syntax DD will be allocated to a temporary data set if you run it in worklist format under AEXEMAIN.

C  To edit the DD statements, enter E.

The DD Statement panel opens (Figure 74).
Figure 74 DD Statement panel

DEAE ------------------------ DD Statement ------------------------------------------
COMMAND =>

DD name: CCBTEST
Sequence: 1

Type: INFILE (INFILE, OUTFILE, REPORT, PGMLIB)
DSN: RDACZB.TEST.INFILE
Member:
Description >
>

DD Statement text (free-form)
> // DISP=SHR
>
>
>
>

Restart DD Statement text (free-form)
> // DISP=OLD

D Edit the information, as follows:

<table>
<thead>
<tr>
<th>At this field</th>
<th>Enter the following information</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD name</td>
<td>ddname, which can contain symbolic variables (Table 31)</td>
</tr>
<tr>
<td></td>
<td>Note: For DDNAMES defined for the USERDEF service, you must edit the JCL to add the procstep to the DDNAME as follows: //procstep.ddname DD.</td>
</tr>
<tr>
<td>Type</td>
<td>Use INFILE for input.</td>
</tr>
<tr>
<td></td>
<td>Use OUTFILE for output file name</td>
</tr>
<tr>
<td></td>
<td>Use REPORT for message output. For the automation component, the type is always allocated to SYSOUT=* . For AEXEMAIN, the type must be a data set. AEXEMAIN copies it to AEXPRINT. If the ddname is SYSPRINT, the standard temporary SYSPRINT is used under AEXEMAIN and this SYSPRINT is ignored.</td>
</tr>
<tr>
<td></td>
<td>Use PGMLIB for program library name. For the automation component or AEXEMAIN, it is TASKLIB. For Standard JCL, it is STEPLIB.</td>
</tr>
<tr>
<td>DSN</td>
<td>data set name, which can contain symbolic variables (see Table 31)</td>
</tr>
<tr>
<td>Member</td>
<td>optional PDS member name and can contain symbolic variables</td>
</tr>
</tbody>
</table>
4 Press END to save your changes and return to the Edit Service panel.

5 To edit service syntax, select Model syntax in the Edit Program Details field and press Enter.

The Service Syntax List panel opens (Figure 75). You can also open this panel by editing an entry on the Action Services List panel (Figure 59).

**TIP**

To limit your search criteria, you can specify a Name and an Owner in the Search Criteria field instead of % (which specifies all).

### Figure 75  Service Syntax List panel

<table>
<thead>
<tr>
<th>ASUUASSL</th>
<th>Service Syntax List</th>
<th>Row 1 to 1 of 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td></td>
<td>SCROLL ==&gt; CSR</td>
</tr>
</tbody>
</table>

Service: CCBPROG1
Current syntax: Clear? N
Default syntax: Clear? N

Please select an option or change Search Criteria and press Enter.

- I = Insert a new syntax
- E = Edit service syntax
- C = Copy
- S = Select
- A = Actions using the syntax
- F = Set as default
- B = Browse service syntax
- V = Preview
- D = Delete
- P = Profile

--> %                                      %      <-- Search Criteria
Act Name ----------------------------------- Owner --- Description (partial) ---

6 Create or edit the service syntax, entering syntax in column 2, and press END to save your changes.
The syntax for user-defined services can contain symbolic variables. For more information, see Table 31.

7 (optional) Select a syntax as the system default by typing F in the Act field and pressing Enter.

The product selects the default syntax when it uses a service in an action but the service does not have any syntax specified for it. Also, you can clear the default syntax selection by typing Y at Clear? and pressing Enter.

## Previewing Generated Services

Use the following steps to preview the command syntax that DASD MANAGER PLUS generates for non-user-defined generated services.

### To preview syntax from the Edit Action Services panel

1. On the DASD MANAGER PLUS main menu, select an Action, and press Enter.

   The Edit Action Services panel appears (Figure 76). This panel shows any existing services that you have specified for the action.

**Figure 76   Edit Action Services panel**

```
DEKK ------------ Edit Action Services  (for DEMO)      ------ Row 1 to 6 of 6
COMMAND ===>                                                  SCROLL ===> CSR

Please select an option and press Enter.

I = Insert a blank step   L = Duplicate a step
A/B = Copy After or Before D = Delete step
E/S = Edit syntax and options M = Manage Object Sets  P = Preview
? = Insert from selection list U = Update Service Z = Zoom object name

JCL Options DSN . .

Act Service  Object Name       Type Part Owner   Name
---------------------------------------------------------------
BMCSTATS ASU102.%               TS
```

**NOTE**

JCL Execution does not execute SQL commands in worklist format. SQL commands must be generated by BMCTRIG in standard JCL format.
2 To preview the Service Syntax, enter P in the Act field for the service.

An ISPF browse panel displays to show the expected Service Syntax generated (Figure 77).

**Figure 77 Service Syntax generated**

```
VIEW    RDAPKM.TSU63312.$19MAR13.$122711.$4964221    Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** *********************************** Top of Data ***********************************
000001
000002  -BMCU 000001 ASUSMAIN
000003  BMCSTATS TABLESPACE ASU102.%
000004  TABLE (ALL)
000005  UPDATEDB2 ALL
000006  RECALL Y
000007  SORTEVT SYSDA
000008  INDEX N
****** *********************************** Bottom of Data ***********************************
-AL AnnOne_Syntax                          RDAALC
 *********************************** Bottom of data ***********************************
```

To preview syntax from the Service Syntax List

1 From the Service Syntax List panel (Figure 78), type V in the Act field for the service.

**Figure 78 Service Syntax List**

```
ASUUASSL --------- Service Syntax List (for DEMO)    -- Row 38 to 65 of 97
COMMAND ===>                                                   SCROLL ===> CSR
Service:                BMCSTATS
Current syntax: Clear? N  DEMO
Default syntax: Clear? N
Please select an option or change Search Criteria and press Enter.
I = Insert a new syntax        E = Edit service syntax
C = Copy                     S = Select
A = Actions using the syntax F = Set as default
D = Delete                   V = Preview

--> %                                        %         <-- Search Criteria
Act Name ----------------------------------- Owner --- Description (partial) ---
DEMO                                     BMC       DEMO
```
2 From the Service Syntax List panel that is displayed (Figure 79), enter the object type to which you want to apply syntax.

**Figure 79  Preview Service Syntax Panel**

ASUUSPOP ----- Preview Service Syntax ----------
COMMAND ===> 

Service: BMCCOPY  
Syntax: MVSTAD1-BMCCOPY

Specify the Object type to apply syntax and press Enter.

Object Type . . 1. Tablespace  
2. Index  
3. Tablespace Set  
4. Index Space  
5. Stogroup  
6. Volume

Press ENTER to accept input and continue.  
Enter CANCEL to disregard input and return.

### Generating actions

Use this procedure to generate or submit an action. Action generation involves specifying data sets for the job and setting various processing options. DASD MANAGER PLUS places the worklist in its own data set and generates the JCL into a separate data set for the JCL to run the worklist. You can assign values to the job generation options or accept the defaults.

**NOTE**

This procedure assumes that the action exists, and that you have specified the services and syntax. For information about creating an action, see “To create an action” on page 271.

**To generate an action**

1 Display an action selection list with the name of the action to generate.

2 Type G in the Act field next to the action that you want to generate.
The first of two Action Job Generation panels appears (Figure 80).

**Figure 80  Action Job Generation panel 1**

```
ASUIAG0 ----------------------- Action Job Generation -------------------------
Command ===> 
Action: RDAPKM.DEMO
Status: GENERATED NOT EXECUTED
Select JCL Type and Build Options, then press Enter.
JCL Type . . . 1. Build Initial JCL
2. Build Restart JCL
3. Build Startover JCL

JCL Build Options
  Record Events 1 1. Yes
                    2. No
  Restart Parm

Commands: HELP END CANCEL
```

The **Action** field displays the action being generated.

The **Status** field shows the current status of the action. The DO_WORKIDS table provides the status. The field can be set to one of the following:

- **Not Generated** — Indicates the action has not been generated
- **Generated Not Executed** — Indicates the action has been generated but not submitted
- **Executed Successfully** — Indicates the action has been generated, submitted, and ran successfully
- **Executed with Errors** — Indicates the action has been generated, submitted, and completed unsuccessfully

3 The **JCL Type** field indicates what type of JCL to generate. The value is based on the **Action** status. Specify the **JCL Type** as follows:

- **Type 1** to initially start the job.
- **Type 2** to restart the job.
Use this option if the execution job failed previously. JCL is created from the old job. The Action Status must be **Executed with Errors** before you can select this option. If you select this JCL type, you can use RESTART PARM to insert parameters into job stream. In addition, you must select build JCL on the next panel to regenerate the JCL.

- **Type 3** to start the execution job over from the beginning. JCL is created from existing JCL.

4 In the **Record Events** field, type 1 to log events in the DASD MANAGER PLUS database.

DASD MANAGER PLUS inserts commands into the job stream that cause the execution program to insert entries in the EVENTS table. This option is valid only when the JCL Type is **Build Initial JCL** or **Build Startover JCL**.

5 In the **Restart Parm** field, specify information to pass to a restart job. The **Restart Parm** field is only available for a JCL Type of **Build Restart JCL**.

6 Press **Enter** to access the second Action Job Generation panel (Figure 81).

**Figure 81  Action Job Generation panel**

| ASUIAG1 --------------- Action Job Generation --------------- |
| Command ====> |
| Action : RDAPKM.DEMO |
| Status : GENERATED NOT EXECUTED |
| JCL PDF: AUS.DOPSEC.CNTL(PFASUB1) |

Specify Dataset Names

- JCL ......... 'RDAPKM.TEST.JCL(DEMO)'
- Worklist ....... 'RDAPKM.TEST.WORKLIST(DEMOW)'
- Diagnostics ....... SYSOUT

Select Processing Options, then press Enter to Continue.

- Override POF Values
- S Build Worklist
- S Edit Worklist
- S Build JCL
- S Edit JCL
- S Submit JCL

Commands: HELP END CANCEL PREVIOUS

The **Action** field displays the action being generated.

The **Status** field shows the current status of the Action. The DO_WORKIDS table provides the status. The field can be set to one of the following statuses:
Generating actions

- **Not Generated** — Indicates the action has not been generated
- **Generated Not Executed** — Indicates the action has been generated but not submitted
- **Executed Successfully** — Indicates the action has been generated, submitted, and ran successfully
- **Executed with Errors** — Indicates the action has been generated, submitted and completed unsuccessfully

**JCL POF** displays the name of the action POF, if one exists. If this field is blank, the product uses the user options.

**NOTE**

You can set default values for this panel by using the User Options panel of DASD MANAGER PLUS. See Chapter 3, “Setting up DASD MANAGER PLUS.”

If you use the &WKID variable in your JCL or worklist data sets, ensure that your action name is 8 characters or less. If the name is longer, &WKID8 uses only the first 8 characters of the action name.

7 In the **JCL** field, type the name of the data set in which to build the JCL file, or accept the default.

   The data set must be sequential or a member of an existing PDS. This field defaults to the data set that you specified in the user options.

8 In the **Worklist** field, type the name of the data set in which to build the worklist file, or accept the default.

   The data set must be sequential or a member of an existing PDS. This field defaults to the data set that you specified in the user options.

9 In the **Diagnostics** field, type the name of the data set in which to place the diagnostic messages that result from generating the JCL and worklist files.

   This file must be sequential. The default is SYSOUT.

10 In the **Override POF Values** field, type S to set the JCL Options overrides to use for this generation only.

   The product resets the variables to the user options immediately after generation.

11 In the **Build Worklist** field, type S to generate (or regenerate) the worklist.
12 In the **Edit Worklist** field, type **S** to display the worklist for viewing or editing after it is generated.

13 In the **Build JCL** field, type **S** to generate (or regenerate) the JCL.

14 In the **Edit JCL** field, type **S** to display the JCL for editing after it is generated.

15 In the **Submit JCL** field, type **S** to submit the JCL to execute the commands in the worklist.

   You can submit the job now or later.

16 Review your entries, and press **Enter** to process the job.

17 Depending on the processing options that you specified, perform one of the following steps:

   - If you chose to edit the worklist, the worklist is displayed. Press **END** to continue.
   
   - If you chose to edit the JCL, the JCL is displayed. Press **END** to continue. Alternatively, you can submit the job from the edit panel by typing **SUB** on the **COMMAND** line.
   
   - If you chose to submit the job by using the Submit Job option, the product generates the worklist and JCL, and displays the Action Generation panel. The message **ENTER** TO SUBMIT appears at the top-right portion of the panel.

18 Press **Enter** to submit the job for processing, or press **END** to exit the Action Generation panel without submitting the job.

---

**Restarting or starting over an action**

Use this procedure to restart an action that has stopped or had an error, or to start over an action that you have defined that contains multiple services. The Restart option starts a job from the last SYNC point or last incomplete worklist command. The Start Over option clears the SYNC table and reruns a job from the beginning. For steps to restart or start over a BMCTRIG-generated job, see Chapter 8, “Analyzing objects by using BMCTRIG.”
To restart an action

1 Analyze the Worklist Execution Log in the job output to determine the action service where the worklist stopped, as follows:

- Check the return code so that you can make any necessary corrections. See “Understanding return codes for batch programs” on page 306.
- Check the sequence number to verify where the worklist will restart.

2 Correct the problem, as follows:

- You might need to edit the worklist (but not rebuild the worklist). For more information about editing worklists, see the Execution chapter of the DASD MANAGER PLUS for DB2 Reference Manual.
- If services completed successfully before the worklist failed, you can restart the worklist with the service that failed, without having to edit the worklist.
- Determine whether to restart the worklist from the point where it failed or start the worklist over from the beginning.

3 Display an Action List panel containing the name of the action to restart.

4 Type G in the Act field beside the action.

The first Action Job Generation panel (Figure 80) appears. The Action field displays the action being generated.

Action Status shows the current status of the action, which it obtains from the DO_WORKIDS table.

- The field is blank if the product has not generated the action.
- A value of N indicates that you have generated the action, but have not run it.
- A value of R indicates that the action completed successfully.
- A value of S indicates that the action started but had an error.

5 In JCL Type, specify to generate Restart JCL or generate Startover JCL.

- To restart a worklist, select the Restart JCL option. DASD MANAGER PLUS inserts RESTART parameters into the JCL job stream. Use this option when the job has terminated abnormally in a previous run. The option is valid only if Action Status is Executed with Errors.
- To start a worklist over from the beginning and clear out the SYNC table for this action, select the StartOver JCL option.
6 In the **Record Events** field, type 1 to log events in the DASD MANAGER PLUS database.

DASD MANAGER PLUS inserts commands into the job stream that cause the Execution program to insert entries in the EVENTS table.

7 (optional) In the **Restart Parm** field, specify the parameters to pass to the utility that you are restarting.

8 Review your entries, and press **Enter** to proceed to the next Action Generation panel.

The second Action Generation panel appears.

The **Action** field displays the action being generated.

The **Status** field shows the current status of the Action. The DO_WORKIDS table provides the status. The field can be set to one of the following:

- **Not Generated** — Indicates the action has not been generated
- **Generated Not Executed** — Indicates the action has been generated but not submitted
- **Executed Successfully** — Indicates the action has been generated, submitted, and ran successfully
- **Executed with Errors** — Indicates the action has been generated, submitted, and completed unsuccessfully

**JCL POF** displays the name of the action POF, if one exists. If this field is blank, the product uses the user options.

9 In the **JCL DSN** field, leave the name as is (to replace the previously generated JCL) or specify a different data set name or member (to place the JCL in a different library or member).

**NOTE**

The **Worklist DSN** field contains the name of the data set into which the product previously generated the worklist. To replace this worklist, leave the name as it is.

10 In the **Diagnostic** field, type the name of the data set in which to place the diagnostic messages that result from generating the JCL and worklist files. This file must be sequential. The default is SYSOUT.
In the **Override POF Values** field, type **S** to set the JCL Options overrides to use for this generation only.

The product resets the variables to the user options immediately after generation.

In the **Edit Worklist** field, type **S** to display the worklist for viewing or editing before generating JCL.

In the **Build JCL** field, type **S** to generate (or regenerate) the JCL.

In the **Edit JCL** field, type **S** to display the JCL for editing after it is generated.

In the **Submit JCL** field, type **S** to submit the JCL to execute the commands in the worklist.

Review your entries, and press **Enter** to begin regenerating the JCL.

The Execution program locates the start sequence number in the worklist and restarts processing at that point.

---

**Understanding return codes for batch programs**

Table 35 describes the return codes for ASUSMAIN, ASUSCPRS, and ASUSUPRS.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>a warning message or an allocation error has occurred</td>
</tr>
<tr>
<td>8</td>
<td>one of the following events has occurred:</td>
</tr>
<tr>
<td></td>
<td>- error gathering table space or index statistics</td>
</tr>
<tr>
<td></td>
<td>- error writing statistics to DASD MANAGER PLUS or DB2 tables</td>
</tr>
<tr>
<td></td>
<td>- initialization error (bad plan, options module, or syntax parse error)</td>
</tr>
<tr>
<td>12</td>
<td>a SYSIN DD statement open error occurred</td>
</tr>
</tbody>
</table>
Table 36 describes the return codes for ASUSTRIG.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>an SQL error occurred when retrieving statistics from DASD MANAGER PLUS tables or when writing statistics to the EXCEPTIONS2 or EVENTS table</td>
</tr>
<tr>
<td>12</td>
<td>a SYSIN DD statement open error occurred, or an SQL error occurred while expanding wildcard object names</td>
</tr>
<tr>
<td>16</td>
<td>a syntax parse error occurred</td>
</tr>
<tr>
<td>20</td>
<td>an invalid option module exists</td>
</tr>
</tbody>
</table>

### Controlling access to actions, services, and service syntax

The OSAUTHCHK=Y/N option in the installation options module controls whether to perform authorization checks. If OSAUTHCHK=N, users who run the product have access to all actions. If OSAUTHCHK=Y, DASD MANAGER PLUS restricts access to actions, services, and service syntax based on the following rules:

- A user can access an action when all of the following conditions exist:
  - The user is the owner of the action.
  - The owner of the action is PUBLIC or blank.

- A user has the following characteristics:
  - primary ID
  - current SQLID

When you enter DASD MANAGER PLUS, your logon ID appears in the SQLID field on the main menu. If you change the field, DASD MANAGER PLUS issues an SQL SET CURRENT SQLID command to check your authorization to use that ID.
Controlling access to actions, services, and service syntax
Using object sets

This chapter contains the following topics:

Before you begin ................................................................. 309
  Overview of object sets ................................................... 310
  Wildcards for object names ............................................. 310
  Patterns for object sets .................................................. 311
  Set Specification panel (for object sets) ........................... 316
  Example of defining object sets ...................................... 318
  Common tasks .............................................................. 320
Using object sets from the main menu ................................. 321
  Creating object sets by using the main menu ..................... 322
  Creating dynamic SQL object sets .................................. 324
  Editing object sets ....................................................... 327
  Viewing object set results ............................................ 329
  Identifying actions associated with an object set ............... 330
  Copying object sets ..................................................... 331
  Deleting object sets ..................................................... 333
  Renaming object sets ................................................... 334
Using object sets from action services ................................. 335
  Creating an object set for an action service ...................... 336
  Associating an existing object set with an action ............... 337
  Associating an object set with an action service by name .... 338
  Viewing object set results ............................................ 339
  Controlling access to object sets ................................... 340

Before you begin

An object set is an ordered set of object name patterns and pattern types. For example, you can specify an object set if you want to apply an action only to certain objects or object types.
Before using object sets, you should have an understanding of DASD MANAGER PLUS actions. (For more information, see Chapter 4, “Maintaining and generating actions.”)

**Overview of object sets**

In an object set, you can enter object names or wildcard names. Object set entries can contain name patterns that specify DB2 objects to include or exclude from service generation.

Object sets are a DASD MANAGER PLUS extension of the DB2 object wildcard facility. Object sets extend and refine wildcards for DB2 object selection for utility generation. You can create and maintain object sets with one of the following methods:

<table>
<thead>
<tr>
<th>Path</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>by using the Object Sets option from the main menu</td>
<td>“Using object sets from the main menu” on page 321</td>
</tr>
<tr>
<td>while defining action services</td>
<td>“Using object sets from action services” on page 335</td>
</tr>
</tbody>
</table>

You can use either method to edit, copy, rename, and delete object sets, expand wildcards, and resolve objects for the set.

DASD MANAGER PLUS provides many types of object name patterns to select the DB2 objects to process. Also, DASD MANAGER PLUS can dynamically run your own SQL to locate objects to process.

**Wildcards for object names**

When you specify an object name, you can include wildcard characters. For most types of services, DASD MANAGER PLUS expands object names that include wildcards into one utility statement for each object that matches the wildcard object name pattern and object type. The characters shown in Table 37 are wildcards and use an SQL LIKE operator for expansion:

<table>
<thead>
<tr>
<th>Wildcard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>* (asterisk)</td>
<td>matches zero to any number of characters</td>
</tr>
<tr>
<td>% (percent)</td>
<td>matches zero to any number of characters</td>
</tr>
</tbody>
</table>
If the qualifier does not contain one of those characters, the product uses an equal (=) operator to locate a match. DASD MANAGER PLUS does not support delimited identifiers when you specify a qualifier in a pattern. However, patterns can resolve to delimited object names.

Although expanding single-pattern wildcards is adequate for many situations, some situations require more control over object selection during utility generation. For example, you might need to specify multiple wildcard object name patterns or exclude specific objects, which makes object sets a better choice than simple patterns.

An object set name cannot contain an asterisk (*), percent (%), or embedded spaces. The owner and object set name together must be a unique name. For more information about the owner name, see “Controlling access to object sets” on page 340.

**Patterns for object sets**

When resolving an object set to produce a list of objects, DASD MANAGER PLUS expands each specification and applies the resulting object set in the order that you define.

You can use the same object set for different types of objects. You can resolve the object set for table spaces, indexes, and storage groups, or a mixture of table spaces and indexes. For example, if you resolve the storage group (SG) pattern SG ABC% for table spaces for utility operations, you can then use TS and IX (a mixed list) to include table spaces and indexes but exclude specific indexes.

---

**Table 37  Wildcard characters (part 2 of 2)**

<table>
<thead>
<tr>
<th>Wildcard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ (underscore)</td>
<td>matches any single character</td>
</tr>
<tr>
<td>? (question mark)</td>
<td>matches any single character</td>
</tr>
</tbody>
</table>

---

*NOTE*

Using wildcards in an object name search can potentially fail to match any characters unless you complete the string with a wildcard. The search can fail if the name is stored in DB2 with trailing blanks. For example, when trying to match database name QZUD16, specifying *ZUD16 might not return any data, but specifying *ZUD16* will produce a match.

Although expanding single-pattern wildcards is adequate for many situations, some situations require more control over object selection during utility generation. For example, you might need to specify multiple wildcard object name patterns or exclude specific objects, which makes object sets a better choice than simple patterns.

An object set name cannot contain an asterisk (*), percent (%), or embedded spaces. The owner and object set name together must be a unique name. For more information about the owner name, see “Controlling access to object sets” on page 340.
DASD MANAGER PLUS supports the types of object patterns as shown in Table 38:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>File formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS</td>
<td>index space pattern</td>
<td>DBNAME.SPNAME</td>
</tr>
<tr>
<td>IX</td>
<td>index name pattern</td>
<td>CREATOR.IXNAME</td>
</tr>
<tr>
<td>OS</td>
<td>DASD MANAGER PLUS</td>
<td>CREATOR.OBJSET</td>
</tr>
<tr>
<td></td>
<td>object set name</td>
<td></td>
</tr>
<tr>
<td>PG</td>
<td>package name pattern</td>
<td>COLLECTIONID.PACKAGE</td>
</tr>
<tr>
<td>PL</td>
<td>plan name pattern</td>
<td>PLAN (one-part name)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CREATOR.PLAN (two-part name)</td>
</tr>
<tr>
<td>RG</td>
<td>RECOVERY MANAGER</td>
<td>CREATOR.RMGROUP</td>
</tr>
<tr>
<td></td>
<td>(RMGR) group name</td>
<td></td>
</tr>
<tr>
<td>SG</td>
<td>storage group name</td>
<td>STOGROUP (one-part name)</td>
</tr>
<tr>
<td></td>
<td>pattern</td>
<td>CREATOR.STOGROUP (two-part name)</td>
</tr>
<tr>
<td>SQ</td>
<td>dynamic SQL statement</td>
<td>blank</td>
</tr>
<tr>
<td>TB</td>
<td>table name pattern</td>
<td>CREATOR.TBNAME</td>
</tr>
<tr>
<td>TS</td>
<td>table space name</td>
<td>DBNAME.TSNAME (two-part name)</td>
</tr>
<tr>
<td></td>
<td>pattern</td>
<td>DBNAME.TSNAME.CREATOR (three-part name)</td>
</tr>
</tbody>
</table>

**NOTE**

If you specify OS in the Obj Type field of the Set Specification panel, DASD MANAGER ignores the specification fields to the right of the Name or Name Pattern field. Therefore, you cannot use the By Part, Begin Part, End Part, Incl IX, and Incl RI fields to further qualify an object set.

For information about valid object pattern lengths, see the *DASD MANAGER PLUS for DB2 Reference Manual*. For detailed information about using type SQ, see “Creating dynamic SQL object sets” on page 324.

Table 39 describes the result set for each pattern type and resolution type. DASD MANAGER PLUS defaults to resolving both table spaces and indexes unless the panel selection or batch syntax indicates a resolution type.

**NOTE**

Object sets will not return any table spaces or indexes for objects that are not defined in the catalog (for example, DSNDB01). To ensure good performance when specifying which objects to exclude from an object set, you should not add excludes for any objects that are not defined in the catalog.
<table>
<thead>
<tr>
<th>Pattern type</th>
<th>Resolution type</th>
<th>Result set</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS TS</td>
<td>no match</td>
<td></td>
</tr>
<tr>
<td>IX TS</td>
<td>indexes that DBNAME and SPNAME identify</td>
<td></td>
</tr>
<tr>
<td>SG TS</td>
<td>storage groups that contain indexes that DBNAME and SPNAME identify</td>
<td></td>
</tr>
<tr>
<td>blank</td>
<td>indexes that DBNAME and SPNAME identify</td>
<td></td>
</tr>
<tr>
<td>IX TS</td>
<td>no match</td>
<td></td>
</tr>
<tr>
<td>IX TS</td>
<td>indexes that CREATOR and IXNAME identify</td>
<td></td>
</tr>
<tr>
<td>SG TS</td>
<td>storage groups that contain indexes that CREATOR and IXNAME identify</td>
<td></td>
</tr>
<tr>
<td>blank</td>
<td>indexes that CREATOR and IXNAME identify</td>
<td></td>
</tr>
<tr>
<td>OS TS</td>
<td>tables spaces that resolve from processing the specifications within the DASD MANAGER object set</td>
<td></td>
</tr>
<tr>
<td>IX TS</td>
<td>indexes that resolve from processing the specifications within the DASD MANAGER object set</td>
<td></td>
</tr>
<tr>
<td>SG TS</td>
<td>storage groups that resolve from processing the specifications within the DASD MANAGER object set</td>
<td></td>
</tr>
<tr>
<td>blank</td>
<td>table spaces and indexes that resolve from processing the specifications within the DASD MANAGER object set and indexes for resulting table spaces if Include IX is Y</td>
<td></td>
</tr>
<tr>
<td>PG TS</td>
<td>table spaces that are referenced within the packages that COLLID and PGNAME identify</td>
<td></td>
</tr>
<tr>
<td>IX TS</td>
<td>indexes that are referenced within the packages that COLLID and PGNAME identify</td>
<td></td>
</tr>
<tr>
<td>SG TS</td>
<td>storage groups that contain table spaces or indexes referenced within the packages (that COLLID and PGNAME identify)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>storage groups for the indexes on the table spaces if Include IX is Y</td>
<td></td>
</tr>
<tr>
<td>blank</td>
<td>table spaces and indexes that are referenced within the packages that COLLID and PGNAME identify</td>
<td></td>
</tr>
</tbody>
</table>

If Include IX is Y, the command also includes indexes that are referenced by the resulting table spaces.
### Table 39  Resulting object sets for pattern and resolution types (part 2 of 3)

<table>
<thead>
<tr>
<th>Pattern type</th>
<th>Resolution type</th>
<th>Result set</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL</td>
<td>TS</td>
<td>table spaces that are referenced within the plans that PLNAME and CREATOR identify. If the plan contains packages, the plan also includes the table spaces that are referenced within the packages (current or last version).</td>
</tr>
<tr>
<td>IX</td>
<td></td>
<td>indexes that are referenced within the plans that PLNAME and CREATOR identify. If the plan contains packages, the plan also includes the indexes that are referenced within the packages (current or last version).</td>
</tr>
<tr>
<td>SG</td>
<td></td>
<td>storage groups that contain table spaces (tables) or indexes referenced within the plans (that PLNAME and CREATOR identify). If Include IX is Yes, it also includes indexes on the table spaces.</td>
</tr>
<tr>
<td>blank</td>
<td></td>
<td>table spaces and indexes that are referenced within the plans that PLNAME and CREATOR identify. If the plan contains packages, the plan also includes the table spaces and indexes that are referenced within the packages (current or last version). If Include IX is Yes, it also includes indexes on the resulting table spaces.</td>
</tr>
<tr>
<td>SG</td>
<td>TS</td>
<td>table spaces using the STOGROUP</td>
</tr>
<tr>
<td>IX</td>
<td></td>
<td>indexes using the STOGROUP</td>
</tr>
<tr>
<td>SG</td>
<td>storage groups that CREATOR and STOGROUP identify</td>
<td></td>
</tr>
<tr>
<td>blank</td>
<td></td>
<td>table spaces and indexes using the STOGROUP and indexes for resulting table spaces if Include IX is Yes</td>
</tr>
<tr>
<td>SQ</td>
<td>TS</td>
<td>tables spaces (type TS is the first variable) that the SQL execution returns. Tables spaces using the storage group (type SG is the first variable) that the SQL execution returns.</td>
</tr>
<tr>
<td>IX</td>
<td>indexes (type IX is the first variable) that the SQL execution returns. Indexes using the storage group (type SG is the first variable) that the SQL execution returns.</td>
<td></td>
</tr>
<tr>
<td>SG</td>
<td>storage groups that contain the table spaces (TS) and indexes (IX) that the SQL execution returns for types TS and IX. Storage groups (type SG is the first variable) that the SQL execution returns.</td>
<td></td>
</tr>
<tr>
<td>blank</td>
<td>table spaces and indexes that executing the SQL execution returns. Table spaces and indexes using the storage group (type SG is the first variable) that the SQL execution returns. Indexes on the resulting table spaces if Include IX is Yes.</td>
<td></td>
</tr>
</tbody>
</table>
If you set `Include RI` to `Y`, DASD MANAGER PLUS uses the table spaces in the result set to locate all referentially related table spaces. If you also specify `Include IX Y`, DASD MANAGER PLUS includes indexes for RI-added objects in the result set.

For better performance, if you plan to use the object set only for DASD MANAGER PLUS batch programs and plan always to include all indexes, do not set `Include IX` in each specification. Instead, use `INDEX Y` in the syntax. Similarly, if you always want to include RI for each specification, use `TABLESPACESET` in the syntax and `RI N` in object set specifications. However, if you need the set for online generation or more granular control, use the object set specification method.

### Table 39 Resulting object sets for pattern and resolution types (part 3 of 3)

<table>
<thead>
<tr>
<th>Pattern type</th>
<th>Resolution type</th>
<th>Result set</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB</td>
<td>TS</td>
<td>table spaces that contain tables that CREATOR and TBNAME identify</td>
</tr>
<tr>
<td></td>
<td>IX</td>
<td>indexes that are defined on tables that CREATOR and TBNAME identify</td>
</tr>
<tr>
<td>SG</td>
<td>storage groups that contain table spaces in which the tables (that CREATOR and TBNAME identify) are defined</td>
<td></td>
</tr>
<tr>
<td></td>
<td>indexes on the table spaces if <code>Include IX</code> is <code>Yes</code></td>
<td></td>
</tr>
<tr>
<td>blank</td>
<td>table spaces that contain tables (that CREATOR and TBNAME identify) and indexes on these tables</td>
<td></td>
</tr>
<tr>
<td></td>
<td>all indexes on the resulting table space if <code>Include IX</code> is <code>Yes</code></td>
<td></td>
</tr>
<tr>
<td>TS</td>
<td>TS</td>
<td>table spaces that the DBNAME, TSNAME, and CREATOR identify</td>
</tr>
<tr>
<td></td>
<td>IX</td>
<td>indexes defined on table spaces that DBNAME, TSNAME, and CREATOR identify</td>
</tr>
<tr>
<td>SG</td>
<td>storage groups that contain table spaces that DBNAME, TSNAME, and CREATOR identify and their indexes if <code>Include IX</code> is <code>Yes</code></td>
<td></td>
</tr>
<tr>
<td>blank</td>
<td>table spaces with DBNAME, TSNAME, and CREATOR and indexes on those tables if <code>Include IX</code> is <code>Yes</code></td>
<td></td>
</tr>
</tbody>
</table>
Set Specification panel (for object sets)

Figure 82 shows an example of the Set Specification panel.

The panel displays the following elements of an object set:

- **Object Set Owner. Object Set Name** at the top of the panel (RDAPKM.DEMO in Figure 82) shows the owner and name of the object set from creation.

- **Act** (also called the action column) indicates an activity (in the form of a one-character or two-character command) to manipulate the object set entry. For example, in the Act field, the R command repeats an entry. Also, Z allows you to zoom into the contents of a package name field to edit it. For example, if the name or name pattern is longer than the field can show on the panel, you will see a plus sign (+). Use Z to see the full name or pattern.

- **Incl/Excl** indicates the object set entry type: + (Include) or - (Exclude).

**NOTE**

Consider the following additional information:

- You must specify at least one Include rule before any Exclude rules.

- Object sets will not return any table spaces or indexes for objects that are not defined in the catalog (for example, DSNDB01). To ensure good performance, when specifying which objects to exclude from an object set, you should not add excludes for any objects that are not defined in the catalog.

- **Obj Type** indicates the type of object pattern for the specification. Table 38 lists valid object patterns.
Name or Name Pattern is the explicit name or pattern (for more information, see “Patterns for object sets” on page 311). If the name or name pattern is longer than the field can show on the panel, you will see a plus sign (+) after the name.

By Part indicates whether to treat partitioned objects by partition. When used in conjunction with the Incl/Excl field, By Part indicates whether to include or exclude individual partitions.

For example, if the value of By Part is Y and you specify Exclude (-) in the Incl/Excl field, you can exclude a partition or partition range by using the Begin Part and End Part fields. However, if the value of By Part is N, no partitions will be excluded. For more information, see “Example of defining object sets.” If Begin Part and End Part are not zero, and you do not set By Part to Y, DASD MANAGER automatically sets By Part to Y.

Begin Part indicates the beginning number in the partition range of 1 through 4096. If you omit this value or use 0, all partitions are assumed.

End Part indicates the end number in the partition range of 1 through 4096. If you omit this value or use 0, all partitions are assumed.

Include IX specifies whether to include associated indexes.

Include RI specifies whether to include referential integrity (RI) related table spaces.

Include LOB specifies how to include associated LOB objects. Valid values are:

- N — specifies not to include auxiliary objects related to name pattern.
- Y — specifies to include LOB base objects and auxiliary objects related to name pattern.
- B — specifies to include LOB base objects related to name pattern only.
- O — specifies to include auxiliary objects related to name pattern only.
**Example of defining object sets**

- **Include XML** specifies how to include associated XML objects. Valid values are:
  - **N** — specifies not to include auxiliary objects related to name pattern.
  - **Y** — specifies to include XML base objects and auxiliary objects related to name pattern.
  - **B** — specifies to include XML base objects related to name pattern only.
  - **O** — specifies to include auxiliary objects related to name pattern only.

- **Include HST** specifies whether to include history related objects. Valid values are:
  - **N** — specifies not to include auxiliary objects related to name pattern.
  - **Y** — specifies to include HST base objects and auxiliary objects related to name pattern.
  - **B** — specifies to include HST base objects related to name pattern only.
  - **O** — specifies to include auxiliary objects related to name pattern only.

**Example of defining object sets**

Figure 83 shows an example of how to define object sets.

**Figure 83  Example Set Specification panel**

<table>
<thead>
<tr>
<th>Incl/Obj</th>
<th>Name or Type</th>
<th>Pattern</th>
<th>By</th>
<th>Begin</th>
<th>End</th>
<th>IX</th>
<th>RI</th>
<th>LOB</th>
<th>XML</th>
<th>HST</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>TS</td>
<td>CCBDBSIM.%</td>
<td>Y</td>
<td>1</td>
<td>1000</td>
<td>Y</td>
<td>N</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>+</td>
<td>IX</td>
<td>RDAJXS.%</td>
<td>N</td>
<td>0</td>
<td>0</td>
<td>N</td>
<td>N</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>-</td>
<td>TS</td>
<td>CCBDBSIM.CCBTSSEG</td>
<td>Y</td>
<td>7</td>
<td>13</td>
<td>Y</td>
<td>N</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>+</td>
<td>TS</td>
<td>CCBP.TS</td>
<td>N</td>
<td>0</td>
<td>0</td>
<td>N</td>
<td>N</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>
To extend and refine object selection, the object set contains the following entries:

- The first include (+) entry adds all table spaces with the name CCBDBSIM.% and all related indexes beginning with partition 1 and ending with partition 1000.

- The second include (+) entry adds all indexes with a name that matches RDAJXS.%.

- To further refine the selection, the first exclude (–) entry excludes partitions 7 through 13 of the CCBDBSIM.CCBTSSEG table space and its indexes.

- The third include (+) entry adds the table space CCBP.TS.

**NOTE**

Actual object results depend on the object set resolution type. For more information, see “Viewing object set results” on page 329.

An exclude specification within an object set for a threshold or corrective action definition does not imply that the object is excluded from BMCTRIG evaluation. BMCTRIG expands the object sets for thresholds and corrective action definitions to determine which objects are in a result set. Because of precedence, BMCTRIG might evaluate the object later because of a more general wildcard threshold or corrective actions definition. Use one of the following methods to ensure that an object is excluded from evaluation when using system triggers:

- Always run BMCTRIG against an object set. That is, use multiple trigger jobs.
- Specify a specific threshold for that object, using high values as the trigger value.

For more information about triggers, see Chapter 8, “Analyzing objects by using BMCTRIG.”
## Common tasks

Table 40 describes some common tasks for setting up and maintaining object sets.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an object set.</td>
<td>“Creating object sets by using the main menu” on page 322</td>
</tr>
<tr>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td>“Creating an object set for an action service” on page 336</td>
</tr>
<tr>
<td>Edit the object set, as follows:</td>
<td>“Editing object sets” on page 327</td>
</tr>
<tr>
<td>■ include additional objects in an object set and remove objects</td>
<td></td>
</tr>
<tr>
<td>■ add object set entry lines</td>
<td></td>
</tr>
<tr>
<td>■ add Include (+) object set entries</td>
<td></td>
</tr>
<tr>
<td>■ add Exclude (–) object set entries</td>
<td></td>
</tr>
<tr>
<td>Work with object sets, as follows:</td>
<td>“Associating an existing object set with an action” on page 337</td>
</tr>
<tr>
<td>■ associate an object set with a service</td>
<td></td>
</tr>
<tr>
<td>■ use the Object Set List</td>
<td></td>
</tr>
<tr>
<td>■ use the Connect command</td>
<td></td>
</tr>
<tr>
<td>■ browse object sets</td>
<td></td>
</tr>
<tr>
<td>■ change object set selection</td>
<td></td>
</tr>
<tr>
<td>■ associate an existing object set with an action by specifying the object set name explicitly</td>
<td></td>
</tr>
<tr>
<td>Copy an object set.</td>
<td>“Copying object sets” on page 331</td>
</tr>
<tr>
<td>Delete an object set.</td>
<td>“Deleting object sets” on page 333</td>
</tr>
<tr>
<td>Rename an object set.</td>
<td>“Renaming object sets” on page 334</td>
</tr>
<tr>
<td>View objects for a particular specification and use the X command to exclude objects from a wildcard specification.</td>
<td>“Viewing object set results” on page 329</td>
</tr>
<tr>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td>“Editing object sets” on page 327</td>
</tr>
<tr>
<td>Identify actions associated with an object set.</td>
<td>“Identifying actions associated with an object set” on page 330</td>
</tr>
<tr>
<td>Creating dynamic SQL in a specification.</td>
<td>“Creating dynamic SQL object sets” on page 324</td>
</tr>
</tbody>
</table>
Using object sets from the main menu

By selecting **Object Sets** from the main menu, you can maintain object sets separately from any action step. This stand-alone path (Figure 84) is similar to the action path, but you cannot associate an object with an action. You can use either path to edit, copy, rename, and delete object sets and perform operations against them. To associate an object set with an action, see “Using object sets from action services” on page 335.

**Figure 84  Object set stand-alone path**

Use the stand-alone path for the following elements:

- multiple actions
- one or more individual actions

The object set provides all object specification.

- system-level thresholds and corrective actions
- automation
Creating object sets by using the main menu

Managing object sets through the stand-alone path also helps to perform the following tasks:

- identify the actions with which the object sets are associated
- create an object set without immediately associating the object set with an action

For example, you might know which objects you want but not the utilities that you want to run against the objects. You could use an object set to specify a set of objects.

Creating object sets by using the main menu

Use this procedure to set up an object set outside an action by using the Object Sets option on the main menu.

To create object sets

1. On the DASD MANAGER PLUS main menu, select Object Sets and press Enter.

   The Object Sets panel appears (Figure 85).

   **Figure 85  Object Sets panel**

   DEAE ------------------------------ Object Sets -------------------------------
   COMMAND ===>  Please select an option and press ENTER.
   ===> 1. List Object Sets
          2. Create Object Set
          3. Edit Object Set Spec
          4. Rename Object Set
          5. Copy Object Set
          6. Delete Object Set
   Set Name . . . %                   (New, existing, or wildcard)
   Set Owner . . . RDAEXK            (Optional, Default=Current User ID)

2. Select Create Object Set, enter an object set name and owner name, and press Enter.

   The object set name cannot contain an asterisk (*), percent sign (%), or embedded spaces. For more information about the owner name, see “Controlling access to object sets” on page 340. Together, the owner and object set name must form a unique name.

   The Create Object Set panel appears (Figure 86).
Creating object sets by using the main menu

Chapter 5 Using object sets 323

Figure 86  Create Object Set panel

On the Create Object Set panel, specify the following information and press Enter:

- object set owner (where the default is your TSO logon ID)
- object set name
- *(optional)* object set description

The Set Specification panel appears *(Figure 82)*.

Enter the following information:

- specification type *(Include or Exclude)*
- object type *(TS, IX, TB, SG, PL, PG, IS, OS, or SQ)*
- name pattern *(SQ does not require a name)*
- partition *(By Part, Begin Part, and End Part)*
- whether to include related indexes
- whether to include RI related table spaces
- whether to include LOB related objects
- whether to include XML related objects
- whether to include HST related objects

**NOTE**

If you specify OS in the **Obj Type** field of the Set Specification panel, DASD MANAGER ignores the specification fields to the right of the **Name or Name Pattern** field. Therefore, you cannot use the **By Part, Begin Part, End Part, Incl IX, Incl RI, Incl LOB, Incl XML, and Incl HST** fields to further qualify an object set.

For more information about how DASD MANAGER PLUS interprets patterns based on object type, see “Patterns for object sets” on page 311.
Creating dynamic SQL object sets

5 (optional) Use the following commands to edit the object set:

- I inserts a line for you to add a new specification.
- R inserts a line that repeats the previous line (for faster editing).
- D deletes a specification line.
- C allows you to add a comment.
- E expands wildcards that you have entered.
- S allows you to add SQL for a SQ-type specification.
- ? opens a pop-up window so that you can add a pattern.
- Z opens a pop-up window so that you can zoom into the contents of a long object name field.

6 Press END to save the object set.

Creating dynamic SQL object sets

Use this procedure to create a dynamic SQL object set. DASD MANAGER PLUS reads the temporary data set and verifies it by performing SQL PREPARE, followed by a cursor OPEN, FETCH of one row, and CLOSE. DASD MANAGER PLUS saves the SQL in the repository with the final specification changes.

To create a dynamic SQL object set

1 On the DASD MANAGER PLUS main menu, select Object Sets and press Enter.

The Object Sets panel appears (Figure 87).

Figure 87 Object Sets panel

2 Select Create Object Set, enter an object set name and owner name, and press Enter.
The object set name cannot contain an asterisk (*), percent sign (%), or embedded spaces. For more information about the owner name, see “Controlling access to object sets” on page 340. Together, the owner and object set name must form a unique name.

The Create Object Set panel appears (Figure 88).

Figure 88 Create Object Set panel

![Create Object Set panel](image)

3 On the Create Object Set panel, specify the following information and press Enter:

- object set owner (where the default is your TSO logon ID)
- object set name
- (optional) object set description

The Set Specification panel appears (Figure 89).

Figure 89 Set Specification panel

![Set Specification panel](image)
4 Enter the following information:

- action (S) for SQL text
- specification type (Include or Exclude)
- SQ for object type (dynamic SQL statement)
- partition information (By Part, Begin Part, End Part)
- whether to include related indexes
- whether to include RI-related table spaces
- whether to include LOB related objects
- whether to include XML related objects
- whether to include HST related objects

DASD MANAGER PLUS opens ISPF Edit on a temporary data set for you to enter SQL statements. If this specification already exists, DASD MANAGER PLUS stores the SQL statements in the temporary data set.

5 Enter a valid SQL SELECT statement in ISPF Edit. The SQL text must return the following variables:

- TS, IX, or SG (two-character literal)
- DBNAME for TS or CREATOR (up to 128 characters) for IX or SG
- TS name (up to eight characters), IX name (up to 128 characters), or STOGROUP name (up to 128 characters)
- optional partition number 0 through 4096

If not specified, the partition number is set to 0.

Following are some SQL text examples for object types TS, IX, and SG:

<table>
<thead>
<tr>
<th>Object type</th>
<th>SQL statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS</td>
<td>SELECT 'TS'. DBNAME, NAME FROM SYSSBM.SYSTABLESPACE WHERE DBNAME LIKE 'QZU%'</td>
</tr>
<tr>
<td></td>
<td>SELECT 'TS'. DBNAME, TSNAME, PARTITION FROM SYSSBM.SYSTABLEPART WHERE DBNAME='QZUDPT22' AND TSNAME='QZUS0122' AND PARTITION IN (4090, 4092, 4094, 4096)</td>
</tr>
<tr>
<td>IX</td>
<td>SELECT 'IX'. CREATOR, NAME FROM SYSSBM.SYINDEXES WHERE TBCREATOR='QZU'</td>
</tr>
<tr>
<td></td>
<td>SELECT 'IX'. IXCREATOR, IXNAME, PARTITION FROM SYSSBM.SYINDEXEXPART WHERE IXCREATOR='QZU' AND PARTITION &gt; 100 AND PARTITION &lt;= 200</td>
</tr>
</tbody>
</table>
Use this procedure to edit and view wildcard expansions for an individual specification and the resulting set. This procedure also shows how to include objects in and exclude objects from an object set.

**To view specification pattern expansions**

1. Access the Set Specification panel as instructed in “To create object sets” on page 322.

2. In the **Act** field, type **E** and press **Enter**.

3. To exclude objects from an expanded Include list (by expanding a wildcard object set entry for an include), proceed as follows:
   
   **A**. In the **Act** field, insert a line by typing **I** (Insert).
   
   **B**. In the **Incl/Excl** field, type a plus (+) character.
   
   **C**. Create a wildcard object set entry by typing a wildcard name.
   
   **D**. In the **Act** field by the wildcard object set entry, type **E** (Expand wildcard), and press **Enter**.

6. Press END to exit the editing panel.

### Editing object sets

<table>
<thead>
<tr>
<th>Object type</th>
<th>SQL statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG</td>
<td><code>SELECT 'SG', 'ANY', STORNAME FROM SYSIBM.SYSTABLEPART WHERE DBNAME='QZUD11' AND TSNAME='QZUS0111'</code></td>
</tr>
<tr>
<td></td>
<td><code>SELECT 'SG', 'ANY', STORNAME, PARTITION FROM SYSIBM.SYSTABLEPART WHERE DBNAME='QZUD11' AND TSNAME='QZUS0111' AND PARTITION BETWEEN 1 AND 4</code></td>
</tr>
</tbody>
</table>
The Specification Expansion panel for the wildcard object appears (Figure 90).

**Figure 90  Excluding objects from an object set on Specification Expansion panel**

---

**E** In the *Act* field, type *X* (Exclude) by any objects that you want to remove from the object set, and press END.

The Set Specification panel reappears with new Exclude (–) entries based on your selections.

**4** To include more objects from an expanded exclude list (by expanding a wildcard object set entry for an exclude), proceed as follows:

**A** In the *Act* field, insert a line by typing *I* (Insert).

**B** In the *Incl/Excl* field, type a minus (–) character.

**C** Create a wildcard object set entry by typing a wildcard name.

**D** In the *Act* field by the wildcard object set entry, type *E* (Expand wildcard), and press Enter.

The Specification Expansion panel for the wildcard object appears.

**E** In the *Act* field, type *X* (Include) by any objects that you want to add to the object set, and press END.

The Set Specification panel reappears with new Include (+) entries based on your selections.
Viewing object set results

Use this procedure to view the results of an object set.

To view object set results

1. On the DASD MANAGER PLUS main menu, select Object Sets and press Enter.

2. On the Object Sets panel, select List Object Sets and press Enter.

   The Object Set List panel appears (Figure 91).

   **Figure 91  Object Set List panel**

   ![Object Set List panel](image)

   DECA ------------------ Object Set List RDAJXS.% -------------- Row 1 to 6 of 6
   COMMAND ===>                                                 Scroll ===> PAGE
   Action column options:
   I = Insert/New  C = Copy  E = Edit Specs  U = Edit Object Set  A = Action XRef
   R = Rename       D = Delete       V = View Objects       T = connecT to Action
   Act Owner     Name                 Description                Updated
   RDAJXS   JENOBSET                                       2005-08-24-11.13
   RDAJXS   JENSET                                         2005-09-14-13.16
   RDAJXS   JENSQL                                         2005-10-04-12.04
   RDAJXS   JESHRS                                        2005-10-03-12.21
   RDAJXS   JXSSET                                         2005-10-03-12.24
   RDAJXS   JXSSQL                                         2005-10-04-12.02
   ************************************************** Bottom of data **************************************************

3. Type V in the Act field, and press Enter.

   The View Objects panel appears (Figure 92).

   **Figure 92  View Objects panel**

   ![View Objects panel](image)

   DECA ------------------ View Objects in RDAJXS .JXSSET -------------- Command ===> Scroll ===> PAGE
   Resolve for objects of type ===>   1. Tablespaces
                                       2. Indexes
                                       3. Stogroups
                                       4. Tablespaces and Indexes
   Additional wildcard pattern: Action column option: X = Generate explicit exclude entry
   Act Type
   ************************************************** Bottom of data **************************************************
4  Select from the types of objects to view.

5  (optional) Enter an additional wildcard pattern, and press Enter.

The additional wildcard pattern adds a temporary specification as the first specification in the list of object set and wildcard specifications.

DASD MANAGER PLUS displays a list of the objects in the object set.

---

NOTE
Alternately, you can enter RESOLVE at the COMMAND line on the Set Specification panel for the same results.

---

Identifying actions associated with an object set

Use this procedure to identify the actions that are associated with a specific object set.

---

NOTE
Always identify associated actions before modifying or deleting an object set.

---

To identify actions that are associated with an object set

1  Access the Object Set List panel as follows:

   A  On the DASD MANAGER PLUS main menu, select Object Sets and press Enter.

   B  On the Object Sets panel, select List Object Sets and press Enter.

   The Object Set List appears (Figure 91).

2  In the Act field, type A and press Enter.

   The Object Set Action Impact List appears (Figure 93).
Figure 93  Object Set Action Impact List panel

The fields on the Object Set Action Impact List are as follows:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Steps for Object Set Owner</td>
<td>name of the object set owner</td>
</tr>
<tr>
<td>Name</td>
<td>name of the object set that you are checking for associations</td>
</tr>
<tr>
<td>Action Name</td>
<td>names of the actions that are associated with this object set</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>numbers of the action steps that reference this object set</td>
</tr>
<tr>
<td>Service Name</td>
<td>names of the services in the designated action step</td>
</tr>
</tbody>
</table>

**Copying object sets**

Use this procedure to copy an object set.

**To copy an object set**

1. On the DASD MANAGER PLUS main menu, select Object Sets and press Enter.
2. On the main object set panel (Figure 85), specify an object set to copy, as follows:
   - A. Select Copy Object Set.
   - B. (*optional*) In the Set Owner field, specify the owner of the object set to copy.

      If you leave this field blank, the default value is your TSO logon ID.
In the **Set Name** field, enter the name of an existing object set to copy.

You cannot enter a wildcard.

Press Enter.

**NOTE**

Alternately, you can open the Object Set List from the main object set panel by selecting **List Object Sets** and specifying a wildcard pattern for the owner or object set name. Then, enter C in the **Act** field next to the object set that you want to copy.

The Copy Object Set panel opens (**Figure 94**).

**Figure 94  Copy Object Set panel**

Enter a new owner name, a name for the copy of the object set, and a short description.

The object set name cannot contain an asterisk (*), percent sign (%), or embedded spaces. Together, the owner and object set name must form a unique name. For more information about setting the owner name, see “Controlling access to object sets” on page 340.

Press Enter.

*NOTE*

Alternately, you can open the Object Set List from the main object set panel by selecting **List Object Sets** and specifying a wildcard pattern for the owner or object set name. Then, enter C in the **Act** field next to the object set that you want to copy.

The Copy Object Set panel opens (**Figure 94**).

**Figure 94  Copy Object Set panel**

Enter a new owner name, a name for the copy of the object set, and a short description.

The object set name cannot contain an asterisk (*), percent sign (%), or embedded spaces. Together, the owner and object set name must form a unique name. For more information about setting the owner name, see “Controlling access to object sets” on page 340.

Press Enter.
Deleting object sets

Use this procedure to delete an object set.

**To delete an object set**

1. On the DASD MANAGER PLUS main menu, select *Object Sets* and press **Enter**.
2. On the main object set panel (Figure 85), specify an object set to delete as follows:
   
   **A** Select **Delete Object Set**.

   **B** *(optional)* In the **Set Owner** field, specify the owner of the object set to delete.

   If you leave this field blank, the default value is your TSO logon ID. For more information about setting the owner name, see “Controlling access to object sets” on page 340.

   **C** In the **Set Name** field, enter the name of an existing object set to delete.

   You cannot enter a wildcard.

   **D** Press **Enter**.

**NOTE**

Alternately, you can open the Object Set List from the main object set panel by selecting **List Object Sets** and specifying a wildcard pattern for the owner or object set name. Then, enter **D** in the **Act** field next to the object set that you want to delete.

The Delete Object Set panel opens (Figure 95).

**Figure 95  Delete Object Set panel**

```
COMMAND ===>

Are you sure you want to delete this Object Set? N (Y=OK to delete)
Set Owner . . . RDAEXK
Set Name . . . BACKUP1
Set Description   BACKUP COPY OF OBJECT SET

Press ENTER key to continue object set delete.
Enter CANCEL or END command to return to previous panel.
```

3. Type **Y** and press **Enter**.
Renaming object sets

Use this procedure to rename an object set.

**To rename an object set**

1. On the DASD MANAGER PLUS main menu, select Object Sets and press Enter.
2. On the main object set panel (Figure 85), specify an object set to rename, as follows:
   
   **A** Select Rename Object Set.
   
   **B** *(optional)* In the **Set Owner** field, specify the owner of the object set to rename.
   
   If you leave this field blank, the default value is your TSO logon ID. For more information about setting the owner name, see “Controlling access to object sets” on page 340.
   
   **C** In the **Set Name** field, enter the name of an existing object set to rename.
   
   You cannot enter a wildcard.
   
   **D** Press Enter.

**NOTE**

Alternately, you can open the Object Set List from the main object set panel by selecting **List Object Sets** and specifying a wildcard pattern for the owner or object set name. Then, enter **R** in the **Act** field next to the object set that you want to rename.

The Rename Object Set panel opens (Figure 96).

**Figure 96  Rename Object Set panel**

```
COMMAND ===> ____________________________________________
Type new Object Set owner or name:

Set Owner . . . RDAEXK_
Set Name . . . NEWNAME___________
Set Description . . ANOTHER TEST

Press ENTER key to complete operation.
Enter CANCEL or END command to return to previous panel.
```
3 In the **Set Name** field, type the new name in the field to rename the object set.

You can also change the name of the set owner. The object set name cannot contain an asterisk (*), percent sign (%), or embedded spaces. Together, the owner and object set name must form a unique name. For more information about setting the owner name, see “Controlling access to object sets” on page 340.

4 Press **Enter** to save your changes.

---

**Using object sets from action services**

One method of maintaining object sets, the action service path, is an extension of the process of specifying action services. Using this method, you can create a new object set, select one from a list, or perform the full range of object set maintenance (Figure 97).

**Figure 97 Object set action path**

If you specify a wildcard for only the object set name, object set management returns only the object sets that you created under your TSO logon ID.
Creating an object set for an action service

Use this procedure to set up an object set for an action service. Creating an object set this way automatically associates the object set with the action.

Before you begin

1 Create an action as instructed in “To create an action” on page 271.

2 Add or edit a service for the action as instructed in “To edit action services” on page 279.

To create an object set for an action service

1 Access object set management from the Edit Action Services panel, as follows:

A In the Act field, type M next to the service name.

B Tab to the Owner field and specify the object set owner.

C Tab to the Name field, type an asterisk (*) or percent sign (%), and press Enter.

The Object Set List panel appears (Figure 91).

D If you are viewing an expanded object set name list, you can enter I in an Act field to create a new object set, or enter E to edit the specifications of a newly created object set.

NOTE
You can leave this field blank to have object set management create the object set under your TSO logon ID.

D If you are viewing an expanded object set name list, you can enter I in an Act field to create a new object set, or enter E to edit the specifications of a newly created object set.

NOTE
You cannot use object sets for object selection within utility option editing. For more information, see “Editing action services” on page 279.

2 (optional) To add and remove object pattern specifications, see “Creating object sets by using the main menu” on page 322.

3 In the Act field, enter T to connect the object set to the action.
4 Press END to save the object set definition and to return to the Edit Action Services panel.

At this point, you have associated the object set with the action services. DASD MANAGER PLUS applies the object set to the service when it generates the action.

**Associating an existing object set with an action**

Use this procedure to associate an existing object set with an action by using the Object Set List. This method is appropriate when you do not know the object set name or want to browse object sets before selecting one. For an alternative method, see “Using object sets from action services” on page 335.

**Before you begin**

1 Create an action as instructed in “To create an action” on page 271.

2 Add or edit a service for the action as instructed in “To edit action services” on page 279.

**To associate an existing object set with an action service**

1 Access the Object Set List from the Edit Action Services panel, as follows:

   A In the Act field, type M next to the service name.

   B In the Owner field, type an owner name (which can include a wildcard).

   C In the Name field, type a name, which can include a wildcard, and press Enter.

The Object Set List panel appears (Figure 98).

**Figure 98 Object Set List panel**

<table>
<thead>
<tr>
<th>Act Owner</th>
<th>Name</th>
<th>Description</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDAEXK</td>
<td>NEW</td>
<td>NEW SET</td>
<td>2004-02-23-15.57</td>
</tr>
</tbody>
</table>

---
2 Review and select an object set to use for this action step, as follows:

A In the Act field, type E (Edit) beside any object set that you want to review and press Enter.

B After reviewing objects, press END to return to the Object Set List.

C In the Act field, type T (connect to action) next to the object set that you want to connect to the action, and press Enter.

D In the Act field beside another object set, type T (connect) and press Enter, to change your selection.

DASD MANAGER PLUS associates the action step with the new object set and automatically deletes the association with the previous object set.

3 Press END to save your changes.

The Edit Action Services panel reappears. The object set name that you selected appears in the Owner and Name fields.

### Associating an object set with an action service by name

Use this procedure to associate an existing object set to an action service, without using the Object Set List. Use this method when you know the object set name and do not need to browse object sets.

**Before you begin**

1 Create an action as instructed in “To create an action” on page 271.

2 Add or edit a service for the action as instructed in “To edit action services” on page 279.

**To associate an object set with an action service by name**

Specify the object set name on the Edit Action Services panel, as follows:

1 In the Owner field, type a object set owner name.

   Your TSO logon ID is the default value.

2 In the Name field, type the name of an existing object set, press END.

   DASD MANAGER PLUS automatically associates the object set with the action.
Viewing object set results

Use this procedure to view object set results.

Before you begin

1. Create an action as instructed in “To create an action” on page 271.

2. Add or edit a service for the action as instructed in “To edit action services” on page 279.

To view object set results

1. Access the Object Set List from the Edit Action Services panel, as follows:
   
   A. In the Act field, type M (manage object sets) next to the service name.
   
   B. Tab to the Owner field and type an owner name, which can include a wildcard (* or %).
   
   C. Tab to the Name field, type a name (which can include wildcard), and press Enter.

   The Object Set List panel appears (Figure 98).

2. In the Act field, type V next to the object set and press Enter.

   The View Objects panel appears (Figure 92).

3. Select from the types of objects to view.

4. (optional) Enter an additional wildcard pattern, and press Enter.

   The additional wildcard pattern adds a temporary specification as the first specification in the list of object set and wildcard specifications.

   DASD MANAGER PLUS displays a list of the objects in the object set.
Controlling access to object sets

In previous versions of DASD MANAGER PLUS, any user with access to the product could modify any filter (now called an *object set*). For backward compatibility, the default authorization setting gives all users open access to object sets.

The OSAUTHCHK=Y/N option in the installation options module controls whether to perform authorization checks. If OSAUTHCHK=N, users who run the product have access to all object sets. If OSAUTHCHK=Y, DASD MANAGER PLUS restricts access to object sets based on the following rules:

- A user can access an object set when all of the following conditions exist:
  - The user is the owner of the object set.
  - The owner of the object set is PUBLIC or blank.

- A user has the following characteristics:
  - primary ID
  - current SQLID

When you enter DASD MANAGER PLUS, your logon ID appears in the SQLID field on the main menu. If you change the field, DASD MANAGER PLUS issues an SQL SET CURRENT SQLID command to check your authorization to use the ID.
Collecting and managing statistics

This chapter contains the following topics:

Before you begin ................................................................. 342
   Overview of collection and update tasks ........................................ 343
   Tips for collecting statistics .................................................... 344
Collecting statistics ............................................................... 346
   IBM RUNSTATS utility ......................................................... 346
   BMCSTATS utility ............................................................. 346
   DASD MANAGER PLUS historical database ............................... 347
   BMCSTATS options ................................................................ 349
Sampling statistics ................................................................. 358
   Minimum size requirements ..................................................... 358
   Specifying sampling percentages ............................................. 359
   Restricting sampling by object type ......................................... 360
   Sampling table columns ....................................................... 360
Accessing the statistics display features .................................... 360
Displaying DB2 object lists ...................................................... 361
   Creating a DB2 object list ..................................................... 361
   Listing data sets used by a database ....................................... 364
   Updating DB2 catalog statistics ............................................. 365
Browsing and updating statistics ............................................. 366
   DB2 and BMCSTATS statistics .............................................. 366
   DB2 table space statistics .................................................... 367
   DB2 table statistics ........................................................... 368
   DB2 table space partition statistics ....................................... 369
   DB2 column statistics ........................................................ 371
   DB2 index statistics ........................................................... 373
   DB2 index partition statistics ............................................... 374
   Updating the DB2 catalog interactively ................................ 375
   Table space statistics update ............................................... 375
   Table statistics update ....................................................... 376
   Table space partition table statistics update ........................... 378
   Column statistics update ..................................................... 379
   Index statistics update ........................................................ 380
   Index partition statistics update .......................................... 382
Before you begin

The DASD MANAGER PLUS product provides a set of statistics tools that maximize your ability to administer DB2 objects by performing the following tasks:

- Providing a historical database
- Collecting statistics
- Updating the DB2 catalog statistics
- Viewing statistics
- Setting threshold values and actions
- Estimating space requirements

DASD MANAGER PLUS can collect statistics while the objects are in use and allow you to view the statistics interactively, as text or in graphs.

**NOTE**

Some BMC utilities can operate concurrently on the same object or partition. For information about which products can operate concurrently, see Chapter 2 of the *DASD MANAGER PLUS for DB2 Reference Manual*.

DASD MANAGER PLUS also provides the following utilities for managing statistics:

- BMCSTATS allows you to monitor growth patterns and other changes through a historical database of statistics.
BMCTRIG (discussed in Chapter 8, “Analyzing objects by using BMCTRIG”) provides these capabilities:

— allows you to set exception values

— scans the historical database to determine whether the exception values have been reached

— generates JCL for utilities and commands to address objects when the exception values are reached

Be sure to run BMCSTATS to collect statistics before you try to use the statistics display, update, and management tasks that this chapter describes. (See “Updating the DB2 catalog by using BMCSTATS” on page 390.)

**NOTE**

After upgrading to a new version of DASD MANAGER PLUS, you must run BMCSTATS on DB2 objects to populate the newly added columns.

A DB2 object list initiates the statistics display, update, and space estimation tasks. If you are not familiar with creating object lists in DASD MANAGER PLUS, see “Creating a DB2 object list” on page 361.

Other statistics tasks in this chapter use actions. If you are not familiar with actions, see Chapter 4, “Maintaining and generating actions.”

### Overview of collection and update tasks

Table 41 shows typical ways to use the statistics tasks that are in this chapter.

<table>
<thead>
<tr>
<th>To populate this database with statistics...</th>
<th>Run these programs...</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS statistics database</td>
<td>BMCSTATS</td>
</tr>
<tr>
<td></td>
<td>BMCCPRS</td>
</tr>
<tr>
<td>DB2 catalog</td>
<td>BMCSTATS</td>
</tr>
<tr>
<td></td>
<td>BMCUPRS</td>
</tr>
<tr>
<td>DB2 catalog history</td>
<td>BMCSTATS</td>
</tr>
<tr>
<td></td>
<td>BMCUPRS</td>
</tr>
</tbody>
</table>

---

Chapter 6  Collecting and managing statistics  343
Table 42 lists the detailed tasks in this chapter.

### Table 42  Statistics collection and update tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>List DB2 objects.</td>
<td>“Creating a DB2 object list” on page 361.</td>
</tr>
<tr>
<td>List DB2 object data sets.</td>
<td>“Listing data sets used by a database” on page 364.</td>
</tr>
<tr>
<td>Specify BMCSTATS job parameters.</td>
<td>“Updating the DB2 catalog by using BMCSTATS” on page 390.</td>
</tr>
<tr>
<td>Collect index statistics.</td>
<td></td>
</tr>
<tr>
<td>Collect space statistics only.</td>
<td></td>
</tr>
<tr>
<td>Save statistics in the DASD MANAGER PLUS database.</td>
<td></td>
</tr>
<tr>
<td>Update the DB2 catalog with BMCSTATS statistics.</td>
<td></td>
</tr>
<tr>
<td>Print a statistics report.</td>
<td></td>
</tr>
<tr>
<td>Delete old statistics.</td>
<td></td>
</tr>
<tr>
<td>Group statistics collected by page.</td>
<td></td>
</tr>
<tr>
<td>Specify the multitasking level for collecting statistics.</td>
<td></td>
</tr>
<tr>
<td>Use sampling for statistics collection.</td>
<td></td>
</tr>
<tr>
<td>Recall archived data sets during statistics collection.</td>
<td></td>
</tr>
<tr>
<td>Specify tables and columns for statistics collection.</td>
<td></td>
</tr>
<tr>
<td>Update DB2 statistics with BMCSTATS statistics.</td>
<td></td>
</tr>
<tr>
<td>Generate a BMCSTATS job.</td>
<td></td>
</tr>
<tr>
<td>Understand the job status codes.</td>
<td>“DB2 and BMCSTATS statistics” on page 366.</td>
</tr>
<tr>
<td>Compare DB2 statistics and BMCSTATS statistics.</td>
<td>“Updating the DB2 catalog interactively” on page 375.</td>
</tr>
<tr>
<td>Update DB2 statistics with your own values.</td>
<td>“Updating the DB2 catalog by using BMCUPRS” on page 398.</td>
</tr>
<tr>
<td>Update DB2 statistics with values from the DASD MANAGER PLUS database.</td>
<td></td>
</tr>
</tbody>
</table>

**Tips for collecting statistics**

**NOTE**

DASD MANAGER PLUS does not collect statistics for the following objects:

- indexes that contain keys with random ordering
- objects with names that do not convert to EBCDIC
Use the following suggestions to improve the performance of BMCSTATS:

- Turn off delete processing by not using the DELETEAGE parameter.
- Delete statistics after physical changes, for example, if you change the index columns.
- Use the supplied ASURRDEL sample SQL to periodically clear old statistics.
- Consider reorganizing the DASD MANAGER PLUS historical database and rebinding the plans to ensure optimization of the access path.
- Consider sampling statistics instead of collecting all statistics. Sampling greatly reduces resource consumption in producing statistics. If you need to use sampling, consider specifying a percentage of pages to sample. For more information, see “BMCSTATS options” on page 349.
- If you experience locking problems when updating the DB2 catalog, consider using the BMCUPRS utility to update catalog tables during a period of low activity. For example, you can use BMCSTATS SaveStats Y and UPDATEDB2 N for normal operation. During a quiet time, you can update the catalog by using BMCUPRS. You also can specify OPTIMIZECOMMIT N for BMCSTATS to increase the number of SQL COMMITS after SQL DELETE, INSERT, and UPDATE operations.
- Executing BMCSTATS against table space sets is resource intensive. You should include only table spaces that are related by referential integrity (RI).
- Consider the following items when collecting column statistics:
  - Collecting column statistics can be a time-consuming process, particularly on large tables. Collect column statistics only when you have a particular need and only on the relevant columns.
  - Collect statistics on all columns periodically, such as when the number of rows changes significantly.
- You can stack BMCSTATS commands for batch execution. To specify different option values for the same object type, place them in a separate command, as follows:

BMCSTATS TABLESPACE options
BMCSTATS TABLESPACE options
Be aware that SQL -530 messages appear when all of the following conditions exist:

— You are collecting history statistics, but not regular DB2 statistics.
— You are collecting history statistics, but have never collected catalog statistics.
— You are writing the SYSTABSTATS_HIST or SYSINDEXSTATS_HIST when no SYSTABSTATS or SYSINDEXSTATS records have ever been written for the requested table.

Under these circumstances, BMCSTATS continues to collect statistics even though the SQL -530 message appears. Only the table that caused the SQL -530 message will not be updated.

Collecting statistics

With DASD MANAGER PLUS, you can manage statistics that the IBM RUNSTATS utility or the BMC Software BMCSTATS utility collects. You can view and compare these statistics. You can also enter statistics from any other source that you choose.

IBM RUNSTATS utility

The RUNSTATS utility places statistical information in the DB2 catalog. DB2 uses RUNSTATS values to determine the optimal access path to DB2 structures. BMCSTATS collects the same statistics as the RUNSTATS utility, in addition to other statistics. The additional information that BMCSTATS provides can help you plan for future system requirements that increasing production activity causes.

NOTE

By analyzing the statistics, you can determine the cases in which you require RUNSTATS. See the IBM DB2 UDB for OS/390 SQL Reference.

BMCSTATS utility

The BMCSTATS utility (Figure 99) populates the DASD MANAGER PLUS database, which is in DB2 tables. You can use the additional statistics that BMCSTATS collects to determine when to run maintenance utilities on the physical objects. For example, BMCSTATS calculates the REORGLEVELS (the number of index levels necessary if you reorganize the object). By comparing REORGLEVELS with the current levels, you can determine whether reorganizing the index space will reduce the number of levels that the index requires.
The BMCSTATS utility also provides the PAGEGROUP facility, which allows you to review the statistics on a specified grouping of pages to uncover additional information about hot spots in the data. When you generate control statements for the BMCSTATS utility, you determine the number of pages to group. This facility and the graphic displays can help you locate areas of concentrated activity within a table space. You can display and compare statistics values for the first, last, and next-to-last time that you ran BMCSTATS.

Figure 99  Gathering statistics using the BMCSTATS utility

Table 43 shows the information that DASD MANAGER PLUS records in the statistics tables of its historical database. BMCSTATS stores this information, and BMCTRIG scans it. For complete information about the DASD MANAGER PLUS historical database, see the DASD MANAGER PLUS for DB2 Reference Manual.

Table 43  DASD MANAGER PLUS historical database (part 1 of 2)

<table>
<thead>
<tr>
<th>Table name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCATSnn.RS_COLDIST</td>
<td>distribution statistics on columns</td>
</tr>
<tr>
<td>BMCATSnn.RS_COLDISTSTAT</td>
<td>distribution statistics on columns at the partition level</td>
</tr>
<tr>
<td>BMCATSnn.RS_COLSTATS</td>
<td>statistics on each column in a partition</td>
</tr>
<tr>
<td>BMCATSnn.RS_COLUMNS</td>
<td>table columns</td>
</tr>
<tr>
<td>BMCATSnn.RS_INDEXES</td>
<td>indexes</td>
</tr>
<tr>
<td>BMCATSnn.RS_INDEXEXPART</td>
<td>index partitions</td>
</tr>
<tr>
<td>BMCATSnn.RS_INDEXPART_DIST</td>
<td>index partition page group</td>
</tr>
<tr>
<td>BMCATSnn.RS_KEYTARGETS</td>
<td>key targets</td>
</tr>
<tr>
<td>BMCATSnn.RS_KEYTARGETSTATS</td>
<td>key targets partitions</td>
</tr>
<tr>
<td>BMCATSnn.RS_KEYTGT_DIST</td>
<td>distribution statistics on key target columns</td>
</tr>
</tbody>
</table>
### Table 43  DASD MANAGER PLUS historical database (part 2 of 2)

<table>
<thead>
<tr>
<th>Table name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCATSnn.RS_KEYTGTDISTSTATS</td>
<td>distribution statistics on key target columns at the partition level</td>
</tr>
<tr>
<td>BMCATSnn.RS_LOBSTATS</td>
<td>LOB statistics</td>
</tr>
<tr>
<td>BMCATSnn.RS_STOGROUP</td>
<td>storage groups</td>
</tr>
<tr>
<td>BMCATSnn.RS_TABLEPART</td>
<td>table space partitions</td>
</tr>
<tr>
<td>BMCATSnn.RS_TABLES</td>
<td>tables</td>
</tr>
<tr>
<td>BMCATSnn.RS_TABLESPACE</td>
<td>table space statistics</td>
</tr>
<tr>
<td>BMCATSnn.RS_TSPART_DIST</td>
<td>table space partition page group</td>
</tr>
<tr>
<td>BMCATSnn.RS_VOLUMES</td>
<td>volumes</td>
</tr>
</tbody>
</table>
BMCSTATS options

When you collect statistics, you choose parameters on the first BMCSTATS panel (Figure 100). These parameters are by category.

Figure 100  BMCSTATS parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE</td>
<td>(Y/N/S Y=ALL Tables, N=No Tables, S=Select Tables)</td>
</tr>
<tr>
<td>INDEX</td>
<td>(Y/N Collect column statistics on all indexes)</td>
</tr>
<tr>
<td>GROUPSIZE</td>
<td>(0-180000 Number of pages in group for statistics)</td>
</tr>
<tr>
<td>SPACEONLY</td>
<td>(Y/N Collect space information only)</td>
</tr>
<tr>
<td>NUMCOLS</td>
<td>(1-64 Max index key columns to concatenate)</td>
</tr>
<tr>
<td>IXNUMQUANTILES</td>
<td>(1-100 Number of quantiles to collect)</td>
</tr>
<tr>
<td>NUMQCOLS</td>
<td>(1-64 Number of columns for quantiles)</td>
</tr>
<tr>
<td>FREQVAL</td>
<td>(Y/N Collect frequent value statistics)</td>
</tr>
<tr>
<td>COUNT</td>
<td>(1-300 Max number of frequent values to collect)</td>
</tr>
<tr>
<td>FREQTYPE</td>
<td>(M/L/B M=Most L=Least B=Both)</td>
</tr>
<tr>
<td>SAVESTATS</td>
<td>(Y/N Save statistics in STATS DB)</td>
</tr>
<tr>
<td>DELETEAGE</td>
<td>(0-32767 Days for deleting old STATS DB entries)</td>
</tr>
<tr>
<td>REPORT</td>
<td>(Y/N Print statistics report)</td>
</tr>
<tr>
<td>UPDATEDB2</td>
<td>(N/A/P/S N=none, A=All, P=Accesspath, S=Space)</td>
</tr>
<tr>
<td>DELETEHISTAGE</td>
<td>(0-32767 Days for deleting history table entries)</td>
</tr>
<tr>
<td>OMITCARDO</td>
<td>(Y/N - Bypass catalog update if cardinality is 0)</td>
</tr>
<tr>
<td>BADDJECTRC</td>
<td>(0-8 Return code when object is bypassed)</td>
</tr>
<tr>
<td>FORCEROLLUP</td>
<td>(Y/N Produce aggregate stats when missing parts)</td>
</tr>
<tr>
<td>MSGLEVEL</td>
<td>(0/1 0-Normal msgs, 1-Additional msgs)</td>
</tr>
<tr>
<td>911ACTION</td>
<td>(I/S Ignore or Stop at -911 SQL errors)</td>
</tr>
<tr>
<td>TASKS</td>
<td>(1-16 Multitasking level)</td>
</tr>
<tr>
<td>TSSAMPLEPCT</td>
<td>(Y/N/1-50 Random sampling tablespace statistics)</td>
</tr>
<tr>
<td>IXSAMPLEPCT</td>
<td>(Y/N/1-50 Random sampling for index statistics)</td>
</tr>
<tr>
<td>ATBWORKAREA</td>
<td>(Y/N Freq/card work areas above the 2G bar)</td>
</tr>
<tr>
<td>OPTIMIZECOMMIT</td>
<td>(Y/N Reduce DELETE, INSERT, UPDATE commits)</td>
</tr>
<tr>
<td>QUIESCEINTERVAL</td>
<td>(0/100000-1000000 Partitions processed before storage reorganization)</td>
</tr>
<tr>
<td>SORTNUM</td>
<td>(0-99 Number of SORTWKs for Histogram Stats)</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>(Device type for sort data sets for Histogram Stats)</td>
</tr>
</tbody>
</table>
**NOTE**

Because collecting column statistics can be expensive in terms of CPU processing time, carefully consider what value you plan to set for each of the following fields: TABLE, NUMCOLS, and COUNT.

- **TABLE** specifies whether to gather statistics on table columns. Options are:
  - **Y** – collect statistics on all table columns (see “BMCSTATS options for tables” on page 355)
  - **N** – do not collect statistics on table columns
  - **S** – collect statistics on specific tables and columns

  You can also select column groups for collection.

- **INDEX** indicates whether to run BMCSTATS on the indexes in the table space. If you specify **Index all Y** when you are collecting statistics from an image copy, DASD MANAGER PLUS uses the DB2 data sets to obtain index statistics. The default is **Y**.

- **GROUPSIZE** is the number of pages (per table space or index) in a group for statistics gathering. If you specify \( n \) pages, you receive a summary of statistics for every group of \( n \) pages. Type a value from 0 through 180000. The default value of zero means no grouping occurs, and the product summarizes statistics by partition.

- **SPACEONLY** specifies whether to gather only space statistics from the ICF catalog. If you need information only about size and extents, specify **Y** for the **SpaceOnly** option. The default is **N**.

- **NUMCOLS** is the number of key columns from left to right that is concatenated to evaluate the value (1 through 64) for each frequent value that the product collects. The default value of 1 collects frequent values for the first key column only.

BMCSTATS returns all combinations up to the NUMCOLS value. For example, if you specify NUMCOLS 5, BMCSTATS returns the following values: column 1; columns 1 and 2; columns 1, 2, and 3; columns 1, 2, 3, and 4; and columns 1, 2, 3, 4, and 5. For this information to be useful to the optimizer, you also must specify UPDATEDB2 **Y**.

**TIP**

If you specify **TABLE S**, review the values on the Select Tables panel the next time that you specify this action. You might need to change the values from an earlier execution.
- **IXNUMQUANTILES** specifies the number of quantiles to collect for index key column processing. Statistics for the specified set of columns are divided into groups based on the number of quantiles specified. Index columns must be in all ascending or all descending order for DASD MANAGER to collect this type of statistic. Otherwise, DASD MANAGER ignores this option.

**NOTE**

BMCSTATS invokes DSNUTILB to collect histogram data for key columns only if you specified UPDATEDB2 A or UPDATEDB2 P on the BMCSTATS parameters panel (see Figure 100). Specifying IXNUMQUANTILES invokes DSNUTILB.

- **NUMQCOLS** specifies the number of index columns on which to collect quantiles.

- **FREQVAL** indicates whether BMCSTATS should collect frequency statistics for columns when using the TABLE option. If you do not specify a value for this option, the value specified by the FREQVAL keyword in the installation options is used. The product is shipped with an installation default of **Y**.

- **COUNT** is the maximum number of frequent values to collect (for table columns, first key columns, concatenated table columns, and concatenated key columns). Type a value from 1 through 300. The default is 10.

- **FREQTYPE** specifies whether to collect the most (M) frequently occurring values in a column, the least (L) frequently occurring values in a column, or both (B) types of values. The default is **M**. The value that you specify for this field also applies to COLGROUP for column group statistics and to Keycard for concatenated key column statistics.

**NOTE**

Specifying a value of L or B in this field can be expensive in terms of CPU processing time and should be considered before choosing these values.

- **SAVESTATS** specifies whether to save the statistics in the DASD MANAGER PLUS database. The default is **Y**.

If you specify SAVESTATS N (to not save BMCSTATS statistics data) and you also specify DELETEAGE to delete statistics rows that are older than a number of days that you specify, DASD MANAGER PLUS does not delete the rows. The SAVESTATS command controls the statistics table handling routines. If you specify SAVESTATS N, no statistics table processing occurs.

- **DELETEAGE** specifies whether to automatically delete statistics for an object after a certain amount of time. Specify the number of days from 0 through 32767 to keep statistics.
The number that you specify is the minimum age of object statistics. For example, to delete all statistics entries on this object that are at least 30 days old, enter 30 for this parameter. The default is 32767, which specifies not to delete any statistics.

---

**NOTE**

This command has no affect if you specify **SAVESTATS N**.

---

- **REPORT** specifies whether to print a report into the job output of the statistics. The default is **Y**.

- **UPDATEDB2** specifies whether to update the DB2 catalog with the statistics that you gather. If the existing DB2 catalog values provide efficient optimizer choices, type **N**. The product ignores this field for volumes and storage groups. Valid values are **A** (All), **P** (Accesspath), **S** (Space), or **N** (None). The default is **N**.

- **HISTORY** specifies whether to update the DB2 History tables.

- **DELETEHISTAGE** specifies how long to keep the history table entries before deleting them.

- **OMITCARD0** specifies whether to bypass updating the DB2 catalog for objects in which BMCSTATS finds a zero cardinality. The default is **N**.

- **BADOBJECTRC** specifies the return code that BMCSTATS will set if it is unable to process a requested object due to object characteristics (unsupported objects), invalid status, object serialization, or object authorization failures. The default is to issue return code 4. Regardless of the setting of this option, BMCSTATS processing continues.

- **RECALL** specifies whether to recall archived data sets to collect statistics on them.

  The **RECALL** option opens the data sets that initiate a recall. **N** skips the object and returns a code 4, but continues with other objects. The default is **N**.

- **FORCEROLLUP** specifies whether to roll up the partition level statistics to the object level in cases in which not all partition statistics are available. The default is **N**.

- **MSGLEVEL** specifies the level of messaging. The default of 0 provides standard messaging. Specifying messaging level 1 provides some additional informative messages about progress and timestamps.
■ **911ACTION** specifies the action BMCSTATS will take if a -911 SQL error occurs during a DELETE, INSERT, or UPDATE operation.

— S specifies if an SQL -911 error occurs, stop all processing, issue error messages, and issue a return code 8 at termination.

— I specifies if an SQL -911 error occurs, stop processing the current object, issue warning messages, continue on with the next available object, and issue a return code 4 at termination. I is the default.

■ **TASKS** is the number of concurrent tasks for gathering statistics. Specify the number of multitasking levels that are used for processing partitioned objects by typing a value from 1 through 16. The default is 5.

■ **TSSAMPLEPCT** specifies whether to use random sampling for statistics.

Sampling greatly reduces resource consumption in producing statistics. If you need to use sampling, consider specifying a percentage of pages to sample.

Table 44 defines the sample table space parameter options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning and use</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Do not sample. Type N to process all pages, including very small tables on a multi-table table space that contains both large and small tables. N is the default.</td>
</tr>
<tr>
<td>Y</td>
<td>Sample 25 percent of the pages. If you type Y, BMCSTATS does not sample objects with fewer than 1000 pages. The numPages/numTables must also be greater than the minimum pages listed, or BMCSTATS will process all pages.</td>
</tr>
<tr>
<td>1-50</td>
<td>Sample the specified percentage of pages for statistics. The numPages/numTables must be greater than the minimum pages listed, or BMCSTATS will process all pages.</td>
</tr>
</tbody>
</table>

For more information about sampling, see “Sampling statistics” on page 358.

■ **IXSAMPLEPCT** specifies whether to use random sampling for statistics. Sampling greatly reduces resource consumption in producing statistics. If you need to use sampling, you should consider specifying a percentage of pages to sample.

While sampling indexes reduces resource consumption, restrictions affect when you can specify this option. In the following instances, DASD MANAGER PLUS suppresses index sampling and processes the entire index:

— specifying an index sampling option in addition to UPDATEDB2 Y

To eliminate this restriction, you can specify the UPDCATIXS=Y installation option during installation to allow the catalog to be updated with index sampled statistics.
— specifying an index sampling option and an option to collect histogram statistics for indexes

When collecting histogram statistics, BMCSTATS invokes DSNUTILB (which also does not support index sampling).

— index has less than 1000 pages

Table 45 defines sample index parameter options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning and use</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Do not sample. Type N to process all pages, including very small indexes. N is the default.</td>
</tr>
<tr>
<td>Y</td>
<td>Sample 25 percent of the pages. If you type Y, BMCSTATS does not sample objects with fewer than 1000 pages.</td>
</tr>
<tr>
<td>1-50</td>
<td>Sample the specified percentage of pages for statistics. BMCSTATS does not sample objects with fewer than 1000 pages.</td>
</tr>
</tbody>
</table>

For more information about sampling, see “Sampling statistics” on page 358.

- **ATBWORKAREA** specifies if the cardinality and frequency work areas will be placed above or below the 2G bar.

  — Y specifies work areas will be placed above the 2G bar.
  — N specifies work areas will be placed below the 2G bar.

  Above the 2G bar can reduce normal 32 bit addressed storage requirements. Below the bar reduces CPU consumption due to additional dynamic address translation overhead. N is the default.

- **OPTIMIZECOMMIT** specifies how SQL COMMIT(s) after SQL DELETE, INSERT, and UPDATE will occur.

  — Y specifies COMMIT strategy for DELETE, INSERT, and UPDATE will be optimized.
  — N specifies COMMIT(s) will occur after every DELETE, INSERT, and UPDATE.
Optimized COMMIT(s) reduce overall SQL overhead. Unoptimized COMMIT(s) reduce deadlocks and deadlock timeouts (SQL error -911). Y is the default.

- **QUIESCEINTERVAL** (10000-1000000) specifies the number of object partitions processed before releasing and reinitializing O/S storage pools.

  - 0 specifies no QUIESCEINTERVAL.
  - 0-1000000 specifies the number of partitions processed before storage pools will be reorganized.

QUIESCEINTERVAL can reduce the risk of storage depletion errors due to storage fragmentation. Storage fragmentation can occur after a large number of object partitions are processed. Use QUIESCEINTERVAL with large object sets and wildcards if storage depletion errors occur (5x78 ABENDs).

- **SORTNUM** specifies the number of temporary data sets that DFSORT can use for sorting and is used only with COLGROUP. Specify this field with the SORTDEV field. You can specify an integer value of 0 through 99.

- **SORTDEV** specifies the device type for dynamic allocation of the sort work files that DFSORT can use for sorting and is only used with COLGROUP. Specify this field with the SORTNUM field.

**BMCSTATS options for tables**

When you specify **TABLE Y** on the BMCSTATS parameters panel (Figure 100), the Select Tables panel is displayed (Figure 101).

**Figure 101  BMCSTATS Select Tables panel**

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Database Tablespace</th>
<th>Owner</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>QZUD10</td>
<td>QZU</td>
<td>QZUT01_D10S01</td>
</tr>
<tr>
<td>QZUD10</td>
<td>QZUS0110</td>
<td>QZU</td>
<td>QZUT01_D10S02</td>
</tr>
<tr>
<td>QZUD11</td>
<td>QZUS0111</td>
<td>QZU</td>
<td>QZUT01_D11S01</td>
</tr>
<tr>
<td>QZUD11</td>
<td>QZUS0211</td>
<td>QZU</td>
<td>QZUT01_D11S02</td>
</tr>
<tr>
<td>QZUD11</td>
<td>QZUS0311</td>
<td>QZU</td>
<td>QZUT01_D11S03</td>
</tr>
<tr>
<td>QZUD12</td>
<td>QZUS0112</td>
<td>QZU</td>
<td>QZUT01_D12S01</td>
</tr>
</tbody>
</table>
From the Select Tables panel, you can specify the tables that you want to include in the column statistics gathering, as follows:

- **To collect statistics for some but not all columns in a table,** type `S` in the Cmd field for the table. The Options for Table `tableName` panel is displayed (Figure 102).

- **To collect statistics for all columns,** type an asterisk (`*`) in the Cmd field for the table.

- **To unselect a table,** blank out any character in the Cmd field.

**Figure 102  BMCSTATS Options for Table `TableName` panel**

```
DEAE           Options for Table TABLE QZU.QZUT01_D10S01      Row 1 to 33 of 33
Command ===>                                                  Scroll ===> CSR
Service Syntax: BMCSTATS.DEMO
Enter data, then press end.                                         More:
------------------------------------------------------------------
-------------------------- What to Collect ------------------------------------
COLUMN . . . . . . * Y      (Y/S Y=ALL Columns, S=Select Columns)
-------------------------- Column ColGroup Options ----------------------------
To generate column lists or column groups, set Table all to "S"
To collect histogram statistics, use UpdateDB2 A or P
Select COLGROUP 1  * N      (N/Y Choose columns for distribution stats)
          COLGROUPFREQVAL (Y/N Collect frequent value statistics)
          COLGROUPFREQUENCY (M/L/B M=Most L=Least B=Both)
          COLGROUPCOUNT (1-300 Number of frequencies to collect)
          NUMQUANTILES . . (1-100 Number of quantiles to collect)
Select COLGROUP 2  * N      (N/Y Choose columns for distribution stats)
          COLGROUPFREQVAL (Y/N Collect frequent value statistics)
          COLGROUPFREQUENCY (M/L/B M=Most L=Least B=Both)
          COLGROUPCOUNT (1-300 Number of frequencies to collect)
          NUMQUANTILES . . (1-100 Number of quantiles to collect)
Select COLGROUP 3  * N      (N/Y Choose columns for distribution stats)
          COLGROUPFREQVAL (Y/N Collect frequent value statistics)
          COLGROUPFREQUENCY (M/L/B M=Most L=Least B=Both)
          COLGROUPCOUNT (1-300 Number of frequencies to collect)
          NUMQUANTILES . . (1-100 Number of quantiles to collect)
Select COLGROUP 4  * N      (N/Y Choose columns for distribution stats)
          COLGROUPFREQVAL (Y/N Collect frequent value statistics)
          COLGROUPFREQUENCY (M/L/B M=Most L=Least B=Both)
          COLGROUPCOUNT (1-300 Number of frequencies to collect)
          NUMQUANTILES . . (1-100 Number of quantiles to collect)
Select COLGROUP 5  * N      (N/Y Choose columns for distribution stats)
          COLGROUPFREQVAL (Y/N Collect frequent value statistics)
          COLGROUPFREQUENCY (M/L/B M=Most L=Least B=Both)
          COLGROUPCOUNT (1-300 Number of frequencies to collect)
          NUMQUANTILES . . (1-100 Number of quantiles to collect)
------------------------------------------------------------------
****************************************************************************** Bottom of data ******************************************************
```
The fields on the Options for Table tableName panel are as follows:

- **COLUMN** specifies whether to collect statistics on all columns or on specific columns.

- **Select COLGROUP** instructs BMCSTATS to calculate correlated column cardinality and frequency statistics for the specified set of columns. If you need more than five COLGROUP sets for a table, you can manually add them to the syntax.

---

**NOTE**

Note the following considerations:

- BMCSTATS invokes DSNUTILB to collect column histogram statistics only if you specified UPDATEDB2 A or UPDATEDB2 P on the BMCSTATS parameters panel. Otherwise, this option is ignored.

- Frequency values are collected as a standard part of regular column statistics and also for first key columns when index statistics are collected. For more information about changing the collection of frequency values, see the FREQVAL and FREQTYPE fields listed in Figure 100. Alternatively, you can specify COLGROUPCOUNT to specify the number of frequent values to collect for the preceding COLGROUP.

- **COLGROUPFREQVAL** indicates whether BMCSTATS should collect frequency statistics for a group of columns. If you do not specify a value for this option, the value specified by the FREQVAL on the BMCSTATS panel (Figure 100) is used for this COLGROUP.

- **COLGROUPFREQTYPE** specifies whether to collect the most (M) frequently occurring values, the least (L) frequently occurring values, or both (B) types of values for the group of columns. If no value is specified, the default is the value you specified for FREQTYPE on the BMCSTATS panel (Figure 100). If no value is specified in either location, the default is M. The value that you specify for this field applies to COLGROUP for column group statistics.

- **COLGROUPCOUNT** specifies the number of frequencies to collect for the preceding group of columns. If no value is specified, the default is the value you specified for COUNT on the BMCSTATS panel (Figure 100). If no value is specified in either location, the default is 10.

- **NUMQUANTILES** specifies the number of quantiles to collect and can be specified when collecting column group statistics. The values are divided into the number of quantiles you specify. If you specify 0, BMCSTATS does not collect histogram statistics.

If you type S in the Select Columns or Select COLGROUP field, the Select Columns panel appears (Figure 103).
Sampling statistics

Sampling large objects makes BMCSTATS run much faster than when gathering complete statistics. The smaller the sample, the faster BMCSTATS runs.

Minimum size requirements

If you specify TSSAMPLEPCT percent or IXSAMPLEPCT percent, BMCSTATS scans all space maps and control pages, and randomly samples the percentage of data pages unless the object has fewer than 1000 pages. If the object has fewer than 1000 pages for sampling, BMCSTATS turns sampling off and reads all data pages. If you need greater precision for index statistics, specify a percentage higher than the default, as in the following example:

<table>
<thead>
<tr>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSSAMPLEPCT</td>
<td>25</td>
</tr>
<tr>
<td>IXSAMPLEPCT</td>
<td>50</td>
</tr>
</tbody>
</table>
BMCSTATS does not sample objects that contain fewer than the specified number of 4-KB pages. DASD MANAGER PLUS enforces a minimum of 1000 pages regardless of page size.

For multi-table table spaces, the \textit{numPages/numTables} must also be greater than the minimum pages listed.

After scanning all space maps, BMCSTATS sampling performs the following tasks:

1. It averages the statistics values for the space map pages.
2. It multiplies the average by the total number of pages.

The two operations calculate statistics at the page level and row level, providing values such as CARD, FARIND, and NEARIND.

While sampling indexes reduces resource consumption, restrictions affect when you can specify this option. In the following instances, DASD MANAGER PLUS suppresses index sampling and processes the entire index:

- specifying an index sampling option in addition to \texttt{UPDATEDB2 Y}

To eliminate this restriction, you can specify the \texttt{UPDCATIXS=Y} installation option during installation to allow the catalog to be updated with index sampled statistics.

\textbf{WARNING}

Index sampling might produce statistics that cause SQL optimizer access selection problems. BMC recommends that you review the index sampled statistics for your indexes before deciding to use them for updating the catalog.

- specifying an index sampling option and an option to collect histogram statistics for indexes

When collecting histogram statistics, BMCSTATS invokes DSNUTILB (which also does not support index sampling).

- using an index that has fewer than 1000 pages

\section*{Specifying sampling percentages}

When you specify a percentage for a value in the panels, 25 percent is conservative and 10 to 15 percent might be adequate. To process all pages, remove the \texttt{TSSAMPLEPCT} keyword.
Restricting sampling by object type

You can further restrict the sample percentage by object type. When you specify the SAMPLE Y keyword, sample processing applies to whatever object type is currently being processed. When processing both table spaces and indexes, SAMPLE T or SAMPLE I can be used to restrict sampling by object type (T for TABLESPACE and I for INDEX) with 25 percent sampling. Entering a percent (from 1 through 50) on the panel for Sample Tablespaces or Sample Indexes on the BMCSTATS panel generates TSSAMPLEPCT and IXSAMPLEPCT, respectively.

Sampling table columns

Sampling is faster on large objects than gathering complete statistics. Sampling on table columns estimates values by using probability. For greater accuracy, do not use sampling.

Accessing the statistics display features

DASD MANAGER PLUS provides a variety of options for collecting and managing statistics on your DB2 objects. DB2 uses object statistics to determine the optimal path to the data. As the volume of data increases and approaches system limits, structural inefficiencies become apparent. For example, backup and recovery windows might shrink, and the performance of your applications and systems might degrade. Systems administrators need statistical information to plan changes that support application growth and activity.

NOTE

When DASD MANAGER PLUS 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 DASD MANAGER PLUS 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 access the statistics display features

On the DASD MANAGER PLUS main menu, select Statistics/Catalog Update/Space Estimation List and press Enter.

The DB2 Object Selection panel appears (Figure 104). You can specify the objects to include in a list on the DB2 Object Selection panel. The Display DB2 Object List is the panel from which you initiate most of the DASD MANAGER PLUS statistics display functions. You can run many of the statistics functions interactively or in batch.
You initiate many of the statistics functions from a list of DB2 objects. For this reason, the first statistics panel that DASD MANAGER PLUS displays is the DB2 Object Selection panel, where you specify the DB2 objects for statistics display. You can use a wildcard character to list the lower-level entries (such as table spaces) that are associated with a higher-level object (such as a database).

When you select the objects for which to display statistics, the DB2 Object List panel appears. You can use the DB2 Object List panel to complete any of the following tasks:

- browse DB2 catalog statistics
- update DB2 catalog statistics
- display BMC Software statistics (BMCSTATS), if available
- display an exceptions report for a BMCTRIG job
- estimate space requirements based on statistics
- display data set information
- zoom the full object name

Creating a DB2 object list

Use this procedure to create a list of one or more DB2 objects on which to perform a variety of statistical functions. From a DB2 object list, you can display BMC Software statistics, estimate space requirements, list the data sets used by a database, and list exceptions. When you select the statistics function from the main menu, DASD MANAGER PLUS guides you through creating a DB2 object list.
To create a DB2 object list

1. On the DASD MANAGER PLUS main menu, select **Statistics/Catalog Update/Space Estimation** and press **Enter**.

   The DB2 Object Selection panel appears (Figure 104).

2. To specify which DB2 objects for which to perform a statistics function, type the qualified object names or a wildcard pattern in the appropriate fields and press **Enter**.

   Except for storage groups and volumes, you can list all lower-level entries that are associated with a higher-level object by specifying the database name and typing an asterisk (*) or percent sign (%) in the lower-level object fields.

   The Display DB2 Object List panel appears (Figure 105). This panel lists the objects that match the information that you specified on the DB2 Object Selection panel. When you specify more than one type of object for the list, the display indents the object types according to level, with database (DB) at the highest level, table space (TS) next, and so on.

   **Figure 105  Display DB2 Object List panel**

```
DEBA ------------------ Display DB2 Object List ------------ Row 1 to 13 of 22
COMMAND ===>                                                  SCROLL ===> PAGE
Please select one or more options and press Enter.
D = Display BMC Stats    S = Space Est           I = Dataset Info by Database
E = Display Excepts      B = Browse DB2 Stats    U = Update DB2 Stats
Z = Zoom Full Object Name
Act Source  Status Object-Type  Object Name
-------------------------------------------------------------------------------
DB . . . QZUD44  .QZUS0144
TS . .   QZUD44  .QZUS0244
TB . .     QZU     .QZUT01_D44S01
IXC        QZU     .QZUX01_D44S01T01
IX         QZU     .QZUX02_D44S01T01
TS . .   QZUD44  .QZUS0244
TB . .     QZU     .QZUT01_D44S02
IXC        QZU     .QZUX01_D44S02T01
IX         QZU     .QZUX02_D44S02T01
```
For each object, you need to know what statistics are available and whether the object still exists. The Source column provides this information, as Table 46 shows.

**Table 46  Source column**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blank</td>
<td>The object exists in the DB2 catalog and has BMCSTATS statistics. For this object, BMCSTATS and DB2 catalog statistics are available.</td>
</tr>
<tr>
<td>B</td>
<td>The object does not exist in the DB2 catalog (because it has been dropped), but it has BMCSTATS statistics. For this object, only BMCSTATS are available.</td>
</tr>
<tr>
<td>C</td>
<td>The object exists in the DB2 catalog but does not have BMCSTATS statistics. (You have not run BMCSTATS against it.) For this object, only DB2 catalog statistics are available.</td>
</tr>
</tbody>
</table>

Table 47 shows possible values in the Status field.

**Table 47  Status field**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blank</td>
<td>You have not entered a command against this object.</td>
</tr>
<tr>
<td>*D</td>
<td>You have entered a command against this object. (See step 3.)</td>
</tr>
</tbody>
</table>

3  In the Act field, specify a statistics operation for an object. Table 48 shows valid commands.

**Table 48  DB2 Object list commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Display BMC Software statistics.</td>
</tr>
<tr>
<td>B</td>
<td>Display (browse) DB2 catalog statistics.</td>
</tr>
<tr>
<td>U</td>
<td>Update DB2 catalog statistics.</td>
</tr>
<tr>
<td>E</td>
<td>Display BMCTRIG exceptions.</td>
</tr>
<tr>
<td>S</td>
<td>Estimate space requirements.</td>
</tr>
<tr>
<td>I</td>
<td>Display a list of data sets for a database.</td>
</tr>
<tr>
<td>Z</td>
<td>Zoom the contents of a long object name.</td>
</tr>
</tbody>
</table>
Listing data sets used by a database

Use this procedure to list all of the data sets that a particular database uses.

To list data sets used by a database


2. On the DB2 Object Selection panel (Figure 104), specify the databases containing the data sets that you want to list.

   You can type the name of a single database or type a wildcard pattern.

   The Display DB2 Object List panel appears (Figure 104). This panel shows all objects from the DB2 subsystem that match the information that you specified on the DB2 Object Selection panel.

3. Create a list of data sets for a database by typing I in the Act field next to a database name and pressing Enter.

   **NOTE**
   If you select more than one database at a time, each subsequent database appears when you press END.

The DB2 Data Set List panel appears (Figure 106). This panel shows all data sets that the specified database uses.

---

**Figure 106 DB2 Data Set List panel**

<table>
<thead>
<tr>
<th>Database Name: QZUD44</th>
<th>Data Set Name</th>
<th>Trks</th>
<th>Ext</th>
<th>AU</th>
<th>Qty</th>
<th>Sqty</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0144.I0001.A001</td>
<td>100 10</td>
<td>10 0</td>
<td>OA0003</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0144.I0001.A002</td>
<td>30 1</td>
<td>2</td>
<td>1 OA0003</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0144.I0001.A003</td>
<td>30 1</td>
<td>30</td>
<td>10 OA0003</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0144.I0001.A004</td>
<td>60 3</td>
<td>2</td>
<td>1 OA0003</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0244.I0001.A001</td>
<td>130 11</td>
<td>T</td>
<td>30 10 OA0003</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0244.I0001.A002</td>
<td>30 1</td>
<td>C</td>
<td>2</td>
<td>1 OA0003</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0244.I0001.A003</td>
<td>30 1</td>
<td>T</td>
<td>30</td>
<td>10 OA0003</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0244.I0001.A004</td>
<td>60 3</td>
<td>C</td>
<td>2</td>
<td>1 OA0003</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0244.I0001.A005</td>
<td>100 10</td>
<td></td>
<td>OA0020</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0244.I0001.A006</td>
<td>30 1</td>
<td>T</td>
<td>30 10 OA0003</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0244.I0001.A007</td>
<td>30 1</td>
<td>C</td>
<td>2</td>
<td>1 OA0003</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEBACAT.DSDNB0.QZUD44.QZUS0244.I0001.A008</td>
<td>30 1</td>
<td>T</td>
<td>30</td>
<td>10 OA0003</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4 Review the information.

Updating DB2 catalog statistics

You can use DASD MANAGER PLUS to update the statistics fields in the DB2 catalog by using the ISPF dialog, the BMCUPRS utility, or the UPDATEDB2 option of the BMCSTATS utility (Figure 107).

Figure 107 Updating the DB2 catalog

While sampling indexes reduces resource consumption, restrictions affect when you can specify this option. In the following instances, DASD MANAGER PLUS suppresses index sampling and processes the entire index:

- specifying an index sampling option in addition to UPDATEDB2 Y

To eliminate this restriction, you can specify the UPDCATIXS=Y installation option during installation to allow the catalog to be updated with index sampled statistics.

**WARNING**

Index sampling might produce statistics that cause SQL optimizer access selection problems. BMC recommends that you review the index sampled statistics for your indexes before deciding to use them for updating the catalog.

- specifying an index sampling option and an option to collect histogram statistics for indexes

  When collecting histogram statistics, BMCSTATS invokes DSNUTILB (which also does not support index sampling).

- index has less than 1000 pages
Browsing and updating statistics

You can use DASD MANAGER PLUS to browse and update DB2 catalog statistics that influence the DB2 Optimizer. You can also compare DB2 catalog statistics with BMCSTATS statistics.

The browse function displays DB2 catalog and BMCSTATS or other statistics on the same panel. You can use this feature to compare sets of statistics for table spaces, indexes, and tables. Chapter 7, “Analyzing statistical trends,” describes the display function for viewing BMCSTATS statistics.

NOTE
You must update the DASD MANAGER PLUS historical database with the BMCSTATS utility at least once before you can compare DB2 catalog statistics with statistics that BMCSTATS produces.

DB2 and BMCSTATS statistics

Use this procedure to display DB2 catalog and BMCSTATS statistics for DB2 objects on the same panel. You can compare the statistics. You can also display BMCSTATS statistics for the first, last, and next-to-last time that you ran BMCSTATS. (See Chapter 7, “Analyzing statistical trends.”)

NOTE
You must run the BMCSTATS utility at least once to populate the DASD MANAGER PLUS database with statistics before you can view any BMC Software statistics.

To display DB2 catalog and BMCSTATS statistics for DB2 objects

1 Create a DB2 object list.

For DB2 catalog statistics, the following object types are valid:

- TB (table)
- TS (table space)
- IX (index)

For more information, see “Creating a DB2 object list” on page 361.

The Display DB2 Object List panel (Figure 104) shows the objects that match the information that you specified on the DB2 Object Selection panel.
2 Type B in the Act field beside the object for which you want to display statistics and press Enter.

Objects for which both BMCSTATS and DB2 catalog statistics are available have a blank in the Source column.

The Statistics Browse panel for the specified object appears.

3 Press END to exit the panels.

**DB2 table space statistics**

Figure 108 is a sample browse panel that displays the DB2 catalog and BMC Software statistics for the specified table space. After viewing the table space statistics, you can display statistics for a particular table.

The product obtains the DB2 catalog value from the SYSIBM.SYSTABLESPACE catalog table. The BMCSTATS value comes from the corresponding statistics table in the DASD MANAGER PLUS database and shows the value from the last run.

**Figure 108  Tablespace Statistics Browse panel**

You can perform the following tasks:

- In the Display Table Statistics? field, type S and press Enter to display updatable table statistics. Typing T on the COMMAND line produces the same result.

  You can also display these statistics by typing B in the Act field beside the table name on the Object Selection List.

- In the Display Partition Statistics? field, type S and press Enter to display updatable partition statistics. Typing P on the COMMAND line produces the same result.
DB2 table statistics

When you select Display Table Statistics on the Tablespace Statistics Browse panel and the table space has more than one table, the Table Selection panel (Figure 109) appears.

Figure 109  Table Selection panel

When you select Display Table Statistics on the Tablespace Statistics Browse panel and the table space has more than one table, the Table Selection panel (Figure 109) appears.

To select a table from the list, type S in the Act field beside the table name and press Enter. The DB2 catalog statistics for the table appear in the Table Statistics Browse panel (Figure 110).

Figure 110  Table Statistics Browse panel

The product obtains the DB2 catalog statistics from the SYSIBM.SYSTABLES catalog table. The BMCSTATS statistics come from the corresponding statistics table in the DASD MANAGER PLUS database (value from the last run).
You can perform the following tasks:

- In the Display Column Statistics? field, type S and press Enter to display a selection list of columns in the table. Typing C at the COMMAND line produces the same list.

- In the Display Index Statistics? field, type S and press Enter to display the index statistics for a table. Typing I at the COMMAND line produces the same panel.

- Press END to return to the previous panel.

### DB2 table space partition statistics

When you select Display Partition Statistics on the Tablespace Statistics Browse panel and more than one partition exists in the table space, the Tablespace Partition Selection panel appears (Figure 111).

**Figure 111  Tablespace Partition Selection panel**

<table>
<thead>
<tr>
<th>Act</th>
<th>Part Nactive</th>
<th>Card</th>
<th>Npages</th>
<th>Pages</th>
<th>Pct</th>
<th>Pct Rowcomp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>360</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>360</td>
<td>19</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>360</td>
<td>350</td>
<td>22</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>360</td>
<td>1667</td>
<td>105</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

To select a table from the list, type S in the Act field beside the table space partition name and press Enter. The Tablespace Partition Table Statistics Browse panel displays the DB2 catalog statistics for the partition (Figure 112).
The product obtains the DB2 catalog statistics from the SYSIBM.SYSTABSTATS catalog table. The BMCSTATS statistics come from the corresponding statistics table in the DASD MANAGER PLUS database (value from the last run).

To display a selection list of columns in the table

1. In the Display Column statistics? field, type S and press Enter.

   Typing C at the COMMAND line produces the same list.

2. Press END to return to the previous panel.
Figure 113 shows a sample Column Selection panel, which lists the columns in the specified table. This panel appears when you choose to view column statistics from the table statistics browse panels or a column selection list. You select the column for which to display statistics.

**Figure 113  Column Selection panel**

```
ASUCTCSE ------------------- Column Selection ---------------- Row 1 to 13 of 21
COMMAND ===>                                                  SCROLL ===> PAGE

Type S and press Enter to select object.

Table Name : QZU.QZUT01_DA1S01

<table>
<thead>
<tr>
<th>Act Column Name</th>
<th>Card</th>
<th>High2key</th>
<th>Low2key</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_1</td>
<td>2036</td>
<td>0.06</td>
<td>0.70</td>
</tr>
<tr>
<td>COLUMN_10</td>
<td>1952</td>
<td>0.21</td>
<td>0.11</td>
</tr>
<tr>
<td>COLUMN_11</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COLUMN_12</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COLUMN_13</td>
<td>1</td>
<td>....</td>
<td>....</td>
</tr>
<tr>
<td>COLUMN_14</td>
<td>2</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>COLUMN_15</td>
<td>2036</td>
<td>....</td>
<td>....</td>
</tr>
</tbody>
</table>
```
To display the DB2 catalog statistics for a column

1 Type S in the Act field beside the column name and press Enter.

The Column Statistics Browse panel appears (Figure 114). The product obtains the DB2 catalog values from the SYSIBM.SYSCOLUMNS catalog table. The BMCSTATS values come from the corresponding statistics table in the DASD MANAGER PLUS database and are the values from the last run.

Figure 114 Column Statistics Browse panel

DEBA ------------------------ Column Statistics Browse ------------------------
COMMAND ===>  
Column Name: CITYSTATE 
Table Name : ZZU.ZZUT12_D34S01

<table>
<thead>
<tr>
<th>DB2 Catalog</th>
<th>BMCSTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardinality :            91</td>
<td>91</td>
</tr>
<tr>
<td>High2key . . :   WEST ORA</td>
<td>WEST ORA</td>
</tr>
<tr>
<td>ECEE4DDC</td>
<td>ECEE4DDC</td>
</tr>
<tr>
<td>65230691</td>
<td>65230691</td>
</tr>
<tr>
<td>Low2key . . :   ALLENTOW</td>
<td>ALLENTOW</td>
</tr>
<tr>
<td>CDDCDEDE</td>
<td>CDDCDEDE</td>
</tr>
<tr>
<td>13355366</td>
<td>13355366</td>
</tr>
</tbody>
</table>

NOTE

Note the following additional considerations about the High2key and Low2key columns:

- The product obtains the values for High2key and Low2key from the SYSIBM.SYSCOLUMNS table and the corresponding table in the DASD MANAGER PLUS database, and displays the decimal and hexadecimal values. If the column has a noncharacter data type, you might not be able to print the data.

- The online display of the High2key and Low2key columns is truncated to 8 characters.

2 Press END to return to the previous panel.
DB2 index statistics

Figure 115 shows a sample Index Statistics Browse panel, which displays the DB2 catalog and BMC Software statistics for a specified index. This panel appears when you choose to view index statistics from the Table Statistics Browse panel.

**Figure 115  Index Statistics Browse panel**

<table>
<thead>
<tr>
<th>ASUCIXPB</th>
<th>Index Statistics Browse</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ====&gt;</td>
<td></td>
</tr>
<tr>
<td>Index Name: QZU.QZUX01_D09S01T01</td>
<td>DB Name : QCHD09</td>
</tr>
<tr>
<td>Table Name: QZU.QZUT01_D09S01</td>
<td></td>
</tr>
<tr>
<td>1stKeyCol : COLUMN_1</td>
<td></td>
</tr>
<tr>
<td>Compress : N</td>
<td></td>
</tr>
<tr>
<td>Hash . . . : N</td>
<td></td>
</tr>
<tr>
<td>IX Extension Type:</td>
<td></td>
</tr>
<tr>
<td>More: +</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DB2 Catalog</th>
<th>BMCSTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstKeyCard: -1</td>
<td>0</td>
</tr>
<tr>
<td>FullKeyCard: -1</td>
<td>0</td>
</tr>
<tr>
<td>Nleaf . . . : 4295M</td>
<td>0</td>
</tr>
<tr>
<td>Nlevels . . : -1</td>
<td>0</td>
</tr>
<tr>
<td>ClusterRatio: 0</td>
<td>0</td>
</tr>
<tr>
<td>DataRepeat : -1</td>
<td>-1</td>
</tr>
<tr>
<td>High2key . : 44444444</td>
<td>44444444</td>
</tr>
<tr>
<td>: 00000000</td>
<td>00000000</td>
</tr>
</tbody>
</table>

Select additional panels to display and press Enter.
Display Partition Statistics? Display Keytargets?

The product obtains the DB2 catalog values from the SYSIBM.SYSINDEXES catalog table. The BMCSTATS values come from the corresponding statistics table in the DASD MANAGER PLUS database.

**NOTE**

Note the following additional considerations about the **High2key** and **Low2key** columns:

- The product obtains the values for **High2key** and **Low2key** from the SYSIBM.SYSCOLUMNS table and the corresponding table in the DASD MANAGER PLUS database, and displays the decimal and hexadecimal values. If the column has a noncharacter data type, you might not be able to print the data.

- The online display of the **High2key** and **Low2key** columns is truncated to 8 characters.

**To display updatable partition statistics**

In the Display Partition Statistics? field, type S and press Enter. Typing P on the COMMAND line produces the same result.
When you select Display Partition Statistics on the Index Statistics Browse panel and the index space has more than one partition, the Index Partition Selection panel appears (Figure 116).

Figure 116  Index Partition Selection panel

<table>
<thead>
<tr>
<th>Index Name: QZU.QZUX03_DA1S03T01</th>
<th>DB Name : QZUDA1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name: QZU.QZUT01_DA1S03</td>
<td></td>
</tr>
<tr>
<td>1stKeyCol : COLUMN_1</td>
<td></td>
</tr>
</tbody>
</table>

Type S in Act field to select the partition.

<table>
<thead>
<tr>
<th>Act Part</th>
<th>Keycount</th>
<th>Cr</th>
<th>FirstKey</th>
<th>FullKey</th>
<th>NLeaf</th>
<th>Nlvls</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
<td>100</td>
<td>19</td>
<td>19</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>350</td>
<td>94</td>
<td>350</td>
<td>350</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1667</td>
<td>85</td>
<td>1667</td>
<td>1667</td>
<td>67</td>
<td>2</td>
</tr>
</tbody>
</table>

To select an index partition from the list

Type S in the Act field beside the index partition name and press Enter. The Index Partition Statistics Browse panel displays the DB2 catalog statistics for the partition (Figure 117).

Figure 117  Index Partition Statistics Browse panel

| Index Name: QZU.QZUX02_D09S01T01 | Partition : 1 |

<table>
<thead>
<tr>
<th>DB2 Catalog</th>
<th>BMCSTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstKeyCard</td>
<td>0</td>
</tr>
<tr>
<td>FullKeyCard</td>
<td>0</td>
</tr>
<tr>
<td>Nleaf . . .</td>
<td>0</td>
</tr>
<tr>
<td>Nlevels . . .</td>
<td>0</td>
</tr>
<tr>
<td>ClusterRatio</td>
<td>0</td>
</tr>
<tr>
<td>DataRepeat</td>
<td>0</td>
</tr>
<tr>
<td>Keycount</td>
<td>0</td>
</tr>
<tr>
<td>2009-08-28-09.19</td>
<td>2009-10-01-01.16</td>
</tr>
</tbody>
</table>

The product obtains the DB2 catalog statistics from the SYSIBM.SYINDEXSTATS catalog table. The BMCSTATS statistics come from the corresponding statistics table in the DASD MANAGER PLUS database (value from the last run).
Updating the DB2 catalog interactively

Use this procedure to display statistics and update the DB2 catalog statistics by selecting Statistics/Catalog Update/Space Estimation from the DASD MANAGER PLUS main menu. You can update the DB2 catalog with the most recent BMCSTATS results from the DASD MANAGER PLUS database or other values that you choose.

To update the DB2 catalog interactively

1 Create a DB2 object list as described in “Creating a DB2 object list” on page 361.

NOTE
To list all of the lower-level entries, such as table spaces, that are associated with a higher-level object, such as a database, type an asterisk in the lower-level object fields.

The Display DB2 Object List panel (Figure 104) shows the objects that match the information that you specified on the DB2 Object Selection panel.

2 Type U in the Act field beside the object to update and press Enter.

The statistics update panel for that object appears. If you are updating statistics for tables, columns, or indexes and more than one exists, another selection list appears. The statistics update panels show both the DB2 and the BMC Software statistics values for you to compare.

Table space statistics update

Figure 118 shows a sample Tablespace Statistics Update panel, which you can use to compare DB2 catalog and BMC Software statistics. From this panel, you can display table statistics for update. You can update the DB2 catalog with the most recent BMCSTATS results from the DASD MANAGER PLUS database or other values that you choose.

To update table space statistics

NOTE
When you make changes to any field on this panel and press Enter, DASD MANAGER PLUS validates the data, updates the catalog, and displays a message indicating whether the update completed successfully.
1 On the Tablespace Statistics Update panel, type new values for the appropriate fields and press Enter.

**Figure 118 Tablespace Statistics Update panel**

![Tablespace Statistics Update panel](image)

2 In the Display Table Statistics? field, type S and press Enter to display statistics for a table in this table space.

   If the table space contains more than one table, the Table Selection panel appears for you to choose a table. After you choose a table, the Table Statistics Update panel appears.

3 In the Display Partition Statistics? field, type S and press Enter to display statistics for a partition in this table space.

   From the Table Space Partition Selection panel, choose a partition. After you choose a partition, the Table Space Partition Table Statistics Update panel appears.

4 Press END to return to the previous panel.

**Table statistics update**

*Figure 119* shows a sample Table Statistics Update panel for a table that you specified on the Table Selection panel. You can update the DB2 catalog with the most recent BMCSTATS results from the DASD MANAGER PLUS database or other values that you choose.
To update table statistics

**NOTE**

When you make changes to any field on this panel and press **Enter**, DASD MANAGER PLUS validates the data, updates the catalog, and displays a message indicating whether the update completed successfully.

1. On the Table Statistics Update panel, type new values for the appropriate fields and press **Enter**.

**Figure 119  Table Statistics Update panel**

```
ASUCTBPA ------------------ Table Statistics Update ------------------
COMMAND ===> 
Type data and press Enter.
Table Name . . : QZU.QZUT01_DACS01
Tablespace Name: QZUDAC.QZUS01AC
Type . . . . : T

<table>
<thead>
<tr>
<th>New Value</th>
<th>DB2 Catalog</th>
<th>BMCSTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card . . 65000</td>
<td>65000</td>
<td>65000</td>
</tr>
<tr>
<td>Npages . 21669</td>
<td>21669</td>
<td>21669</td>
</tr>
<tr>
<td>PctPages . 93</td>
<td>93</td>
<td>93</td>
</tr>
<tr>
<td>PctRowComp . 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2010-11-15-10.47</td>
<td>2010-12-23-12.54</td>
<td></td>
</tr>
</tbody>
</table>

Select additional panels to display and press Enter.
Display Column statistics? Display Index statistics?
```

2. In the **Display Column Statistics?** field, type **S** and press **Enter** to display the Column Selection panel.

3. In the **Display Index Statistics?** field, type **S** and press **Enter**, to display the Index Selection panel.

4. Press END to return to the previous panel.
Table space partition table statistics update

Figure 120 shows a sample Table Space Partition Table Statistics Update panel for a table that you specified on the Table Space Selection panel. You can use this panel to display column and index statistics for update. You can update the DB2 catalog with the most recent BMCSTATS results from the DASD MANAGER PLUS database or other values that you choose.

To update table space partition table statistics

**NOTE**

When you make changes to any field on this panel and press Enter, DASD MANAGER PLUS validates the data, updates the catalog, and displays a message indicating whether the update completed successfully.

1. Type new values for the appropriate fields and press Enter.

   **Figure 120  Table Space Partition Table Statistics Update panel**

   DEBA --------- Tablespace Partition Table Statistics Update -------------------
   COMMAND ===>
   Type data and press Enter.
   Table Name . . : ZZU.ZZUT01_D11S01
   Tablespace Name: ZZUD11.ZZUS0111
   Partition . . : 4
   New Value | DB2 Catalog | BMCSTATS
   Card . . . . : 19 | 19 | 19
   Npages . . . : 2 | 2 | 2
   PctPages . . : 0 | 0 | 0
   Nactive . . : 360 | 360 | 360
   PctRowComp . : | 0 | 0 | 0
   Select additional panels to display and press Enter.
   Display Column statistics?

2. In the Display Column Statistics? field, type S and press Enter to display the Column Selection panel.

3. Press END to return to the previous panel.
Column statistics update

Figure 121 shows a sample Column Statistics Update panel for a column that you specified on the Column Selection panel. You can update the DB2 catalog with the most recent BMCSTATS results from the DASD MANAGER PLUS database or other values that you choose.

To update column statistics

NOTE

When you make changes to any field on this panel and press Enter, DASD MANAGER PLUS validates the data, updates the catalog, and displays a message indicating whether the update completed successfully.

1 Type a value in the New Value column beside the appropriate fields and press Enter.

Figure 121  Column Statistics Update panel
Index statistics update

NOTE
Note the following additional considerations about the High2key and Low2key columns:

- The product obtains the values for High2key and Low2key from the SYSIBM.SYSCOLUMNS table and the corresponding table in the DASD MANAGER PLUS database, and displays the decimal and hexadecimal values. If the column has a noncharacter data type, you might not be able to print the data.

- The online display of the High2key and Low2key columns is truncated to 8 characters.

- You cannot update values for the High2key and Low2key columns from this panel.

2 Press END to return to the previous panel.

Index statistics update

Figure 122 shows a sample Index Statistics Update panel, which displays the DB2 catalog and BMC Software statistics for the index that you specified. You can update the DB2 catalog with the most recent BMCSTATS results from the DASD MANAGER PLUS database or other values that you choose.

To update index statistics

NOTE
When you make changes to any field on this panel and press Enter, DASD MANAGER PLUS validates the data, updates the catalog, and displays a message indicating whether the update completed successfully.

1 Type a value in the New Value column beside the appropriate fields.
Figure 122 Index Statistics Update panel

| ASUCIXPA ---------------- Index Statistics Update ----------------------------- |
| Type data and press Enter.                                                      |
| Index Name: QZU.QZUX01_D09S01T01                                              |
| Table Name: QZU.QZUT01_D09S01                                                  |
| 1stKeyCol : COLUMN_1                                                            |
| Compress : N                                                                    |
| Hash ...: N                                                                    |
| IX Extension Type:                                                              |
| Sparse ... . . . . : N                                                          |

<table>
<thead>
<tr>
<th>New Value</th>
<th>DB2 Catalog</th>
<th>BMCSTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstKeyCard</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>FullKeyCard</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>Nleaf.</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>Nlevels.</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>ClusterRatio</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DataRepeat</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>High2key</td>
<td>44444444</td>
<td>44444444</td>
</tr>
</tbody>
</table>

Select additional panels to display and press Enter.
Display Partition Statistics?

**NOTE**

Note the following additional considerations about the **High2key** and **Low2key** columns:

- The product obtains the values for **High2key** and **Low2key** from the SYSIBM.SYSCOLUMNS table and the corresponding table in the DASD MANAGER PLUS database, and displays the decimal and hexadecimal values. If the column has a noncharacter data type, you might not be able to print the data.

- The online display of the **High2key** and **Low2key** columns is truncated to 8 characters.

- You cannot update values for the **High2key** and **Low2key** columns from this panel.

2 In the **Display Partition Statistics?** field, type **S** and press **Enter**, to display the Index Partition Selection panel.

3 Press **END** to return to the previous panel.
Index partition statistics update

Figure 123 shows a sample Index Partition Statistics Update panel, which displays the DB2 catalog and BMC Software statistics for the index partition that you specified. You access this panel from the Index Statistics Update panel (Figure 122). If the index space contains multiple index partitions, choose an index partition on the Index Partition Selection panel. You can update the DB2 catalog with the most recent BMCSTATS results from the DASD MANAGER PLUS database or other values that you choose.

To update index partition statistics

NOTE
When you make changes to any field on this panel and press Enter, DASD MANAGER PLUS validates the data, updates the catalog, and displays a message indicating whether the update completed successfully.

1 Type a value in the New Value column beside the appropriate fields.

Figure 123  Index Partition Statistics Update panel

<table>
<thead>
<tr>
<th>New Value</th>
<th>DB2 Catalog</th>
<th>BMCSTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstKeyCard . 10719</td>
<td>10719</td>
<td>10719</td>
</tr>
<tr>
<td>FullKeyCard . 10719</td>
<td>10719</td>
<td>10719</td>
</tr>
<tr>
<td>Nleaf . . . . 62</td>
<td>62</td>
<td>62</td>
</tr>
<tr>
<td>Nlevels. . . . 2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>ClusterRatio . 100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DataRepeat . 190</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>Keycount . . . 32396</td>
<td>32396</td>
<td>32396</td>
</tr>
<tr>
<td></td>
<td>2009-10-29-02.26</td>
<td>2009-10-28-01.25</td>
</tr>
</tbody>
</table>

2 Press END to return to the previous panel.
Managing DB2 and BMC Software statistics

You can use DASD MANAGER PLUS to update DB2 catalog statistics for table spaces, indexes, and tables. You can update these statistics interactively or by using a batch utility job. The interactive update function enables you to compare and update the DB2 catalog with the most recent BMCSTATS or other statistical values that you choose. For example, you can update the DB2 catalog statistics with values from the DASD MANAGER PLUS statistics tables by using the DASD MANAGER PLUS update panels or in batch by using the BMCSTATS and BMCUPRS utilities.

Using BMCSTATS, you can choose not to update the DB2 catalog and continue collecting statistics (without affecting the DB2 Optimizer path selection). Alternatively, you can update the DASD MANAGER PLUS database with values from the DB2 catalog by using BMCCPRS.

NOTE
If you set the UPDNUCAT option in the installation options module to N, the BMCSTATS utility updates only the DB2 catalog fields that allow SQL update. If you set this option to Y, the BMCSTATS utility updates all statistics columns.

You can also delete object statistics that are no longer useful. You can automatically delete BMCSTATS statistics when you collect new statistics or manually delete them by using the user interface.

Having authority to collect statistics

To run BMCSTATS, BMCUPRS, and BMCCPRS, you must have one of the types of authority that Table 49 shows. If using DB2 security exits for authorization, the programs invoke the exit to verify access.

<table>
<thead>
<tr>
<th>Authority</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSADM</td>
<td>SYSUSERAUTH.SYSADMAUTH = G or Y</td>
</tr>
<tr>
<td>SYSCTRL</td>
<td>SYSUSERAUTH.SYSCTRLAUTH = G or Y</td>
</tr>
<tr>
<td>SDBADM</td>
<td>SYSUSERAUTH.SYSADMAUTH = G or Y</td>
</tr>
<tr>
<td>SQLADM</td>
<td>SYSUSERAUTH.SYSSLADMAUTH = G or Y</td>
</tr>
<tr>
<td>STATS for the database</td>
<td>SYSDBAUTH.STATSAUTH = G or Y</td>
</tr>
<tr>
<td>DBADM</td>
<td>SYSDBAUTH.DBADMAUTH = G or Y</td>
</tr>
<tr>
<td>DBCTRL</td>
<td>SYSDBAUTH.DBCTRLAUTH = G or Y</td>
</tr>
<tr>
<td>DBMAINT</td>
<td>SYSDBAUTH.DBMAINTAUTH = G or Y</td>
</tr>
</tbody>
</table>
Be aware of the following items:

- When `OPNDB2ID = Y` in the installation options module, users with STATS authority can collect statistics even if their logon IDs do not have Resource Access Control Facility (RACF) authority to read the data set.

  BMCSTATS uses RACF to check secondary authorizations. If the user ID submitting the BMCSTATS job does not have DB2 authority to run BMCSTATS against the object, BMCSTATS uses the RACF authority to process the objects in the utility.

- Users with Execute authority for the DASD MANAGER PLUS statistics collection plan can view and update the DB2 catalog online.

- For more information about the plans, see the installation guide.

**Updating DB2 columns (by DASD MANAGER PLUS)**

Table 50 and Table 51 show the columns in the DB2 catalog that DASD MANAGER PLUS updates. When the UPDNUCAT installation option is set to `N`, the parameter indicates that you do not want to update all columns in the DB2 catalog, but only the columns for which DB2 allows updates using SQL. Also, BMCSTATS updates the same columns for ACCESSPATH and SPACE statistics as the IBM RUNSTATS utility. For more information, see the chapter about managing statistics in the catalog in the Administration Guide for DB2 for OS/390 and z/OS, volume 2.
Table 50  Table columns that are updated in the DB2 catalog  (part 1 of 3)

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIBM.SYSCOLDIST</td>
<td>CARDF, COLGROUPCOLNO, COLVALUE, FREQUENCY, FREQUENCYF, HIGHVALUE, IBMREQD, LOWVALUE, NAME, NUMCOLUMNNS, QUANTILENO, STATSTIME, TBOWNER, TBNAME, TYPE</td>
</tr>
<tr>
<td>SYSIBM.SYSCOLSTATS</td>
<td>COLCARD, HIGHKEY, HIGH2KEY, LOWKEY, LOW2KEY, STATS_FORMAT, STATSTIME</td>
</tr>
<tr>
<td>SYSIBM.SYSCOLUMNS</td>
<td>COLCARD, COLCARDF, HIGH2KEY, LOW2KEY, STATS_FORMAT, STATSTIME</td>
</tr>
<tr>
<td>SYSIBM.SYSINDEXES</td>
<td>AVGKEYLEN, CLUSTERED, CLUSTERRATIO, CLUSTERRATIOF, DATAREPEATFACTORF, FIRSTKEYCARD, FIRSTKEYCARDF, FULLKEYCARD, FULLKEYCARDF, NLEAF, NLEVELS, SPACE, SPACEF, STATSTIME</td>
</tr>
</tbody>
</table>
Table 50  Table columns that are updated in the DB2 catalog (part 2 of 3)

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIBM.SYSINDEXPART</td>
<td>AVGKEYLEN, CARDF, DSNUM, EXTENTS, FAROFFPOSF, LEAFDIST, LEAFFAR, LEAFNEAR, NEAROFFPOSF, PQTY, PSEUDO_DEL_ENTRIES, SECQTYI, SPACE, SPACEF, SQTY, STATSTIME</td>
</tr>
<tr>
<td>SYSIBM.SYSKEYTARGETS</td>
<td>CARDF, HIGH2KEY, LOW2KEY, STATS_FORMAT, STATSTIME</td>
</tr>
<tr>
<td>SYSIBM.SYSKEYTGTDIST</td>
<td>CARDF, FREQUENCYF, HIGHVALUE, IBMREQD, IXSCHEMA, IXNAME, KEYGROUPKEYNO, KEYSEQ, KEYVALUE, LOWVALUE, NUMKEYS, QUANTILENO, STATSTIME, TYPE</td>
</tr>
<tr>
<td>SYSIBM.SYSTABLES</td>
<td>AVGROWLEN, CARD, CARDF, NPAGES, NPAGESF, PCTPAGES, PCTROWCOMP, SPACEF, STATSTIME</td>
</tr>
</tbody>
</table>
Table 50  Table columns that are updated in the DB2 catalog (part 3 of 3)

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIBM.SYSTABLEPART</td>
<td>AVGROWLEN</td>
</tr>
<tr>
<td></td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>CARDF</td>
</tr>
<tr>
<td></td>
<td>DSNUM</td>
</tr>
<tr>
<td></td>
<td>EXTENTS</td>
</tr>
<tr>
<td></td>
<td>FARINDREF</td>
</tr>
<tr>
<td></td>
<td>NEARINDREF</td>
</tr>
<tr>
<td></td>
<td>PAGESAVE</td>
</tr>
<tr>
<td></td>
<td>PERACTIVE</td>
</tr>
<tr>
<td></td>
<td>PERCDROP</td>
</tr>
<tr>
<td></td>
<td>PQTY</td>
</tr>
<tr>
<td></td>
<td>SECQTY1</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
</tr>
<tr>
<td></td>
<td>SPACEF</td>
</tr>
<tr>
<td></td>
<td>SQTY</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td>SYSIBM.SYSTABLESPACE</td>
<td>AVGROWLEN</td>
</tr>
<tr>
<td></td>
<td>NACTIVE</td>
</tr>
<tr>
<td></td>
<td>NACTIVEF</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
</tr>
<tr>
<td></td>
<td>SPACEF</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
</tbody>
</table>
### Table 51  Partitioned Table Space/Index Space Columns Updated in the DB2 Catalog (part 1 of 2)

<table>
<thead>
<tr>
<th>Partitioned Table/Index Space</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIBM.SYSCOLDISTSTATS</td>
<td>CARDF</td>
</tr>
<tr>
<td></td>
<td>COLGROUPCOLNO</td>
</tr>
<tr>
<td></td>
<td>COLVALUE</td>
</tr>
<tr>
<td></td>
<td>FREQUENCY</td>
</tr>
<tr>
<td></td>
<td>FREQUENCYF</td>
</tr>
<tr>
<td></td>
<td>HIGHVALUE</td>
</tr>
<tr>
<td></td>
<td>IBMREQD</td>
</tr>
<tr>
<td></td>
<td>KEYCARD DATA</td>
</tr>
<tr>
<td></td>
<td>LOWVALUE</td>
</tr>
<tr>
<td></td>
<td>NAME</td>
</tr>
<tr>
<td></td>
<td>NUMCOLUMNS</td>
</tr>
<tr>
<td></td>
<td>PARTITION</td>
</tr>
<tr>
<td></td>
<td>QUANTILENO</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td></td>
<td>TBNAME</td>
</tr>
<tr>
<td></td>
<td>TOWNER</td>
</tr>
<tr>
<td></td>
<td>TYPE</td>
</tr>
<tr>
<td>SYSIBM.SYSINDEXSTATS</td>
<td>CLUSTER RATIO</td>
</tr>
<tr>
<td></td>
<td>CLUSTER RATIOF</td>
</tr>
<tr>
<td></td>
<td>FIRSTKEYCARDF</td>
</tr>
<tr>
<td></td>
<td>FULLKEYCARDF</td>
</tr>
<tr>
<td></td>
<td>KEYCOUNT</td>
</tr>
<tr>
<td></td>
<td>KEYCOUNTF</td>
</tr>
<tr>
<td></td>
<td>NLEAF</td>
</tr>
<tr>
<td></td>
<td>NLEVELS</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td>SYSIBM.SYSKEYTARGETSTATS</td>
<td>HIGHKEY</td>
</tr>
<tr>
<td></td>
<td>HIGH2KEY</td>
</tr>
<tr>
<td></td>
<td>IBMREQD</td>
</tr>
<tr>
<td></td>
<td>IXSCHEMA</td>
</tr>
<tr>
<td></td>
<td>IXNAME</td>
</tr>
<tr>
<td></td>
<td>KEYSEQ</td>
</tr>
<tr>
<td></td>
<td>LOWKEY</td>
</tr>
<tr>
<td></td>
<td>LOW2KEY</td>
</tr>
<tr>
<td></td>
<td>PARTITION</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td></td>
<td>STATS_FORMAT</td>
</tr>
</tbody>
</table>
When you run BMCSTATS and specify UPDATEDB2, BMCSTATS resets the real-time statistics and timestamp values (in both the DSNRTSDB and DB2 memory). The BMCUPRS utility resets the real-time statistics and timestamp values only when you specify RESETRTS Y.

Table 52 lists the columns in table SYSIBM.SYSTABLESPACESTATS that BMCSTATS and BMCUPRS reset for each partition and the reset value.

### Table 52 Statistics reset in table SYSIBM.SYSTABLESPACESTATS (part 1 of 2)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Reset value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPDATESTATSTIME</td>
<td>timestamp of the reset</td>
</tr>
<tr>
<td>STATSLASTTIME</td>
<td>timestamp of the reset</td>
</tr>
<tr>
<td>STATSINSERTS</td>
<td>zero</td>
</tr>
</tbody>
</table>
Table 52  Statistics reset in table SYSIBM.SYSTABLESPACESTATS (part 2 of 2)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Reset value</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATSDELETES</td>
<td>zero</td>
</tr>
<tr>
<td>STATSUPDATES</td>
<td>zero</td>
</tr>
<tr>
<td>STATSMASSDELETE</td>
<td>zero</td>
</tr>
</tbody>
</table>

Table 53 lists the columns in table SYSIBM.SYSINDEXSPACESTATS that BMCSTATS and BMCUPRS reset for each partition and the reset value.

Table 53  Statistics reset in table SYSIBM.SYSINDEXSPACESTATS

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Reset value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPDATESTATSTIME</td>
<td>Timestamp of the reset</td>
</tr>
<tr>
<td>LASTTIME</td>
<td>Timestamp of the reset</td>
</tr>
<tr>
<td>STATSINSERTS</td>
<td>zero</td>
</tr>
<tr>
<td>STATSDELETES</td>
<td>zero</td>
</tr>
<tr>
<td>STATSMASSDELETE</td>
<td>zero</td>
</tr>
</tbody>
</table>

Updating the DB2 catalog by using BMCSTATS

Use this procedure to create a utility job that updates the DB2 catalog with statistics from the DASD MANAGER PLUS database. With this procedure, you create a BMCSTATS utility job that collects statistics and updates the DB2 catalog at the same time. If you have recently run the BMCSTATS utility with the UPDATEDB2 option set to No, you can update the DB2 catalog statistics without collecting statistics again. (See “Updating the DB2 catalog by using BMCUPRS” on page 398.)

**NOTE**

If you set the UPDNUCAT option in the installation options module to N, BMCSTATS updates only those columns in the DB2 catalog that IBM identifies as updatable; otherwise, BMCSTATS updates all statistics columns. For information about updatable columns, see the DB2 catalog tables in the *IBM DB2 UDB for OS/390 SQL Reference*.

To update the DB2 catalog by using BMCSTATS

1  Create an action as described in “Creating an action” on page 271.

A user-defined action is a group of services (utilities) within a DASD MANAGER PLUS job. This action creates a worklist to run the BMCSTATS utility. For more information about actions, see Chapter 4, “Maintaining and generating actions.”

After you specify the action and press Enter, the Action List appears.
2 Add the BMCSTATS service to the action, as follows:

(Use the Edit Action Services panel to select and edit services and service syntax for the action. See “Generating actions” on page 299.)

A Type I in the Act field next to the services that should precede the BMCSTATS service and press Enter to add a blank line.

B In the blank line, type an E (or S) in the Act field.

C In the Object Name field, type the fully qualified object name.

You can use wildcards. For example, QZUD40.% specifies all table spaces in the QZUD40 database.

D In the Type field, type the two-letter abbreviation for the specified object type and press Enter.

Valid entries are TS (table space), IX (index), IS (index space) TT (table space set), SG (storage group), and VL (volume).

**NOTE**

Alternatively, specify the object name and object type by using an object set. See “Creating an object set for an action service” on page 336.

If you need to insert a service, decide where to add the step, type I in the Act field of the line before, and press Enter.

The Service List panel appears and lists all services from which you can select. From this panel you can also edit a service.

E Select the BMCSTATS utility as a service in your new action by typing S in the Act field beside the service name and pressing Enter.

F Press END to return to the Edit Action Services panel.

3 On the Edit Action Services panel, type E in the Act field beside the BMCSTATS service to edit the service syntax.

The Service Syntax List appears.

4 Edit service syntax by typing E in the Act field beside service syntax. Optionally, you can add new syntax by using I.
The BMCSTATS panel appears (Figure 124). Use this panel to customize how to collect statistics on specific objects. You must have specified the object name and object type on the Edit Action Services panel or when you created an object set.

**NOTE**

You can use more than one BMCSTATS service for additional objects, specifying different parameters for each step.
**Figure 124** BMCSTATS panel

<table>
<thead>
<tr>
<th>Command</th>
<th>BMCSTATS TABLESPACE QZUD40.%</th>
<th>Scroll</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Syntax:</td>
<td>BMCSTATS.DEMO</td>
<td>Enter data, then press end.</td>
<td>More: +</td>
</tr>
</tbody>
</table>

**What to Collect**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE</td>
<td><em>(Y/N/S)</em></td>
</tr>
<tr>
<td>INDEX</td>
<td>Y</td>
</tr>
<tr>
<td>GROUPSIZE</td>
<td>0</td>
</tr>
<tr>
<td>SPACEONLY</td>
<td>N</td>
</tr>
</tbody>
</table>

**Index Space Options**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMCOLS</td>
<td>1</td>
</tr>
<tr>
<td>IXNUMQUANTILES</td>
<td></td>
</tr>
<tr>
<td>NUMCOLS</td>
<td></td>
</tr>
</tbody>
</table>

**Table Space Options**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQVAL</td>
<td></td>
</tr>
</tbody>
</table>

**BMC Stats Reporting and Update Options**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVESTATS</td>
<td>Y</td>
</tr>
<tr>
<td>DELETEAGE</td>
<td>32767</td>
</tr>
<tr>
<td>REPORT</td>
<td>Y</td>
</tr>
</tbody>
</table>

**DB2 Catalog Update Options**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPDATEDB2</td>
<td>N</td>
</tr>
<tr>
<td>HISTORY</td>
<td>N</td>
</tr>
<tr>
<td>OMITCARDO</td>
<td>N</td>
</tr>
</tbody>
</table>

**Stats Processing Options**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BADOBJECTRC</td>
<td>4</td>
</tr>
<tr>
<td>RECALL</td>
<td>N</td>
</tr>
<tr>
<td>FORCEROLLUP</td>
<td>N</td>
</tr>
<tr>
<td>MSGLEVEL</td>
<td>0</td>
</tr>
<tr>
<td>911ACTION</td>
<td>I</td>
</tr>
</tbody>
</table>

**Stats Tuning Options**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASKS</td>
<td>5</td>
</tr>
<tr>
<td>TSSAMPLEPCT</td>
<td>N</td>
</tr>
<tr>
<td>IXSAMPLEPCT</td>
<td>N</td>
</tr>
<tr>
<td>ATBWORKAREA</td>
<td></td>
</tr>
<tr>
<td>OPTIMIZECOMMIT</td>
<td></td>
</tr>
<tr>
<td>QUIESCEINTERNAL</td>
<td></td>
</tr>
</tbody>
</table>

**Sort Options**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORTNUM</td>
<td></td>
</tr>
<tr>
<td>SORTDEVT</td>
<td></td>
</tr>
</tbody>
</table>
5 Set the BMCSTATS utility parameters on the main panel (or accept the defaults). For an explanation of each parameter, see “BMCSTATS options” on page 349.

6 When you return to the service syntax list, select the syntax by typing S in the Act field.

7 After you complete the final optional BMCSTATS panel, press END until you return to the Edit Action Services panel.

The Edit Action Services panel reappears and shows the BMCSTATS utility as a service (Figure 125).

Figure 125 Edit Action Services panel for BMCSTATS

8 (optional) Add other services to the action, as follows:

A Insert a blank service after BMCSTATS by typing I in the Act field and pressing Enter.

B Display the Service List by typing ? in the Act field of the new blank line.

C Select a service by typing S to the left of the service and pressing Enter. (For more information about editing action services, see “Creating an action by copying an existing action” on page 273.)

The Edit Action Services panel reappears.

D Press END to return to the Action List panel.

9 To generate JCL for your action, type G in the Act field and press Enter.

The Action Generation panel appears. The Action Generation panel controls generating, editing, and submitting JCL. The default values come from the user options file.
The **Status** field shows the current status of the Action. The DO_WORKIDS table provides the status. The field can be set to one of the following:

- **Not Generated** — Indicates the Action has not been generated
- **Generated Not Executed** — Indicates the Action has been generated but not submitted
- **Executed Successfully** — Indicates the Action has been generated, submitted, and ran successfully
- **Executed with Errors** — Indicates the Action has been generated, submitted, and completed unsuccessfully

10 Specify the action generation parameters and press **Enter**. If you need instructions for completing this panel, see “Generating actions” on page 299.

11 Depending on the options that you specified on the second Action Generation panel, choose one of the following actions:

- If you selected **Edit Worklist**, the worklist appears. Review the worklist. Advanced users might want to edit the worklist. Press END to continue.

  For detailed information about the worklist commands, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

- If you selected **Edit JCL**, the JCL appears. Review the JCL. Advanced users might want to edit the JCL. Press END to continue.

- If you selected **Submit JCL**, the Action Generation panel reappears. Asterisks (*) in the **Build Worklist** and **Build JCL** fields indicate that the product has generated them.

  **TIP**
  Alternately, you can enter **SUB** on the ISPF Edit panel.

12 Press **Enter** to confirm that you want to submit the BMCSTATS job. If you do not want to submit the job now, press **END** to return to the Edit Action Services panel.
Figure 126 shows the JCL for a sample BMCSTATS utility job.

**Figure 126  JCL for BMCSTATS utility job (part 1 of 2)**

```plaintext
//RDAJXN4U JOB (5213),'UTILITY-JENBMCS',
  // CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
  // NOTIFY=RDAJXN4
  /*ROUTE XEQ BMCPLEX1
  /*JOBPARM SYSAFF=DB2A
  //***************************************************
  //*  CREATED BY :  RDAJXN4
 //*  TIMESTAMP  :  11/17/2005.11.20.38
 //*  ENVIRONMENT:  ISPF 5.2MVS     TSO
 //*  RELEASE    :  V08.02.00 01/30/2006
 //*  DB2 VERSION:  815
  //***************************************************
  //*--------------------------------------------------------------------
  //*        DASD MANAGER WORKLIST EXECUTION
  //*--------------------------------------------------------------------
  //STEP1 EXEC PGM=AEXEMAIN,REGION=0M,
  //             PARM='DS815ECA'
  //STEPLIB DD DISP=SHR,DSN=DB2.MSTRPLAN.LOAD
  //        DD DISP=SHR,DSN=SYS3.DEBA.DSNEXIT
  //        DD DISP=SHR,DSN=SYS2.DB2V81M.DSNLOAD
  //********************************************************************
  //* LOADLIB CONCATENATION FOR RUNNING AT STAGE 2
  //*  AJXSTEPU -- SEE AEX.V71S2.SLIB(ASUSTEPU)
  //* ID:  ASU
  //********************************************************************
  // DD DISP=SHR,DSN=ASU.QATEST.LOAD
  // DD DISP=SHR,DSN=ASH.QATEST.LOAD
  // DD DISP=SHR,DSN=CSGI.SASC.V650.LOAD
  // DD DISP=SHR,DSN=AUS.DOPSEC.LOAD
  // DD DISP=SHR,DSN=AUS.EDITPR.LOADLIB
  // DD DISP=SHR,DSN=DB2.DPD.LOAD
  //********************************************************************
  //ABNLIGNR DD DUMMY
  //DSSPRINT DD SYSOUT=*  
  //SYSUDUMP DD SYSOUT=*  
  //SYSTERM DD SYSOUT=*  
  //UTPRINT DD SYSOUT=*  
  //SYSOUT DD SYSOUT=*  
```

396   DASD MANAGER PLUS for DB2 User Guide
Figure 126  JCL for BMCSTATS utility job (part 2 of 2)

Figure 127 provides an example of a BMCSTATS worklist.

Figure 127  BMCSTATS worklist

*************** Top of Data ***********************
* TIME 000000 '2004-06-24-17.41.50.00005'
* GENERIC BMCSTATS
* SSD 000001 DEBA
* WKID 000002 JENBMCS
* SYNC 000003

* BMCSTATS TABLESPACE JEND30.%
  TABLE (ALL)
  EVENTS N
  TASKS 2
  KEYCARD Y
  NUMCOLS 2
  COUNT 5

*************************** Bottom of Data ***************************
Updating the DB2 catalog by using BMCUPRS

Use this procedure to create a BMCUPRS action that updates the DB2 catalog with the BMC Software statistics from the DASD MANAGER PLUS database. If you have current BMCSTATS statistics, you might want to use BMCUPRS instead of BMCSTATS because BMCUPRS enables you to update the DB2 catalog during low activity times.

**NOTE**
DASD MANAGER PLUS does not collect statistics for the following objects:
- indexes that contain keys with random ordering
- objects with names that do not convert to EBCDIC

**To update the DB2 catalog by using BMCUPRS**

1. Create an action as described in “Creating an action” on page 271.

   A user-defined action is a group of services (utilities) within a DASD MANAGER PLUS job. This action creates a worklist to run the BMCUPRS utility. For more information about actions, see Chapter 4, “Maintaining and generating actions.”

   After you specify the action and press Enter, the Action List panel appears.

2. On the Action List panel, select the action by typing E (or S) in the Act field beside the action and pressing Enter.

3. On the Edit Action Services panel, add the BMCUPRS service to the action, as follows:

   A. Type I in the Act field next to the services that should precede the BMCUPRS service and press Enter to add a blank line.

   B. In the blank line, type the following information:

       — Type E (or S) in the Act field.
       — In the Service field, type BMCUPRS.
       — In the Object Name field, type the fully qualified object name.

       You can use wildcards. For example, QZUD40.% specifies all table spaces in the QZUD40 database.

       — In the Type field, type the two-letter abbreviation for the object type.
Valid entries are TS (table space), IX (index), IS (index space), and TT (table space set). Alternatively, specify the object name and object type by using an object set. See “Creating an object set for an action service” on page 336.

— Press Enter to display the Service Syntax List panel.

4 Type E in the Act field next to the syntax and press Enter to display the BMCUPRS panel (Figure 128). Optionally, you can add new syntax by using I.

**Figure 128  BMCUPRS parameters panel for a table space**

5 Specify the BMCUPRS utility parameters, as follows:

A In the TABLE field, type one of the following actions:

- Type Y to update column statistics on all tables in the tablespace.
- Type S to display a selection list of tables to update. The options for S are similar to the options for using BMCSTATS (page 349).
- Type N not to update column statistics on all tables in the tablespace. The default is N.

B In the INDEX field, indicate whether to update column statistics for all of the indexes in the table space and for the table space itself. The default is Y.
C In the UPDATEDB2 field, indicate whether to update the DB2 catalog with All, Accesspath, or Space statistics that you gather.

The default is A (for all). BMCSTATS updates the same columns for ACCESSPATH and SPACE statistics as IBM’s RUNSTATS utility. For more information, see the chapter about managing statistics in the catalog in the Administration Guide for DB2 for OS/390 and z/OS, volume 2.

D In the History field, indicate whether to update the DB2 tables. The default is N.

E In the Delete History Age field, indicate how long to keep the history table entries before deleting them. The default is 32767, which means that the entries will not be deleted.

F In the OMITCARD0 field, indicate whether to bypass updating the DB2 catalog for objects in which BMCSTATS finds a zero cardinality. The default is N, which means that the DB2 catalog will be updated.

G In the RESETRTS field, indicate whether to reset the statistics in the DB2 real-time statistics tables. The default is N.

H In the REPORT field, indicate whether to print a report into the job output of the statistics that you copied or updated. The default is Y.

I In BADOBJECTRC and MSGLEVEL fields, accept the defaults.

J When you finish editing the BMCUPRS parameters, press END to save your changes and press END again until you return to the Edit Action Services panel.

K Enter S on the Service Syntax List panel to select it.

6 Access the Action Generation panel by typing G beside an action on the Action List panel and pressing Enter.

The Action Generation panel appears.

7 Complete the fields on the Action Generation panel, as follows:

A In the Build Worklist, Build JCL, and Submit Job fields type S.

B Type values for any other options that you want, and press Enter.

See Figure 80 for information about the fields of the Action Generation panel.

C Press Enter.
8 Depending on the options that you specify on the Action Generation panel, complete the following tasks:

- If you selected **Edit Worklist**, the worklist appears. Review the worklist. Advanced users might want to edit the worklist. Press END to continue.
  
  For detailed information about the worklist commands, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

- If you selected **Edit JCL**, the JCL appears. Review the JCL. Advanced users might want to edit the JCL. Press END to continue.

- If you selected **Submit JCL**, the Action Generation panel reappears. Asterisks (*) in the **Build Worklist** and **Build JCL** fields indicate that you have generated them.

9 Press **Enter** to confirm that you want to submit the BMCCPRS job. If you do not want to submit the job now, press END to return to the Edit Action Services panel.

---

**TIP**

Alternately, when the JCL appears for you to review it, you can enter SUB on the ISPF Edit panel to submit the JCL.

---

### Updating DASD MANAGER PLUS statistics by using BMCCPRS

Use this procedure to update the DASD MANAGER PLUS database with statistics from the DB2 catalog.

---

**NOTE**

DASD MANAGER PLUS does not collect statistics for the following objects:

- indexes that contain keys with random ordering
- objects with names that do not convert to EBCDIC

---

### To update DASD MANAGER PLUS statistics by using BMCCPRS

1 Create an action as described in “Creating an action” on page 271.

A user-defined action is a group of services (utilities) within a DASD MANAGER PLUS job. This action creates a worklist to run the BMCCPRS utility. For more information about actions, see Chapter 4, “Maintaining and generating actions.”

After you specify the action and press **Enter**, the Action List appears.
2 On the Action List, select the action by typing E (or S) in the Act field beside the action and pressing Enter.

3 Add the BMCCPRS service to the action, as follows:

A Type I in the Act field next to the services that should precede the BMCCPRS service and press Enter to add a blank line.

B On the blank line, type the following information:

—— Type E (or S) in the Act field.
—— In the Service field, type BMCCPRS.
—— In the Object Name field, type the fully qualified object name.

You can use wildcards. For example, QZUD40.% specifies all table spaces in the QZUD40 database.

—— In the Type field, type the two-letter abbreviation for the object type.

Valid entries are TS (table space), IX (index), and TT (table space set). Alternatively, specify the object name and object type by using an object set. See “Creating an object set for an action service” on page 336.

—— Press Enter to display the Service Syntax List panel.

—— Type E in the Act field next to the syntax and press Enter to display the BMCCPRS panel (Figure 129). Optionally, you can add new syntax using I.

Figure 129 BMCCPRS Parameters panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>Syntax created</th>
<th>More:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASUDUM</td>
<td>CSR</td>
<td>Syntax created</td>
<td></td>
</tr>
<tr>
<td>BMCCPRS TABLESPACE QZUD40.%</td>
<td>CSR</td>
<td>Syntax created</td>
<td></td>
</tr>
<tr>
<td>Service Syntax: BMCCPRS.DEMO</td>
<td>CSR</td>
<td>Syntax created</td>
<td></td>
</tr>
<tr>
<td>Enter data, then press end.</td>
<td>CSR</td>
<td>Syntax created</td>
<td></td>
</tr>
</tbody>
</table>

More:

What to Migrate

Tablespace Options:
- TABLE . . . . . * N (Y/N/S Y=ALL Tables, N=No Tables, S=Select Tables)
- INDEX . . . . . Y (Y/N Copy column stats on all indexes)

BMC Statistics Report Options
- REPORT . . . . . Y (Y/N - Print statistics report)

Stats Processing Options
- BADOBJECTRC . . . 4 (0-8 Return code when object is bypassed)
- MSGLEVEL . . . . 0 (0/1 0-Normal msgs, 1-Additional msgs)
4 Specify the BMCCPRS utility parameters, as follows:

A In the **TABLE** field, type one of the following options:

- Type **Y** to update column statistics on all tables in the tablespace.
- Type **S** to display a selection list of tables to update. The options for S are similar to the options for using BMCSTATS (page 349).
- Type **N** not to update column statistics on all tables in the tablespace. The default is N.

B In the **INDEX** field, indicate whether to update column statistics for all of the indexes in the table space and for the table space itself. The default is **Y**.

C In the **REPORT** field, indicate whether to print a report into the job output of the statistics that you copied or updated. The default is **Y**.

D In **BADOBJECTRC** and **MSGLEVEL** fields, accept the defaults.

E When you finish editing the BMCCPRS parameters, press **END** to save your changes, and press **END** again until you return to the Edit Action Services panel.

F Enter **S** on the Service Syntax List panel to select it.

5 Access the Action Generation panel by typing **G** next to an action on the Action List panel and pressing **Enter**.

The Action Generation panel appears.

6 Complete the fields on the Action Generation panel and press **Enter**.

A In the **Build Worklist**, **Build JCL**, and **Submit JCL** fields, type **S**.

B Type values for any other options that you want, and press **Enter**.

Figure 80 describes the fields of the Action Generation panel.

7 Depending on the options that you specified on the Action Generation panel, complete the following tasks:

- If you selected **Edit Worklist**, the worklist appears. Review the worklist. Advanced users might want to edit the worklist. Press **END** to continue.

For a detailed description of the worklist commands and how to use them, see the *DASD MANAGER PLUS for DB2 Reference Manual*. 
Altering, renaming, or dropping and re-creating objects

- If you selected **Edit JCL**, the JCL appears. Review the JCL. Advanced users might want to edit the JCL. Press END to continue.

- If you selected **Submit JCL**, the Action Generation panel reappears. Asterisks (*) in the **Build Worklist** and **Build JCL** fields indicate that you have generated them.

Press **Enter** to confirm that you want to submit the BMCCPRS job. If you do not want to submit the job now, press END to return to the Edit Action Services panel.

**TIP**

Alternately, when the JCL appears for you to review it, you can enter **SUB** on the ISPF Edit panel to submit the JCL.

---

**Altering, renaming, or dropping and re-creating objects**

When you alter or drop and then re-create an object (for example, if you change the keys of an index, or drop partitions from a table space), BMCSTATS statistics remain in the database for these objects until you delete them. To ensure that your statistics accurately reflect the objects in your database, use one of the following methods to delete obsolete statistics:

- Run BMCSTATS on the altered or dropped object with DELETEAGE set to 0 to delete all statistics relating to the dropped object. For more information, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

- Delete individual statistics through the online interface by following the procedure “Deleting BMC Software statistics” on page 405.

- Manually alter and run the sample SQL that BMC provides in the *hlq.CNTL* (ASURSDEL) data set.

In DB2 Version 9, when you rename a column using RENAME COLUMN, the column names in the BMCSTATS statistics tables are not renamed automatically. To ensure that the column names are renamed, you can manually alter and run the sample SQL that BMC provides in the *hlq.CNTL* (ASURENAM) data set. Running this SQL updates the column name in the following tables:

- RS_COLUMNS
- RS_COLDIST
- RS_COLSTATS
- RS_COLDISTSTAT
- RS_INDEXES
- RS_LOBSTATS
For more information on managing DASD MANAGER PLUS repositories, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

**Deleting BMC Software statistics**

Use this procedure to delete individual, outdated BMC Software object statistics from the DASD MANAGER PLUS database.

---

**NOTE**

You can also delete groups of statistics according to their age by using the BMCSTATS DeleteAge option.

---

**To delete BMC Software statistics**

1. Create a list of DB2 objects that contains the object whose statistics you want to delete. If necessary, follow the instructions in “Creating a DB2 object list” on page 361.

   The Display DB2 Object List appears (Figure 104).

2. To display the BMC Software Statistics for the object, type D in the Act field beside the object name and press Enter. See Chapter 7, “Analyzing statistical trends.”

   The statistics panel for the specify object appears.

3. List the BMC Software statistics detail by typing S in the Display Entry List field at the bottom of the panel and pressing Enter.

   **NOTE**

   You can also list entries by typing L at the COMMAND line and pressing Enter.

   The statistics list panel for the object appears.

4. Type D in the Act field beside the statistic to delete and press Enter.

   The Delete objectType Statistics panel appears. Use this panel to confirm your selection.

5. On the Delete objectType Statistics panel, you can delete the specific entry by typing E in the Delete Oper field, or you can delete all entries older than the Delete Date you chose by typing L in the Delete Oper field.
Deleting BMC Software statistics

Eliminating all old entries means that you do not need to remove each entry separately from the BMCSTATS tables.

6 Press Enter to delete the statistics or END to return to the previous panel without deleting statistics.

**NOTE**

BMC Software provides sample SQL in the hlq.CNTL (ASURSDEL) data set that can help you delete the objects from the BMCSTATS tables that no longer exist in the catalog.
Analyzing statistical trends

This chapter contains the following topics:

Before you begin ................................................................. 408
Analyzing BMC Software statistics ............................................ 408
  BMC Software statistics for DB2 objects ............................... 409
  Table space statistics ...................................................... 409
  Table space partition statistics ......................................... 411
  Table statistics for a partitioned table space ......................... 414
  Table space partition pagegroup statistics ......................... 414
  Table statistics ......................................................... 416
  Column statistics ........................................................ 417
  Column value statistics ................................................. 419
  Index statistics .......................................................... 420
  Index partition statistics ............................................... 422
  Index partition pagegroup statistics ................................ 424
  Storage group statistics ................................................. 425
  Volume statistics ......................................................... 427
Graphing statistics ............................................................. 428
  Creating a statistics graph .............................................. 428
  Setting graphic display options ...................................... 431
Estimating space requirements by using statistics .................. 436
  Simulating changes to DB2 objects ................................... 436
  Estimating space requirements for a table space ................... 437
  Estimating space requirements for an index ......................... 444
Estimating space requirements based on user-specified values .... 447
  Estimating space requirements for a table space ................... 448
  Estimating space requirements for an index ......................... 451
Before you begin

Before you perform the steps in this section, you must have collected statistics from BMCSTATS, RUNSTATS, or some other source. Chapter 6, “Collecting and managing statistics,” describes the statistics and how to collect them.

**NOTE**

When DASD MANAGER PLUS 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 DASD MANAGER PLUS sorts the data retrieved from the DB2 Catalog queries, the product displays sorted data on panels and in reports in an EBCDIC collating sequence.

Analyzing BMC Software statistics

You can access DASD MANAGER PLUS statistics display features by using the Statistics/Catalog Update/Space Estimation option on the main menu. You can analyze statistics to determine the physical design of your DB2 structures and when to run maintenance utilities. Because the statistical profile of the objects can change, view the trends to determine when and what actions to take.

DASD MANAGER PLUS timestamps BMCSTATS statistics and stores them in a separate database, not in the DB2 catalog. DASD MANAGER PLUS provides two ways to view BMC Software statistics. You can display current and historic BMC Software statistics for the first, last, and previous time the BMCSTATS utility was run. You can also browse BMC Software statistics beside DB2 catalog statistics to make comparisons between the values. (See “Browsing and updating statistics” on page 366.) The additional statistics that BMCSTATS collects are useful in determining when to schedule maintenance utilities.

If your terminal supports IBM GDDM graphics, you can analyze trends in DB2 objects by displaying 18 graphs of statistical relationships for all of the data in the historical database. You can tailor graphs while viewing them to focus on specific trends. You can also generate hardcopy reports.

Because the statistics database is in DB2 tables, you can use a query facility to report on DB2 statistics and produce trend reports. DASD MANAGER PLUS provides many reports and sample QMF queries that you can run against the historical database and that let you produce trend analysis reports. See Chapter 10, “Producing reports.”
BMC Software statistics for DB2 objects

Use this procedure to display BMC Software statistics for DB2 objects. You can display statistics from the historical database for the first, last, and previous times that you ran BMCSTATS on an object. You can also display the percentage of change among these values since the last time you ran BMCSTATS. To display BMC Software statistics for comparison with DB2 statistics, see Chapter 6, “Collecting and managing statistics.”

**NOTE**

Before the product can display any BMC Software statistics, you must run the BMCSTATS utility at least once to populate the DASD MANAGER PLUS historical database with statistics.

**To display BMC Software statistics**

1. Create a DB2 object list as instructed in “Creating a DB2 object list” on page 361.

2. Type D in the Act field beside the object for which you want to display statistics and press Enter.

3. To select an object, type S in the Act field beside the object name and press Enter.

   The product displays BMC Software statistics for the specified object.

**Table space statistics**

Figure 130 shows a sample Tablespace Statistics panel, which is a high-level view of the statistics. The panel displays the last, previous, and first statistics from the DASD MANAGER PLUS statistics database for the specified table. This panel also displays statistics that BMCSTATS generates and statistics that BMCCPRS adds to the DASD MANAGER PLUS database.
To display table space statistics

You can perform the following tasks:

NOTE

If BMCSTATS encounters negative data points, it will not display them in the page statistics graphs. Negative data points occur when BMCSTATS gathers selected partition statistics on a table space in which the Force rollup field is set to N (meaning the statistics are not rolled up for the entire table space).

- In the Display Partition information? field, type S (or type P at the COMMAND line) and press Enter to display table space partition statistics. The Table Space Partition Selection panel appears for you to make a selection.

- In the Graph Pages? field, type S (or type GP at the COMMAND line) and press Enter to display the page statistics graph.

- In the Display Table information? field, type S (or type T at the COMMAND line) and press Enter to display the Table Statistics panel.

- In the Display Entry List? field, type S (or type L at the COMMAND line) and press Enter to list statistics for dates other than those shown. The Table Space Statistics List panel appears.
Tips for analysis

The following tips can help you analyze statistics:

- Compare the Space and Reorgspace values. If the Reorgspace value is a small fraction of the Space value, you might recover a large percentage of space by reorganizing the table space.

- Display the Pages Graph to see the relationship over time between active pages, allocated pages, and the pages required if you reorganized the table space. Reorganization restores PCTFREE and FREEPAGE, so the REORGED value might be more or less than the number of active pages. If the value is more than the allocated pages, a reorganization can cause one or more secondary extents to be allocated (unless you alter the space parameters first).

Table space partition statistics

Figure 131 shows a sample Tablespace Partition Statistics panel, which is a more detailed view of the table space statistics. The panel displays the last, previous, and first statistics from the DASD MANAGER PLUS statistics database for the selected table space partition.

---

**NOTE**

A zero in the Partition field indicates that the table space is not partitioned.
Figure 131  Tablespace Partition Statistics panel

To display table space partition statistics

You can perform any of the following tasks:

- In the **More Part Stats** field, type S to display additional table space partition statistics (or type T at the COMMAND line) and press Enter.

- In the **Column Stats** field, type S to display column statistics (or type C at the COMMAND line) and press Enter.

- In the **Graph Extents** field, type S to display a graph of the extent statistics (or type GE at the COMMAND line) and press Enter.

- In the **Graph Cardinality** field, type S to display a graph of the cardinality statistics (or type GC at the COMMAND line) and press Enter.

- In the **Graph Pages** field, type S to display a graph of the page statistics (or type GP at the COMMAND line) and press Enter.

- In the **Graph PctActive/Drop** field, type S to display a graph of the percentage of active and drop statistics (or type G% at the COMMAND line) and press Enter.

- In the **Graph PageGroup Stats** field, type S to display a graph of the page group statistics (or type GG at the COMMAND line) and press Enter.
■ In the List Entries field, type S to display Table Space Partition Statistics for dates other than those shown (or type L at the COMMAND line) and press Enter. The Table Space Partition Statistics panel appears.

■ In the PageGroup Stats field, type S to display Table Space Partition Pagegroup Statistics (or type G at the COMMAND line) and press Enter. The panel displays the statistics for the partition by page group for the last BMCSTATS execution on the object.

**Tips for analysis**

The following tips can be helpful for analyzing statistics:

■ A large Farindref value can indicate an increase in I/O to the data set, which means that an indirect reference was caused by an update to VARCHAR columns. Consider a reorganization to improve performance.

■ Reorgspace indicates the amount of space the table space should occupy after a reorganization. This value can be greater than or less than the current used space. The value might be greater if you have added a large number of rows to the table. A reorganization adds free space.

■ Any data sets with a large number of Extents might indicate a potential problem. Consider performing a reorganization in which you reallocate space to consolidate extents. Display the Extents Graph to analyze the number of extents over time.

■ To see the cardinality over time, display the Cardinality Graph. If NEARINDREF or FARINDREF increases in relation to the CARD, the table space is becoming more disorganized. Consider a reorganization. With the BMCTRIG utility, you can set a threshold for the percentage of pages that require indirect references and initiate a reorganization.

■ The number of Dirty Pages can help determine when to take an image copy and whether to take an incremental or full image copy. With BMCTRIG, you can set a threshold for the percentage of dirty pages and initiate a COPY. Dirty=0 initiates an exception on one or more dirty pages.

■ Display the Percent Active/Drop Graph to see the percentage of active and dropped pages in the partition over time. Immediately after a reorganization, PCT ACTIVE should be approximately equal to the amount that PCTFREE and FREEPAGE leave. PCT ACTIVE should approach 100 percent as the amount of free space for new or expanded rows decreases. The PCT DROPPED can be a nonzero value only in a nonsegmented table space. You must reorganize to reclaim this space.
Display the Pages Graph to see the relationship of page statistics over time: allocated pages, active pages, pages if reorganized, dirty pages, and full pages. If the difference between allocated pages and active pages is large, the data sets might be over-allocated. A high number of dirty pages relative to active pages indicates an image copy is necessary. If the number of full pages approaches the number of active pages, you need to add more space for new rows.

To analyze page group information (the distribution of data in the partition), display the Page Group Graph. This graph shows row distribution and the number of dirty and full pages in the data set.

Table statistics for a partitioned table space

Figure 132 displays the last, previous, and first statistics from the DASD MANAGER PLUS database for the specified table in a table space partition.

Figure 132 Additional Tablespace Partition Statistics panel

Table space partition pagegroup statistics

Figure 133 shows a sample Tablespace Partition Pagegroup Statistics panel. The panel lists all page group statistics in the DASD MANAGER PLUS database for the specified table space partition. To enter page group statistics in the historical database, you must run BMCSTATS with a nonzero value for the GROUPSIZE keyword. You access this panel through the Table Space Partition Statistics panel.
**NOTE**

A zero in the **Partition** field indicates that the table space is not partitioned.

---

**Figure 133  Tablespace Partition Pagegroup Statistics panel**

![Figure 133  Tablespace Partition Pagegroup Statistics panel](image)

**Tips for analysis**

The following tips can be helpful for analyzing statistics:

- The Cardinality (number of data rows) in each page group differs partly because the first page contains DB2 information as opposed to table data. For a larger object, scroll through the page groups to determine the distribution of data and locate data hot spots.

- The product collects Faroffpos and Nearoffpos statistics for indexes, but these statistics apply to the data. They indicate the degree of clustering, and the DB2 Optimizer uses them for clustering and nonclustering indexes. Use these statistics as thresholds only for clustering indexes; Pctcluster is preferred. Both statistics will approach zero for clustered data.
Table statistics

Figure 134 is a sample Table Statistics panel. The panel displays the last, previous, and first statistics from the DASD MANAGER PLUS statistics database for the specified table. You access this panel through the Table Space Statistics panel.

To display table statistics

You can perform any of the following tasks:

- In the Graph Cardinality? field, type S to display a graph of the cardinality statistics (or type GC at the COMMAND line) and press Enter.
- In the Graph Pages? field, type S to display a graph of the page statistics (or type GP at the COMMAND line) and press Enter.

**NOTE**

If BMCSTATS encounters negative data points, it will not display them in the cardinality, pages, average row length, or percent pages statistics graphs. Negative data points occur when BMCSTATS gathers selected partition statistics on a table space in which the Force rollup field is set to N (so the statistics are not rolled up for the entire table space).
In the Graph Row-Avg? field, type S to display a graph of average row length statistics (or type GR at the COMMAND line) press Enter.

In the Graph PctPages field, type S to display a graph of the percent pages (or type GT at the COMMAND line) and press Enter.

In the Display Column Information? field, type S (or type C at the COMMAND line) and press Enter. The Column Selection panel appears.

In the Display Index Information? field, type S (or type I at the COMMAND line) and press Enter to display the Index Statistics panel.

In the Display Entry List? field, type S (or type L at the COMMAND line) and press Enter to display a list of table statistics entries. The Table Statistics List panel appears.

**Tips for analysis**

The following tips can be helpful for analyzing statistics:

- If the table space contains multiple tables, you can determine the number of rows for this table by the value for Cardinality. Cardinality also shows the growth trend for the table over time.

- To analyze cardinality over time, display the Cardinality Graph. A higher percentage of indirect references (INDREFs) indicates the table is becoming more disorganized. If the value is more than 10 percent, consider reorganizing the table.

- From a DB2 performance perspective, review the column and index information by using the options at the bottom of the Table Statistics panel.

- To see how the percentage of pages changes over time, display the Percent Pages Graph.

- To see how the average row length changes over time, display the Average Row Length Graph.

**Column statistics**

Figure 135 shows a sample Column Statistics panel. The panel displays the last, previous, and first statistics from the DASD MANAGER PLUS statistics database for the specified column.
To access the Column Statistics panel

1 Display the Table Space Statistics panel. For more information, see “BMC Software statistics for DB2 objects” on page 409.

2 Display the Table Statistics panel.

3 In the Display Column Information field, type S (or type C at the COMMAND line) and press Enter.

You can also display column statistics for table space partitions.

Figure 135  Column Statistics panel

### DECA ----------------------- Column Statistics --------------------------------
COMMAND ===>

Table Name : QZU.QZUT01_DA1S01  Coltype: INTEGER  Length : 4
Column Name: COLUMN_1

<table>
<thead>
<tr>
<th>Last Run</th>
<th>Prev Run</th>
<th>%Change</th>
<th>1st Run</th>
<th>%Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardinality</td>
<td>2036</td>
<td>2036</td>
<td>0</td>
<td>2036</td>
</tr>
<tr>
<td>Avg Len</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Min Len</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Max Len</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>#Nulls</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>High2key</td>
<td>Ø.0p....</td>
<td>Ø.0p....</td>
<td>Ø.0p....</td>
<td></td>
</tr>
<tr>
<td>Low2key</td>
<td>Ø.7b....</td>
<td>Ø.7b....</td>
<td>Ø.7b....</td>
<td></td>
</tr>
</tbody>
</table>


Select additional panels to display and press Enter.
Display Most Frequent Values?  Display Entry List?

**NOTE**

Note the following additional considerations about the High2key and Low2key columns:

- The product obtains the values for High2key and Low2key from the SYSIBM.SYSCOLUMNS table and the corresponding table in the DASD MANAGER PLUS database, and displays the decimal and hexadecimal values. If the column has a noncharacter data type, you might not be able to print the data.

- The online display of the High2key and Low2key columns is truncated to 8 characters.
You can perform the following tasks:

- In the **Display Most Frequent Values** field, type S to display the Column Value Statistics panel which shows the ten most frequent column values (or type V at the COMMAND line) and press Enter. Figure 135 shows this panel.

- In the **Display Entry List?** field, type S to display the Column Statistics List panel (or type L at the COMMAND line) and press Enter.

**Tips for analysis**

The following tips can help you analyze statistics:

- The Column Selection and Column Statistics panels show column statistics that can help you when evaluating whether to define a column as VARCHAR or fixed length. If the average length of a column is 20 and the maximum length is 60, you might have large potential savings in DASD by defining the table as VARCHAR. However, if the average Length is 50 and the maximum length is 60, the DASD savings might not be justified when compared to the additional programming and performance costs that are associated with VARCHAR.

- If the number of nulls represents a large percentage of the column values, an index is not useful, because DB2 treats nulls as duplicate keys. However, it might be appropriate to create a UNIQUE WHERE NOT NULL index.

- Use the values for High2key and Low2key for evaluating predicates with constant values. DB2 uses these values when it assumes a uniform distribution of data in the column. If the data is not uniformly distributed and the predicate contains a constant value in the SYSIBM.SYSFIELDS table, DB2 ignores these values.

**Column value statistics**

*Figure 136* is a sample Column Value Statistics panel. BMCSTATS automatically collects the ten most frequent occurrences of the values in the column and displays a frequency list of the unique values for the current table. You access this panel through the **Display Most Frequent Values** field of the Column Statistics panel (*Figure 135*).
Index statistics

Figure 136  Column Value Statistics panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Table Name</th>
<th>Column Name</th>
<th>Coltype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECA</td>
<td>QZU.QZUT01_DAIS01</td>
<td>COLUMN_1</td>
<td>INTEGER</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05%</td>
<td>1818610</td>
</tr>
<tr>
<td>0.05%</td>
<td>1818609</td>
</tr>
<tr>
<td>0.05%</td>
<td>1818598</td>
</tr>
<tr>
<td>0.05%</td>
<td>1818597</td>
</tr>
<tr>
<td>0.05%</td>
<td>1446291</td>
</tr>
<tr>
<td>0.05%</td>
<td>1818606</td>
</tr>
<tr>
<td>0.05%</td>
<td>1818604</td>
</tr>
<tr>
<td>0.05%</td>
<td>1818603</td>
</tr>
<tr>
<td>0.05%</td>
<td>1818602</td>
</tr>
<tr>
<td>0.05%</td>
<td>1818605</td>
</tr>
</tbody>
</table>

Tips for analysis

The following tips can help you analyze statistics:

- If the column is a key index column and the data is not uniformly distributed, you might want to consider not using this column for a key column.

- You can establish a threshold in a BMCTRIG utility job to indicate when column values are not uniformly distributed. The threshold parameter is NonUniform. If an exception is initiated, you can review the data to determine whether to log an exception.

Index statistics

Figure 137 shows a sample Index Statistics panel. The panel displays the last, previous, and first statistics from the DASD MANAGER PLUS database for the specified index. You access this panel through the Table Statistics panel (Figure 134) or through the Object List panel (by typing D in the Act field beside the object name).

This panel displays statistics entries generated by BMCSTATS as well as any included in the DASD MANAGER PLUS database by BMCCPRS.
To display index statistics

You can perform any of the following tasks:

**NOTE**

If BMCSSTATS encounters negative data points, it will not display them in the cardinality statistics, cluster ratio, or pages graphs. Negative data points occur when BMCSSTATS gathers selected partition statistics on a table space in which the **Force rollup** field is set to N (so the statistics are not rolled up for the entire table space).

- In the **Display Partition info?** field, type S to display Index Partition Statistics (or type P at the COMMAND line) and press Enter. The Index Partition Selection panel appears.

- In the **Graph Cardinality?** field, type S to display a graph of the cardinality statistics (or type GC at the COMMAND line) and press Enter.

- In the **Graph Clusterratio?** field, type S to display a graph of the cluster ratio statistics (or type GR at the COMMAND line) and press Enter.

- In the **Graph Pages?** field, type S to display a graph of the page statistics (or type GP at the COMMAND line) and press Enter.
**Index partition statistics**

- In the **Display Entry List?** field, type **S** to display the Index Statistics List (or type **L** at the **COMMAND** line) and press **Enter**.

- In the **Display Keytargets?** field, type **S** to display the Keytargets List.

### Tips for analysis

The following tips can help you analyze statistics:

- **BMCSTATS** reports the number of index levels and the percent clustered value and calculates the number of levels for a reorganized index. For very large indexes, this indicates that you can improve performance of SQL statements that use the index if you reorganize the index. Guidelines are difficult, but you should review this statistic when performance degrades to determine an appropriate threshold for your system. Both **LevelMin** and **PctCluster** are thresholds that can be monitored using **BMCTRIG**.

- To see the relationship between **Cardinality** and **Faroffpos** and **Nearoffpos**, display the **Cardinality Graph**. If **Faroffpos** and **Nearoffpos** increase relatively to **Cardinality**, the table is becoming more disorganized. A reorganization should be considered.

- To see the relationship between the number of allocated pages and the number of active pages over time, display the **Pages Graph**. The **IF REORGED** line indicates how many pages will be active if the index is reorganized. The number shown can be more or less than the current number of active pages. If the value is more than the number of allocated pages, a reorganization will cause allocation of secondary extents. If the value approaches the active value, you should consider allocating more space for new rows.

- To analyze the clustering ratio of the index over time, display the **Clusterratio Graph**.

---

**Index partition statistics**

**Figure 138** is a sample Index Partition Statistics panel. The panel displays the last, previous, and first statistics found in the DASD MANAGER PLUS database for the specified index partition.

---

**NOTE**

A zero in the **Partition** field indicates the index is not partitioned.
To display index partition statistics

You can perform any of the following tasks:

- In the **Graph Cardinality?** field, type S to display a graph of the cardinality statistics (or type GC at the COMMAND line) and press Enter.

- In the **List PageGroup?** field, type S to display a list of pagegroup statistics and press Enter.

- In the **Graph Leafdist?** field, type S to display a graph showing the leaf distribution (or type LP at the COMMAND line) and press Enter.

- In the **Graph Extents?** field, type S to display a graph of the extent statistics (or type GE at the COMMAND line) and press Enter.

- In the **Graph PageGroup?** field, type S to display a graph of the page group statistics (or type GG at the COMMAND line) and press Enter.

- In the **Display Entry List?** field, type S to display Table Space Partition Statistics (or type L at the COMMAND line) and press Enter. The Table Space Partition Statistics panel appears for you to make a selection.

- In the **Graph Pages?** field, type S to display a graph of the page statistics (or type GP at the COMMAND line) and press Enter.
Tips for analysis

The following tips can help you analyze statistics:

- Any data sets with a large number of extents might indicate a potential problem that a reorganization can help resolve. Display the Extents Graph and look for a sharp increase in the number of extents over time.

- The product collects Faroffpos and Nearoffpos statistics for indexes although these statistics apply to the table space data. They indicate the degree of clustering. The DB2 Optimizer uses them for clustering and nonclustering indexes. Use these statistics as thresholds only for clustering indexes, but note that BMC recommends using Pctcluster instead. Both statistics approach zero for clustered data.

- To see the relationship between Cardinality and Faroffpos and Nearoffpos, view the Cardinality Graph. If Faroffpos and Nearoffpos increase relatively to Cardinality, the table is becoming more disorganized. Consider a reorganization.

- To see how the leaf pages are distributed over time, view the Leafdist Graph. Optimum Leafdist is 100, indicating all leaf pages are adjacent to each other. A value of 200 indicates an average of one page between leaf pages. A larger number indicates the need for an index reorganization.

- To see the relationship between the number of allocated pages and the number of active pages over time, view the Pages Graph. The IF REORGED line indicates how many pages will be active if you reorganize the space. The number can be more or less than the current number of active pages. If the value is more than the number of allocated pages, a reorganization can cause allocation of secondary extents. If the value approaches the active value, consider allocating more space for new rows.

Index partition pagegroup statistics

Figure 139 shows a sample Index Partition Pagegroup Statistics panel. The panel lists all page group statistics in the database for the specified index partition. To enter page group statistics in the historical database, you must run BMCSTATS with a nonzero value for the GROUPSIZE keyword. Access this panel through the Table Space Partition Statistics panel.

**NOTE**

A zero in the Partition field indicates the table space is not partitioned.
Figure 139  Index Partition Pagegroup Statistics panel

![Index Partition Pagegroup Statistics panel]

Storage group statistics

Figure 140 shows a sample Storage Group Statistics panel. The panel shows the last, previous, and first statistics in the DASD MANAGER PLUS database for the specified storage group. You access this panel through the DB2 Object Selection panel. See “Creating a DB2 object list” on page 361.

This panel displays statistics entries that BMCSTATS generates. If the database contains only one statistics entry for this object, values appear only in Last Run. If the database contains only two entries, values appear only in Last Run and Prev Run.

To display storage group statistics

You can perform the following tasks:

■ In the Display Entry List? field, type S to display the Storage Group Statistics List panel (or type L at the COMMAND line) and press Enter.

■ In the Delete entries? field, type S to display the Delete Storage Group Statistics panel (or type D at the COMMAND line) and press Enter.
Figure 140  Storage Group Statistics panel

<table>
<thead>
<tr>
<th>Last Run</th>
<th>Prev Run</th>
<th>%Change</th>
<th>1st Run</th>
<th>%Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracks</td>
<td>29126</td>
<td>0</td>
<td>15750</td>
<td>84</td>
</tr>
<tr>
<td>TS Space</td>
<td>13742</td>
<td>0</td>
<td>6908</td>
<td>98</td>
</tr>
<tr>
<td>TS Count</td>
<td>177</td>
<td>0</td>
<td>136</td>
<td>30</td>
</tr>
<tr>
<td>IX Space</td>
<td>15384</td>
<td>0</td>
<td>8842</td>
<td>73</td>
</tr>
<tr>
<td>IX Count</td>
<td>369</td>
<td>0</td>
<td>242</td>
<td>52</td>
</tr>
</tbody>
</table>

Select additional panels to display and press Enter.
Display Entry List?  Delete entries?

Tips for analysis

The Storage Group Statistics panel displays statistics at a high level and indicates the amount of space within a STOGROUP that DB2 objects occupy. If your installation uses a charge-back method for allocating DASD expenses, you can assign a cost for the DASD space by accumulating the total tracks of DASD containing DB2 data. Additionally, if you segregate DB2 applications into separate storage groups, you can accurately assign DASD expenses to those application groups, which helps if your installation defines table spaces and indexes in storage groups. For installations that have user-defined VSAM data sets, you can obtain similar information from the Volume Statistics display.
Volume statistics

Figure 141 shows a sample Volume Statistics panel. The panel displays the last, previous, and first statistics in the DASD MANAGER PLUS database for the specified volume.

Figure 141  Volume Statistics panel

<p>| DEAE-------------------------- Volume Statistics ------------------------------ |
| COMMAND ==&gt;                  |
| Volume: AUS001              |
|                               |</p>
<table>
<thead>
<tr>
<th>Last Run</th>
<th>Prev Run</th>
<th>1st Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Data Sets : 1959</td>
<td>1617</td>
<td>1121</td>
</tr>
<tr>
<td>DB2 Tracks . . : 41830</td>
<td>36254</td>
<td>33906</td>
</tr>
<tr>
<td>% DB2 Data Sets : 83</td>
<td>72</td>
<td>67</td>
</tr>
<tr>
<td>Total Data Sets : 1969</td>
<td>1627</td>
<td>1131</td>
</tr>
<tr>
<td>% Volume Used : 99</td>
<td>88</td>
<td>83</td>
</tr>
<tr>
<td>DSCBS . . . . : 4499</td>
<td>4499</td>
<td>4499</td>
</tr>
<tr>
<td>% DSCBS Used : 47</td>
<td>40</td>
<td>26</td>
</tr>
<tr>
<td>Free DSCBS . . : 2342</td>
<td>2690</td>
<td>3319</td>
</tr>
<tr>
<td>Free Cyls . . : 0</td>
<td>358</td>
<td>518</td>
</tr>
<tr>
<td>Free Trks . . : 104</td>
<td>5680</td>
<td>8028</td>
</tr>
<tr>
<td>Free Extents : 38</td>
<td>56</td>
<td>61</td>
</tr>
</tbody>
</table>

Select additional panels to display and press Enter.
Display Entry List?  Delete entries?

To display volume statistics

You can perform the following tasks:

- In the Display Entry List? field, type S to display the Volume Statistics List panel (or type L at the COMMAND line) and press Enter.

- In the Delete entries? field, type S to display the Delete Volume Statistics panel (or type D at the COMMAND line) and press Enter.

Tips for analysis

The Volume Statistics provides information similar to the Storage Group Statistics panel. However, the percentage of a DASD volume that DB2 objects occupy can provide an alternative to installations that use a charge-back method to allocate DASD expenses.
Graphing statistics

If you have a graphics-capable terminal or printer, DASD MANAGER PLUS displays or prints a variety of statistical graphs by using IBM GDDM Presentation Graphics Facility (PGF). Use these graphs to compare and analyze DB2 catalog and BMCSTATS statistics.

When you use the graphs with the statistical displays, the displays show statistical trends that allow for planning of maintenance schedules and, if necessary, redesign of the physical objects. For this reason, you can access the graph displays from the statistics panels (see Figure 130).

Many statistics graphs depend on the number of statistical entries in the BMCSTATS tables to show a change in the statistics over time. BMC Software recommends that you collect the statistics several times to obtain graphs that display noticeable trends.

Creating a statistics graph

Use this procedure to display or print a statistics graph. The options specify whether to print or display the graph. (For more information, see “Setting graphic display options” on page 431.) You must have a graphics-capable terminal to display statistics graphs.

**Before you begin**

Table 54 describes the graphs that you can display or print.

<table>
<thead>
<tr>
<th>Graph</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Partition Cardinality</td>
<td>graphs the number of rows of the index partition in relation to the number of NEAROFFPOS and FAROFFPOS references over time (months)</td>
</tr>
<tr>
<td></td>
<td>If NEAROFFPOS and FAROFFPOS increase in relation to the cardinality, the table becomes more disorganized, and you should consider a reorganization.</td>
</tr>
<tr>
<td>Index Partition Extents</td>
<td>graphs the number of extents in the partition over time (months)</td>
</tr>
<tr>
<td></td>
<td>The maximum number that VSAM allows is 7257 per data set.</td>
</tr>
</tbody>
</table>
### Table 54  Statistics graphs summary  (part 2 of 3)

<table>
<thead>
<tr>
<th>Graph</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Partition PageGroup</td>
<td>graphs the distribution of data in the partition</td>
</tr>
<tr>
<td></td>
<td>This dual axis graph shows the number of pages in the data set in relation to the cardinality (rows and keys) and the number of leaf pages and full pages in the page group. The size of the page group when you last ran BMCSTATS with a page group value greater than zero determines the maximum page value on the right axis.</td>
</tr>
<tr>
<td>Index Partition Leafdist</td>
<td>graphs the leaf distribution in the partition over time (months)</td>
</tr>
<tr>
<td></td>
<td>Optimum LEAFDIST is 100, indicating all leaf pages are adjacent to each other. A value of 200 would indicate an average of one page between leaf pages. A larger number might indicate the need for an index reorganization.</td>
</tr>
<tr>
<td>Index Partition Pages</td>
<td>graphs the relationship of the number of allocated pages and the number of active pages over time (months)</td>
</tr>
<tr>
<td></td>
<td>A large difference between allocated pages and active pages indicates the data sets might be over-allocated. If the FULL value approaches the ACTIVE value, consider allocating more space for new rows. The IF REORGED value might be more or less than the current number of ACTIVE pages. If the IF REORGED value is more than the ALLOCATED value, a reorganization will cause the allocation of a secondary extent.</td>
</tr>
<tr>
<td>Index Cardinality</td>
<td>graphs the FIRSTKEY and FULLKEY cardinality over time (months)</td>
</tr>
<tr>
<td>Index Pages</td>
<td>graphs over time (months) the active pages, allocated pages, and the pages that are required if you run the REORG utility</td>
</tr>
<tr>
<td></td>
<td>If the difference between ALLOCATED and ACTIVE is large, the data sets might be over-allocated. The IF REORGED value might be more or less than the current number of ACTIVE pages. If the IF REORGED value is more than the ALLOCATED value, a reorganization will cause the allocation of a secondary extent.</td>
</tr>
<tr>
<td>Index Cluster Ratio</td>
<td>graphs the CLUSTERRATIO of the index over time (months)</td>
</tr>
<tr>
<td>Table Cardinality</td>
<td>graphs the cardinality (number of rows) and the number of indirect references to these rows over time</td>
</tr>
<tr>
<td></td>
<td>An increasingly high percentage of indirect references indicates the table is becoming more disorganized. If the value is greater than 10%, consider reorganizing the table space.</td>
</tr>
<tr>
<td>Table Pages</td>
<td>graphs the number of pages in the table over time (months)</td>
</tr>
<tr>
<td>Table Average Row Length</td>
<td>graphs the average row length for the table over time (months)</td>
</tr>
<tr>
<td>Table Percent Pages</td>
<td>graphs the percent pages for the table over time (months)</td>
</tr>
</tbody>
</table>
### Table 54  Statistics graphs summary  (part 3 of 3)

<table>
<thead>
<tr>
<th>Graph</th>
<th>Description</th>
</tr>
</thead>
</table>
| Table Space Partition Cardinality | graphs the cardinality (number of rows) of the table space partition over time  
This graph also plots the number of near and far indirect references for comparison with the cardinality. If NEARINDREF or FARINDREF increases in relation to CARD, the table is becoming more disorganized and you should consider reorganizing the partition. |
| Table Space Partition Extents  | graphs the number of extents in the partition over time (months)  
The maximum extents that VSAM allows is 7257 per data set.                                                                                                                                                                                                                     |
| Table Space Partition PageGroup | graphs the distribution of data in the partition  
This dual axis graph shows the number of pages in the data set in relation to the cardinality (rows and keys) and the number of leaf pages and full pages in the page group. The size of the page group when you last ran BMCSTATS with a page group value greater than zero determines the maximum page value on the right axis. |
| Table Space Partition Pages    | graphs over time the page statistics: allocated pages, active pages, pages that are required if reorganized, dirty pages, and full pages  
If the difference between ALLOCATED and ACTIVE is large, the data sets might be over-allocated. The IF REORGED value might be more or less than the current number of ACTIVE pages. If the IF REORGED value is more than the ALLOCATED value, a reorganization will cause the allocation of a secondary extent. A high DIRTY value in relation to ACTIVE indicates an image copy is necessary. If FULL is approaching ACTIVE, you need to allocate more space for new rows. |
| Table Space Partition Percent Active/Drop | graphs the percentage of active and dropped pages in the partition over time (months)  
Immediately after reorganization, PCT ACTIVE is approximately equal to the amount that PCTFREE and FREEPAGE leave. PCT ACTIVE should approach 100% as the amount of free space for new or expanded rows decreases. PCT DROPPED can be nonzero only in nonsegmented table spaces. You must perform a reorganization to reclaim this space. |
| Table Space Pages              | graphs over time the active pages, allocated pages, and pages that are required if reorganized  
If the difference between ALLOCATED and ACTIVE is large, the data sets might be over-allocated. The IF REORGED value might be more or less than the current number of ACTIVE pages. If the IF REORGED value is more than the ALLOCATED value, a reorganization will cause the allocation of a secondary extent. |
To create a statistics graph

1. Create a list of DB2 objects with the objects for which you want to graph statistics.

   For more information, see “Creating a DB2 object list” on page 361.

   The DB2 Objection Selection List appears. From this panel, you select the object
   statistics to graph.

2. Type D in the Act field beside the object for which you want to display statistics and
   press Enter.

   The statistics panel appears. The kinds of graphs that you can create depend on the
   object type. The bottom of the statistics panels shows the graph display options.

3. Tab to the graph type that you want to display, type S beside it, and press Enter.

   The graph appears. Figure 142 shows a sample Index Information graph. Table 54
   describes all of the graphs and offers tips for analyzing the data.

4. Press END to return to the statistics panel.

Setting graphic display options

Use this procedure to set the printing options that the graphics facility uses to print
statistics graphs. These options also determine whether to print the graphics or
display them on a graphics terminal.

Before you begin

You can specify graphic display and printing options to obtain the most desirable
display or print. Figure 142 shows the components of a statistics graph for which you
can set options.
The numbers on the statistics graph represent the following elements:

1. a header showing the graph title
2. markers showing the data points
3. lines connecting the markers
4. X-axis title of the horizontal axis
5. Y-axis title of the vertical axis or axes
6. labels for horizontal (X) and vertical (Y) axes
7. the legend that describes the plotted data (markers and lines)

**To set graphic display options**

1  Access the Graphics Options panel as follows:

   **A** On the DASD MANAGER PLUS main menu, select User Options panel and press Enter.

   **B** On the User Options panel, select Graphic Display options and press Enter.

   The first DASD MANAGER PLUS Graphics Options panel appears (Figure 143).
Figure 143  Graphics Options panel

2 Specify the Header Attributes to determine how to display graph headers as follows:

A Choose a color from the Color Key at the bottom right side of the panel and type its number in the Color field. Choose a value from 0 to 8.

B Specify the character size multiplier in the Char Size field. The product divides this multiplier by 100, and multiplies the spacing sizes by this factor.

C In the Char H/W field, specify the character height-to-width multiplier to indicate the height of the character relative to its width. For example, a value of 200 multiplies the height by twice the width.

3 Specify the Legend Key Attributes to determine how to display the legends as follows:

A Specify the Color Key, Char Size, and Char H/W the same as you did in step 2.

B In the Orient field, specify the orientation of the legend, as follows:
   - H for horizontal
   - V for vertical
C In the **Margin** field, specify the placement of the legend, as follows:

<table>
<thead>
<tr>
<th>For orientation</th>
<th>Use</th>
</tr>
</thead>
</table>
| Orient=H        | B for bottom  
|                 | T for top |
| Orient=V        | R for right  
|                 | L for left |

D In the **Justify** field, specify the alignment of the legend, as follows:

- C for center
- R for right
- L for left
- T for top
- B for bottom

E In the **Order** field, specify the order of legend, as follows:

- KNOR for normal (left to right for horizontal, top to bottom for vertical)
- KREV for reversed

F In the **Box** field, specify whether to place a box around the legend, as follows:

- KBOX for a box
- NKBO for no box

4 In the **Markers** field, specify whether to use place markers, as follows:

- MARK to use markers
- NOMA for no markers

5 In the **Lines** field, specify whether to draw lines, as follows:

- LINE to draw lines
- NOLI for no lines

6 Specify the X/Y Title Attributes to determine the attributes of the X and Y axes of the graphs.

A Specify the **Color Key**, **Char Size**, and **Char H/W** the same as you did in step 2 on page 433.

B In the **Axis Color** field, specify the color of the X or Y axis. Choose a color from the Color Key at the bottom right of the panel and type its number here.
7 Specify the Label Attributes the same as you did in step 2 on page 433 to determine
the appearance of the axes labels: the Color Key, Char Size, and Char H/W.

8 If you specified Box=KBOX under Legend Key Attributes, specify the Box
Attributes to control the appearance of a box around the legend. Choose a color
from the Color Key at the bottom right side of the panel and type its number from 0
to 8 in the Color field.

When you press Enter, the second graphics options panel appears (Figure 144).

Figure 144 Printer Graphics Options panel

9 Complete the fields on this panel as follows:

A In the Device Name field, type the LUNAME of the printer.

B In the Copies field, type the number of copies to print.

C In the Page Size field, specify the number of Rows per page and the number of
Columns per row.

D In the Print Only field, type Y if you want to print the graphs on a printer instead
of displaying them on the screen, which allows you to print graphs from
nongraphics terminals. Type N to display graphs.

E Press Enter.

10 Press END to save the options and return to the main menu.
Estimating space requirements by using statistics

You can use DASD MANAGER PLUS statistics to estimate the space requirements of existing objects for predicting storage capacity. DASD MANAGER PLUS estimates space based on current object statistics from the DASD MANAGER PLUS tables. You can modify the statistics values and let DASD MANAGER PLUS recalculate the space estimates to predict the space requirements of a table space or index from recent growth trends.

**NOTE**
You must run BMCSTATS before using the space estimation function.

Simulating changes to DB2 objects

Space Estimation lets you display space statistics for DB2 objects and then simulate changes to them to see the effect on related objects. In this way, you can estimate the effects of changes to the database. For example, using the DASD MANAGER PLUS space estimation capability, you can test “what-if” scenarios, as follows:

- Increase the number of rows in a table to simulate data growth.
- Increase or decrease the primary allocation quantity for a table space to determine the effect on the number of extents.

**NOTE**
When you change the values, DASD MANAGER PLUS calculates new space requirements for the object based on those values, but it does not store the new values.
For a table space, you can simulate changes to any of the following statistics:

- page size
- segment size
- number of rows
- average row length
- primary and secondary quantity
- allocation unit
- device type
- percent free
- free pages
- table space type
- maxrows

For an index, you can simulate changes to any of the following statistics:

- key length
- whether unique or nonunique
- number of rows
- rows per key
- primary and secondary quantity
- allocation unit
- device type
- percent free
- free pages
- piecesize
- table space type

**Estimating space requirements for a table space**

Use this procedure to estimate the DASD requirements for existing DB2 table spaces by using the most current values in the DASD MANAGER PLUS statistics tables or any other values that you choose.

**Before you begin**

The space estimations (Estimated) are space that is required if you reorganize the table space. The product bases the estimates on the values for Pagesize, Segsize, Avg Row Length, Number of Rows, Priqty, Secqty, Alloc Unit, Device Type, Pctfree, and Freepage (Figure 145). The default values are the most current values in the DASD MANAGER PLUS statistics tables for the object.
Estimating space requirements for a table space

**NOTE**

Space estimation is not available for XML objects or for objects that have partial statistics.

You can edit some of these fields and press **Enter** to display new space estimates using different values. Some fields are informational only. The **Tab** key moves the cursor to fields that you can modify.

**Figure 145 Tablespace Estimation panel (nonpartitioned)**

<table>
<thead>
<tr>
<th>ASUZTBES</th>
<th>Tablespace Estimation</th>
<th>Row 1 to 1 of 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>SCROLL====&gt; PAGE</td>
<td></td>
</tr>
</tbody>
</table>

Type data and press Enter.

Tablespace Name QZUDA1.QZUS01A1

| Pagesize . . . . . . . . . . | 4 (4,8,16,32) Maxrows 255 (0-255) |
| Segsize . . . . . . . . . . . | 4 (0-64) Dssize . . 0 |
| Pages . . . . . . . . . . . | 130 MaxParts 0 |
| Space (Trks) . . . . . . . . . | 90 |
| Extents . . . . . . . . . . . | 1 |
| Priqty . . . . . . . . . . . | 6 |
| Secqty . . . . . . . . . . . | 1 |
| Pctfree . . . . . . . . . . . | 0 (0-99) Freepage 0 (0-3) |
| Alloc Unit . . . . . . . . . | C (T or C) Compress (Y or blank) |
| Device Type . . . . . . . . | 3390 |

-Row Length- --- Estimated ---

<table>
<thead>
<tr>
<th>Owner</th>
<th>Table Name</th>
<th>Max</th>
<th>#Rows</th>
<th>Pages</th>
<th>Rows/Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Avg</td>
<td>/Pages</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QZU.QZUT01_DA1S01

| 241 | 2036 | 128 | 16 |

The following information pertains to nonpartitioned table spaces:

- The panel shows current used space on the left:
  - **Pages** is the number of 4-KB, 8-KB, 16-KB, or 32-KB blocks of physical storage that the table space uses.
  
  - **Space (Trks)** is the space, in tracks, that the table space uses.
  
  - **Extents** is the number of total data set extents that the table space uses when you collected statistics. If this value is greater than the VSAM maximum (7257 per data set), the product displays a message.
Estimating space requirements for a table space

Estimations for the entire table space (if reorganized) appear in the box on the right side of the panel, as follows:

— **Pages** is the number of pages that the table space requires.
— **Space (Trks)** is the space in tracks that the table space requires.
— **Space (KB)** is the space in kilobytes that the table space requires.
— **Extents** is the number of extents that the table space requires.
— **Data Sets** is the number of data sets that the table space contains.

Estimations per table appear in edit list format, at bottom right, under the label **Estimated**, as follows:

— **Rows/Page** is the number of rows per page that the table uses.
— **Pages** is the number of pages that the table uses.

Information about the table itself appears at the bottom of the panel:

— **Owner** and **Table Name** are self-explanatory.
— **Row Length Max** is the longest row length.
— **Row Length /Avg** is the average length of the rows.
— **#Rows** is the number of rows in the table.
— **Pages** is the number of pages in the table.

**Figure 146  Tablespace Estimation (Partitioned) panel**
The following information appears for partitioned table spaces:

- **Figure 146** shows currently used space, growth information, and the type of table space on the left and center of the panel:
  
  - **Pages** is the number of 4-KB, 8-KB, 16-KB, or 32-KB blocks of physical storage that the table space uses.
  
  - **Space (Trks)** is the space in tracks that the table space uses.
  
  - **Type** indicates the type of table space:
    
    - **L** for large table spaces
    - *(DB2 Version 10 and later)* **G** for partition-by-growth members
    - *(DB2 Version 9 and earlier)* **K** for large cluster members
    - *(DB2 Version 9 and earlier)* **I** for normal cluster member
    - *(DB2 Version 9 and later)* **R** for range-partitioned universal table space
    - **blank** for normal
  
  - **Maxrows** displays the maximum number of rows per page (0 through 255).
  
  - **Dssize** is the maximum size of a data set in kilobytes.
  
  - **Segsize** is the number of pages in the segment for this table space. Segsize can only be non-zero if **Type** is **R**. If **Type** is **R** and you enter 0 in this field, the default segment size will be set to 4.

- The **Estimated** box shows estimations for the entire table space (if reorganized):
  
  - **Pages** is the number of 4-KB, 8-KB, 16-KB, or 32-KB blocks of physical storage that the table space uses.
  
  - **Space (Trks)** is the space in tracks that the table space requires.
  
  - **Space (KB)** is the space in kilobytes that the table space requires.

- Estimations for each partition appear in edit list form at bottom right under the word **Estimated**:
  
  - **Rows/Page** is the rows per page that the partition required.
  
  - **Tracks/Kbytes** are the tracks and kilobytes that the partition required.
  
  - **Ext** is the total number of data set extents that the table space partition used when you collected statistics.
If the **Extents** value is greater than the VSAM maximum (7257 per data set), the product displays a message.

- The bottom of the panel shows information about the table itself:
  - **Part** is the partition number.
  - **#Rows** is the number of rows in the partition.
  - **AvgRow** is the average row length for the partition.
  - **Priqty** is the primary allocation VSAM value from the ICF catalog.
  - **Secqty** is the secondary allocation VSAM value from the ICF catalog.
  - **Aloc unit** is the allocation unit specified for the table space.
  - **Devt** is the disk device type for the table space.
  - **Pct Free** is the percentage of free space per page for the table space.
  - **Free Page** is the page increment for inserting a free page.
  - **Compress** is whether the partition is compressed.
  - **Space** is the amount of space in tracks in the partition.
  - **Ext** is the number of data set extents in the partition.

---

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

---

**Figure 147  Tablespace Estimation (Partitioned by Growth) panel**
The following information appears for partition-by-growth table spaces:

- **Figure 147** shows currently used space, growth information, and the type of table space on the left and center of the panel:

  - **Type** indicates the type of table space:
    - *(DB2 Version 9 and later)* G for partition-by-growth table spaces
    - *(DB2 Version 9 and earlier)* I for normal cluster member
    - *blank* for normal

  - **#Rows** is the number of rows in the partition.

  - **Pagesize** is the size of the 4-KB, 8-KB, 16-KB, or 32-KB blocks of physical storage that the table space uses.

  - **Maxrows** displays the maximum number of rows per page (0 through 255).

  - **Segsize** is the number of pages in the segment for this table space.

  - **Dssize** is the maximum size of a data set in kilobytes. If you specify this field, you must also specify a value for the MaxParts field. The following values are valid: 0, 1, 2, 4, 8, 16, 32, 64. For partition-by-growth table spaces, the value for this field must be greater than 0.

  - **Pages** is the number of 4-KB, 8-KB, 16-KB, or 32-KB blocks of physical storage that the table space uses.

  - **Max Parts** is the maximum number of partitions to which the partition-by-growth table space can grow. The value must be in the range from 1 through 4096 and depends on the corresponding value for the Dssize field.

  - **Space (Trks)** is the amount of space in tracks in the partition.

  - **Alloc Parts** is the number of partitions currently allocated for the table space.

  - **MaxRow Len** is the maximum row length for the table. The value does not include overhead (header 6 bytes and map ID 2 bytes).

  - **AvgRow Len** is the average row length for the partition.

  - **Extents** is the number of data set extents in the partition.

  - **Priqty** is the primary allocation VSAM value from the ICF catalog.

  - **Secqty** is the secondary allocation VSAM value from the ICF catalog.

  - **Pctfree** is the percentage of free space per page for the table space.
— **Freepage** is the page increment for inserting a free page.

— **Alloc Unit** is the allocation unit specified for the table space.

— **Compress** is whether the partition is compressed.

— **Device Type** is the disk device type for the table space.

■ The **Estimated** box shows estimations for the entire table space (if reorganized):

— **Pages** is the number of 4-KB, 8-KB, 16-KB, or 32-KB blocks of physical storage that the table space uses.

— **Space (Trks)** is the space in tracks that the table space requires.

— **Space (KB)** is the space in kilobytes that the table space requires.

— **Extents** is the total number of data set extents, based on the primary and secondary quantities, that the table space will require to accommodate the entered specifications.

  If the **Extents** value is greater than the VSAM maximum (7257 per data set), the product displays a message.

— **Est. Parts** is the estimated number of partitions for the table space.

■ Estimations for each partition appear in edit list form at bottom right under the word **Estimated**:

— **Parts** is the partition number. A less than sign (<) followed by a partition number indicates that all the partitions less than that number have the values listed in that row. The partition that is currently active is shown on the next line.

— **#Rows** is the number of rows in the partition.

— **Rows/Page** is the rows per page that the partition required.

— **Tracks/Kbytes** are the tracks and kilobytes that the partition required.

— **Ext** is the estimate for the number of secondary allocations required in allocation units specified. The value depends on the amount of space estimates and primary and secondary allocation quantities.
To estimate space requirements for a table space

1 Create a DB2 object list that contains the table space for which you want to estimate space as instructed in “Creating a DB2 object list” on page 361.

2 Select the Space Estimation function by typing S in the Act field beside the table space for which you want space estimates.

The Tablespace Estimation panel appears. (Figure 145 shows the panel for nonpartitioned table spaces, Figure 146 shows the panel for partitioned table spaces, and Figure 147 shows the panel for partition-by-growth table spaces.)

3 Estimate space requirements for the entire table space or for individual tables or partitions (or both) as follows:

   A Type over the existing values and press Enter to update the Estimated fields.

   B You can repeat this step as many times as necessary.

4 When you have finished estimating space requirements, press END to exit.

Estimating space requirements for an index

Use this procedure to estimate the DASD requirements for existing DB2 indexes by using the most current values in the DASD MANAGER PLUS statistics tables or any other values that you choose.

**NOTE**

Space estimation is not available for XML objects or for objects that have partial statistics.

To estimate space requirements for an index

1 Create a DB2 object list that contains the index for which you want to estimate space as instructed in “Creating a DB2 object list” on page 361.

The Display DB2 Object List panel appears showing the indexes that match your specification.

2 Select the Space Estimation function by typing S in the Act field beside the index for which you are estimating space.
The Index Estimation panel appears. (Figure 148 shows the panel for nonpartitioned index spaces and Figure 149 shows the panel for partitioned index spaces.) The panel shows how much space might be required if you reorganized the index. **Index Name** shows the name of the index that the product used to estimate space. **Table Name** is the name of the table where the product created the index. **DB Name** is the name of the database containing the index. The name of the first column of the columns in the index appears in the **1stKey Col** field. If the index is partitioned, the panel lists each partition. A zero in the **Part** field indicates a nonpartitioned index.

The product estimates space (at bottom right under the label **Estimated**) based on values in the **Keylength**, **Unique**, **Number of Rows**, **Rows per Key**, **Priqty**, **Secqty**, **Alloc Unit**, **Devt**, **Pctfree**, and **Free Page** fields. These values default to the most current values in the DASD MANAGER PLUS statistics tables for the object.

3 You can edit some of the fields on this panel, press **Enter**.

Other fields are for information only. The **Tab** key moves between fields that you can modify.

**Figure 148  Index Space Estimation panel (nonpartitioned)**

The following information appears for a nonpartitioned index as Figure 148 shows:

- Actual current space is shown in the upper left. **Space (Trks)** is the space in tracks that the index space uses.
The **Estimated** box shows estimations for the entire index (if reorganized):

- **Pages** is the number of pages that the object requires.
- **Space (Trks)** is the space in tracks that the index requires.
- **Space (KB)** is the space in kilobytes that the index requires.
- **Levels** is the number of levels in the index.
- **Data Sets** is the number of data sets for a nonpartitioned index.

Estimations per partition appear in edit list format at bottom right:

- **Tracks/Kbytes** is the tracks and kilobytes that a specific partition requires.
- **Ext** is the number of extents that a specific partition requires, based on primary and secondary quantities. If this value is greater than the VSAM maximum (7257 per data set), the panel displays a message.

4. Estimate space requirements for the index. Type new values over the existing values and press **Enter** to update the **Estimated** fields. You can repeat this step as necessary.

5. When you have finished estimating space requirements, press **END** to exit.

**Figure 149  Index Space Estimation panel (partitioned)**

<table>
<thead>
<tr>
<th>ASUZIPES</th>
<th>Index Space Estimation</th>
<th>Row 1 to 1 of 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ====&gt;</td>
<td></td>
<td>SCROLL====&gt; PAGE</td>
</tr>
</tbody>
</table>

Type data and press Enter.

Index Name . . QZU.QZUX02_DB6S01T01
Table Name . . QZU.QZUT01_DB6S01
1stKey Col . C007_DATE

Pagesize . . . 4
Unique . . . D (U/P/N/D/C/R/G)
Compress . . N (Y or blank)
Pct Compressed 0 (0 - 99)
Keylength. . . 4 (1-2000)
Avgkeylen . . 4
Pages . . 234
Space (Trks) 300

| -------- Estimated ------- |
| Pages . . 228 |
| Space (Trks) 60 |
| Space (KB) . 2880 |
| Levels . . . 2 |

| Level Avgnlkeylen Avgkeyln |
|---------------------------|---------------------------|

| Part #Rows /Key Priqty Secqty U Devt Free Page Space Ext Kbytes Tracks Ext |
|---------------------------|---------------------------|
| Avgnlkeylen Avgkeyln |

| 1 | 0 | 0 | 5 | 1 | C | 3390 | 1 | 0 | 75 | 1 | 720 |
The following information appears for a partitioned index, as Figure 149 shows:

- Current used space appears in the upper left. **Space (Trks)** is the space in tracks that the index space uses.

- The **Estimated** box shows estimations for the entire index (if reorganized):
  - **Pages** is the number of pages that the index requires.
  - **Space (Trks)** is the space in tracks that the index requires.
  - **Space (KB)** is the space in kilobytes that the index requires.
  - **Levels** is the number of levels in the index.

- Estimations per partition appear in edit list format at bottom right:
  - **Tracks/Kbytes** is the tracks and kilobytes that a specific partition requires.

  — **Ext** is the number of extents that a specific partition requires based on primary and secondary quantities. If this value is greater than the VSAM maximum (7257 per data set), the panel displays a message.

6 Estimate space requirements for the index. Type new values over the existing values and press **Enter** to update the **Estimated** fields. You can repeat this step as necessary.

7 When you have finished estimating space requirements, press END to exit.

---

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

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

**NOTE**

You do not need to run BMCSTATS before using SSE.
For example, for a non-partitioned table space estimate, you can change the fields highlighted in Figure 150. 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 150 Tablespace Estimation panel**

Estimates for entire table

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pageno</td>
<td>16</td>
<td>(4, 8, 16, 32)</td>
</tr>
<tr>
<td>Segsize</td>
<td>4</td>
<td>(0-64)</td>
</tr>
<tr>
<td>Maxrows</td>
<td>255</td>
<td>(0-255)</td>
</tr>
<tr>
<td>Dssize</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Pctfree</td>
<td>0</td>
<td>(0-99)</td>
</tr>
<tr>
<td>Freepage</td>
<td>0</td>
<td>(0-255)</td>
</tr>
<tr>
<td>Compress</td>
<td>Y</td>
<td>(Y or blank)</td>
</tr>
<tr>
<td>Priqty</td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>Secqty</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Alloc Unit</td>
<td>T (T or C)</td>
<td></td>
</tr>
<tr>
<td>Device Type</td>
<td>3390</td>
<td></td>
</tr>
</tbody>
</table>

Tables: Enter I to Insert, R to Repeat, or D to Delete a line.

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Length</th>
<th>#Rows</th>
<th>Avg Row</th>
<th>---- Estimated ----</th>
<th>Rows/Page</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Estimating space requirements for a table space**

You can use SSE to estimate space requirements for non-partitioned, partitioned, and partition-by-growth table spaces:

- “To estimate space requirements for a non-partitioned table space” on page 449
- “To estimate space requirements for a partitioned table space” on page 449
- “To estimate space requirements for a partition-by-growth table space” on page 450

**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 (highlighted in Figure 150) as needed to correspond to your table space.

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

B Press Enter to update the Estimated fields:

- The Estimated box on the right side of the panel displays estimates for the *entire table space*.

- The Estimated list in the bottom right corner shows estimates *per table*.

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

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

5 When finished, press END to exit.

To estimate space requirements for a partitioned table space

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

2 On the Space Estimation panel, enter `TS` in the `Object Type` field, and `P` in the `Tablespace Type` field. Then, press Enter.
Estimating space requirements for a table space

3 Estimate space requirements for the entire table space or for individual partitions as follows:

A Change any of the modifiable values (highlighted in Figure 150) as needed to correspond to your table space.

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

B Press Enter to update the Estimated fields:

- The Estimated box on the right side of the panel displays estimates for the entire table space.
- The Estimated list in the bottom right corner shows estimates per partition.

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

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

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

5 When finished, press END to exit.

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

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

2 On the Space Estimation panel, enter TS in the Object Type field, and G in the Tablespace Type field. Then, press Enter.
3 Estimate space requirements for the entire table space or for individual partitions as follows:

A Change any of the modifiable values (highlighted in Figure 150) as needed to correspond to your table space.

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

B Press Enter to update the Estimated fields:

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

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

5 When finished, press END to exit.

---

**Estimating space requirements for an index**

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

**NOTE**
Space estimation is not available for XML objects, 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 as needed to correspond to your index.

---

**NOTE**

For more information about a specific field, press F1.

---

B Press Enter to update the **Estimated** fields:

- The **Estimated** box on the right side of the panel displays estimates for the index.

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

---

**NOTE**

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

---

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

5 When finished, press END to exit.
Chapter 8  Analyzing objects by using BMCTRIG

This chapter contains the following topics:

Before you begin .................................................. 454
Overview of the main features of BMCTRIG ........................................ 456
  System triggers ............................................. 457
  Exception conditions ...................................... 458
  Suppressing categories of exceptions from analysis ..................... 461
  Job generation ............................................. 462
  Methods of running BMCTRIG .................................. 465
Setting up BMCTRIG ........................................... 470
  Selecting objects to evaluate ................................ 470
  Automating corrective actions ................................ 470
  Using BMCTRIG processing options .......................... 471
  Sample workflows ......................................... 472
Setting exception thresholds ....................................... 476
  Understanding reorganization-related exceptions .................. 476
  Understanding statistics-related exceptions ....................... 482
  Understanding copy-related exceptions ........................ 483
  Setting thresholds in BMCTRIG syntax .......................... 485
  Defining system-level thresholds ................................ 488
  Creating user-defined exceptions ................................ 494
  Identifying threshold values .................................. 496
Managing corrective actions ........................................ 496
  Setting a corrective action in BMCTRIG syntax .................... 496
  Defining system-level corrective actions ........................ 497
  Defining system-level object-action priorities ....................... 500
Setting up a BMCTRIG job ....................................... 504
  Creating an action with BMCTRIG .............................. 504
  Specifying BMCTRIG syntax options .............................. 507
  Generating a BMCTRIG job .................................... 524
  Using worklist format or standard JCL ............................. 526
Viewing exceptions that BMCTRIG produces .......................... 545
Working with reorganization guidelines for thresholds .................. 549
Generating an automatic reorganization with BMCTRIG ............... 553
Setting up a REORG corrective action .................................. 556
Before you begin

The tasks in this chapter build on the information from previous chapters. You should be familiar with creating and editing actions (Chapter 4, “Maintaining and generating actions”), working with objects sets (see Chapter 5, “Using object sets”), and BMCSTATS (Chapter 6, “Collecting and managing statistics”).

When you run the BMCTRIG utility, it can evaluate the statistics in the DASD MANGER PLUS statistical database, the real-time statistics (RTS) catalog tables, or both. Because real-time statistics are gathered continuously as part of a DB2 background process that automatically writes them to the RTS catalog tables, you do not need to collect these statistics before running BMCTRIG. If you are using DB2 RTS only and not any of the DASD MANAGER statistics for evaluation, you do not need to run BMCSTATS on the objects before running BMCTRIG. If you are not using DB2 RTS, to ensure that you are monitoring current BMCSTATS values, collect or update the BMCSTATS statistics in the database before running BMCTRIG.

By using BMCTRIG, you can monitor changes in your database statistics. BMCTRIG reports significant changes (exceptions) and automatically creates maintenance jobs to perform the actions that you specify, based on the statistical data in the database. When encountering an exception, BMCTRIG logs it in an exceptions table for monitoring and trend analysis. The exception can also initiate corrective actions that contain services such as utilities.

Figure 151 shows how BMCTRIG can produce a report and a reorganization job if specific table spaces or indexes exceed a specified threshold (such as five extents) or if a percentage of modified rows is greater than 20 percent.
Figure 151  Initiating a reorganization based on exception values

For the contents of the DASD MANAGER PLUS statistical database, see Table 43 and the statistics tables in the DASD MANAGER PLUS for DB2 Reference Manual.

You can set BMCTRIG to evaluate exception thresholds and then generate and (optionally) submit the following services:

- statistics utilities
- image copy utilities
- reorganization utilities
- unload and load utilities
- database commands and utilities
- check and report utilities
- worklist commands
- user-defined services
- resizing commands and utilities

For specific information about exceptions and BMCTRIG rules, see the DASD MANAGER PLUS for DB2 Reference Manual.
Overview of the main features of BMCTRIG

Because every DB2 application has different requirements for availability and performance, the amount of update activity differs for each application. These requirements affect when you should perform maintenance (such as reorganizations) on objects. DB2 does not provide an automated way of determining when an object needs maintenance. Often, you need to use SQL and IDCAMS reports to try to determine which objects require maintenance. Running reorganizations and copies on a scheduled basis rather than only when the object is exceeding site guidelines can be costly in terms of wasted CPU cycles and work hours. DASD MANAGER PLUS allows you to test for user-defined thresholds, and if exceptions occur, to take user-defined corrective action.

By using the BMCTRIG utility, you can monitor changes in your database statistics. BMCTRIG reports significant changes (exceptions) and automatically creates maintenance jobs to perform the actions that you specify based on the condition of your DB2 objects.

You define the exception thresholds that BMCTRIG monitors. You can set different thresholds for different objects in your subsystem and you can assign a priority to specific thresholds to indicate the severity of the condition. When executed, BMCTRIG checks for objects that meet or exceed the thresholds that you have defined.

If thresholds are exceeded, BMCTRIG takes a user-defined action. The following user-defined actions are examples:

- Record the exception in the Exceptions Table, logging when the exceptions occurred, and what the threshold and actual values were for the exception.

- Produce a batch report when BMCTRIG encounters an exception condition that contains the following information:
  - exception condition
  - object name and type
  - current value of the statistic
  - compare value and threshold value
  - corrective action taken (such as a user-defined REORG utility job)

- Issue a TSO notify command to the appropriate person.

- Generate maintenance jobs that use the corrective actions that you defined.

BMCTRIG uses priorities to order work and to limit generation of corrective action work. The corrective actions can include any utilities that DASD MANAGER PLUS supports or your own (user-defined) program.
System triggers

System-level triggers provide the following benefits:

— You can set thresholds by using wildcard patterns, object set names, or specific object names.

— System-level triggers allow you to assign a priority value for each exception that you define to indicate the severity of the condition encountered. BMCTRIG uses priorities to order the objects when generating maintenance jobs. You can also use priorities to limit which corrective actions are generated.

— Using system-level triggers is the only method that allows you to define and use your own exceptions. To create your own exceptions, you need to provide REXX routines to perform the evaluation. You then need to define your exception to DASD MANAGER PLUS. For more information, see “Creating user-defined exceptions” on page 494.

— You determine what to monitor. Define as many or as few thresholds as you want.

**TIP**

BMC installs a set of common thresholds that you can use as an initial set of conditions to monitor.

You can specify thresholds and corrective actions in BMCTRIG command syntax, or you can define them in a central repository as system-level thresholds for use by all BMCTRIG jobs, as follows:

- thresholds in the BMCTRIG command syntax

  Specifying thresholds and corrective actions in command syntax makes them applicable to that specific job only. (To specify an option in the command syntax, typically you select that option by using the BMCTRIG panel.)

- system-level thresholds

  Specifying thresholds and corrective actions in the central repository makes them available to all BMCTRIG jobs and provides a central point to administer exception thresholds and corrective actions. Using system triggers also provides a greater level of flexibility and control than is available when thresholds and actions are specified in BMCTRIG command syntax. For more information about the flexibility and control that system triggers provide, see “Exception conditions” on page 458 and “Job generation” on page 462.
If you administer thresholds at the system level, you can also choose whether to allow syntax overrides at the BMCTRIG level. The installation option SYSTRIGS, in conjunction with the command syntax option SYSTEMTRIGGERS, provides control over which definitions BMCTRIG uses.

Using the installation option SYSTRIGS=F forces BMCTRIG to use system triggers. When you specify this option, BMCTRIG ignores any thresholds or corrective actions that you specified in BMCTRIG command syntax. When this option is in effect, BMCTRIG also ignores the SYSTEMTRIGGERS option in the command syntax.

When you set SYSTRIGS to Y or N in the installation options module, this option specifies whether BMCTRIG uses system triggers as the default behavior. In this situation, you can use the SYSTEMTRIGGERS option in the command syntax to override the default behavior for individual BMCTRIG jobs.

If you do not set the SYSTRIGS option in the installation options module, BMCTRIG ignores system triggers by default (which is similar to setting it to N). In this case, BMCTRIG uses system triggers only when you specify the SYSTEMTRIGGERS option in the command syntax.

**NOTE**
You must define system triggers by selecting *Thresholds, Corrective Actions, and Priorities* from the DASD MANAGER PLUS main menu before you execute the BMCTRIG job to use the triggers.

---

**Exception conditions**

DASD MANAGER PLUS provides a robust set of statistics and conditions for monitoring DB2 objects. The exceptions define conditions that are related to typical DB2 utility needs such as collecting statistics, making image copies, and performing data reorganization.

You should review the exceptions that BMCTRIG monitors to determine which exceptions to check against threshold values, either for reporting or for automating corrective actions. After reviewing the exceptions that BMC Software provides, determine whether there are other site-specific conditions that you want to monitor. To create your own exceptions (called user-defined exceptions), you need to provide REXX routines to perform the evaluation and define your exceptions for DASD MANAGER PLUS. For more information, see “Creating user-defined exceptions” on page 494.
For the exceptions that you decide to monitor, you also need to provide threshold values that indicate when a problem occurs. BMC Software supplies recommended thresholds, but you can set the thresholds to values that are appropriate for your environment.

You can specify exception conditions as follows:

- exceptions without system triggers (with the command syntax)

  If you do not use system triggers, you must define exceptions in the BMCTRIG command syntax. Use the Override System Trigger Exceptions options on the BMCTRIG service syntax dialog to define these exceptions (and specify System Triggers N, which is the default value). BMCTRIG uses these thresholds to analyze all objects that this BMCTRIG job processes.

- system-level triggers (without command syntax overrides)

  To use system-level triggers, select Thresholds, Corrective Actions, and Priorities from the DASD MANAGER PLUS main menu. System-level triggers allow you to set different threshold values for different objects in your subsystem. For example, each application might have a different tolerance level.

- command syntax with system-level triggers

  If you do not set the SYSTRIGS installation option to F (that is, FORCE), you can specify thresholds in your command syntax in addition to the system-level thresholds. To specify thresholds in the command syntax, use the BMCTRIG Override Thresholds product panels.

  Thresholds that you specify in the command syntax override corresponding thresholds that are specified in the repository. Also, you can use override thresholds to specify thresholds for exceptions that are not defined in the system triggers repository. When you override a system threshold, BMCTRIG ignores the priority values that are defined in the repository for that threshold.

---

**NOTE**

BMCTRIG will not evaluate exceptions on clone tables because clone tables have no associated statistics. BMCTRIG generates utilities for the base table space and its associated tables and indexes. However, you can specify the CLONE keyword to run on the clone objects instead of the base objects.
Threshold priorities

When you set different thresholds for different objects in your subsystem, you can assign a priority (0-255) to a threshold to indicate the severity of the condition. BMCTRIG uses priorities to order work and to limit generation of corrective action work. The following example shows a possible way of defining exceptions, thresholds, and priorities:

<table>
<thead>
<tr>
<th>Exception</th>
<th>Threshold</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTENTS</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>EXTENTS</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>EXTENTS</td>
<td>75</td>
<td>100</td>
</tr>
</tbody>
</table>

In this example, an object with 55 extents receives an exception with a priority of 50. An object with 80 extents receives an exception with a priority of 100. BMCTRIG considers these priorities when determining which objects to process first in the generated jobs. Prioritizing ensures that BMCTRIG addresses the most severe conditions first when executing the generated jobs.

BMCTRIG allows you to define thresholds at a subsystem level based on object names, patterns, or sets. BMCTRIG also allows you to define thresholds in the syntax. Defining thresholds at the subsystem level gives a central point to administer exception handling. If you decide to administer thresholds at the subsystem level, you can also decide whether to allow threshold overrides in the command syntax.

For information about how to define your own exceptions or define thresholds at the subsystem level, see “Setting exception thresholds” on page 476.

Corrective actions for exceptions

After you determine which exceptions to monitor, you need to define the corrective actions that address the problems that BMCTRIG identifies. Typically, you define a corrective action for each category of exceptions that you want to monitor. For example, you can create a corrective action that runs a BMCSTATS service when the product encounters a NOSTATS exception; or if the product encounters the DIRTY exception, it generates a BMCCOPY service.

However, if you are defining thresholds and corrective actions at the subsystem level, you set the appropriate corrective action for each exception that you monitor. Additionally, you can determine whether to use different corrective actions for different sets of objects. BMC Software supplies a set of initial corrective actions that you can use, or you can set up your own actions to meet your particular needs.
If you are using subsystem-level thresholds and corrective actions, you can establish priorities for objects when a particular corrective action occurs. The priority on the object-action occurrence takes precedence over the threshold priority, which allows you to establish work priority for a combination of objects and actions. For example, critical objects that need a reorganization might have a high priority while test objects that need a reorganization might have a low priority. BMCTRIG orders the corrective action work by priority and optionally limits work generation by priority.

Chapter 4, “Maintaining and generating actions,” discusses creating and editing actions. For information about linking corrective actions to exceptions, see “Managing corrective actions” on page 496.

Suppressing categories of exceptions from analysis

Most of the DASD MANAGER PLUS exceptions are classified in one of the following categories:

- statistics conditions
- copy conditions
- reorganization conditions

When executing BMCTRIG, you can bypass evaluations based on one of these three categories. This action is particularly useful when you are including system-defined triggers. For example, suppose you have system-defined triggers to detect REORG, COPY, and STATS exceptions. You want to run a BMCTRIG job using the system-defined triggers, but you are only interested in REORG conditions. You can bypass the analysis of exceptions for the COPY and STATS conditions.

You use the following keywords to suppress exceptions from being analyzed based on category type:

- TRIGNOREORG allows bypassing all exceptions categorized with a type of REORG during the trigger analysis phase.
- TRIGNOSTATS allows bypassing all exceptions categorized with a type of STATS during the trigger analysis phase.
- TRIGNOCOPY allows bypassing all exceptions categorized with a type of COPY during the trigger analysis phase.
Job generation

For job generation, first determine what to generate. That is, select corrective actions to be in response to exceptions that BMCTRIG encounters. Whether or not you use system triggers, you must already have defined actions for various maintenance tasks. (For more information about defining actions, see Chapter 4, “Maintaining and generating actions.”) For example, you might have defined the following types of actions:

- statistics maintenance actions
- copy maintenance actions
- reorganization maintenance actions

When you use your defined actions with BMCTRIG in response to triggers and exceptions, actions are referred to as corrective actions. You can use corrective actions as follows:

- without system triggers (command syntax only)

  When you are not using system triggers, you identify the name of the corrective action that you want to have generated in the Default Action field of the BMCTRIG JCL Generation dialog panel. You can specify only one corrective action, which is generated for all exceptions that are identified by that BMCTRIG job.

- with system triggers

  When you use system triggers, you can associate them with specific exceptions, as in the following examples:

  — Associate a statistics corrective action with the BSTATAGE exception to generate a statistics job whenever BMCTRIG encounters this exception.

  — Associate a copy action with copy related exceptions such as DIRTY, COPYAGE, or COPYPEND.

  — Associate reorganization actions with related exceptions such as EXTENTS, SPACE, and PCTCLUS.
System triggers enable you to choose different corrective actions for different sets of objects, as in the following examples:

— Use an Online Reorg action for some applications and a Shrlevel None Reorg for other applications.

— Use a statistics action that updates the DB2 catalog for some applications and one that does not update the catalog for other applications.

— You can use wildcard patterns, object set names, or specific object names to identify which corrective action to take for specific exceptions.

Using system triggers enables you to identify all exception situations, but generate work only for specific objects or exceptions. When objects or exceptions that are not defined in the Corrective Actions repository are triggered, BMCTRIG logs them in the Exceptions table and notes them in the report, but does not generate a corrective action.

**Duplicate actions**

BMCTRIG offers the option for eliminating duplicate actions. Specify ELIMDUPACTS Y and BMCTRIG eliminates an action if it contains the same services in the same order as another action. Table 55 shows three example actions:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>COPY named “weekly copy”</td>
</tr>
<tr>
<td>2</td>
<td>QUIESCE REORG Shrlevel N reorganization COPY named “weekly copy” STATS named “update catalog”</td>
</tr>
<tr>
<td>3</td>
<td>STATS named “update catalog”</td>
</tr>
</tbody>
</table>

In this example, if one object generates all three of the corrective actions and you have specified ELIMDUPACTS Y, BMCTRIG eliminates both actions 1 and 3 for that object. In this case, action 2 contains exactly the same actions with the same syntax as actions 1 and 3. BMCTRIG analyzes each object individually to determine whether duplicate actions exist. When BMCTRIG eliminates an action, you receive an information message.

If BMCTRIG eliminates an action with a higher priority than the remaining action, it raises the priority of the remaining action.
BMCTRIG also provides the ELIMDUPOPTS option which causes BMCTRIG to eliminate actions if they are functionally duplicated by an inline option of another service. When generating BMCREORG or IBM REORG, BMCTRIG detects inline copy (BMCCOPY or FULLCOPY). When generating IBMREORG, BMCTRIG detects inline statistics collection services (BMCSTATS or RUNSTATS). BMCTRIG does not eliminate individual services from an action.

**TIP**
Whenever possible, make COPY and STATS actions individual actions so that BMCTRIG can eliminate duplicates.

### Object-action priorities

Using system triggers also enables you to establish priorities for specific objects when a particular corrective action is initiated. These object-action priorities are not the same as the priorities for thresholds. The object-action priorities take precedence over the threshold priorities. Object-action priorities allow you to establish work priority for a combination of objects and actions. For example, you can set a lower priority for an application that is already highly-tuned.

BMCTRIG assigns object-action priorities as follows:

- If there is no active exception for the object-exception in the EXCEPTIONS2 table and if the object-action table has an applicable row with a priority, BMCTRIG assigns the object-action priority.

  If there is no active exception for the object-exception in the EXCEPTIONS2 table but the object-action table does not have an applicable row with a priority, BMCTRIG assigns the threshold priority.

- BMCTRIG increments or ages the exception priority by taking the priority in the table and adding the aging number multiplied by the number of days since the exception row was last updated.

- For GROUP YES, the priority of all objects for an action is set to the highest priority of any exception row for all objects with that action. For GROUP NO, the priority of an object-action is the highest priority of an exception row for that object and action. For more information about grouped services, see “Grouped services” on page 466.

For procedures for using object-action priorities, see “Defining system-level object-action priorities” on page 500.
BMCTRIG generates work for the objects and actions that it identifies as follows:

- If any priorities exist or you have specified JCLWLB Y, BMCTRIG sorts by job, priority, object ID, action name. Otherwise, if you have specified GROUP Y, BMCTRIG sorts by action name and object ID. (By default, BMCTRIG sorts work by object ID and action name).

- BMCTRIG sorts Object IDs by DBID, object type, PSID, and partition number.

- Other rules apply for ordering work or keeping objects together within the same job for workload balancing. See “Workload balancing” on page 468.

- BMCTRIG reorganizes nonpartitioning indexes before their corresponding table space partitions.

- BMCTRIG performs index REBUILDs after performing the corresponding table space recoveries.

**Default actions**

You can also specify a default corrective action in command syntax (by using the Default Action field of the BMCTRIG JCL Generation dialog panel) for BMCTRIG to use in addition to the Corrective Actions defined in the repository. You can use a default action even when SYSTRIGS=F.

When you specify a default action, BMCTRIG generates that action in the following situations:

- for any exception that does not have a corrective action already defined for it in the repository table

- for any object that does not have exceptions and corrective actions defined in the repository

**Methods of running BMCTRIG**

BMCTRIG can identify exceptions and generate jobs in a single invocation. Alternately, you can also split the processing into two independent phases as follows:

- In the first execution, allow BMCTRIG to analyze objects and identify exceptions conditions. Make sure you specify SAVE Y on the BMCTRIG Exception Options dialog panel to write the exceptions to the Exceptions table.
Review the exceptions that were initiated and optionally modify them on the Active Exceptions dialog panel (by choosing Thresholds, Corrective Actions, and Priorities from the DASD MANAGER PLUS main menu). You can use this dialog to do the following tasks:

— Remove exceptions.
— Inactivate exceptions.
— Change the priority of an exception.
— Add a corrective action or modify the corrective action for an exception.

Execute BMCTRIG a second time by using the RESUME Y option. This option reads the Exceptions table and generates corrective actions for any exception where the Active indicator is set to Y.

If a corrective action is not logged in the Exceptions table, BMCTRIG uses the default action if you have specified one in the command syntax. If you have not specified a default action, BMCTRIG ignores the exception and does not generate an action for it.

To use system triggers in one area but not both, run a two-step BMCTRIG process by using RESUME Y for the second invocation. For example, specify SYSTEMTRIGGERS N for the exception identification process. Then use SYSTEMTRIGGERS Y during the RESUME processing.

**Grouped services**

Grouped services allow you to process multiple table spaces at one time or multiple partitions (for some utilities) when you run a utility. Grouped services are more efficient and lead to greater performance. BMCTRIG can generate the services in Table 56 as grouped services when the Execution type is G (grouped) (see Figure 63 for an example):

**Table 56 Grouped services**

<table>
<thead>
<tr>
<th>Supported services for grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCCOPY</td>
</tr>
<tr>
<td>BMCCOPYI</td>
</tr>
<tr>
<td>BMCMOD</td>
</tr>
<tr>
<td>BMCREORG</td>
</tr>
<tr>
<td>BMSTATS</td>
</tr>
<tr>
<td>BMCUPRS</td>
</tr>
<tr>
<td>BMCCPRS</td>
</tr>
</tbody>
</table>

---

Methods of running BMCTRIG

**Review the exceptions that were initiated and optionally modify them on the Active Exceptions dialog panel (by choosing Thresholds, Corrective Actions, and Priorities from the DASD MANAGER PLUS main menu).** You can use this dialog to do the following tasks:

— Remove exceptions.
— Inactivate exceptions.
— Change the priority of an exception.
— Add a corrective action or modify the corrective action for an exception.

Execute BMCTRIG a second time by using the RESUME Y option. This option reads the Exceptions table and generates corrective actions for any exception where the Active indicator is set to Y.

If a corrective action is not logged in the Exceptions table, BMCTRIG uses the default action if you have specified one in the command syntax. If you have not specified a default action, BMCTRIG ignores the exception and does not generate an action for it.

To use system triggers in one area but not both, run a two-step BMCTRIG process by using RESUME Y for the second invocation. For example, specify SYSTEMTRIGGERS N for the exception identification process. Then use SYSTEMTRIGGERS Y during the RESUME processing.

**Grouped services**

Grouped services allow you to process multiple table spaces at one time or multiple partitions (for some utilities) when you run a utility. Grouped services are more efficient and lead to greater performance. BMCTRIG can generate the services in Table 56 as grouped services when the Execution type is G (grouped) (see Figure 63 for an example):

<table>
<thead>
<tr>
<th>Supported services for grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCCOPY</td>
</tr>
<tr>
<td>BMCCOPYI</td>
</tr>
<tr>
<td>BMCMOD</td>
</tr>
<tr>
<td>BMCREORG</td>
</tr>
<tr>
<td>BMSTATS</td>
</tr>
<tr>
<td>BMCUPRS</td>
</tr>
<tr>
<td>BMCCPRS</td>
</tr>
</tbody>
</table>
Methods of running BMCTRIG

Chapter 8  Analyzing objects by using BMCTRIG

In the following example, the first three table spaces in the worklist are ungrouped:

- BMCC COPY TABLESPACE DB.TS1
- BMCC COPY TABLESPACE DB.TS2
- BMCC COPY TABLESPACE DB.TS3

Grouping them as follows leads to greater efficiency:

- BMCC COPY TABLESPACE DB.TS1 TABLESPACE DB.TS2 TABLESPACE DB.TS3

Grouped services allow you to process multiple partitions as well.

**NOTE**

For REORG, BMCREORG, COPY, and BMCCOPY grouped services, if any object within the group is specified as not logged, SHRLEVEL CHANGE will not be generated. Instead, SHRLEVEL CHANGE will be changed automatically to SHRLEVEL REFERENCE.

In the following example, partitions 1, 7, and 10 are ungrouped:

- BMCR REORG TABLESPACE DB.TS1 PART 1
- BMCR REORG TABLESPACE DB.TS1 PART 7
- BMCR REORG TABLESPACE DB.TS1 PART 10

You can group them in the worklist as follows:

- BMCR BMCREORG TABLESPACE DB.TS1 PART 1,7,10

BMCTRIG generates grouped services only for contiguous services (that is, more than one grouped service in a sequence) that are at the beginning or end of actions as grouped services. In the following example, BMCTRIG groups the first and last services:

1. grouped service
2. service that cannot be grouped
3. service that cannot be grouped
4. grouped service
For example, if a corrective action contains several services, and BMCTRIG is generating these services for more than one object, the table spaces appear ungrouped in the worklist, as follows:

<table>
<thead>
<tr>
<th>Service Details</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCC COPY TABLESPACE DB.TS1</td>
<td>grouped service</td>
</tr>
<tr>
<td>BMCL LOAD TABLESPACE DB.TS1</td>
<td>service that cannot be grouped</td>
</tr>
<tr>
<td>BMCU BMCSTATS TABLESPACE DB.TS1</td>
<td>grouped service</td>
</tr>
<tr>
<td>BMCC COPY TABLESPACE DB.TS2</td>
<td>grouped service</td>
</tr>
<tr>
<td>BMCL LOAD TABLESPACE DB.TS2</td>
<td>service that cannot be grouped</td>
</tr>
<tr>
<td>BMCU BMCSTATS TABLESPACE DB.TS2</td>
<td>grouped service</td>
</tr>
</tbody>
</table>

BMCTRIG processes at the action service level first and then determines groups at the object level. In this example, BMCTRIG determines that both the first and last services (BMCCCOPY and BMCSTATS) can be grouped for the TABLESPACE DB.TS1 objects. BMCTRIG then determines that the first and last services (BMCCCOPY and BMCSTATS) can be grouped for the TABLESPACE DB.TS2 objects. In this scenario, BMCTRIG generates the following worklist:

<table>
<thead>
<tr>
<th>Service Details</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCC COPY TABLESPACE DB.TS1 TABLESPACE DB.TS2</td>
<td>grouped service</td>
</tr>
<tr>
<td>BMCL LOAD TABLESPACE DB.TS1</td>
<td>service that cannot be grouped</td>
</tr>
<tr>
<td>BMCL LOAD TABLESPACE DB.TS2</td>
<td>service that cannot be grouped</td>
</tr>
<tr>
<td>BMCU BMCSTATS TABLESPACE DB.TS1 TABLESPACE DB.TS2</td>
<td>grouped service</td>
</tr>
</tbody>
</table>

In the next example, BMCTRIG ignores the middle grouped service because it falls between two services that cannot be grouped. BMCTRIG groups only the first and last services, as follows:

<table>
<thead>
<tr>
<th>Line</th>
<th>Service</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>QUIESCE</td>
<td>grouped service</td>
</tr>
<tr>
<td>2.</td>
<td>UNLOAD</td>
<td>service that cannot be grouped</td>
</tr>
<tr>
<td>3.</td>
<td>QUIESCE</td>
<td>grouped service [ignored]</td>
</tr>
<tr>
<td>4.</td>
<td>LOAD</td>
<td>service that cannot be grouped</td>
</tr>
<tr>
<td>5.</td>
<td>COPY</td>
<td>grouped service</td>
</tr>
</tbody>
</table>

If lines 2 and 4 were removed, the QUIESCE could be grouped.

**Workload balancing**

Workload balancing is available only when you are generating standard JCL format. You specify JCLWLB Y and a maximum number of jobs with NUMJOBS or MINJOBS.

BMCTRIG performs work balancing by ordering the object-action occurrences by priority and then by cost. BMCTRIG determines the cost by adding the number of active pages that the service will process for each service in the action for that object. If the service is at the space level, the cost is the number of pages for the space. If the service is at the partition level, the cost is the number of pages for the partition. If the
service will process indexes along with the table space or partition, it adds the pages for the table space and index partitions. The BMCSTATS tables provide the number of active pages. If no statistics exist, BMCTRIG retrieves the physical size from the ICF catalog.

Some services have no associated cost, such as ALTERSEC, BMCMOD, BR14, IDCAMS, MODIFY, QUIESCE, REPAIR, REPORT, RESIZE, START, STOP, and SYNC.

The objects are assigned to a job by assigning them to the lowest cost job. The job is the lowest cost if the sum of the object-action costs in it is the lowest of the jobs. When a job exceeds the maximum number of steps, no more work can be assigned to it. If objects cannot be assigned to any job due to exceeding the number of steps, BMCTRIG issues a warning message, and the exceptions for that object-action will remain active.

For workload balancing, the following rules for generating actions apply to BMCTRIG:

- All actions for an object are kept together in the same job.
- All partitions for an object are kept together if any action contains a service that does not support part level processing.
- Related indexes and the corresponding table space are kept together when any of the following conditions exist:
  - resizing is specified in any action
  - an index will be escalated to the table space for a service
  - RECOVER is in any action
  - REORG or BMCREORG and the table space have at least one nonpartitioned index
Setting up BMCTRIG

This section describes considerations for setting up BMCTRIG to analyze the DB2 objects in your system and automate corrective actions. This section also provides sample workflows.

Selecting objects to evaluate

When you select the objects to evaluate for exceptions and corrective action automation, consider how many objects to evaluate in one job. For a very large DB2 environment, you might want to divide the work into more than one BMCTRIG job.

Also, consider dividing the work into logical sets of objects. After you determine how to divide the work in your subsystem, you can create object sets for complex object selection or use a simple wildcard pattern. For more information about object sets, see Chapter 5, “Using object sets.”

Automating corrective actions

When using BMCTRIG to evaluate exceptions and perform corrective actions, you can select from the following processing options:

- Evaluate exceptions only (and do not generate corrective actions).
- Evaluate and generate corrective actions.
- Evaluate in one step, manually review and modify corrective actions, and generate corrective actions in a separate step.
- Evaluate and register corrective actions to the Automation component.
- Generate corrective actions without evaluating exceptions.
Using BMCTRIG processing options

BMCTRIG provides a robust set of options for you to control its processing and output. For more information about the processing options, including the following topics, see “Creating an action with BMCTRIG” on page 504:

- excluding objects
- worklist format or standard JCL
- workload balancing
- resizing
- rebinding
- JCL cleanup

Table 57 shows the general BMCTRIG processing options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluate and Generate</td>
<td>This option detects exceptions for objects that meet or exceed thresholds that you define. If BMCTRIG finds exceptions, it generates jobs for the corresponding objects.</td>
</tr>
<tr>
<td>Evaluate Exceptions</td>
<td>This option detects exceptions for objects that meet or exceed thresholds that you define, but does not generate utility jobs. If BMCTRIG finds exceptions, it records them in the exceptions table and writes them to the exceptions report in the job output. You can review the exceptions and create customized exceptions reports to analyze the condition of the data in your database.</td>
</tr>
<tr>
<td></td>
<td>With this option, BMCTRIG does not generate any utility jobs. Later, you can use the Resume Generation option to generate utility jobs.</td>
</tr>
<tr>
<td>Generate Jobs</td>
<td>This option generates JCL for all objects that are specified in the Object Name field or in the object set. BMCTRIG does not perform exception evaluation. This option expands the wildcards and generates jobs for each object in the wildcard. This option does not perform object exclusion processing. That is, BMCTRIG does not exclude archived or empty objects from the generated job. However, it does perform REBIND and resize processing, if you specify it.</td>
</tr>
<tr>
<td>Evaluate Exceptions for Automation</td>
<td>This option detects exceptions for objects that exceed thresholds and identifies the associated corrective actions. If you have a license for the Database Performance for DB2 solution, this option registers the corrective actions in the Automation candidate table. You can then execute the corrective actions by using the automation component of the Database Performance solution.</td>
</tr>
<tr>
<td>Resume Generation</td>
<td>This option reads the exceptions table for active exceptions and generates the actions that it identifies in the table. This process allows you to evaluate, view, or modify exceptions, and then generate actions (in separate steps).</td>
</tr>
</tbody>
</table>
The following sections show examples of workflows for the majority of BMCTRIG processing options (Table 57). Before you evaluate any exceptions, you can perform the following steps to define system thresholds:

- Review or set up thresholds (page 489).
- Set up shop-specific exceptions and thresholds for the exceptions (page 491).
- Set up corrective actions (page 496).
- Associate corrective actions with the exceptions (page 496).
- Set priorities for objects and corrective actions (page 488).

**Evaluating exceptions and generating corrective actions**

Figure 152 shows how to set up a job to evaluate exceptions and generate one or more corrective actions to perform maintenance, based on exceptions. For this functionality, use the **Evaluate and Generate** processing option from the BMCTRIG main menu.

**Figure 152  Evaluating exceptions and generating corrective actions**
Evaluating BMCTRIG exceptions

Figure 153 shows how to set up a job to monitor object statistics and report exceptions. For this functionality, use the Evaluate Exceptions processing option from the BMCTRIG main menu.

**Figure 153  Evaluating exceptions**

- **Evaluation**
  - START
  - Decide which objects need to be monitored (page 504).

- **Generation**
  - Set up a BMCTRIG job that evaluates exceptions and generate the job (page 545).
  - Analyze the exceptions report in your job output (page 546) or create an exceptions report (page 617).
  - Take the necessary corrective action (page 553) or use the Resume option (page 571).
Generating jobs without evaluating exceptions

Figure 154 shows how to set up a job that generates maintenance utilities without evaluating exceptions. For this functionality, use the Generate Jobs processing option from the BMCTRIG main menu.

Figure 154 Generating jobs

1. Decide which objects need utility maintenance (page 504).
2. Set up the action that contains the maintenance utilities to generate (page 270).
3. Set up a BMCTRIG job that references the action as the default action to perform (page 504).
4. Generate the BMCTRIG job according to your maintenance schedule (page 504).
Evaluating exceptions for automation

If you have installed the Database Performance for DB2 solution, DASD MANAGER PLUS can register objects and actions to the automation component. The automation component then uses the corrective actions that you have defined. For this functionality, use the **Evaluate Exceptions for Automation** processing option from the BMCTRIG main menu.

**Figure 155** shows how to set up a job that evaluates exceptions for the Database Performance for DB2 automation component.

**Figure 155 Evaluating exceptions for automation**

- **Evaluation**
  - **START**
  - Decide which objects need utility maintenance (page 504).
  - Set up a BMCTRIG job that evaluates exceptions and generate it (page 504).

- **Generation**
  - Generate the BMCTRIG job according to your maintenance schedule (page 504).
  - Analyze the exceptions report in your job output (page 546) or create an Exceptions Report (page 581).
  - Review registered candidates and completed work with the Automation component (Database Performance for DB2).
  - Run the automation component to generate jobs.
Setting exception thresholds

This section describes the thresholds that you can set for detecting reorganization-related, statistical-related, copy-related, and other miscellaneous exceptions. By using the BMCTRIG utility, you can set thresholds to identify changes in BMCSTATS and DB2 RTS statistical values. The thresholds that you use for reporting might differ from the thresholds that you use for generating corrective actions.

Upon encountering an object that initiates an exception, BMCTRIG records the exception in the DASD MANAGER PLUS exceptions table. To determine what (if any) corrective action to take, you should analyze statistical trends and the exceptions report. You can also define your own site-specific exceptions for DASD MANAGER PLUS to monitor. For more information, see “To create a user-defined exception” on page 494. For information about threshold evaluation and BMCTRIG rules, see the DASD MANAGER PLUS for DB2 Reference Manual.

Understanding reorganization-related exceptions

Exceptions in this category identify conditions that indicate the need to reorganize an object, including the following situations:

- The amount of data in an object has changed significantly.
- The data is disorganized.
- Data set fragmentation conditions exist.
- DB2 status conditions exist.

**NOTE**

If you specify TRIGNOREORG Y, BMCTRIG will not analyze these exceptions.

Table 58 lists the reorganization-related exceptions. Figure 177 shows the options on the Reorg-Related Override Exceptions panel.
<table>
<thead>
<tr>
<th>Condition</th>
<th>Exception definition name</th>
<th>How to use as an exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append inserts</td>
<td>APPNDINS(^a)</td>
<td>Use this exception to initiate a reorganization of an index based on the percentage of index entries that have been inserted since the last REORG, REBUILD INDEX, or LOAD REPLACE on the index space or partition that have a key value that is greater than the maximum key value in the index or partition.</td>
</tr>
<tr>
<td>AREO* pending</td>
<td>AREOPEND</td>
<td>Use this exception to initiate a reorganization of an index or a table space based on whether the index or table space is in advisory REORG-pending status (AREO*).</td>
</tr>
<tr>
<td>AREOR pending</td>
<td>ARERPEND</td>
<td>Use this exception to initiate a reorganization of an index or a table space for optimal performance and to apply pending definition changes (AREOR status).</td>
</tr>
<tr>
<td>CARD</td>
<td>CARD</td>
<td>Use this exception to identify unexpected growth of a table space or index based on a percentage increase in the number of rows. This exception shows whether a part has increased by more than the specified percentage of rows since the previous statistics. This exception is effective only if a previous entry exists in the statistical database.</td>
</tr>
<tr>
<td>Dataset Extents</td>
<td>DSEXTENT(^b)</td>
<td>Use this exception to identify data sets within a table space or index that have reached a certain number of extents (evaluated at the data set level).</td>
</tr>
<tr>
<td>Del since reorg</td>
<td>REORDELS(^a)</td>
<td>Use this exception to initiate a reorganization of a table space or index in which there has been a percentage of rows deleted since the last REORG (evaluated at the partition level).</td>
</tr>
<tr>
<td>EXTENTS</td>
<td>EXTENTS</td>
<td>Use this exception to identify table spaces or indexes that have reached a certain number of extents (evaluated at the partition level).</td>
</tr>
<tr>
<td>FARIND</td>
<td>FARIND(^b)</td>
<td>Use this exception to initiate a reorganization of a table space based on the percentage of variable length rows that table updates have caused to be relocated. When an update operation increases the length of a record by more than the amount of space available in the page that stores the record, DB2 moves the record to another page. Until you reorganize the table space, the record needs an additional page reference when you access it. See also TOTALIND.</td>
</tr>
</tbody>
</table>
### Table 58  Reorganization-related exceptions (part 2 of 5)

<table>
<thead>
<tr>
<th>Condition</th>
<th>Exception definition name</th>
<th>How to use as an exception</th>
</tr>
</thead>
</table>
| FAROFF    | FAROFF                    | Use this exception to initiate a reorganization of a table space in which a high percentage of rows is far from their optimal positions due to inserts into a full page. For a clustering index, a high Faroff percentage indicates that clustering might be degrading.  
Note the following additional information:  
■ This exception is evaluated only for clustering indexes.  
■ You must specify INDEX Y or INDEX followed by an index name for this exception to be initiated. |
| Ins since reorg | REORGINS\(^a\) | Use this exception to initiate a reorganization of a table space or index in which there has been a percentage of rows inserted since the last REORG (evaluated at the partition level). |
| LEAFDIST  | LEAFDIST                  | Use this exception to initiate a reorganization of an index based on the distance in page IDs between successive leaf pages during a sequential access of the index.  
You can also use this exception to report on index organization. An increase over time indicates that you should reorganize the index. The value must be between 100 and 99999998. The minimum distribution is 100 times the average number of pages between successive leaf pages. |
| LEAFFAROFF| LEAFFOFF\(^b\)            | Use this exception to initiate a reorganization of an index based on the percentage of leaf pages that are physically far from the previous leaf page (evaluated at the partition level). |
| LEAFTOTOFF| LEAFTOFF\(^b\)            | Use this exception to initiate a reorganization of an index based on the percentage of leaf pages that are not in their optimal positions (evaluated at the partition level). |
| LEVELINC  | LEVELINC\(^b\)            | Use this exception to identify when the number of index levels has increased compared with the previous statistics entry (evaluated at the index partition level if you specify Partlvl Y, otherwise evaluated at the index level). |
## Understanding reorganization-related exceptions

Chapter 8 Analyzing objects by using BMCTRIG

<table>
<thead>
<tr>
<th>Condition</th>
<th>Exception definition name</th>
<th>How to use as an exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVELMIN</td>
<td>LEVELMIN(^b)</td>
<td>Use this exception to initiate a reorganization of an index when current statistics indicate that the number of index levels is greater than the number that are required to reorganize the index (evaluated at the index partition level if you specify Partlvl Y, otherwise evaluated at the index level).</td>
</tr>
<tr>
<td>number of index levels that are greater than the minimum number required</td>
<td></td>
<td>Index levels are a function of the number of table rows and the size of the index key. Inserting or changing keys can create splits that take one full index page and create two pages that are only half full. Eventually, these splits cause an index tree split and create another index level. If you are not sure what value to use, try setting a LevelInc or LevelMin exception.</td>
</tr>
<tr>
<td>LEVELS</td>
<td>LEVELS(^b)</td>
<td>Use this exception to identify reaching a specific number of index levels (evaluated at the index partition level if you specify Partlvl Y, otherwise evaluated at the index level). The default is 99.</td>
</tr>
<tr>
<td>number of index levels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOB Freespace</td>
<td>LOBFRSPC</td>
<td>Use this exception to initiate a reorganization based on the percentage of the LOB that is freespace.</td>
</tr>
<tr>
<td>LOB ORGRATIO</td>
<td>LOBORGR</td>
<td>Use this exception to initiate a reorganization based on the percentage of organization of the table space.</td>
</tr>
<tr>
<td>Mass del reorg</td>
<td>REORMDEL(^a)</td>
<td>Use this exception to initiate a reorganization of a table space or index in which there has been mass deletion since the last REORG (evaluated at the partition level).</td>
</tr>
<tr>
<td>Mods since reorg</td>
<td>REORMODS(^a)</td>
<td>Use this exception to initiate a reorganization of a table space or index in which there is a large percentage of rows or keys modified since the last REORG (evaluated at the partition level).</td>
</tr>
<tr>
<td>percentage of rows modified since last Reorg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pct Dropped Rows</td>
<td>PCTDROP</td>
<td>Use this exception to initiate a reorganization based on the percentage of space in the table space that is occupied by dropped rows. (This exception applies only to simple spaces.)</td>
</tr>
<tr>
<td>percentage of space that dropped rows used</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pct over alloc</td>
<td>REORDSPC(^a)</td>
<td>Use this exception to determine the percentage by which space allocated to a table space or partition exceeds space occupied by row data.</td>
</tr>
</tbody>
</table>
### Understanding reorganization-related exceptions

#### Table 58  Reorganization-related exceptions (part 4 of 5)

<table>
<thead>
<tr>
<th>Condition</th>
<th>Exception definition name</th>
<th>How to use as an exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>PctActivHi</td>
<td>PACTHI</td>
<td>Use this exception for either of the following purposes:</td>
</tr>
<tr>
<td>maximum percentage of active pages</td>
<td></td>
<td>- to initiate a reorganization on a table space containing a high percentage of active pages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- to validate design assumptions by determining how much of the allocated space that the table space is using</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can use this exception to identify table spaces that are running out of space.</td>
</tr>
<tr>
<td>PctActivLo</td>
<td>PACTLO</td>
<td>Use this exception to initiate a reorganization on table spaces containing a low percentage of active pages. You can use this exception to identify table spaces containing a large number of empty pages.</td>
</tr>
<tr>
<td>minimum percentage of active pages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCTCLUST</td>
<td>PCTCLUS</td>
<td>Use this exception to initiate a reorganization of a table space based on the cluster ratio of the clustering index. The ratio is optimal after reorganization. An index is in clustering order if the access of data is in ascending row identifier (RID) sequence.</td>
</tr>
<tr>
<td>cluster ratio of the object</td>
<td></td>
<td>Note the following additional information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- This exception is evaluated only for clustering indexes at the partition level if you specify Partlvl Y, otherwise evaluated at the index level.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- You must specify INDEX Y or INDEX followed by an index name for this exception to be initiated.</td>
</tr>
<tr>
<td>Pseudo Del Key</td>
<td>PSEUDODL</td>
<td>Use this exception to initiate a reorganization of an index based on the percentage of keys that are pseudo-deleted in an index partition. (This exception is evaluated at the partition level.)</td>
</tr>
<tr>
<td>percentage of pseudo-deleted keys</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorg DISORG LOB</td>
<td>REORGLOB</td>
<td>Use this exception to initiate a reorganization of a table space in which a percentage of the LOBs that were inserted since the last REORG or LOAD REPLACE were not perfectly chunked.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorg Leaf</td>
<td>REORLEAF</td>
<td>Use this exception to initiate a reorganization of an index based on a percentage of total pages in comparison to number of active leaf pages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorg Pend</td>
<td>REORPEND</td>
<td>Use this exception to initiate a reorganization of a table space in reorg pending status (REORP). This exception is evaluated at the partition level if you specify Partlvl Y. Otherwise, the exception is evaluated at the object level.</td>
</tr>
<tr>
<td>reorganization pending</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Condition</td>
<td>Exception definition name</td>
<td>How to use as an exception</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REORGSPACE</td>
<td>REORSPACE</td>
<td>Use this exception to initiate a reorganization of a table space or index. The exception is based on the ratio of the percentage of used tracks to the number of tracks that the object would use if it were reorganized. MINSPACE is a synonym for REORSPACE. If the REORGSPACE_KB statistic is less than or equal to zero (indicating that no statistics were collected), the REORSPACE exception is not available.</td>
</tr>
<tr>
<td>SPACE</td>
<td>SPACE</td>
<td>Use this exception to identify unexpected growth of a table space or index based on an increase of used space. This exception works only if a previous entry exists in the statistical database.</td>
</tr>
<tr>
<td>TOTALIND</td>
<td>TOTALIND(^b)</td>
<td>Use this exception to initiate a reorganization of a table space based on the percentage of variable length rows that have been relocated due to table updates. The sum of the columns Farindref and Nearindref is the total number of such relocated rows. See also FARIND.</td>
</tr>
</tbody>
</table>
| TOTALOFF           | TOTALOFF                  | Use this exception to initiate a reorganization of a table space in which table rows are not stored in the same order as the entries in their index. Note the following additional information:  
  - This exception is evaluated only for clustering indexes.  
  - You must specify INDEX Y or INDEX followed by an index name for this exception to be initiated. |
| Unclust inserts    | UNCLUST\(^a\)            | Use this exception to initiate a reorganization of a table space that has a high number of unclustered inserts. |

\(^a\) You must specify DB2 RTS Y or O in the Exception options panel (to indicate that you want to use DB2 RTS for analysis) for this exception to be evaluated.

\(^b\) This exception can be evaluated using DB2 RTS or BMC statistics. You must specify DB2 RTS Y or O in the Exception options panel to have BMCTRIG evaluate the exception using DB2 RTS data. For more information, see the DASD MANAGER PLUS for DB2 Reference Manual.
Understanding statistics-related exceptions

Exceptions in this category can be used to identify conditions that indicate the need to collect statistics and potentially update the DB2 catalog for an object. This includes situations where the existing statistics are out of date due to a reorg, rebuild, or copy of an object, and situations where the object data has changed significantly.

**NOTE**

If you specify TRIGNOSTATS Y, BMCTRIG will not analyze these exceptions.

Table 59 lists the statistics-related exceptions. Figure 178 shows the options on the Statistics-Related and Misc. Override Exceptions panel.

**Table 59  Statistics-related exceptions (part 1 of 2)**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Exception definition name</th>
<th>How to use as an exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCSTATS Age</td>
<td>BSTATAGE</td>
<td>Use this exception to identify the number of days since you last collected statistics and saved them in the DASD MANAGER PLUS statistics tables for this object (evaluated at the partition level).</td>
</tr>
<tr>
<td>number of days since you collected statistics</td>
<td>BSTATAGE</td>
<td>Use this exception to identify the number of days since you last collected statistics and saved them in the DASD MANAGER PLUS statistics tables for this object (evaluated at the partition level).</td>
</tr>
<tr>
<td>Catalog Upd Age</td>
<td>CSTATAGE&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Use this exception to identify the number of days since the product updated statistics for this object in the DB2 catalog (evaluated at the partition level).</td>
</tr>
<tr>
<td>number of days since the statistics were updated in the DB2 catalog</td>
<td>CSTATAGE&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Use this exception to identify the number of days since the product updated statistics for this object in the DB2 catalog (evaluated at the partition level).</td>
</tr>
<tr>
<td>Load after stats</td>
<td>LOADSTAT&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Use this exception to identify tables spaces or indexes in which there has been a LOAD REPLACE after the last time statistics were collected (evaluated at the partition level).</td>
</tr>
<tr>
<td>Mass del stats</td>
<td>STATMDEL&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Use this exception to identify tables spaces or indexes in which there has been mass deletion since the last time statistics were collected (evaluated at the partition level).</td>
</tr>
<tr>
<td>Mods since stats</td>
<td>STATMODS&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Use this exception to initiate a reorganization of a table space or index in which there is a large percentage of rows or keys modified since the last time statistics were collected (evaluated at the partition level).</td>
</tr>
<tr>
<td>percentage of rows modified since last stats</td>
<td>STATMODS&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Use this exception to initiate a reorganization of a table space or index in which there is a large percentage of rows or keys modified since the last time statistics were collected (evaluated at the partition level).</td>
</tr>
<tr>
<td>NO RTS stats dat</td>
<td>NORTSSTAT&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Use this exception to identify objects for which no DB2 RTS RUNSTATS statistics exist.</td>
</tr>
<tr>
<td>objects for which no DB2 RTS RUNSTATS statistics are detected</td>
<td>NORTSSTAT&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Use this exception to identify objects for which no DB2 RTS RUNSTATS statistics exist.</td>
</tr>
</tbody>
</table>
Understanding copy-related exceptions

Exceptions in this category can be used to identify conditions that indicate the need to copy an object. This includes situations where the existing COPY is out of date due to a reorg, rebuild, or load utility, situations where the object data has changed significantly, and DB2 status conditions.

**NOTE**

If you specify TRIGNOSTATS Y, BMCTRIG will not analyze these exceptions.

Table 60 lists the Copy-related exceptions. Figure 179 shows the options on the Copy-Related Override Exceptions panel.
### Table 60  Copy-related exceptions (part 1 of 2)

<table>
<thead>
<tr>
<th>Condition</th>
<th>Exception definition name</th>
<th>How to use as an exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy Pend</td>
<td>COPYPEND</td>
<td>Use this exception to initiate an image copy on table spaces in copy pending status. This exception is evaluated at the partition level if you specify <code>Partlvl Y</code>. Otherwise, the exception is evaluated at the object level.</td>
</tr>
<tr>
<td>Dirty Pages (IX Dirty Pages)</td>
<td>IXDIRTY\textsuperscript{b}</td>
<td>Use this exception to initiate a copy of an index based on the percentage of index pages modified since the last image copy.</td>
</tr>
<tr>
<td>Dirty Pages (TS Dirty Pages)</td>
<td>DIRTY\textsuperscript{a}</td>
<td>Use this exception to initiate a full image copy based on the percentage of modified pages.</td>
</tr>
<tr>
<td>DtyIncremt</td>
<td>DIRTYIC\textsuperscript{a}</td>
<td>Use this exception with the DIRTY exception to initiate an incremental copy on objects based on the percentage of modified pages.</td>
</tr>
<tr>
<td>Full Copy Age</td>
<td>FICAGE\textsuperscript{a}</td>
<td>Use this exception to initiate a full image copy of this object (evaluated at the partition level).</td>
</tr>
<tr>
<td>Image Copy Age (IX Copy Age)</td>
<td>IXICAGE\textsuperscript{a}</td>
<td>Use this exception to initiate a copy of an index based on the number of days since the last image copy of this object (evaluated at the partition level).</td>
</tr>
<tr>
<td>Image Copy Age (TS Copy Age)</td>
<td>COPYAGE\textsuperscript{a}</td>
<td>Use this exception to initiate a full or an incremental image copy of this object if a copy has not been taken in more than the specified number of days (evaluated at the partition level).</td>
</tr>
<tr>
<td>Load after copy</td>
<td>LOADCOPY\textsuperscript{b}</td>
<td>Use this exception to identify table spaces or indexes in which there has been a LOAD REPLACE after the last copy (evaluated at the partition level).</td>
</tr>
<tr>
<td>Mods since copy</td>
<td>COPYMODS\textsuperscript{b}</td>
<td>Use this exception to identify the percentage of rows or keys that have been modified since the last copy was made (evaluated at the partition level).</td>
</tr>
<tr>
<td>No RTS copy data</td>
<td>NORTSCPY\textsuperscript{b}</td>
<td>Use this exception to identify objects for which no DB2 RTS image copy statistics exist (evaluated at the partition level).</td>
</tr>
</tbody>
</table>
Setting thresholds in BMCTRIG syntax

Before you begin

To set the thresholds by using the BMCTRIG syntax, you must have already performed the following steps:

1. Create an action that contains BMCTRIG as instructed in “Creating an action” on page 271.

2. Create syntax for BMCTRIG to use with the BMCTRIG action as instructed in “Creating an action by copying an existing action” on page 273.

3. Specify threshold values for the exceptions that you want to monitor.

DASD MANAGER PLUS displays panels for the exceptions for you to enter threshold values. When you generate the BMCTRIG action, the syntax indicates the exceptions and thresholds that you specified. Unless you set the SYSTRIGS option to F (FORCE) in the installation options module, BMCTRIG uses the thresholds that the syntax specifies for the objects that it processes.

Table 60  Copy-related exceptions (part 2 of 2)

<table>
<thead>
<tr>
<th>Condition</th>
<th>Exception definition name</th>
<th>How to use as an exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumIncrnt</td>
<td>COPY#IIIC</td>
<td>Use this exception to initiate a full image copy when you have made a specific number of incremental copies or more.</td>
</tr>
<tr>
<td>Rebd after copy</td>
<td>REBDCOPYb</td>
<td>Use this exception to initiate a copy of an index based on whether a REBUILD occurred after the last copy.</td>
</tr>
<tr>
<td>Reorg after copy</td>
<td>REORCOPYb</td>
<td>Use this exception to initiate a copy of a table space or index in which there has been a REORG after the last copy (evaluated at the partition level).</td>
</tr>
</tbody>
</table>

a  This exception can be evaluated using DB2 RTS or BMC statistics. You must specify DB2 RTS Y or O in the Exception options panel to have BMCTRIG evaluate the exception using DB2 RTS data. For more information, see the DASD MANAGER PLUS for DB2 Reference Manual.

b  You must specify DB 2RTS Y or O in the Exception options panel (to indicate that you want to use DB2 real time statistics for analysis) for this exception to be evaluated.
If the SYSTRIGS option is Y in the installation options module or SYSTEMTRIGGERS is Y, BMCTRIG also evaluates the exceptions that you have defined for the subsystem that you have not specified in the syntax. In this situation, the thresholds that you specify in the syntax override the thresholds in the system triggers.

**To set thresholds in syntax from the main BMCTRIG panel**

1. Access the main BMCTRIG panel as follows:

   A. Create an action for BMCTRIG or select the one that you have already defined as instructed in “Creating an action” on page 271.

   B. Edit the BMCTRIG service as instructed in “Creating an action by copying an existing action” on page 273.

   The main BMCTRIG panel appears (Figure 156).

   **NOTE**

   If you plan to analyze DB2 RTS exceptions, change the DB2 RTS option to Y or O in the What to Analyze section of the BMCTRIG Service Syntax panel.
Figure 156  BMCTRIG main panel

<table>
<thead>
<tr>
<th>Command</th>
<th>BMCTRIG TABLESPACE QZUD40.%</th>
<th>More: +</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Syntax: BMCTRIG.TRIGDEMO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enter data, then press end.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Processing Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select . . . . 1</td>
</tr>
<tr>
<td>1. Evaluate and Generate</td>
</tr>
<tr>
<td>2. Evaluate Exceptions</td>
</tr>
<tr>
<td>3. Generate Jobs</td>
</tr>
<tr>
<td>4. Evaluate Exceptions for Automation</td>
</tr>
<tr>
<td>5. Resume Generation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What to Analyze</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Triggers: (Y/N - use exceptions and actions in DASD tables)</td>
</tr>
<tr>
<td>DB2 RTS . . . . N (Y/N/O - Y-Use RTS, N-Ignore RTS, O-Use Only RTS)</td>
</tr>
<tr>
<td>Bypass Exceptions:</td>
</tr>
<tr>
<td>Reorg Exceptions (Y/N - Bypass analysis of Reorg-Related Exceptions)</td>
</tr>
<tr>
<td>Stats Exceptions (Y/N - Bypass analysis of Stats-Related Exceptions)</td>
</tr>
<tr>
<td>Copy Exceptions (Y/N - Bypass analysis of Copy-Related Exceptions)</td>
</tr>
<tr>
<td>Indexes . . . . Y (Y/N - Include all indexes in tablespace)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Saving and Reporting Exception Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save . . . . Y (Y/N - Save Exceptions in Exceptions table)</td>
</tr>
<tr>
<td>DELETEAGE . . . 32767 (Delete Exceptions older than this many days)</td>
</tr>
<tr>
<td>REPORT . . . . Y (Y/N - Print Report on objects with exceptions)</td>
</tr>
<tr>
<td>FKEYOMSG . . . . Y (Y/N - Allow notification of empty indexes)</td>
</tr>
<tr>
<td>NOTIFY . . . . . (Notify this User id on Exceptions)</td>
</tr>
<tr>
<td>MSGLEVEL . . . 0 (0/1 0-Normal msgs, 1-Additional msgs)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Eliminate Duplicates and Index Escalation Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX Escalate . . . Y (Y/N/A - Escalate IX's to TS's during generation)</td>
</tr>
<tr>
<td>IX Copy . . . . N (Y/N - Copy indexes)</td>
</tr>
<tr>
<td>Remove dup actions Y (Y/N - Eliminate duplicate actions)</td>
</tr>
<tr>
<td>Detect inline opts Y (Y/N - Eliminate actions based on inline options)</td>
</tr>
</tbody>
</table>

2 Scroll down until **Override System Trigger Exceptions** appears (Figure 157).
Defining system-level thresholds

System thresholds provide a central point for administering the exceptions and thresholds that you monitor in your subsystem. BMCTRIG applies system-level thresholds if you set the SYSTEMTRIGGERS option to Y in syntax, or if the SYSTRIGS option in the installation options module is Y or F (FORCE). For more information about the installation options module, see Chapter 3, “Setting up DASD MANAGER PLUS.”
To access system thresholds

1 On the DASD MANAGER PLUS main menu, select Thresholds, Corrective Actions, and Priorities.

The System Thresholds and Corrective Actions panel appears (Figure 158).

Figure 158  Exception thresholds on System Thresholds and Corrective Actions panel

2 Select Exception thresholds.

The Exception Thresholds panel appears (Figure 159).

Figure 159  Exception Thresholds panel
Defining system-level thresholds

The Exception Thresholds panel displays all of the thresholds that are defined for your subsystem. In the **Sort by** field, you can choose to sort the thresholds either by exception or object.

From this panel, you can perform full maintenance on the system thresholds by completing any of the following procedures:

- add a threshold ([page 491](#))
- edit an existing threshold ([page 492](#))
- delete a threshold ([page 492](#))
- view the long description of the exception for a threshold ([page 493](#))
- edit the exception definition for a threshold ([page 493](#))

Before proceeding, note that BMCTRIG applies rules in the order shown in **Table 61** when determining which system-level thresholds to apply to each object:

**Table 61 Applying system-level thresholds to objects**

<table>
<thead>
<tr>
<th>Qualifier</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>equal</td>
<td>equal</td>
</tr>
<tr>
<td>2</td>
<td>set owner</td>
<td>set name</td>
</tr>
<tr>
<td>3</td>
<td>equal</td>
<td>wildcard match</td>
</tr>
<tr>
<td>4</td>
<td>wildcard match</td>
<td>equal</td>
</tr>
<tr>
<td>5</td>
<td>wildcard match</td>
<td>equal</td>
</tr>
</tbody>
</table>


* The object is within the expanded object set results when resolving it for both table spaces and indexes.

The Qualifier, Name, Type, and Exception name identify the keys that BMCTRIG uses to retrieve thresholds for the object for a particular exception. For more information about applying thresholds and corrective actions, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

**NOTE**

An exclude specification within an object set for a threshold or corrective action definition does not imply that the object is excluded from BMCTRIG evaluation. BMCTRIG expands the object sets for thresholds and corrective action definitions to determine which objects are in a result set. Due to precedence, BMCTRIG might evaluate the object later because of a more general wildcard threshold or corrective actions definition. Use one of the following methods to ensure that an object is excluded from evaluation when using system triggers:

- Always run BMCTRIG against an object set. That is, use multiple trigger jobs.
- Code a specific threshold for that object, using a high value as the trigger value.

For more information about object sets, see Chapter 5, “Using object sets.”
Defining system-level thresholds

To add a new threshold

1 If you know the exception name, add a new threshold as follows. (If you do not know the exception name, go to step 2 on page 491.)
   
   **A** Type I in the Act field of a threshold entry that has the exception name for which you want to set a threshold.

   **B** Press Enter.

   The Edit Exception Threshold pop-up panel appears (Figure 160). DASD MANAGER PLUS automatically enters the exception name and description.

   **C** Go to step 3 on page 491.

2 If you do not know the name of the exception that you want to add, proceed as follows:

   **A** Enter ? in the Act field next to any threshold.

   **B** Select an exception from the Exceptions Definition List.

   The Edit Threshold pop-up panel appears (Figure 160). DASD MANAGER PLUS automatically enters the exception name and description.

Figure 160 Edit Exception Threshold panel

<table>
<thead>
<tr>
<th>Exception name: DSEXTENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description: MAXIMUM NUMBER OF EXTENTS PER DATASET</td>
</tr>
<tr>
<td>Threshold: 75 (Range=Low Value 1 High Value 7257)</td>
</tr>
<tr>
<td>Priority: 100 (0-255 to indicate the priority)</td>
</tr>
<tr>
<td>Enabled: Y (Y/N to indicate whether exception is enabled)</td>
</tr>
<tr>
<td>Object Pattern: %.%</td>
</tr>
<tr>
<td>Pattern type: TS (TS, IX, or OS)</td>
</tr>
<tr>
<td>Press END to update or CANCEL to exit without saving.</td>
</tr>
</tbody>
</table>

3 Complete the panel as follows:

   - In the **Threshold** field, enter a threshold within the valid range that the panel shows.

   - *(optional)* In the **Priority** field, set a priority for the threshold. (0 is the lowest priority, and 255 is the highest priority.)
Defining system-level thresholds

- In the **Enabled** field, indicate whether the threshold is currently enabled for analysis. BMCTTRIG uses only enabled thresholds.

- **Object Pattern** indicates the objects to which the threshold applies.

- In the **Pattern Type** field, specify the type of object (TS, IX, or OS). Use the object type OS to specify an object set name.

4 Press END to save the threshold and return to the previous panel.

**To edit an existing threshold**

1 On the Exception Thresholds panel (Figure 159), type **E** in the **Act** field next to the threshold that you want to edit.

The Edit Threshold pop-up panel appears (Figure 160).

2 You can edit the threshold as follows:

   - In the **Threshold** field, enter a threshold within the valid range that the panel shows.

   - *(optional)* In the **Priority** field, set a priority for the threshold. (0 is the lowest priority, and 255 is the highest priority.)

   - In the **Enabled** field, indicate whether the threshold is currently enabled for analysis. BMCTTRIG uses only enabled thresholds.

   - **Object Pattern** indicates the objects to which the threshold applies.

   - In the **Pattern Type** field, specify the type of object (TS, IX, or OS). Use the object type OS to specify an object set name.

3 For a user-defined exception, the threshold is the name of a REXX program. For more information about user-defined exceptions, see “Creating user-defined exceptions” on page 494.

4 Press END to save your changes and return to the previous panel.

**To delete a threshold**

1 On the Exception Thresholds panel (Figure 159), type **D** in the **Act** field next to the threshold that you want to delete.

   The Delete confirmation pop-up panel appears.

2 Type **Y** in the confirmation field and press **Enter**.
To view the long description of the exception for a threshold

1 On the Exception Thresholds panel (Figure 159), type T in the Act field next to the threshold.

2 View the description.

   For user-defined exceptions, you can also update the long description.

3 Press END to return to the previous panel.

To edit the exception definition for a threshold

1 On the Exception Thresholds panel (Figure 159), type X in the Act field next to the threshold.

   The Edit Exception pop-up panel appears (Figure 161).

Figure 161  Edit Exception panel

DEAE ------------------------ Edit Exception -----------------------
COMMAND ===> Exception name: DSEXTENT
Description:   MAXIMUM NUMBER OF EXTENTS PER DATASET
Type:           REORG
Applies to TS:  Y
Applies to IX:  Y
Aging
Increment:      25  (0-255 to increment priority per day)

Press END to update or CANCEL to exit without saving.

2 Change the exceptions as necessary.

   For exceptions that BMC Software supplies, you can change the priority aging increment. The value is the number that BMCTRIG adds to the exception priority for each day that the exception remains active in the exceptions table. For user-defined exceptions, you can also update the other fields.

3 Press END to save your changes and return to the previous panel.
Creating user-defined exceptions

A user-defined exception consists of a name and one or more REXX programs that are used to evaluate exceptions.

Before you use a user-defined exception in BMCTRIG, however, you must create one or more REXX programs that will evaluate the thresholds for the exception. Also, you must ensure that the SYSEXEC DD that BMCTRIG uses refers to the data set that contains the REXX program. You set the value for SYSEXEC in the JCL Generation POF. For more information, see Chapter 3, “Setting up DASD MANAGER PLUS.” For more information and samples of the REXX program requirements for user-defined exceptions, see the DASD MANAGER PLUS for DB2 Reference Manual.

NOTE
You must code the SYSTSPRT DD as a SYSOUT file if SYSTRIGGERS is Y and you have specified user-defined exceptions. The REXX driver always writes out at least two lines to the SYSTSPRT DD for every REXX program executed. If the REXX program executes a SAY instruction, then the output goes to the SYSTSPRT DD. If BMCTRIG executes under AEXEMAIN, then you must specify SYSTSPRT DD SYSOUT.

To create a user-defined exception

1 On the DASD MANAGER PLUS main menu, select Thresholds, Corrective Actions, and Priorities.

The System Thresholds and Corrective Actions panel appears (Figure 162).

Figure 162 Exception definitions on System Thresholds and Corrective Actions panel

Use the following options to define the exception condition thresholds and the corrective actions for use within the subsystem. You may specify priorities for executing utility actions against objects. You may also view and edit active exceptions.

Option: 1
1. Exception definitions
2. Exception thresholds
3. Corrective actions
4. Object-action priorities
5. Active exceptions
2. Select Exception definitions.

The Exception Definitions List panel appears (Figure 163).

**Figure 163  Exception Definitions List panel**

<table>
<thead>
<tr>
<th>Object</th>
<th>Aging</th>
<th>Exception</th>
<th>Type</th>
<th>Type</th>
<th>Increment</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSTATAGE</td>
<td>STATS</td>
<td>%</td>
<td>%</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CARD</td>
<td>REORG</td>
<td>%</td>
<td>%</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>COPY#IIC</td>
<td>COPY</td>
<td>TS</td>
<td>TS</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>COPYAGE</td>
<td>COPY</td>
<td>TS</td>
<td>TS</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>COPYPEND</td>
<td>COPY</td>
<td>TS</td>
<td>TS</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

3. Add a new exception by typing I in the Act field next to an exception.

The Edit User-defined Exception panel appears (Figure 164).

**Figure 164  Edit User-defined Exception panel**

<table>
<thead>
<tr>
<th>Exception name: RTS-COPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description: Analyze Realtime Stats for Copy Exceptions</td>
</tr>
<tr>
<td>Type: COPY</td>
</tr>
<tr>
<td>Applies to TS: Y (Y/N applies to table spaces)</td>
</tr>
<tr>
<td>Applies to IX: Y (Y/N applies to indexes)</td>
</tr>
<tr>
<td>Aging Increment: 25 (0-255 to increment priority per day)</td>
</tr>
</tbody>
</table>

4. Enter a unique name for the exception, keeping in mind the following considerations:

- BMCTTRIG uses the name for reporting the exception.
- You should specify whether the exception applies to table spaces, indexes, or both.
Identifying threshold values

- **(optional)** You can specify the aging priority (the same as you would for exceptions that BMC Software supplies).

5 Press END to save your changes and return to the previous panel.

Identifying threshold values

Use either of the following methods to identify the appropriate threshold values for objects:

- Run BMCSTATS on all significant objects, and study the statistics reports to learn the current status of the objects. Chapter 7, “Analyzing statistical trends,” describes the BMCSTATS panels.

- Run a benchmark application before and after a reorganization. Next, display BMCSTATS to monitor the status of objects. Then, run the benchmark application periodically, and compare the statistics to determine the appropriate values for initiating a reorganization.

Managing corrective actions

When BMCTRIG encounters exceptions, it can generate and (optionally) submit any of the utilities, commands (Table 28), or user-defined services. For example, you can set up an action that contains a BMCCOPY service and define it as the corrective action for the DIRTY exception. You can also create an action that contains a BMCREORG service and define it as the corrective action for the DSEXTENTS exception. When BMCTRIG identifies objects that exceed the DIRTY or DSEXTENTS thresholds, it generates the corrective action that is associated with that specific exception. You can also specify default actions for exceptions that do not have corrective actions that are associated with them.

Setting a corrective action in BMCTRIG syntax

You can specify a default corrective action by using the BMCTRIG syntax.

**To set the default corrective action in the syntax**

1 Create an action that contains BMCTRIG.

2 Create the service syntax for BMCTRIG to use with the BMCTRIG action.
Defining system-level corrective actions

1. Specify the name of a corrective action (such as a reorganization) that you have created on the JCL Generation Options panel (at Default Action) as follows:

- If you are not using system-level triggers (SYSTEMTRIGGER Y or installation option SYSTRIGS Y or F), you must specify an action name in the BMCTRIG syntax in order to generate a corrective action.

- If you are using system-level triggers, the DASD MANAGER PLUS tables contain definitions for the corrective actions. However, you can also specify an action name in the BMCTRIG syntax to use when the system tables do not provide a definition for a corrective action.

### Defining system-level corrective actions

System-level corrective actions provide a central point for administering the exceptions and corrective actions that you monitor in your subsystem. BMCTRIG applies system-level corrective actions if you specify the SYSTEMTRIGGERS Y option in the BMCTRIG syntax or if the SYSTRIGS option in the installation options module is set to Y or F.

You can perform the following tasks for corrective actions:

- access corrective actions (page 498)
- add a corrective action (page 499)
- edit an existing corrective action (page 499)
- delete a corrective action (page 500)
To access system corrective actions

1 On the DASD MANAGER PLUS main menu, select Thresholds, Corrective Actions, and Priorities.

The System Thresholds and Corrective Actions panel appears (Figure 165).

Figure 165 Corrective actions on System Thresholds and Corrective Actions panel

To access system corrective actions

2 Select Corrective actions.

The Corrective Actions panel appears (Figure 166).

Figure 166 Corrective Actions panel

The panel displays all of the corrective actions that are defined for your subsystem. In the Sort by field, you can choose to sort the corrective actions by exception, object, or by corrective action. From this panel you can perform full maintenance on the corrective actions in the system.
To add a new corrective action

1. On the Corrective Action panel (Figure 166), type I in the Act field of a corrective action entry, and press Enter.

The Edit Corrective Action panel appears (Figure 167).

Figure 167  Edit Corrective Action panel

```
DEAE ------------------ Edit Corrective Action ------------------
COMMAND ===> 
Enter ? on Exception or Corrective Action for a selection list.
Press END to save changes or CANCEL to exit without saving.

Exception: 
Corrective Action: 
Object Type:     TS     (TS, IX, or OS) 
Object Pattern: 
```

2. If you know the names of the exception and corrective action, type them in the appropriate fields. To select from a list, enter ? in the field.

3. Keep in mind the following items:
   - You must specify the type of object to which the corrective action applies (TS, IX, or OS). Use OS to name an object set.
   - You can enter an object name or a wildcard pattern.
   - BMCTRIG applies the hierarchy shown in Table 61 when locating objects to evaluate system-level thresholds.

The Qualifier, Name, Type, and Exception name identify the key that BMCTRIG uses to retrieve corrective actions for the object for a particular exception. For more information about applying thresholds and corrective actions, see the DASD MANAGER PLUS for DB2 Reference Manual.

4. Press END to save your changes and return to the previous panel.

To edit an existing corrective action

1. On the Corrective Action panel (Figure 166), type E in the Act field next to the corrective action that you want to edit.

The Edit Corrective Action pop-up panel appears (Figure 167).

2. Edit the corrective actions as necessary.
3  Press END to save your changes and return to the previous panel.

**To delete a corrective action**

1  On the Corrective Action panel (Figure 166), type D in the Act field next to the corrective action that you want to delete.

   The Delete confirmation pop-up panel appears.

2  Type Y in the confirmation field and press Enter.

**Defining system-level object-action priorities**

System-level object-action priorities provide a central point for administering the priorities for corrective actions within your subsystem. Each object-action has a priority. Work in a worklist is ordered by object-action priority. BMCTRIG applies system-level corrective actions, thresholds, and priorities if you specify the SYSTEMTRIGGERS Y option in the BMCTRIG syntax, or if the SYSTRIGS option in the installation options module is set to Y or F.

You can perform the following tasks for object-action priorities:

- access object-action priorities (page 501)
- add an object-action priority (page 502)
- edit an existing object-action priority (page 503)
- delete an object-action priority (page 503)
To access system object-action priorities

1 On the DASD MANAGER PLUS main menu, select Thresholds, Corrective Actions, and Priorities.

The System Thresholds and Corrective Actions panel appears (Figure 168).

Figure 168 Object-action priorities on System Thresholds and Corrective Actions panel

2 Select Object-action priorities.

The Object-Action Priorities panel appears (Figure 169).

Figure 169 Object-Action Priorities panel
Defining system-level object-action priorities

NOTE
You will see a plus sign (+) after a long name that has been truncated. You can view or modify the full name by typing I or E.

The Object-Action Priorities panel shows all of the priorities that are defined for actions on specific objects or object name patterns that are defined for your subsystem. You can use this panel to create or edit object-action priorities. BMCTRIG orders or controls the work generated based on a priority limit. From this panel, you can perform full maintenance on the object-action priorities in the system.

BMCTRIG uses the hierarchy shown in Table 62 when applying priority.

Table 62 Applying object-action priority

<table>
<thead>
<tr>
<th>Qualifier</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>equal</td>
<td>equal</td>
</tr>
<tr>
<td>2</td>
<td>equal</td>
<td>equal</td>
</tr>
<tr>
<td>3</td>
<td>wildcard match</td>
<td>equal</td>
</tr>
<tr>
<td>4</td>
<td>wildcard match</td>
<td>equal</td>
</tr>
</tbody>
</table>

The Qualifier, Name, Type, and Action name identify the key that BMCTRIG uses to retrieve object-action priorities for the object for a particular action. For more information, see the DASD MANAGER PLUS for DB2 Reference Manual.

To add a new object-action priority

1 On the Object-Action Priorities panel (Figure 169), type I in the Act field of an object-action priority, and press Enter.

The Edit Object-Action pop-up panel appears (Figure 170).

Figure 170 Edit Object-Action panel

DECA ---------------------- Edit Object-Action ----------------------
COMMAND =>
Enter ? in the Action field for a selection list.
Press END to update or CANCEL to exit without updating.

Object Pattern:
Object Type: TS (TS or IX)
Action: (may use wildcards)
Priority: 0 (0-255)
Part Number: 0 (0-4096)

2 Enter the object name or pattern for which the priority will apply and the object type.
3 If you know the action name, enter it in the Act field. To select from a list, enter ? in the field.

4 Enter the priority.

5 (optional) Enter a partition number to limit the priority to a specific partition.

--- NOTE ---
The partition values range from 0 through 4096.

6 Press END to save your changes and return to the previous panel.

To edit an existing object-action priority

1 On the Object-Action Priority panel (Figure 169), type E in the Act field next to the object-action priority that you want to edit.

The Edit Object-Action Priorities pop-up panel appears (Figure 170).

2 Edit the object-action priorities as necessary.

3 Press END to save your changes and return to the previous panel.

--- NOTE ---
You cannot edit an object-action priority that is required by BMC.

To delete an object-action priority

1 On Object-Action Priority panel (Figure 169), type D in the Act field next to the object-action priority that you want to delete.

The Delete confirmation pop-up panel appears.

2 Type Y in the confirmation field and press Enter.

--- NOTE ---
You cannot delete an object-action priority that is required by BMC.
Setting up a BMCTRIG job

Use these procedures as guidelines for creating a BMCTRIG job to detect exceptions. The procedures in this section show how to include BMCSTATS and BMCTRIG in the same action in worklist JCL format.

If you plan to analyze any exceptions from the DASD MANAGER PLUS statistics database, run the BMCSTATS utility before the BMCTRIG utility to ensure that you are analyzing current statistics. If you do not plan to analyze any BMCSTATS specific exceptions, this step is not necessary. For more information about the analysis of various thresholds, see the BMCTRIG rules in the DASD MANAGER PLUS for DB2 Reference Manual.

To specify which objects to monitor with the BMCTRIG job

1. Set up an object set for complex object requirements, or use a simple pattern for the service. (See Chapter 5, “Using object sets.”)

2. Identify which exceptions to monitor (by using system triggers or overrides).

   To set thresholds that apply to both table spaces and indexes, see “Setting thresholds in BMCTRIG syntax” on page 485.

3. Identify which actions to generate or specify a default action in the BMCTRIG syntax. (To generate actions that differ based on exceptions, use system triggers.)

Creating an action with BMCTRIG

You set up a BMCTRIG action to generate JCL to run BMCTRIG the same way as you set up other actions in DASD MANAGER PLUS. The following procedures use an example action named EXCEPT1.

To create an action with BMCTRIG

1. Create an action as instructed in “Creating an action” on page 271.

2. Add the service BMCSTATS in the action (before BMCTRIG so that your BMC Software statistics are current) as instructed in “Creating an action by copying an existing action” on page 273.

   If you edit the BMCSTATS parameters, be sure to specify SaveStats Y (the default value).

3. Add the service BMCTRIG.
4 Use the same object name pattern or object set for both the BMCSTATS service and the BMCTRIG service.

5 For the BMCSTATS and BMCTRIG services, either select an existing service syntax for each or create new ones.

When viewing the action that you created, the Edit Action Services panel should look similar to Figure 171.

**Figure 171  Example of Edit Action Services panel with BMCTRIG**

```
ASUUAEDS -------- Edit Action Services (for EXCEPT1) ------ Row 1 to 2 of 2
COMMAND ===>                                                  SCROLL ===> CSR

Please select an option and press Enter.

I = Insert a blank step         L = Duplicate a step
A/B = Copy After or Before      D = Delete step
E/S = Edit syntax and options   M = Manage Object Sets   P = Preview
? = Insert from selection list  U = Update Service       Z = Zoom object Name

JCL Options DSN . .

<table>
<thead>
<tr>
<th>Act Service</th>
<th>Object Name</th>
<th>Object Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCSTATS</td>
<td>TZU%.%</td>
<td>TS</td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>TZU%.%</td>
<td>TS</td>
</tr>
</tbody>
</table>

******************************* Bottom of data ****************************
```

6 To view the syntax options for the BMCTRIG step, type E in the Act field next to the BMCTRIG service.

The Service Syntax List for BMCTRIG appears. If you have already selected syntax, the name appears in the Current Syntax field.

7 To view or edit the syntax, type E in the Act field next to it.

8 If you have not selected syntax for this BMCTRIG service and you want to create new syntax, perform the following actions:

A Type I in the Act field next to any entry.

B Enter information in the pop-up panel, and then press Enter to continue.
Creating an action with BMCTRIG

The BMCTRIG main panel appears (Figure 172).

**Figure 172  BMCTRIG main panel for example job EXCEPT1**

<table>
<thead>
<tr>
<th>Command</th>
<th>BMCTRIG TABLESPACE TZU%.%</th>
<th>Scroll</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEGG</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Service Syntax: BMCTRIG.TRIGDEMO

Enter data, then press end. More: +

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Select . . . . . . 1</td>
<td>1. Evaluate and Generate</td>
</tr>
<tr>
<td></td>
<td>2. Evaluate Exceptions</td>
</tr>
<tr>
<td></td>
<td>3. Generate Jobs</td>
</tr>
<tr>
<td></td>
<td>4. Evaluate Exceptions for Automation</td>
</tr>
<tr>
<td></td>
<td>5. Resume Generation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>System Triggers Y</td>
<td>(Y/N - use exceptions and actions in DASD tables)</td>
</tr>
<tr>
<td>DB2 RTS . . . . . Y</td>
<td>(Y/N/O - Y-Use RTS, N-Ignore RTS, O-Use Only RTS)</td>
</tr>
<tr>
<td>Bypass Exceptions:</td>
<td></td>
</tr>
<tr>
<td>Reorg Exceptions N</td>
<td>(Y/N - Bypass analysis of Reorg-Related Exceptions)</td>
</tr>
<tr>
<td>Stats Exceptions N</td>
<td>(Y/N - Bypass analysis of Stats-Related Exceptions)</td>
</tr>
<tr>
<td>Copy Exceptions N</td>
<td>(Y/N - Bypass analysis of Copy-Related Exceptions)</td>
</tr>
<tr>
<td>Indexes . . . . . Y</td>
<td>(Y/N - Include all indexes in tablespace)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Save . . . . . Y</td>
<td>(Y/N - Save Exceptions in Exceptions table)</td>
</tr>
<tr>
<td>DELETEAGE . . . 32767</td>
<td>(Delete Exceptions older than this many days)</td>
</tr>
<tr>
<td>REPORT . . . . Y</td>
<td>(Y/N - Print Report on objects with exceptions)</td>
</tr>
<tr>
<td>FKEYOMSG . . . . Y</td>
<td>(Y/N - Allow notification of empty indexes)</td>
</tr>
<tr>
<td>NOTIFY . . . . . .</td>
<td>(Notify this User id on Exceptions)</td>
</tr>
<tr>
<td>MSGLEVEL . . . . 0</td>
<td>(0/1 0-Normal msgs, 1-Additional msgs)</td>
</tr>
</tbody>
</table>
Specifying BMCTRIG syntax options

This section describes how to specify BMCTRIG syntax options. The options are displayed on a single, scrollable dialog and are divided into sections by category and labeled with subheadings. The BMCTRIG options subheadings are:

- Processing options
- What to Analyze options
- Saving and Reporting Exception options
- Eliminate Duplicates and Index Escalation options
- Exclude Objects from Generation options
- Corrective Action Generation options
- Rebind and Resize options
- Job and JCL Generation options
- Reorg-Related Override Exceptions
- Statistics-Related Override Exceptions
- Copy-Related Override Exceptions

Selecting the Processing options

For the action with BMCTRIG that you created, select one of the following processing options from the BMCTRIG menu (Figure 172):

- Evaluate and Generate
- Evaluate Exceptions
- Generate Jobs
- Evaluate Exceptions for Automation
- Resume Generation

For more information about these options, see Table 57.

Specifying What to Analyze options

These options specify what trigger analysis uses during the evaluation process. The options answer questions such as whether System Trigger definitions should be used to determine the conditions to analyze. They indicate whether DB2 RTS data should be evaluated for specific exceptions. The options also give you the ability to suppress exceptions from evaluation based on categories such as REORG, STATISTICS, and COPY. An example of these options may be found in Figure 172.

Specify these options as shown in Table 63.
### Table 63  What to Analyze options (part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Triggers</td>
<td>This option indicates whether to apply thresholds and corrective actions that are defined in the system.</td>
</tr>
<tr>
<td></td>
<td>If you specify <code>SYSTEM TRIGGERS Y</code> in the command syntax (or <code>SYSTRIGS Y</code> in the installation options module), BMCTRIG evaluates these exceptions in conjunction with system-level thresholds. Values that you specify on the BMCTRIG panels override values in the THRESHOLDS table for that exception. If you specify <code>SYSTEM TRIGGERS N</code> in the command syntax, BMCTRIG evaluates only thresholds that you specify on the BMCTRIG panels.</td>
</tr>
<tr>
<td>DB2 RTS</td>
<td>This option indicates whether to use the DB2 Real Time Statistics (RTS) for exception analysis. The choices follow:</td>
</tr>
<tr>
<td></td>
<td>■ Y – Use the RTS tables for exception evaluation in addition to the BMCSTATS statistics.</td>
</tr>
<tr>
<td></td>
<td>■ N – Do not use RTS tables.</td>
</tr>
<tr>
<td></td>
<td>■ O – Use the RTS tables for exception evaluation only.</td>
</tr>
<tr>
<td></td>
<td>If you specify Y, BMCTRIG accesses the RTS tables and uses those for exception evaluation in addition to the BMCSTATS statistics. If the same statistic is in both, BMCTRIG uses the RTS statistic value.</td>
</tr>
<tr>
<td></td>
<td>If you specify N, BMCTRIG does not evaluate RTS-based exceptions. If you specify N and also specify a DB2RTS exception, BMCTRIG cannot evaluate the exception.</td>
</tr>
<tr>
<td></td>
<td>If you specify O, BMCTRIG accesses the RTS tables and uses those values for exception evaluation. The BMCSTATS tables are not evaluated.</td>
</tr>
</tbody>
</table>
Specifying Saving and Reporting Exception options

These options include whether to write exceptions in the DASD MANAGER Exceptions table and to report on the exceptions. An example of these options may be found in Figure 172.

Specify these options as shown in Table 64.

Table 64  Saving and Reporting options (part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save</td>
<td>This option indicates whether to save found exceptions to the DASD MANAGER Exception table. Specify <strong>Save Y</strong> to write to the exceptions table or <strong>Save N</strong> to skip writing exceptions to the exceptions table.</td>
</tr>
<tr>
<td>DELETEAGE</td>
<td>This option is used to maintain the number of entries in the DASD MANAGER Exception table by providing an expiring number of days. Specify the number of days to keep the exceptions. The exceptions are removed from the exception table by the first trigger executed after the number of days has expired.</td>
</tr>
<tr>
<td>REPORT</td>
<td>This option is used to print a statistics report on the objects that have exceptions. Specify <strong>REPORT Y</strong> to generate the report.</td>
</tr>
</tbody>
</table>
Specifying BMCTRIG syntax options

Specifying Eliminate Duplicates and Index Escalation options

These options are applicable if you are generating corrective actions. They are used to remove redundant utilities and to control whether non-index type utilities are escalated from index or index partition to the table space or table space partition. An example of these options may be found in Figure 173.
Figure 173  Example of options to eliminate duplicates and set index escalation

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX Escalate . . . Y</td>
<td>(Y/N/A - Escalate IX's to TS's during generation)</td>
</tr>
<tr>
<td>IX Copy . . . N</td>
<td>(Y/N - Copy indexes)</td>
</tr>
<tr>
<td>Remove dup actions Y</td>
<td>(Y/N - Eliminate duplicate actions)</td>
</tr>
<tr>
<td>Detect inline opts Y</td>
<td>(Y/N - Eliminate actions based on inline options)</td>
</tr>
<tr>
<td>TS Tracks &lt; . . . 0</td>
<td>(0-99999999 Exclude TS/TP's with &lt; this many tracks)</td>
</tr>
<tr>
<td>TS Tracks &gt; . . . 99999999</td>
<td>(0-99999999 Exclude TS/TP's with &gt; this many tracks)</td>
</tr>
<tr>
<td>IX Tracks &lt; . . . 0</td>
<td>(0-99999999 Exclude IX/IP's with &lt; this many tracks)</td>
</tr>
<tr>
<td>IX Tracks &gt; . . . 99999999</td>
<td>(0-99999999 Exclude IX/IP's with &gt; this many tracks)</td>
</tr>
<tr>
<td>ExcludeEmpty . . . N</td>
<td>(Y/N - Exclude TS/TSP objects with 0 cardinality)</td>
</tr>
<tr>
<td>Archived . . . N</td>
<td>(Y/N - Exclude objects whose datasets are archived)</td>
</tr>
<tr>
<td>Bad status . . . N</td>
<td>(Y/N - Exclude objects not in RW status)</td>
</tr>
<tr>
<td>PARTLVL . . . N</td>
<td>(Y/N - Generate actions at a partition level)</td>
</tr>
<tr>
<td>Generate priority 0</td>
<td>(0-255 Min. exception priority to generate)</td>
</tr>
<tr>
<td>Group objects . . N</td>
<td>(Y/N - Generate objects as a group)</td>
</tr>
<tr>
<td>Max reorg parts</td>
<td>(0-4096 Max parts to group for reorg)</td>
</tr>
</tbody>
</table>

Specify these options as shown in Table 65.

Table 65  Eliminate Duplicates and Index Escalation options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX Escalate</td>
<td>Some utilities (like QUIESCE) cannot run against indexes. To run these utilities against the corresponding table spaces instead, when an index receives an exception, type Y in the IX Escalate field. Otherwise, type N to skip that utility for indexes. Use A to process the corresponding table space always, even if the utility supports indexes.</td>
</tr>
<tr>
<td>IX Copy</td>
<td>This option tells BMCTRIG to generate index copies whenever the exception specifies an IBM copy. To qualify for an index copy, the copy must be copyable. You can make an index copyable with an ALTER command. If you specify IX Copy Y and IX Escalate Y, BMCTRIG generates copies for copyable indexes and escalates noncopyable indexes to the table space level.</td>
</tr>
<tr>
<td>Remove dup actions</td>
<td>Typing Y automatically removes any actions that are duplicates. (A duplicate is the same services and syntax in the same order including subsets.) For more information, see “Duplicate actions” on page 463.</td>
</tr>
<tr>
<td>Detect inline opts</td>
<td>This option indicates whether BMCTRIG should eliminate actions if they are functionally duplicated by an inline option of another service.</td>
</tr>
</tbody>
</table>
Specifying Exclude Objects from Generation options

If you are evaluating exceptions, you can specify options to exclude objects from corrective actions as shown in Figure 173. BMCTRIG writes exception rows for these objects but marks them as EXCLUDED in the ACTION_TAKEN column (and sets the ACTIVE column to N).

To exclude certain objects completely, you might want to use an object set. For more information, see Chapter 5, “Using object sets.” These exclusion rules are applied during exception analysis so they are not applied to objects included in generation due to escalation rules that are not included in exception analysis.

Specifying Corrective Action Generation options

The options in Table 66 are applicable if you are generating corrective actions.

### Table 66 Corrective Action Generation options (part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARTLVL</td>
<td>Typing Y generates a corrective action (where possible) for each partition that receives an exception. N generates a corrective action for the entire table space or index.</td>
</tr>
<tr>
<td></td>
<td>Index exception evaluation occurs at the index level or at the index partition level. At the index level, BMCTRIG gathers statistics for indexes and evaluates exceptions based on these statistics. At the index partition level, BMCTRIG gathers statistics for every index partition within the index. If the index is not partitioned, BMCTRIG gathers statistics for an object that is referred to as partition zero.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Some statistics exist at both the index and index partition levels. BMCTRIG can evaluate certain exceptions at either level, but cannot evaluate both levels simultaneously.</td>
</tr>
<tr>
<td></td>
<td>- The exceptions that BMCTRIG can evaluate at either level are LEVELINC, LEVELMIN, LEVELS, and PCTCLUS as follows:</td>
</tr>
<tr>
<td></td>
<td>— If you do not specify PARTLVL Y, BMCTRIG evaluates LEVELINC, LEVELMIN, LEVELS, and PCTCLUS at the index level. That is, BMCTRIG uses index statistics only.</td>
</tr>
<tr>
<td></td>
<td>— If you specify PARTLVL Y and the index is partitioned, BMCTRIG evaluates LEVELINC, LEVELMIN, LEVELS, and PCTCLUS at the index partition level.</td>
</tr>
<tr>
<td></td>
<td>— If you specify PARTLVL Y and the index is not partitioned, BMCTRIG still evaluates LEVELINC, LEVELMIN, LEVELS, and PCTCLUS at the index level.</td>
</tr>
<tr>
<td></td>
<td>- NOSTATS is evaluated at the partition level for PARTLVL Y. Otherwise, NOSTATS is evaluated at the object level.</td>
</tr>
<tr>
<td>Generate priority</td>
<td>Typing a value in this field limits the generated work to object-actions with this priority or higher.</td>
</tr>
</tbody>
</table>
**Specifying Rebind and Resize Object options**

Figure 174 shows the Rebind and Resize Object options.

**Figure 174  Rebind options**

---

Rebind Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebind</td>
<td>N</td>
</tr>
<tr>
<td>Rebind Version</td>
<td>L</td>
</tr>
</tbody>
</table>

Resize Object Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resize jobs</td>
<td>Y</td>
</tr>
<tr>
<td>Resize Down</td>
<td>Y</td>
</tr>
<tr>
<td>Resize % pqty</td>
<td>0</td>
</tr>
<tr>
<td>Resize % sqty</td>
<td>0</td>
</tr>
<tr>
<td>Round qty by</td>
<td>A</td>
</tr>
<tr>
<td>Maximum pqty</td>
<td>0</td>
</tr>
<tr>
<td>Pqty % if max</td>
<td>0</td>
</tr>
</tbody>
</table>

**To use the Rebind options**

The BMCTRIG Rebind option lets you generate Rebind commands for packages and plans for table spaces or indexes that corrective action jobs affect. An affected object is one for which the utility updates the DB2 catalog statistics during execution of the corrective action job (for example, BMCREORG with **UpdateDB2stats=Y**). Table 67 lists the utilities that update DB2 catalog statistics. The utilities provide rebinds if you specify them as follows:

- in the corrective action
- with the parameters shown in Table 67

**Table 67  Triggered Rebind utilities (part 1 of 2)**

<table>
<thead>
<tr>
<th>Utility</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCCOPY</td>
<td>RUNSTATS = Yes</td>
</tr>
<tr>
<td></td>
<td>or UPDATE = Accesspath</td>
</tr>
<tr>
<td>BMCLOAD</td>
<td>UPDATEDB2STATS = Yes</td>
</tr>
</tbody>
</table>
Specifying BMCTRIG syntax options

Each table space or index space for which a utility runs is a **rebind candidate**. When the BMCTRIG job runs, BMCTRIG notes each rebinding candidate object that uses the utilities and parameters that Table 67 lists. BMCTRIG generates the rebinds last. If **Workload Balance** is **Y**, the rebinds are placed in a separate job. At the end of utility generation, BMCTRIG checks each object for dependencies to find the related packages or plans to rebind.

Each plan or package is at the end of the worklist as part of a REBIND command (-REBD). A single –REBD command can contain up to 400 rebinding statements. If a command requires more statements, the utility generates additional –REBD commands. If BMCTRIG does not find plans or packages for any of the rebinding candidates, it issues the following message:

```
BMC17167I ** NO REBIND OBJECTS DETECTED **
```

The options in Table 68 are applicable if you are generating corrective actions.

### Table 67  Triggered Rebind utilities (part 2 of 2)

<table>
<thead>
<tr>
<th>Utility</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCREORG</td>
<td>UPDATEDB2STATS = Yes</td>
</tr>
<tr>
<td>BMCSTATS</td>
<td>UPDATE = All or Accesspath</td>
</tr>
<tr>
<td>BMCUPRS</td>
<td>UPDATEDB2STATS = Yes (default), All, or Accesspath</td>
</tr>
<tr>
<td>LOAD</td>
<td>ALL</td>
</tr>
<tr>
<td></td>
<td>or UPDATEDB2STATS = Accesspath</td>
</tr>
<tr>
<td>REORG</td>
<td>STATISTICS</td>
</tr>
<tr>
<td></td>
<td>or UPDATE = Accesspath</td>
</tr>
<tr>
<td>RUNSTATS</td>
<td>ALL</td>
</tr>
<tr>
<td></td>
<td>or UPDATE = All or Accesspath</td>
</tr>
</tbody>
</table>

### Table 68  Rebind options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebind</td>
<td>To rebind affected packages and plans, type Y in the <strong>Rebind</strong> field. To rebind packages only, type <strong>PG</strong>. To rebind plans only, type <strong>PL</strong>. Otherwise, type <strong>N</strong>. For more information, see “To use the Rebind options” on page 513.</td>
</tr>
<tr>
<td>Rebind Version</td>
<td>To include only the latest version for the value selected for <strong>Rebind</strong>, type <strong>L</strong> in the <strong>Rebind Version</strong> field. To include all versions, type <strong>A</strong>. For more information, see “To use the Rebind options” on page 513.</td>
</tr>
</tbody>
</table>
To specify Resize options

For all Resize options, the option is valid only when the service in the corrective action is a reorganization and the Resize option in the REORG utility is set to any value except N, or the service is RESIZE.

The options in Table 69 are applicable if you are generating corrective actions.

Table 69  Resize options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resize jobs</td>
<td>If you specify Y for Resize jobs, BMCTRIG forces PARTLVL Y and MAXOBJECTS 1 and splits the REORG and RESIZE operations into separate jobs for each object and partition combination. Type N to consolidate REORG and RESIZE operations with other utilities into a single job in accordance with the Max Objects parameter on the JCL Generation options panel. Notes: Consider the following additional information:  ■ This option is valid only with Standard JCL.  ■ If you specify Y for Resize jobs and N for Workload Balance, BMCTRIG cannot perform a grouped reorganization.</td>
</tr>
<tr>
<td>Resize Down</td>
<td>To allow objects to be resized down, specify Y in the Resize Down field. The Resize Down option in the BMCTRIG action specifies whether to allow downward resizing of objects.</td>
</tr>
<tr>
<td>Resize % pqty</td>
<td>To increase the Reorgspace by a certain percentage, type a number between 0 and 999 in the Resize % field. If you use 0, BMCTRIG uses the standard formula. The specified percentage overrides the standard formula for resizing, which is ( \text{NEW PRIQTY} = \text{REORGSPACE} + (2\times SQTY) ).</td>
</tr>
<tr>
<td>Resize % sqty</td>
<td>Use this option to change the secondary quantity allocation to be a percent of the calculated primary quantity. If you do not specify a value, the secondary quantity is not changed.</td>
</tr>
<tr>
<td>Round qty by</td>
<td>Type A to round by the allocated type as noted in the BMCSTATS tables (the default) or C to calculate on a cylinder boundary.</td>
</tr>
<tr>
<td>Maximum pqty</td>
<td>Use this option to provide an upper limit for the calculated primary quantity when resizing. Note that DB2 rules for the maximum data set size always apply no matter what value you set (subject to DB2 rules). If you use this option, you can specify Pqty % if max as an alternate percent to apply to Reorgspace, if the calculated primary quantity exceeds this limit.</td>
</tr>
<tr>
<td>Pqty % if max</td>
<td>Type a value to specify the percent to apply to the Reorgspace to adjust the primary quantity, if the calculated primary quantity meets or exceeds Maximum pqty.</td>
</tr>
</tbody>
</table>
The **Resize** option in the corrective action for the IBM REORG and BMCREORG functions specifies the scope of the resizing. **Figure 175** shows the **Resize** option on the first IBM REORG utility panel. If the corrective action contains a RESIZE service, it will apply to all generated VCAT or storage group objects.

**Figure 175  REORG with Resize option**

Table 70 lists valid values for the **Resize** option and the corresponding instruction to the BMCTRIG program.

**Table 70  BMCTRIG interpretation of Resize option values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Instruction to BMCTRIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td><em>(default value)</em> Do not resize any table space or index objects. Use this value when generating an index or table space object.</td>
</tr>
<tr>
<td>T</td>
<td>Evaluate all table space partitions qualified for generation to determine whether to resize them. If so, generate the appropriate utility commands to resize the data sets during the reorganization.</td>
</tr>
<tr>
<td>I</td>
<td>Evaluate all index partitions qualified for generation to determine whether to resize them. If so, generate the appropriate utility commands to resize the data sets during the reorganization.</td>
</tr>
<tr>
<td>A</td>
<td>Evaluate all objects qualified for generation (table space and index partitions) to determine whether to resize them. If so, generate the appropriate utility commands to resize the data sets during the reorganization. If the table space or table space partition is generated, associated indexes or partitions are resized.</td>
</tr>
</tbody>
</table>
Selection of DB2 data sets for resizing is based on the latest BMCSTATS data. The primary space quantity is expanded or contracted based on the primary quantity that BMCTRIG calculates. You can set or change the quantity by providing a resize exit. If the return value is not zero, BMCTRIG uses it as the primary quantity. If BMCTRIG does not find a resize exit, it uses calculations and user options. The shipped default for the resize exit uses the following formula to calculate primary quantity:

\[ \text{PRIQTY} = \text{REORGSPACE} + (2 \times \text{SQTY}) \]

### Resize space considerations

The maximum primary quantity is determined by DB2 rules. For more information about data set allocation rules for the ALTER and CREATE statements, see the *IBM SQL Reference*. For more information about table space and index design, see the *IBM DB2 Administration Guide*.

If the value for SQTY is zero, the new primary quantity is a percentage of the old primary quantity. If the new primary quantity is equal to the old primary quantity value, the product will not resize the object.

For more information about resize options, see “To specify Resize options” on page 515.

### Resize limitations

The following resize limitations apply:

- You must have run BMCSTATS on the objects before using the Resize option.

- The elapsed time for the BMCTRIG job might increase if you resize many VCAT-defined objects due to ICF Catalog access.

- BMCTRIG does not resize multi-data set, VCAT-defined objects.

- The product does not insert passwords for VCAT-defined data sets in the Define statements.

- When you run BMCTRIG with PARTLVL Y, BMCTRIG examines table space and index partitions that have exceptions to determine whether they meet resize criteria. For more information about resize criteria, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

- Resize is not available if the REORGSPACE or REORGSPACE_KB statistic is less than or equal to zero (indicating that no statistics were gathered).

- The Resize option is not available on a table space that is partition-by-growth.
### Specifying Job and JCL Generation options

Figure 176 shows the Job and JCL Generation options.

#### Figure 176  Job and JCL Generation options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Action</td>
<td>(Default corrective action to generate)</td>
</tr>
<tr>
<td>AUTOSUBMIT</td>
<td>(Y/N - Automatically Submit the Util DSN)</td>
</tr>
<tr>
<td>Worklist only</td>
<td>(Y/N - Generate only the worklist)</td>
</tr>
<tr>
<td>When No Objects</td>
<td>(S/C - S-Stop, C-Continue)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard JCL</td>
<td>(Y/N - Standard JCL, N=Worklist JCL)</td>
</tr>
<tr>
<td>Max Objects</td>
<td>(1-255 - Maximum number of objects per job)</td>
</tr>
<tr>
<td>Max Steps</td>
<td>(1-255 - Maximum number of steps per job)</td>
</tr>
<tr>
<td>Number of Jobs</td>
<td>(0-46655 - Number of jobs to generate)</td>
</tr>
<tr>
<td>Workload Balance</td>
<td>(Y/N - Balance work across jobs)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Util DSN</td>
<td></td>
</tr>
<tr>
<td>Interim WL DSN</td>
<td>Note: Interim WL DSN is required when Standard JCL=Y. Util DSN (JCL target dataset) is always required.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Sequence Type</td>
<td>(N/A/blank - N=Numeric A=Alpha blank=Numeric)</td>
</tr>
<tr>
<td>Beginning Seq</td>
<td>(1-999/ZZZ Starting sequence value for &amp;JOBSEQ)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobcard1</td>
<td>//JOBC JOB (&amp;ZACCTNUM),'&amp;PGMR'.</td>
</tr>
<tr>
<td>Jobcard2</td>
<td>// CLASS=A,MSGLEVEL=(1,1)</td>
</tr>
<tr>
<td>Jobcard3</td>
<td>//</td>
</tr>
<tr>
<td>Jobcard4</td>
<td>//</td>
</tr>
<tr>
<td>Jobcard5</td>
<td>//</td>
</tr>
</tbody>
</table>

If you are generating corrective actions, you can specify job processing options.

1. In the Standard JCL field, type N for worklist format.

   For information about all of the other JCL generation options, see the DASD MANAGER PLUS for DB2 Reference Manual.

2. In the Util DSN field, specify a data set name to hold the JCL that BMCTRIG generates. For more information, see “To specify data set names (Util DSN)” on page 519.
To specify the maximum number of DB2 objects (Max Objects)

To generate standard JCL, specify the **Max Objects** option on the JCL Generation panel. **Max Objects** specifies the maximum number of DB2 objects to place in each generated job.

Set this option to 1 if you are generating one object per job. Do not use this option if you are using workload balancing. When the number that you specify is reached, a new job is generated. The default number is 20 if you are not using workload balancing. Until you become familiar with this feature, specify as few objects per job as you can.

To specify data set names (Util DSN)

To generate standard JCL, you must specify data set names, as follows:

- **Util DSN** is the name of the data set in which to place the generated utility jobs. When **Standard JCL** is **Yes**, **Util DSN** must be an existing partitioned data set. You do not have to specify a member name because BMCTRIG automatically generates the name based on the job name from the job card option. If **Standard JCL** is **No**, you must specify a PDS with a member or specify a sequential data set.

Before generating a BMCTRIG job that uses standard JCL, allocate this data set by using the following parameters:

```
DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB,DSORG=PO)
```

- **Interim WL DSN** is the name of the data set into which to place the interim worklist statements when **Standard JCL** is **Yes**. BMCTRIG generates the interim worklist statements and inputs them to batch JCL generation.

**Interim WL DSN** must be a sequential data set or a partitioned data set with a member name. Before generating a BMCTRIG job that uses standard JCL, allocate this data set by using the following parameters:

```
DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB,DSORG=PS or PO)
```

To specify the minimum number of jobs (Number of Jobs)

When generating standard JCL, you can specify the minimum number of jobs to generate (**Number of Jobs**). This option is useful when registering the generated utility jobs in a job scheduling package. Blank is the default. Blank or 0 generates as many jobs as needed to process all of the objects that meet the exception criteria.
For workload balancing, this option must be greater than 1 and indicates the maximum and minimum number of concurrent jobs that are generated for each action.

**NOTE**

If the action contains grouped services, you can have a grouped service at the beginning of the action and one at the end. Grouped services do not count in the total number of services. Also, if you are rebinding, the rebind commands are built as a separate job at the end and do not count in the total number of jobs.

Indicate in this option the number of jobs for the job scheduler, and match the job-naming convention that the job card uses. BMCTRIG ensures that all matching jobs are current and accurate, based on the newest exceptions.

If not enough objects receive exceptions to require the specified minimum number of jobs, BMCTRIG places IEFBR14 utilities in the extra jobs.

If BMCTRIG encounters more exceptions than the specified minimum number of jobs can contain, BMCTRIG responds as follows:

- If you are not using workload balancing, BMCTRIG continues producing jobs with the naming convention from the job card until it has placed all objects with exceptions in jobs.

- If you are using workload balancing, BMCTRIG generates only the number of concurrent jobs that you specify per action. As stated earlier, BMCTRIG might also generate a grouped object job before and after the concurrent jobs and a rebind job after the concurrent jobs.

For objects that BMCTRIG cannot fit into jobs, it does not generate them and their exceptions remain active.

If more jobs exist than the minimum number of jobs that you specified, you have the following choices:

- Change **Number of Jobs** to the number of jobs generated for future executions of the same BMCTRIG action.

- Treat the extra jobs as one-time-only request jobs.

- Ignore the extra jobs in the schedule and delete them from the utility job data set (**Util DSN**). If you ignore the extra jobs, the next time that BMCTRIG runs, it identifies the objects in these jobs as exceptions and builds the utility jobs again.
If specific BMCTRIG jobs regularly exceed the **Number of Jobs** value, perform one of the following tasks:

- Increase the **Number of Jobs** value to create more jobs.
- Increase the **Max Objects per job** value to perform more work in each job. (Note that specifying **Max Objects** is ignored for workload balancing.)
- Refine the thresholds so that fewer objects meet exceptions.
- Use priorities on thresholds and object-actions to order important work.

**Specifying Reorg-Related Override Exceptions**

Figure 177 shows the Reorg-Related Override Exceptions options. Table 58 provides descriptions of the exceptions. The *DASD MANAGER PLUS for DB2 Reference Manual* provides formulas for the exceptions.

---

**NOTE**

If you are using system triggers, override exceptions are not required. When system triggers are defined, the exceptions that you specify on this panel override the system trigger definitions.
Specifying BMCTRIG syntax options

---

**Figure 177  Reorg-Related Override Exceptions**

<table>
<thead>
<tr>
<th>DB2 RTS Specific Reorg Exceptions:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shared Tablespace/Index Exceptions:</strong></td>
</tr>
<tr>
<td>Mods since reorg</td>
</tr>
<tr>
<td>Ins since reorg</td>
</tr>
<tr>
<td>Del since reorg</td>
</tr>
<tr>
<td>Mass del reorg N</td>
</tr>
<tr>
<td><strong>Tablespace Exceptions:</strong></td>
</tr>
<tr>
<td>Unclust inserts</td>
</tr>
<tr>
<td>Reorg DISORG LOB</td>
</tr>
<tr>
<td>Pct over alloc</td>
</tr>
<tr>
<td><strong>Index Exceptions:</strong></td>
</tr>
<tr>
<td>Append inserts</td>
</tr>
<tr>
<td>Reorg Leaf..</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DB2 RTS or BMCSTATS Reorg Exceptions:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shared Tablespace/Index Exceptions:</strong></td>
</tr>
<tr>
<td>Dataset Extents</td>
</tr>
<tr>
<td>AREO* pending N</td>
</tr>
<tr>
<td>AREOR pending N</td>
</tr>
<tr>
<td><strong>Tablespace Exceptions:</strong></td>
</tr>
<tr>
<td>FARIND ...</td>
</tr>
<tr>
<td>TOTALIND ...</td>
</tr>
<tr>
<td>Reorg Pend ... N</td>
</tr>
<tr>
<td><strong>Index Exceptions:</strong></td>
</tr>
<tr>
<td>LEAFTOTOFF ...</td>
</tr>
<tr>
<td>LEAFFAROFF ...</td>
</tr>
<tr>
<td>LEVELMIN ...</td>
</tr>
<tr>
<td>Pseudo Del Key</td>
</tr>
<tr>
<td>LEVELINC ...</td>
</tr>
<tr>
<td>LEVELS ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BMCSTATS Specific Reorg Exceptions:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shared Tablespace/Index Exceptions:</strong></td>
</tr>
<tr>
<td>SPACE ...</td>
</tr>
<tr>
<td>REORGSPACE ...</td>
</tr>
<tr>
<td>EXTENTS ...</td>
</tr>
<tr>
<td>CARD ...</td>
</tr>
<tr>
<td><strong>Tablespace Exceptions:</strong></td>
</tr>
<tr>
<td>FAROFF ...</td>
</tr>
<tr>
<td>TOTALOFF ...</td>
</tr>
<tr>
<td>PCTCLUST ...</td>
</tr>
<tr>
<td>Pct Dropped Rows</td>
</tr>
<tr>
<td>PctActiveHI ...</td>
</tr>
<tr>
<td>PctActiveLo ...</td>
</tr>
<tr>
<td>LOB ORGRATIO ...</td>
</tr>
<tr>
<td>LOB Freespace ...</td>
</tr>
<tr>
<td><strong>Index Exceptions:</strong></td>
</tr>
<tr>
<td>LEAFDIST ...</td>
</tr>
</tbody>
</table>

---

**Specifying Statistics-Related Override Exceptions**

**Figure 178** shows the Statistics-Related Override Exceptions options. **Table 59** provides descriptions of the exceptions. The *DASD MANAGER PLUS for DB2 Reference Manual* provides formulas for the exceptions.
Specifying BMCTRIG syntax options

**NOTE**
If you are using system triggers, override exceptions are not required. When system triggers are defined, the exceptions that you specify on this panel override the system trigger definitions.

**Figure 178  Statistics-Related Exceptions options**

<table>
<thead>
<tr>
<th>Statistics-Related and Misc. Override Exceptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DB2 RTS Specific Statistics/Miscellaneous Exceptions:</strong></td>
</tr>
<tr>
<td>Reorg after stat N  (Y/N - Reorg since last stats)</td>
</tr>
<tr>
<td>Load after stats N  (Y/N - Load after last stats)</td>
</tr>
<tr>
<td>Mods since stats  (1-100 % rows modified since stats)</td>
</tr>
<tr>
<td>Mass del stats N  (Y/N - Mass delete since last stats)</td>
</tr>
<tr>
<td>N0 RTS stats dat N  (Y/N No statistics data in the RTS tables)</td>
</tr>
<tr>
<td><strong>DB2 RTS or BMCSTATS Statistics/Miscellaneous Exceptions:</strong></td>
</tr>
<tr>
<td>Catalog Upd Age  (1-32767 Days since last catalog update)</td>
</tr>
<tr>
<td><strong>BMCSTATS Specific Statistics/Miscellaneous Exceptions:</strong></td>
</tr>
<tr>
<td>NOSTATS . . . . N  (Y/N BMCSTATS do not exist)</td>
</tr>
<tr>
<td>BMCSTATS Age . .  (1-32767 Days since last BMCSTATS)</td>
</tr>
<tr>
<td><strong>Index Exceptions:</strong></td>
</tr>
<tr>
<td>NONUNIFORM . . .  (0-99% Non Uniformity of Index)</td>
</tr>
<tr>
<td>ROWS/KEY . . .    (1-99999999 number of rows per key)</td>
</tr>
</tbody>
</table>

Specifying Copy-Related Override Exceptions

**Figure 179** shows the Copy-Related Override Exceptions options. **Table 60** provides descriptions of the exceptions. The *DASD MANAGER PLUS for DB2 Reference Manual* provides formulas for the exceptions.

**NOTE**
If you are using system triggers, override exceptions are not required. When system triggers are defined, the exceptions that you specify on this panel override the system trigger definitions.
Generating a BMCTRIG job

This section describes how to specify and generate the JCL for a BMCTRIG job.

To generate JCL to run BMCTRIG

After you create an action to run BMCTRIG (and BMCSTATS), you generate the JCL.

1 From the DASD MANAGER PLUS main menu, select Utility Actions (WORKIDS).

2 On the Actions menu, select List Actions, type the action name or a pattern, and press Enter.

3 On the Action List panel, type G in the Act field next to the BMCTRIG action that you created.

   The Action Generation panel appears. (See “Generating actions” on page 299).

4 To generate the worklist and JCL to run your BMCTRIG job, enter the following information:

   A In the JCL DSN field, specify the JCL data set in which to build the BMCTRIG JCL.

   B In the Worklist DSN field, specify the worklist data set in which to build the BMCTRIG worklist and press Enter.
The worklist appears (Figure 180).

**Figure 180 BMCTRIG worklist**

```plaintext
-TIME 000000 '2004-08-03-13.46.42.00004'
-SSID 000001 DEAE
-WKID 000002 EXCEPT1
-SYNC 000003
-BMCU 000004 ASUSMAIN
  BMCSTATS TABLESPACE TZU%.%
  TABLE (ALL)
-BMCU 000005 ASUSTRIG
  BMCSTATS TABLESPACE TZU%.%
  SYSTEMTRIGGERS Y
  SPACE 30 COPYAGE 7 FAROFF 20 LEVELINC Y IXCOPYAGE 10 CATSTATAGE 30
  STDJCL Y JCLGEN Y MINJOBS 5 BEGJOBSEQ 001 UTILJOB MYREORG UTILSEQ 0001
  STDJLDSN 'ASU.DEAE.V710QA.RGRTRIG.TRIGNWJ'
  UTILDSN 'ASU.DEAE.V710QA.RGRTRIG.TRIGIWL(EXCEPT1)'
  TRIGJC1
  //EXCPT&JOBSEQ JOB (&ZACCTNUM),',&PGMR',
  TRIGJC2
  //  CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
  TRIGJC3
  //  NOTIFY=&USERID
  TRIGJC4
  /*
  TRIGJC5
  /*
```

5 Edit the worklist and press END.

6 Edit the JCL as required, submit the job, and view the job output.

The BMCSTATS report and a list of the exceptions that the BMCTRIG job finds are in the AEXPRINT data set in the System Display and Search Facility (SDSF) output. If you specified `Report=Y` on the Exception Options panel, the BMCSTATS report appears at the end of the data set.

7 To view the exceptions that the BMCTRIG job detected, access the SDSF AEXPRINT data set and enter `FIND EXCEPTIONS` on the COMMAND line (Figure 187).

You can also create an exceptions report (see “Statistics exception report” on page 617), view exceptions online (see “To view exceptions by object online” on page 547), or view active exceptions (see “To view active exceptions” on page 547).

8 *(optional)* Add the job to your scheduler to run routinely.

If you plan to run BMCTRIG in two phases (one to evaluate exceptions and a second to generate corrective actions), add the RESUME Y option to your second BMCTRIG job. To specify that DASD MANAGER PLUS generate RESUME Y, select the **Resume Generation** option on the BMCTRIG syntax option panel.
The BMCTRIG utility can generate standard JCL or worklist JCL. Setting the Standard JCL option to Y instructs BMCTRIG to generate standard JCL for corrective actions. Setting the Standard JCL option to N instructs BMCTRIG to generate worklist format JCL in which programs are run by a single job step that runs AEXEMAIN (Execution monitor).

### Standard JCL

The BMCTRIG utility can generate standard JCL jobs for the objects that meet or exceed a threshold. You can generate standard-format JCL only by using the BMCTRIG utility. Standard JCL offers the benefits of its familiarity and flexibility. You can accomplish the tasks shown in Table 71 when you use standard JCL:

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>balance work across multiple jobs</td>
<td>Balancing the workload reduces overall elapsed time.</td>
</tr>
<tr>
<td>restart jobs more easily</td>
<td>With standard JCL, use the new or restart parameter for the utility that failed and switch out the data sets marked **INITIAL in the JCL with the data sets marked **RESTART. Standard JCL makes it easier for system operators to restart the jobs because, in most cases, the operators know how to access DASD MANAGER PLUS and use its worklist restart option.</td>
</tr>
<tr>
<td>generate multistep jobs</td>
<td>Standard JCL puts each utility into its own step, making it easier to restart at the failing step, if necessary.</td>
</tr>
<tr>
<td>control job names and other job card parameters</td>
<td>Standard JCL provides job-sequence variables that allow more than 999 job names and a job card exit. See “Job card exit and job name” on page 531.</td>
</tr>
<tr>
<td>specify the minimum number of jobs to generate</td>
<td>Specifying the minimum number of jobs to generate is useful when using a job scheduler. If the full number of jobs is not required, BMCTRIG generates null IEFBR14 jobs to meet the minimum number. Because you can specify a minimum number of jobs, you do not have to change your scheduler as often. See “To specify the minimum number of jobs (Number of Jobs)” on page 519.</td>
</tr>
</tbody>
</table>

To generate standard JCL, specify **Standard JCL Y** when specifying the BMCTRIG parameters.

**WARNING**

If you upgrade your version of DASD MANAGER PLUS, BMC recommends that you regenerate your BMCTRIG JCL before you run BMCTRIG jobs. Otherwise, you should review DD requirements for potential changes.
Figure 181 shows an overview of generating standard JCL.

**Figure 181  Generating standard JCL**

**JCL generation options**

The BMCTRIG JCL Generation panel (Figure 181) shows the JCL generation options that are required for creating standard JCL. The required options are as follows:

- Standard JCL = Y
- data set names:
  - Util DSN
  - Interim WL DSN
- jobcards
Example procedures

The interim worklist and generated JCL that Figure 182 and Figure 183 show are the output of procedures in the following tasks:

- “To set up a REORG corrective action” on page 557
- “To generate the BMCTRIG service that uses the REORG corrective action” on page 565

Interim worklist

When you specify standard JCL, BMCTRIG produces an interim worklist from the BMCTRIG parameters, including the job generation options. Unlike the worklist for worklist-format utility jobs, the interim worklist is not an input data set for the generated JCL. Instead, JCL Generation processes the interim worklist and incorporates its information directly into the services of the standard JCL. For more information about JCL Generation, see Chapter 3, “Setting up DASD MANAGER PLUS.”

The interim worklist contains worklist commands, including the commands shown in Table 72 for standard JCL.

Table 72  Standard JCL worklist commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-NEWJ</td>
<td>-NEWJ causes JCL Generation to begin creation of a new job. Each time the maximum number of objects per job is reached, DASD MANAGER PLUS writes the -NEWJ command to the interim worklist. DASD MANAGER PLUS also writes the -NEWJ command whenever the number of services in a job exceeds 255.</td>
</tr>
<tr>
<td>-UDSJ</td>
<td>-UDSJ identifies the data set that contains the generated JCL.</td>
</tr>
<tr>
<td>-BR14</td>
<td>-BR14 generates IEFBR14 if the minimum number of jobs that you specified is not reached. For example, if you specify a minimum of four jobs, but BMCTRIG processes all objects that meet threshold specifications in the first three jobs, BMCTRIG writes the -BR14 command once with a -NEWJ preceding it. JCL Generation builds a fourth job that contains a IEFBR14 step in the worklist.</td>
</tr>
</tbody>
</table>

Figure 182 shows an interim worklist that assumes Max Objects = 2 and Number of Jobs = 4.

As specified in the BMCTRIG job generation options, the worklist includes four jobs. Each job begins with the -NEWJ worklist command. The first job specifies COPY and REORG services for the first two objects that meet threshold specifications:

- QZUD40.QZUS0140
- QZUD41.QZUS0141
The second job specifies COPY and REORG services for the last object that meets threshold specifications, QZUD42.QZUS0142.

The third and fourth jobs specify a null IEFBR14 utility because the following statements are true:

- Only three objects meet threshold specifications and were generated into the first two jobs.
- You have specified a minimum of four jobs.

**Figure 182  Interim worklist (part 1 of 2)**

| -TIME 000001 '2003-05-19-15.49.41.416895' |
| 17635602 |
| * FOR TRIGGER EXCEPTION |
| 83821694 |
| -UDSJ 000001 'RDAJXN3.TEST9.JCL' |
| -NEWJ |
| //TBMC&JOBSEQ JOB (5213), '&PGMR'. |
| 29267175 |
| -COPY 000002 |
| COPY TABLESPACE QZUD40.QZUS0140 |
| SHRLEVEL REFERENCE |
| COPYDDN (C0001) |
| RECOVERYDDN (R0001) |
| 52773357 |
| -JCLP 000002 COPY DDNAME C0001 DSNPREF |
| RDAJXN.DEAE.LP.&OBNOD |
| -JCLP 000002 COPY DDNAME C0001 DEVTYPE SYSDA |
| -JCLP 000002 COPY DDNAME R0001 DSNPREF |
| RDAJXN.DEAE.RP.&OBNOD |
| -JCLP 000002 COPY DDNAME R0001 DEVTYPE SYSDA |
| 15244851 |
| -REOR 000003 |
| REORG TABLESPACE QZUD40.QZUS0140 |
| UNLDDN S0002 |
| WORKDDN SYSUT001 |
| SORTDATA |
| SORTDEVT SYSDA |
| UNLOAD CONTINUE |
| 99885114 |
| -COPY 000004 |
| COPY TABLESPACE QZUD41.QZUS0141 |
| SHRLEVEL REFERENCE |
| COPYDDN (C0003) |
| RECOVERYDDN (R0003) |
| 52772074 |
| -JCLP 000004 COPY DDNAME C0003 DSNPREF |
| RDAJXN.DEAE.LP.&OBNOD |
| -JCLP 000004 COPY DDNAME C0003 DEVTYPE SYSDA |
| -JCLP 000004 COPY DDNAME R0003 DSNPREF |
Figure 182  Interim worklist (part 2 of 2)
Job card exit and job name

When building the interim worklist, BMCTRIG calls the job card exit before generating each utility instance. The default source code for the exit is in the HLQ.CNTL data set member ASUXJBCD, where HLQ is the high-level qualifier of your DASD MANAGER PLUS control library. If the load module does not exist, BMCTRIG continues.

The default code for the job card exit returns the job card that is passed to it. The job card is usually the job card image of the first job card that you specify when you define the BMCTRIG utility in DASD MANAGER PLUS. The resulting job name can contain variables. The resolved job name must be no more than eight characters, whether the product resolves it through the job card user exit or through variable substitution. For more information about job card exits, see the DASD MANAGER PLUS for DB2 Reference Manual.

Be aware of the following requirements:

- job sequence variables

BMC Software recommends using one of the &JOBSEQ variables as part of your job name specification when generating standard JCL format jobs. Table 73 describes the variables and how to use them.

Table 73 JOBSEQ variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Generates</th>
<th>Job names with variables</th>
<th>First job name generated</th>
<th>Maximum number of services</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JOBSEQ alpha</td>
</tr>
<tr>
<td>&amp;JOBSEQ or &amp;JS</td>
<td>3-digit job sequence number</td>
<td>AABB&amp;JOBSEQ or AA&amp;JOBSEQBB</td>
<td>AABB001 or AA001BB</td>
<td>46,655</td>
</tr>
<tr>
<td>&amp;JOBSEQ1 or &amp;JS1</td>
<td>1-digit job sequence number</td>
<td>AABB&amp;JOBSEQ1 or AA&amp;JOBSEQ1BB</td>
<td>AABB1 or AA1BB</td>
<td>35</td>
</tr>
<tr>
<td>&amp;JOBSEQ2 or &amp;JS2 or &amp;2</td>
<td>2-digit job sequence number</td>
<td>AABB&amp;JOBSEQ2 or AA&amp;JOBSEQ2BB</td>
<td>AABB01 or AA01BB</td>
<td>1,295</td>
</tr>
</tbody>
</table>

a When specifying Standard JCL=Y and using job name variables, avoid using 1 and 2 after &JOBSEQ unless specifying &JOBSEQ1 or &JOBSEQ2. Also, avoid using the variable &WKID.

JCL Generation increments the job sequence number by 1 for each BMCTRIG-generated standard job, to a maximum count of 46,655 (for &JOBSEQ).

b This column indicates the first job name unless you specify a different beginning sequence number by using the Beginning Seq Num option.
Using worklist format or standard JCL

- **job names for BMCTRIG actions**

  BMC Software recommends using a five-character name for the BMCTRIG jobs that use the standard JCL format. Using a five-character name for the action helps relate the BMCTRIG action to the jobs that the action generates.

- **new jobs in the worklist**

  BMCTRIG uses options and rules to determine when to issue a new job. If BMCTRIG does not need to issue a new job, the job card exit can specify when to issue a new job unless you are using workload balancing. The default job card exit begins a new job when the object count for the current job exceeds the `Max Objects` parameter value. The job card also starts a new job if the number of steps in the current job exceeds 255, the maximum number of steps allowed by OS/390 and z/OS.

---

**WARNING**

If you are using workload balancing, BMCTRIG ignores the exit for determining a new job. The workload balancing routine automatically specifies when to begin a new job.

---

**Generated standard JCL**

The generated standard JCL in Figure 183 shows the structure of the interim worklist, as follows:

<table>
<thead>
<tr>
<th>Job</th>
<th>Description</th>
</tr>
</thead>
</table>
| first job  | The job in Figure 183 is the first of the four jobs of the worklist. The job consists of four services, one for each utility and object combination:  
  - COPY QZUD40.QZUS0140  
  - REORG QZUD40.QZUS0140  
  - COPY QZUD41.QZUS0141  
  - REORG QZUD41.QZUS0141  
  The COPY and REORG utility parameters appear under the SYSIN DDs. A data set cleanup step follows each step. Note that the parameters for restarting the job are in comment lines. |
| second job | The second job consists of the following services:  
  - COPY QZUD42.QZUS0142  
  - REORG QZUD42.QZUS0142 |
| third job  | The third job has one step, a IEFBR14. |
Figure 183  Generated standard JCL (part 1 of 6)

```plaintext
//TBMC001  JOB (5213),'DASDTRIG-STDJCL'.
// CLASS=A,MSGCLASS=X
//*
/*JOBPARM SYSAFF=DB2A
//*
/*******************--------------------------**********
//* CREATED BY :  RDAJXN2
//* TIMESTAMP  :  10/10/2005.15.49.53
//* ENVIRONMENT:  ISPF 5.2MVS  BATCH
//* RELEASE :  V08.02.00 01/30/2006
//* DB2 VERSION:  710
/*******************--------------------------**********
//STEP1 EXEC PGM=DSNUTILB,
/* PARM=(DEAE,'TBMC001.BMCTR',
/*       RESTART),
/* PARM=(DEAE,'TBMC001.BMCTR'),
/* REGION=0M,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DB2.MSTRPLAN.LOAD
//   DD DISP=SHR,DSN=SYS3.DEAE.DSNEXIT
//   DD DISP=SHR,DSN=SYS2.DB2V71M.DSNLOAD
//   DD DISP=SHR,DSN=AUS.ASU810.D71.LOAD
//ABNLIGNR DD DUMMY
//DSSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//UTPRINT  DD SYSOUT=*  
//SYSOUT   DD SYSOUT=*  
//SYSIN DD *
COPY TABLESPACE QZUD40.QZUS0140
SHRLEVEL REFERENCE
COPYDDN (C0001)
RECOVERYDDN (R0001)

/*------------------------------*/
/* DATA SET SIZE REQUIRED FOR DD C0001 IS AN ESTIMATE */
//C0001 DD DSN=RDAJXN.DEAE.LP.QZUD40.QZUS0140.C0001,
/* DISP=SHR,**RESTART*/
// DISP=(NEW,CATLG,CATLG),  **INITIAL*/
// SPACE=(CYL,(21,5),RLSE). ESTIMATE-C/H
// UNIT=SYSDA

/*------------------------------*/
/* DATA SET SIZE REQUIRED FOR DD R0001 IS AN ESTIMATE */
//R0001 DD DSN=RDAJXN.DEAE.RP.QZUD40.QZUS0140.R0001,
/* DISP=SHR,**RESTART*/
// DISP=(NEW,CATLG,CATLG),  **INITIAL*/
// SPACE=(CYL,(21,5),RLSE). ESTIMATE-C/H
// UNIT=SYSDA
```

Chapter 8  Analyzing objects by using BMCTRIG  533
Using worklist format or standard JCL

Figure 183  Generated standard JCL (part 2 of 6)

```plaintext
//**----------------------------------------------
//** END OF JOBSTEP
//**----------------------------------------------
//STEP2 EXEC PGM=DSNUTILB,
//** PARM=(DEAE,'TBMC001.BMCTR',
//**       RESTART),
//** PARM=(DEAE,'TBMC001.BMCTR'),
//** REGION=OM,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DB2.MSTRPLAN.LOAD
// DD DISP=SHR,DSN=SYS3.DEAE.DSNEXIT
// DD DISP=SHR,DSN=SYS2.DB2V71M.DSNLOAD
// DD DISP=SHR,DSN=AUS.ASU810.D71.LOAD
//ABNLIGNR DD DUMMY
//DSSPRINT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//UTPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//SYSIN DD *
REORG TABLESPACE QZUD40.QZUS0140
UNLODN S0002
WORKDDN SYSUT001
SORTDATA
SORTDEVT SYSDA
UNLOAD CONTINUE
//**----------------------------------------------
//** UTILITY UNLOAD DD STATEMENTS
//**----------------------------------------------
//** DATA SET SIZE REQUIRED FOR DD S0002 IS AN ESTIMATE
//S0002 DD DSN=RDAJXN2.TBMC001.SYSREC.S0002,
//** DISP=SHR,       ***RESTART*
// DD DISP=(NEW,CATLG,CATLG), ***INITIAL*
// DD SPACE=(CYL,(24,6),RLSE), ESTIMATE-C/C
// DD UNIT=SYSDA
//**----------------------------------------------
//** UTILITY WORK DD STATEMENTS
//**----------------------------------------------
//** DATA SET SIZE REQUIRED FOR DD SYSUT001 IS AN ESTIMATE
//SYSUT001 DD DSN=RDAJXN2.TBMC001.STEP2.SYSUT001,
//** DISP=SHR,       ***RESTART*
// DD DISP=(NEW,CATLG,CATLG), ***INITIAL*
// DD SPACE=(CYL,(45,11)), ESTIMATE-C/C
// DD UNIT=SYSDA
//**----------------------------------------------
//** UTILITY SORTOUT WORK DD STATEMENTS
//**----------------------------------------------
//** DATA SET SIZE REQUIRED FOR DD SORTOUT IS AN ESTIMATE
//SORTOUT DD DSN=RDAJXN2.TBMC001.STEP2.SORTOUT,
//** DISP=SHR,       ***RESTART*
// DD DISP=(NEW,CATLG,CATLG), ***INITIAL*
```

DASD MANAGER PLUS for DB2 User Guide
```
// SPACE=(CYL,(45,11)), ESTIMATE=C/C
// UNIT=SYSDA
// END OF JOBSTEP

// DATASET CLEANUP STEP WILL EXECUTE IF PREVIOUS CONDITION CODE
// IS 4 OR LESS.

//CLEANUP  EXEC PGM=IEFBR14,
// COND=(4,LT)

// UTILITY UNLOAD DD STATEMENTS
// ALTER UNLOAD AND BMCUNLOAD DATASETS ARE COMMENTED OUT.
// IBM REORG AND BMC REORG DATASETS ARE NOT COMMENTED OUT
// UNLESS STACKED ON TAPE AND MULTI-STEP JOB.

//S0002 DD DSN=RDAJXN2.TBMC001.SYSREC.S0002,
// DISP=(MOD,DELETE,DELETE),
// SPACE=(TRK,(1,1)),
// UNIT=SYSDA

// UTILITY WORK DD STATEMENTS

//SYSUT001 DD DSN=RDAJXN2.TBMC001.STEP2.SYSUT001,
// DISP=(MOD,DELETE,DELETE),
// SPACE=(TRK,(1,1)),
// UNIT=SYSDA

// UTILITY SORTOUT WORK DD STATEMENTS

//SORTOUT DD DSN=RDAJXN2.TBMC001.STEP2.SORTOUT,
// DISP=(MOD,DELETE,DELETE),
// SPACE=(TRK,(1,1)),
// UNIT=SYSDA

// END OF DATASET CLEANUP STEP

//STEP3 EXEC PGM=DSNUTILB,
// PARM=(DEAE,'TBMC001.BMCTR',
// RESTART),
// PARM=(DEAE,'TBMC001.BMCTR'),
// REGION=0M,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DB2.MSTRPLAN.LOAD
// DD DISP=SHR,DSN=SYS3.DEAE.DSNEXIT
// DD DISP=SHR,DSN=SYS2.DB2V61M.DSNLOAD
// DD DISP=SHR,DSN=AUS.ASU621.D71.LOAD
//ABNLIGNR DD DUMMY
//DSSPRINT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
```
Using worklist format or standard JCL

Figure 183 Generated standard JCL (part 4 of 6)

```plaintext
//UTPRINT DD SYSOUT=*  
//SYSOUT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
COPY TABLESPACE QZUD41.QZUS0141 
SHRLEVEL REFERENCE 
COPYDDN (C0003) 
RECOVERYDDN (R0003) 

// ------------------------------ 
// UTILITY COPY DD STATEMENTS 
// ------------------------------ 
// DATA SET SIZE REQUIRED FOR DD C0003 IS AN ESTIMATE 
//C0003 DD DSN=RDAJXN.DEAE.LP.QZUD41.QZUS0141.C0003, 
//   DISP=SHR, ***RESTART* 
//     DISP=(NEW,CATLG, CATLG), ***INITIAL* 
//     SPACE=(CYL,(37,9),RLSE), ESTIMATE-C/H 
//     UNIT=SYSDA 
// ------------------------------ 
// UTILITY RECOVERY SITE IMAGE COPY DD STATEMENTS 
// ------------------------------ 
// DATA SET SIZE REQUIRED FOR DD R0003 IS AN ESTIMATE 
//R0003 DD DSN=RDAJXN.DEAE.RP.QZUD41.QZUS0141.R0003, 
//   DISP=SHR, ***RESTART* 
//     DISP=(NEW,CATLG, CATLG), ***INITIAL* 
//     SPACE=(CYL,(37,9),RLSE), ESTIMATE-C/H 
//     UNIT=SYSDA 
// ------------------------------ 
// END OF JOBSTEP 
// *-----------------------------* 
//STEP4 EXEC PGM=DSNUTILB, 
//  PARM=(DEAE,'TBMC001.BMCTR', 
//  RESTART), 
//     PARM=(DEAE,'TBMC001.BMCTR'), 
//     REGION=0M,COND=(4,LT) 
//STEPLIB DD DISP=SHR,DSN=DB2.MSTRPLAN.LOAD 
//        DD DISP=SHR,DSN=SYS3.DEAE.DSNEXIT 
//        DD DISP=SHR,DSN=SYS2.DB2V61M.DSNLOAD 
//        DD DISP=SHR,DSN=AUS.ASU621.D71.LOAD 
//ABNLIGNR DD DUMMY 
//DSSPRINT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSDUMP DD SYSOUT=* 
//UTPRINT DD SYSOUT=* 
//SYSLIB DD SYSOUT=* 
//REORG TABLESPACE QZUD41.QZUS0141 
UNLODN 50004 
WORKDDN SYSUT001 
SORTDATA 
SORTDEVT SYSDA 
UNLOAD CONTINUE 
// *-----------------------------* 
```
Figure 183  Generated standard JCL (part 5 of 6)

```c
/*  UTILITY UNLOAD DD STATEMENTS */
/*--------------------------------------------------------------------*/
/* DATA SET SIZE REQUIRED FOR DD $0004 IS AN ESTIMATE */
/$0004 DD DSN=RDAJXN2.TBMC001.SYSREC.$0004,
  /* DISP=SHR, ***RESTART*/
  /* DISP=(NEW,CATLG,CATLG),***INITIAL*/
  /* SPACE=(CYL,(34,8),RLSE), ESTIMATE=C/C */
  /* UNIT=SYSDA */
/*--------------------------------------------------------------------*/
/* UTILITY WORK DD STATEMENTS */
/*--------------------------------------------------------------------*/
/* DATA SET SIZE REQUIRED FOR DD $YSUT001 IS AN ESTIMATE */
/$YSUT001 DD DSN=RDAJXN2.TBMC001.STEP4.$YSUT001,
  /* DISP=SHR, ***RESTART*/
  /* DISP=(NEW,CATLG,CATLG),***INITIAL*/
  /* SPACE=(CYL,(14,3)), ESTIMATE=C/C */
  /* UNIT=SYSDA */
/*--------------------------------------------------------------------*/
/* UTILITY SORTOUT WORK DD STATEMENTS */
/*--------------------------------------------------------------------*/
/* DATA SET SIZE REQUIRED FOR DD SORTOUT IS AN ESTIMATE */
/$SORTOUT DD DSN=RDAJXN2.TBMC001.STEP4.$SORTOUT,
  /* DISP=SHR, ***RESTART*/
  /* DISP=(NEW,CATLG,CATLG),***INITIAL*/
  /* SPACE=(CYL,(14,3)), ESTIMATE=C/C */
  /* UNIT=SYSDA */
/*--------------------------------------------------------------------*/
/* END OF JOBSTEP */
/*--------------------------------------------------------------------*/
/* DATASET CLEANUP STEP WILL EXECUTE IF PREVIOUS CONDITION CODE */
/* IS 4 OR LESS. */
/*--------------------------------------------------------------------*/
/$CLEANUP EXEC PGM=IEFBR14,
  /* COND=(4,LT) */
/*--------------------------------------------------------------------*/
/* UTILITY UNLOAD DD STATEMENTS */
/* ALTER UNLOAD AND BMCUNLOAD DATASETS ARE Commented OUT. */
/* IBM REORG AND BMC REORG DATASETS ARE NOT Commented OUT */
/* UNLESS STACKED ON TAPE AND MULTI-STEP JOB. */
/*--------------------------------------------------------------------*/
/$0004 DD DSN=RDAJXN2.TBMC001.SYSREC.$0004,
  /* DISP=(MOD,DELETE,DELETE), */
  /* SPACE=(TRK,(1,1)), */
  /* UNIT=SYSDA */
/*--------------------------------------------------------------------*/
/* UTILITY WORK DD STATEMENTS */
/*--------------------------------------------------------------------*/
/$SYSUT001 DD DSN=RDAJXN2.TBMC001.STEP4.$SYSUT001,
  /* DISP=(MOD,DELETE,DELETE), */
  /* SPACE=(TRK,(1,1)), */
```
Worklist format JCL

The BMCTRIG utility can generate worklist-format JCL that is exclusive to the administrative products. You can also generate worklist-format jobs independent of the BMCTRIG utility. Worklist JCL offers the benefits of having one single job, no step or object limit, and event logging, as shown in Table 74.

Table 74  Worklist JCL benefits

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a single job</td>
<td>With worklist JCL, you generate one job that contains all utility steps for all objects that met or exceeded an exception in the BMCTRIG utility. Because you generate only one job, you plan only one job for the scheduler.</td>
</tr>
<tr>
<td>no step or object limit</td>
<td>Unlike standard JCL, JCL in worklist format does not limit the number of steps to generate in one job. Because worklist JCL does not have steps, one job can contain hundreds of utilities and objects (or more).</td>
</tr>
<tr>
<td>actions are logged in the EVENTS table</td>
<td>Because the product logs all utilities in the EVENTS table, the user can query the EVENTS table to display the various utilities that have run against specific objects. The EVENTS table contains a history of the utilities that have run in DASD MANAGER PLUS.</td>
</tr>
<tr>
<td>control job names</td>
<td>With worklist JCL, you can name the job that DASD MANAGER PLUS generates.</td>
</tr>
<tr>
<td>restart jobs easily</td>
<td>With worklist JCL, you need only to add the restart parameters (see the DASD MANAGER PLUS for DB2 Reference Manual) to restart a job.</td>
</tr>
</tbody>
</table>

To generate worklist JCL instead of standard JCL, specify Standard JCL N on the JCL Generation panel of the BMCTRIG utility when you are setting the BMCTRIG parameters.
NOTE
DASD MANAGER PLUS bypasses generating a SYNC command in the worklist if the previous command in the worklist was also a SYNC.

Generating worklist JCL

Figure 184 illustrates worklist JCL generation.

Figure 184  Generating worklist JCL
**JCL generation options**

The BMCTRIG JCL generation options are display under the multiple sections beginning with Job Generation Options on the BMCTRIG Syntax Options panel (Figure 185).

**Figure 185  Worklist JCL Generation options panel**

1. In the **Standard JCL** field, type **N** for worklist format.

For information about all of the other JCL generation options, see the *DASD MANAGER PLUS for DB2 Reference Manual*. 

---

540  DASD MANAGER PLUS for DB2 User Guide
To generate worklist format JCL, specify the Util DSN data set name. Util DSN is the name of the data set in which to place the generated utility job. When Standard JCL is N, Util DSN must be an existing partitioned data set, or a physical sequential data set. Specify a member name if the data set is partitioned because the product builds the utility job into that data set as the member name that you specify.

---

**WARNING**

To avoid potential job errors, regenerate your BMCTRIG JCL before you run BMCTRIG jobs. Otherwise, you should review DD requirements for potential changes. With DASD MANAGER PLUS version 8.1 or later, the UTDEF DD must reference the XML data set by default instead of the CLIST.

---

**Example procedures**

The worklist format job that Figure 186 shows is the output of procedures that are described in the following tasks:

- “To set up a REORG corrective action” on page 557
- “To generate the BMCTRIG service that uses the REORG corrective action” on page 565

**Figure 186  Generated worklist JCL (part 1 of 5)**

```
//RDAJXN2U JOB (5213), 'UTILITY-JENTEST'.
// CLASS=A, MSGCLASS=X
//*
/*. JOBPARM SYSAFF=DB2A
/*. //***************************************************
/*. /* Created by: RDAJXN2
/*. /* Timestamp: 03/06/2005.11.20.30
/*. /* Environment: ISPF 5.0MVS BATCH
/*. /* Release: V8.1.00 10/10/2005
/*. /* DB2 Version: 711
/*. //***************************************************
/*. //--------------------------------------------------------------------
/*. /* DASD MANAGER WORKLIST 'EXECUTION
/*. //--------------------------------------------------------------------
/*. //STEP1 EXEC PGM=AEXEMAIN, REGION=0M,
/*. // PARM='DS622EAE'
/*. //STEPLIB DD DISP=SHR, DSN=AUS.ASU810.D71.LOAD
/*. // DD DISP=SHR, DSN=SYS3.DEAE.DSNEXIT
/*. // DD DISP=SHR, DSN=SYS2.DB2V71M.DSNLOAD
/*. // DD DISP=SHR, DSN=DB2.MSTRPLAN.LOAD
/*. // ABNLIGNR DD DUMMY
/*. // DSSPRINT DD SYSOUT=* 
/*. // SYSUDUMP DD SYSOUT=* 
/*. // SYSTERM DD SYSOUT=* 
/*. // UTPRINT DD SYSOUT=* 
```
Figure 186  Generated worklist JCL (part 2 of 5)

```plaintext
//SYSOUT DD SYSOUT="*
//AEXIN DD * ASU
SSID DEAE
DASDTRIG
   LINES 54 STATS
UTILITYID TJENTEST.S001
COPYOPT ACP$OPTS
UNLOADOPT ADU$OPTS
LOADOPT AMU$OPTS
REORGPT ARU$OPTS
CHECKOPT ACK$OPTS
RECOVEROPT AFR$OPTS
//SYSIN DD SPACE=(CYL,(15,15)),UNIT=SYSDA,DISP=(NEW,PASS),
   DSORG=PS,LRECL=80,BLKSIZE=3200,RECFM=FB
//SYSPRINT DD SPACE=(CYL,(15,15)),UNIT=SYSDA,DISP=(NEW,PASS)
//*JGENSRT DD SYSOUT="*,
//*         DSORG=PS,LRECL=80,BLKSIZE=6160,RECFM=FB
//SYSTSIN DD SPACE=(TRK,(1,1)),UNIT=SYSDA,DISP=(NEW,PASS),
   //DSORG=PS,LRECL=80,BLKSIZE=3200,RECFM=FB
//SYSTSPRT DD DISP=(NEW,PASS),
   //DSN=&&TSPRT1,
   //SPACE=(CYL,(5,5)),UNIT=SYSDA,
   //DSORG=PS,LRECL=137,BLKSIZE=3155,RECFM=VBA
//AEXPRINT DD SYSOUT="*
//ACPDDGLP DD *
   DEFINE GDG (NAME(&BASE) LIMIT(10) SCR)
//ACPDDGLB DD *
   DEFINE GDG (NAME(&BASE) LIMIT(10) SCR)
//ACPDDGRP DD *
   DEFINE GDG (NAME(&BASE) LIMIT(10) SCR)
//ACPDDGRB DD *
   DEFINE GDG (NAME(&BASE) LIMIT(10) SCR)
//WORKLIST DD *
   -TIME 000001 '2003-03-06-11.19.25.628396'
   57489450
   * FOR TRIGGER EXCEPTION
   25145407
   74734598
   -COPY 000001
   COPY TABLESPACE JEND30.C791278
   SHRLEVEL REFERENCE
   CONCURRENT
   COPYDDN (C0001)
   11279953
   -JCLP 000001 COPY DDNAME C0001 DSNPREF
   &PREFIX..&SSID..&OBNOD
   -JCLP 000001 COPY DDNAME C0001 DEVTYPE SYSDA
   20787392
   -REOR 000002
   REORG TABLESPACE JEND30.C791278
   UNLDDN S0002
```
Figure 186  Generated worklist JCL (part 3 of 5)

<table>
<thead>
<tr>
<th>WORKDDN</th>
<th>SYST001</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORTDATA</td>
<td></td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>SYSDA</td>
</tr>
<tr>
<td>UNLOAD CONTINUE</td>
<td>99726329</td>
</tr>
</tbody>
</table>

-COPY 000003
COPY TABLESPACE JEND3O.JENS0130
SHRLEVEL REFERENCE
CONCURRENT
COPYDDN (C0003) 08926708

-JCLP 000003 COPY DDNAME C0003 DSNPREF
&PREFIX..&SSID..&OBNOD
-JCLP 000003 COPY DDNAME C0003 DEVTYPE SYSDA 20525244

-REOR 000004
REORG TABLESPACE JEND3O.JENS0130
UNLDDN S0004
WORKDDN SYST001
SORTDATA
SORTDEVT SYSDA
UNLOAD CONTINUE 59782702

//--*----------------------------------------------------------
//--*  UTILITY UNLOAD DD STATEMENTS
//--*----------------------------------------------------------
//--* M = ESTIMATE IS THE MAXIMUM DATASET SIZE FOUND FOR THIS WORKDD TYPE
//--* S0002 DD DSN=RDAJXN2.JENTEST.SYSREC.S0002,
//--*   DISP=(NEW,CATLG,CATLG),
//--*   SPACE=(CYL,(13,5),RLSE), ESTIMATE-C/M
//--*     UNIT=SYSDA
//--* M = ESTIMATE IS THE MAXIMUM DATASET SIZE FOUND FOR THIS WORKDD TYPE
//--* S0004 DD DSN=RDAJXN2.JENTEST.SYSREC.S0004,
//--*   DISP=(NEW,CATLG,CATLG),
//--*   SPACE=(CYL,(12,5),RLSE), ESTIMATE-C/M
//--*     UNIT=SYSDA
//--*----------------------------------------------------------

//--*----------------------------------------------------------
//--*  UTILITY WORK DD STATEMENTS
//--*----------------------------------------------------------
//--* M = ESTIMATE IS THE MAXIMUM DATASET SIZE FOUND FOR THIS WORKDD TYPE
//--* SYSUT001 DD DSN=RDAJXN2.JENTEST.STEP1.SYSUT001,
//--*   DISP=(NEW,CATLG,CATLG),
//--*   SPACE=(CYL,(10,2)), ESTIMATE-C/M
//--*     UNIT=SYSDA
//--*----------------------------------------------------------

//--*----------------------------------------------------------
//--*  UTILITY SORTOUT WORK DD STATEMENTS
//--*----------------------------------------------------------
//--* M = ESTIMATE IS THE MAXIMUM DATASET SIZE FOUND FOR THIS WORKDD TYPE
//--* SORTOUT DD DSN=RDAJXN2.JENTEST.STEP1.SORTOUT,
//--*   DISP=(NEW,CATLG,CATLG),
//--*   SPACE=(CYL,(10,2)), ESTIMATE-C/M

Chapter 8  Analyzing objects by using BMCTRIG  543
Figure 186  Generated worklist JCL (part 4 of 5)

```jcl
// ** UTILITY COPY DD STATEMENTS
// *--------------------------------------------------------------------
// ** DATA SET SIZE REQUIRED FOR DD C0001 IS AN ESTIMATE
//C0001 DD DSN=RDAJXN2.DEAE.JEND30.C791278.C0001,
//         DISP=(NEW,CATLG,CATLG),
//         SPACE=(CYL,(3,1),RLSE), ESTIMATE=C/H
//         UNIT=SYSDA
// ** DATA SET SIZE REQUIRED FOR DD C0003 IS AN ESTIMATE
//C0003 DD DSN=RDAJXN2.DEAE.JEND30.JENS0130.C0003,
//         DISP=(NEW,CATLG,CATLG),
//         SPACE=(CYL,(2,1),RLSE), ESTIMATE=C/H
//         UNIT=SYSDA
// *--------------------------------------------------------------------
// ** END OF JOBSTEP
// *--------------------------------------------------------------------
// ** DATASET CLEANUP STEP WILL EXECUTE IF PREVIOUS CONDITION CODE
// ** IS 4 OR LESS.
// *--------------------------------------------------------------------
//CLEANUP  EXEC PGM=IEFBR14,
//         COND=(4,LT)
// *--------------------------------------------------------------------
// ** UTILITY UNLOAD DD STATEMENTS
// ** ALTER UNLOAD AND BMCUNLOAD DATASETS ARE COMMENTED OUT.
// ** IBM REORG AND BMC REORG DATASETS ARE NOT COMMENTED OUT
// ** UNLESS STACKED ON TAPE AND MULTI-STEP JOB.
// *--------------------------------------------------------------------
//S0002 DD DSN=RDAJXN2.JENTEST.SYSREC.S0002,
//         DISP=(MOD,DELETE,DELETE),
//         SPACE=(TRK,(1,1)),
//         UNIT=SYSDA
//S0004 DD DSN=RDAJXN2.JENTEST.SYSREC.S0004,
//         DISP=(MOD,DELETE,DELETE),
//         SPACE=(TRK,(1,1)),
//         UNIT=SYSDA
// *--------------------------------------------------------------------
// ** UTILITY WORK DD STATEMENTS
// *--------------------------------------------------------------------
//SYSUT001 DD DSN=RDAJXN2.JENTEST.STEP1.SYSUT001,
//         DISP=(MOD,DELETE,DELETE),
//         SPACE=(TRK,(1,1)),
//         UNIT=SYSDA
// *--------------------------------------------------------------------
// ** UTILITY SORTOUT WORK DD STATEMENTS
// *--------------------------------------------------------------------
//SORTOUT DD DSN=RDAJXN2.JENTEST.STEP1.SORTOUT,
//         DISP=(MOD,DELETE,DELETE),
//         SPACE=(TRK,(1,1)),
//         UNIT=SYSDA
```
Generating worklist JCL

The generated worklist JCL that Figure 184 shows contains all worklist commands and objects to run the utility against, in addition to all of the data sets necessary to run the utility.

Viewing exceptions that BMCTRIG produces

DASD MANAGER PLUS provides many methods for viewing the exceptions that BMCTRIG detects. For example, you can view the exceptions that appear in the BMCTRIG job output. BMCTRIG stores the exceptions in the exceptions table. You can view the contents of the exceptions table online or from batch reports. Additionally, you can modify some of the columns of active exceptions rows online to alter the output that a subsequent BMCTRIG job will produce when using the RESUME Y processing option.

To view the exceptions report in the BMCTRIG output

On the BMCTRIG main menu, specify Y at Report.

When BMCTRIG runs, it reports any exceptions that it identifies by writing messages to the AEXPRINT DD data set (when you run it under AEXEMAIN). You can check the BMCSTATS report to view the statistics for that object. Figure 187 shows a sample list of exceptions.
Figure 187  Exceptions reported in AEXPRINT

<table>
<thead>
<tr>
<th>Exception</th>
<th>Qualifier/Object</th>
<th>Part</th>
<th>Type</th>
<th>Current</th>
<th>Compare</th>
<th>Trigger</th>
<th>Action Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCTRIG</td>
<td>T2U001</td>
<td>000</td>
<td>IX</td>
<td>412</td>
<td>N/A</td>
<td>00000030 DRGSK006</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U001</td>
<td>000</td>
<td>IX</td>
<td>90</td>
<td>2</td>
<td>00000030 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U004</td>
<td>000</td>
<td>IX</td>
<td>240000</td>
<td>30000</td>
<td>00000030 DRGSK002</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U004</td>
<td>000</td>
<td>IX</td>
<td>99</td>
<td>N/A</td>
<td>00000020 DRGSK002</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>001</td>
<td>IX</td>
<td>61</td>
<td>N/A</td>
<td>00000050 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>001</td>
<td>IX</td>
<td>44740</td>
<td>99448</td>
<td>00000020 DRGSK002</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>001</td>
<td>IX</td>
<td>207</td>
<td>126</td>
<td>00000030 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>002</td>
<td>IX</td>
<td>99</td>
<td>N/A</td>
<td>00000020 DRGSK002</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>002</td>
<td>IX</td>
<td>65</td>
<td>N/A</td>
<td>00000050 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>002</td>
<td>IX</td>
<td>47549</td>
<td>105552</td>
<td>00000020 DRGSK002</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>002</td>
<td>IX</td>
<td>219</td>
<td>132</td>
<td>00000030 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>3</td>
<td>1</td>
<td>00000030 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>3</td>
<td>2</td>
<td>00000000 DRGSK002</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>240000</td>
<td>30000</td>
<td>00000030 DRGSK002</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>39</td>
<td>27</td>
<td>00000030 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>3</td>
<td>2</td>
<td>00000000 DRGSK002</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>240000</td>
<td>30000</td>
<td>00000030 DRGSK002</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>66</td>
<td>N/A</td>
<td>00000020 DRGSK002</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>97</td>
<td>N/A</td>
<td>00000050 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>42</td>
<td>30</td>
<td>00000030 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>42</td>
<td>15</td>
<td>00000030 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>88</td>
<td>N/A</td>
<td>00000050 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>88</td>
<td>N/A</td>
<td>00000005 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>333</td>
<td>206</td>
<td>00000030 DRGSK003</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>T2U011</td>
<td>003</td>
<td>IX</td>
<td>333</td>
<td>72</td>
<td>00000030 DRGSK003</td>
<td></td>
</tr>
</tbody>
</table>

Figure 187 contains the following fields:

- **Exception** is the name of the exception.
- **Qualifier/Object** is the fully qualified object name.
- **Part** is the partition number, if any.
- **Type** is the two-character abbreviation for the DB2 object type.
- **Current** is the object’s current attribute value.
- **Compare** is the previous value, if applicable, that BMCTRIG compares with the current value. BMCTRIG uses this value to calculate some exceptions based on ratios of current and previous values.
- **Trigger** is the value that you specified for the exception.
- **Action Taken** is the latest action, if any (for example, whether you generated a REORG or COPY utility, or whichever action is specified in the corrective action).
To view the exceptions report

Another method for viewing the information in the exceptions table is by using the exceptions report, which you can run in batch.

1. On the DASD MANAGER PLUS main menu, select Reports.


3. (optional) Enter criteria to limit the rows reported.

You can limit the rows by object name or pattern, object type, exception date, and exception type. You can run the report in foreground or in batch.

For more information about DASD MANAGER PLUS reports, see Chapter 10, “Producing reports.”

To view exceptions by object online

1. On the DASD MANAGER PLUS main menu, select Statistics/Catalog Update/Space Estimation List.

2. Type the object name or pattern for objects that you want to view, and press Enter.

The Display DB2 Object List panel appears.

3. Type E in the Act field next to the object to display exceptions table data for that object.

You can view both active and inactive exceptions for an object.

To view active exceptions

You can view active exceptions and optionally modify the priority and corrective actions for them.

1. On the DASD MANAGER PLUS main menu, select Thresholds, Corrective Actions, and Priorities.

2. Then select Active Exceptions.

The Active Exceptions panel appears (Figure 188).
3 After making updates to the selection criteria, press **Enter** to retrieve the rows.

By default, you see all active exceptions. You might want to limit the selection criteria.

4 To view or edit an exception, type **S** or **E** on the Act field next to it.

The Object Exception Detail panel appears (Figure 189).

**Figure 189 Object Exception Detail panel**

| DEAE ------------------------ Object Exception Detail ------------------------- |
|-----------------------------|-----------------------------|-----------------------------|
| Object Type: **TS**          | Object Qualifier: **CCBDBSIM** | Object Name: **CCBTSSIM** |
| Partition: 0                 | Exception Name: **EXTENTS**  | Exception Type: **REORG**   |
| Current Value: 0             | Compare Value: 0            | Trigger Value: 0            |
| JCL Dataset:                 | Action Taken:              | Registration ID: 0          |
| Active: **Y**                | (Y/N)                      | Priority: 0                 |
| Corrective Action: (Enter ? for a selection list) | |

5 *(optional)* In the **Active** field, to deactivate the exception, specify **N**.
Working with reorganization guidelines for thresholds

No strict rules exist for determining when objects require a reorganization, but guidelines can help you identify the need for a reorganization. Often, the statistical trends are more useful than the statistics themselves.

Consider the following statistics to determine when to initiate a reorganization for a particular index or table space:

- index reorganization
  
  Percent Pseudo-Deleted RIDs (PSEUDODL) is the percentage of rows that have been pseudo-deleted in an index. As delete processing occurs, index entries are deleted from leaf pages. However, DB2 continues to keep leaf pages as long as one index entry exists on that page. If heavy insert processing occurs followed by a large number of deletes, the number of pages that DB2 scans when searching for an index entry can be high because only a few index entries might exist on each leaf page. In this case, you might need to reorganize the index.

  IBM recommends reorganizing an index when %Pseudo-deleted Keys is greater than ten percent. IBM also recommends to consider using the REUSE option with IBM REORG in this situation.

This exception is true if

\[
\frac{RS\_INDEXPART\_PSEUDO\_DEL\_ENTRIES}{RS\_INDEXPART\_CARDF} \geq specifiedValue
\]
— Leaf TotalOff (LEAFTOFF) is the percentage of leaf pages that are not in their optimal positions. This value includes leaf pages that are physically located near the previous leaf pages as well as leaf pages that are physically located far from the previous leaf page.

This exception is a much more effective measure of the disorganization of an index than the traditional LEAFDIST statistic. LEAFDIST measures the number of pages between successive leaf pages. This value can become quite large with a single page split (such as a single page being out of position). Because LEAFDIST is the average distance between successive leaf pages, the value might not change significantly from the first page split to the 100th page split. Instead, LeafTotalOff measures the percentage of pages in the index that are out of position, giving a more effective view of the overall organization of an index.

The exception is true if

\[
\frac{(RS\_INDEXPART.LEAFNEAR + RS\_INDEXPART.LEAFFAR)}{RS\_INDEXPART.NLEAF} \geq \text{specifiedValue}
\]

— Leaf FarOff (LEAFFOFF) is the percentage of leaf pages that are located physically far away from previous leaf pages (for successive active leaf pages accessed in an index scan).

The exception is true if

\[
\frac{RS\_INDEXPART.LEAFFAR}{RS\_INDEXPART.NLEAF} \geq \text{specifiedValue}
\]

— The Leafdist (LEAFDIST) value is 100 times the average number of pages between successive leaf pages. The minimum number depends primarily on the FREEPAGE value, which is the leaf distribution immediately after a reorganization. The larger the number beyond the minimum, the more disorganized the index is. Consider reorganizing the index when the leaf distribution increases 200 beyond the minimum.

— Dataset Extents (DSEXTENTS) is the number of extents in a data set. The maximum number of supported extents is 7257. Data sets with a large number of extents might indicate a potential problem that a reorganization can help resolve. You might not want to reorganize based solely on the number of extents, but when you do reorganize, try to eliminate extents by using the RESIZE parameter on the DB2 REORG utility or the BMC Software REORG PLUS utility.

— The AREO* pending (AREOPEND) value indicates whether to raise an exception if the table space is in advisory REORG-pending status.
— The AREOR pending (ARERPEND) value indicates whether a table space, index, or partition needs to be reorganized for optimal performance and whether to apply pending definition changes.

— The Append inserts (APPNDINS) is the percentage of index entries that have been inserted since the last REORG, REBUILD INDEX, or LOAD REPLACE on the index space or partition that have a key value that is greater than the maximum key value in the index or partition.

— The Mass del reorg (REORMDEL) value indicates whether to evaluate for any mass deletion since the last REORG (evaluated at the partition level).

— The Mods since reorg (REORMODS) value identifies whether to evaluate the percentage of keys that have been modified since the last REORG (evaluated at the partition level).

— The Ins since reorg (REORINS) value identifies whether to evaluate the percentage of keys that have been inserted since the last REORG (evaluated at the partition level).

— The Del since reorg (REORDEL) value identifies whether to evaluate the percentage of keys that have been deleted since the last REORG (evaluated at the partition level).

— The Reorg Leaf (REORLEAF) value identifies whether to evaluate the percentage of total pages in comparison to number of active leaf pages (evaluated at the partition level).

— The Level Min (LEVELMIN) exception identifies indexes whose levels are greater than the number that are required to reorganize the index (evaluated at the index partition level if you specify Partlvl Y; otherwise, it is evaluated at the index level).

— Reorgspace (REORSPACE) indicates the estimated amount of space that the index should occupy after a reorganization. This value can be greater than or less than the space that is being used. The value might be greater if a large number of rows has been added to the table. A reorganization can add free space back into the index, based on the FREEPAGE and PCTFREE parameters.
table space reorganization

— Reorg Pend (REORPEND) is the indicator that the table space is in REORG PENDING status.

— FARINDREF (used in Farind (FARIND) and Totalind (TOTALIND)) is the number of referenced rows that are more than 16 pages from the original page. NEARINDREF (used in Totalind (TOTALIND)) is the number of referenced rows that are less than 16 pages from the original page. A large FARINDREF value can indicate an increase in I/O to the data set that are caused by indirect references to updated VARCHAR columns.

— FAROFFPOS (used in Faroff (FAROFF) and Totaloff (TOTALOFF)) is the number of referenced rows that are greater than or equal to 16 pages from their optimal positions. NEAROFFPOS (used in Totaloff (TOTALOFF)) is the number of referenced rows that are less than 16 pages from their optimal positions. The product collects these statistics for indexes, but the statistics apply to the data. The statistics indicate the degree of clustering, and the DB2 Optimizer uses them for clustering and nonclustering indexes. FAROFFPOS and NEAROFFPOS are used as exceptions only for clustering indexes, and PCTCLUST is preferred. Both statistics will approach zero for clustered data.

— Reorgspace (REORSPACE) indicates the estimated amount of space that the table space should occupy after a reorganization. This value can be greater than or less than the space that is being used. The value might be greater if a large number of rows has been added to the table. A reorganization can add free space back into the table space, based on the FREEPAGE and PCTFREE parameters.

— Pctclus (PCTCLUS) applies to the degree of clustering of the data. This value is collected for a clustering index. You might want to begin by setting a low threshold, such as 85 percent, and handle the worst-case table spaces first.

— Dataset Extents (DSEXTENTS) is the number of extents in a data set. The maximum number of extents supported is 7257. Data sets with a large number of extents might indicate a potential problem that a reorganization can help resolve. You might not want to reorganize based solely on the number of extents, but when you do reorganize, you should attempt to eliminate extents at that time. You can do this by using the RESIZE parameter on the DB2 REORG utility or the BMC Software REORG PLUS utility.

— The AREO* pending (AREOPEND) value indicates whether to raise an exception if the table space is in advisory REORG-pending status.

— The AREOR pending (ARERPEND) value indicates whether a table space, index, or partition needs to be reorganized for optimal performance and whether to apply pending definition changes.
— The Mass del reorg (REORMDEL) value indicates whether to evaluate for any mass deletion since the last REORG (evaluated at the partition level).

— The LOB ORGRATIO (ORGRATIO) value indicates the percentage of organization in the LOB table space and this value is triggered if it is lower than the value that you entered.

— The LOB Freespace (LOBFRSPC) value indicates whether a table space needs to be reorganized based on the percentage of the LOB that is freespace.

— Unclust inserts (UNCLUST) is the percentages of unclustered inserts. Use this exception to initiate a reorganization of a table space that has a high number of unclustered inserts.

— The Mods since reorg (REORMODS) value identifies whether to evaluate the percentage of rows that have been modified since the last REORG (evaluated at the partition level).

— The Ins since reorg (REORINS) value identifies whether to evaluate the percentage of rows that have been inserted since the last REORG (evaluated at the partition level).

— The Del since reorg (REORDEL) value identifies whether to evaluate the percentage of rows that have been deleted since the last REORG (evaluated at the partition level).

— Pct over alloc (REORDSPC) value identifies the percentage by which space allocated to a table space or partition exceeds space occupied by row data.

**TIP**

BMC provides a set of thresholds for typical exceptions. You can use this set as an initial set with SYSTEMTRIGGERS Y. For additional information about creating and adding thresholds, see “Setting thresholds in BMCTRIG syntax” on page 485.

---

**Generating an automatic reorganization with BMCTRIG**

Use this procedure as a guideline to automate utility generation. This procedure sets up and generates an action job that collects current statistics (BMCSTATS), sets statistical thresholds for specific DB2 objects (BMCTRIG), and generates a reorganization maintenance utility job if the objects exceed the thresholds. BMCTRIG will apply thresholds using RTS and BMCSTATS data. (Specify the maintenance utility job in a separate action.)
This job first updates the DASD MANAGER PLUS database with BMCSTATS. This example job uses BMCTRIG to scan the DASD MANAGER PLUS database to detect DB2 objects when any of the following exception conditions exist:

- 10 percent of rows more than 16 pages from their original pages
- 10 percent of table rows that are not in optimal positions
- at least 20 percent larger or smaller than required
- in 5 or more extents
- 5 percent of rows over 16 pages from their optimal positions
- a cluster ratio of 95 percent or less

**To generate an automatic reorganization with BMCTRIG**

1. Create an action as described in “Creating an action” on page 271.

2. Edit the action services to include BMCSTATS and BMCTRIG services, or copy an action to use its BMCSTATS and BMCTRIG services.

   For more information, see “Creating an action by copying an existing action” on page 273 or “Copying Services from one action into another” on page 285.

3. On the Action List panel, type E (or S) in the Act field beside the action name and press Enter.

   The Edit Action Services panel appears.

4. Type E (Edit) in the Act field by the BMCTRIG service, and press Enter.

   The Service Syntax List panel appears.

5. Type E next to the service syntax, and press Enter.

   The BMCTRIG main panel appears.

6. In the Processing Options section, select Evaluate and Generate.

7. In the What to Analyze section, specify the following options:

   A. In the System Triggers field, type Y.

   B. In the DB2 RTS field, type Y.

8. Press DOWN to scroll to the Exclude Objects from Generation Options section.
9 In the **Exclude Objects from Generation Options** section, specify the following options:

**A** Exclude table spaces and table space partitions below and above a certain size from BMCTRIG utility generation, as follows:
- In the **TS Tracks <** field, specify a minimum size.
- In the **TS Tracks >** field, specify a maximum size.

**B** Exclude indexes and index partitions below and above a certain size, as follows:
- In the **IX Tracks <** field, specify a minimum size.
- In the **IX Tracks >** field, specify a maximum size.

**C** In the **ExcludeEmpty** field, type Y to exclude all table spaces and table space partitions that have zero cardinality from BMCTRIG utility generation.

**D** In the ** Archived** field, type Y to exclude utility generation for archived or migrated objects, or type N to include archived objects.

10 Press **DOWN** to scroll to the **Job Generation Options** section.

11 In the following sections, specify the options as described in “Generating a BMCTRIG job” on page 524:

- Job Generation Options
- Standard JCL Options
- Output Dataset Options
- Jobcard(s) for BMCTRIG generated job(s)

12 Press **DOWN** to scroll to the **DB2 RTS or BMCSTATS Reorg Exceptions** segment of the **Reorg-Related Override Exceptions** section.

13 In the **DB2 RTS or BMCSTATS Reorg Exceptions** segment, specify the following options:

**A** In the **FARIND** field, type 10.

**B** In the **TOTALIND** field, type 10.

14 In the **BMCSTATS Specific Reorg Exceptions** segment, specify the following options:

**A** Specify shared table space and index exceptions, as follows:
- In the **REORGSPACE** field, type 20.
- In the **EXTENTS** field, type 5.
NOTE
If the index or table space is not partitioned, the product retrieves the value of Extents from the ICF Catalog at execution time and applies it to each data set.

B Specify table space exceptions, as follows:

- In the FAROFF field, type 5.
- In the PCTCLUST field, type 95.

15 Press END to return to the DASD MANAGER PLUS main menu.

16 Set up the related corrective action as described in “To set up a REORG corrective action” on page 557.

17 Press END to return to the DASD MANAGER PLUS main menu.

18 On the DASD MANAGER PLUS main menu, select Utility Actions (WORKIDs) and press Enter.

19 On the Actions panel, select the action that you created in step 1 on page 554.

20 Generate the BMCTRIG job as described in “To generate the BMCTRIG service that uses the REORG corrective action” on page 565.

21 Analyze the exceptions, as follows:

- To view exceptions in the job output, enter FIND EXCEPTIONS on the COMMAND line.

- To produce an exceptions report later, see “Statistics exception report” on page 617.

Setting up a REORG corrective action

This section describes how to set up a corrective action that contains the services (utilities) to run when a specified DB2 object exceeds the thresholds. You establish the thresholds by using a BMCTRIG utility. (See “Generating an automatic reorganization with BMCTRIG” on page 553.) Be aware that, unlike a regular action, a corrective action specifies actions but often does not specify the DB2 objects to process.
To set up a REORG corrective action

1. Create a BMCTRIG job as described in “Generating an automatic reorganization with BMCTRIG” on page 553.

2. Create an action as described in “Creating an action” on page 271.

3. Edit the action services to add the REORG utility to the action, as follows:
   - **A** Type E (or S) in the Act field of the blank service.
   - **B** Tab to the Service field and type REORG.
   - **C** (optional) In the Type field, type the two-letter abbreviation for the object type, and press Enter.

**NOTE**

This procedure uses an action named SKREORG.
Setting up a REORG corrective action

**NOTE**
You can specify **Object Name** and **Object Type** by using an object set. See “Creating an object set for an action service” on page 336.

The object type, object name, and partition are automatically supplied during execution of the BMCTRIG action step that references this corrective action. If you specify object type, object name, or partition here with or without an object set, BMCTRIG ignores them.

For more information about editing action services, see “Creating an action by copying an existing action” on page 273.

4 Type **E** (or **S**) in the **Act** field of the REORG service.

The Service Syntax List appears.

5 Create the syntax or select existing syntax and press **Enter**.

The REORG panel appears (Figure 190).

**Figure 190  REORG panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>REORG</th>
<th>Service Syntax: REORG.SKHREORG</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDK</td>
<td></td>
<td>Scroll ===&gt; CSR</td>
</tr>
<tr>
<td>Enter data, then press end.</td>
<td>More: +</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------</td>
<td></td>
</tr>
</tbody>
</table>
| What to Reorg | __________________________ansi
| Beginning PART . .  | (1-4096 First or only partition in inclusive range) |
| Ending PART . . . | (1-4096 Ending partition in inclusive range) |
| CLONE . . . . . N | (Y/N Process Clone objects instead of Base objects) |
| Shlevel Options | (N/R/C N-None, R-Ref, C-Change) |
| Change Options | |
| DEADLINE . . . NONE | |
| MAXRO . . . 300 | (0-86400/DEFER Max Seconds in Read Only) |
| DRAIN . . . W | (W/A W-Writers, A-All) |
| LONLGO . . . C | (C/T/D C-Continue, T-Term, D-Drain) |
| DELAY . . . 1200 | (Number of seconds) |
| TIMOUT . . . A | (A/T A-Abend, T-Term) |
| DRAIN_WAIT . . | (0-1800 seconds before drain timeout) |
| RETRY . . . | (0-255 maximum number of retries) |
| RETRY_DELAY . . | (1-1800 seconds between retries) |
| FASTSWITCH . . | (Y/N Specify fast switch) |
| MAPPINGTABLE . | |
6 Specify **Shrlevel** options.

The SHRLEVEL field on the REORG panel indicates how the reorganization occurs and determines the type of access allowed during reorganization. Table 76 lists the choices for this field.

### Table 76  Shrlevel access during reorganization

<table>
<thead>
<tr>
<th>Choice</th>
<th>Access during Unload</th>
<th>Access during Reload</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>read only</td>
<td>no access</td>
</tr>
<tr>
<td>R (Reference)</td>
<td>read only</td>
<td>read only</td>
</tr>
<tr>
<td>C (Change)</td>
<td>read/write</td>
<td>read/write</td>
</tr>
</tbody>
</table>

**NOTE**
- You must set **Shrlevel=N** for LOB table spaces.
- **Shrlevel=C** (Change) will not be generated on an object that is not logged. Instead, this option will be changed automatically to **Shrlevel=R** (Reference).

Table 77 describes the Shrlevel options.

### Table 77  Shrlevel options (part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEADLINE</td>
<td>Type the deadline for the completion of the SWITCH phase. If DB2 estimates that the SWITCH phase will not finish by this date or time, DB2 issues the messages that the –DISPLAY UTILITIES command would issue and then stops the reorganization.</td>
</tr>
<tr>
<td>MAXRO</td>
<td>Type the maximum number of seconds for the last iteration of log processing, during which applications have read-only access. The default is 300 seconds.</td>
</tr>
<tr>
<td>DRAIN</td>
<td>Specify the applications to drain at the end of the log phase after the MAXRO threshold is reached, and when the last iteration of the log is done.</td>
</tr>
<tr>
<td>LONGLOG</td>
<td>Type the action to take if the reorganization’s reading of the log is not catching up to the application’s writing of the log.</td>
</tr>
<tr>
<td>DELAY</td>
<td>Type the number of seconds to wait after the LONGLOG condition occurs before the LONGLOG action is initiated.</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>Specify an action to perform if the reorganization experiences a timeout condition during a drain in either the LOG or SWITCH phase.</td>
</tr>
<tr>
<td>DRAIN_WAIT</td>
<td>Type the number of seconds that the utility should wait for DML when draining. If you specify 0, regular draining occurs, using the IRLMRWT and UTIMOUT values. The range of values is 0 through 1800.</td>
</tr>
<tr>
<td>RETRY</td>
<td>Type the maximum number of times to retry.</td>
</tr>
</tbody>
</table>
Setting up a REORG corrective action

### Table 77  Shrlevel options (part 2 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETRY_DELAY</td>
<td>Type the time in seconds between retrying.</td>
</tr>
<tr>
<td>FASTSWITCH</td>
<td>Specify whether the SWITCH phase methodology is enabled. <strong>Yes</strong> is the default. If the methodology is enabled, the fifth qualifier of the data set alternates between I0001 and J0001.</td>
</tr>
<tr>
<td>MAPPINGTABLE</td>
<td>Type the name of a previously defined mapping table for use in mapping between RIDs of data records in the original copy of the area and the corresponding RIDs in the shadow copy. This field supports up to 250 characters.</td>
</tr>
</tbody>
</table>

7 Specify Reorg options.

The Reorg Options panel appears (Figure 191).

**Figure 191  Reorg options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETRY_DELAY</td>
<td>Type the time in seconds between retrying.</td>
</tr>
<tr>
<td>FASTSWITCH</td>
<td>Specify whether the SWITCH phase methodology is enabled. <strong>Yes</strong> is the default. If the methodology is enabled, the fifth qualifier of the data set alternates between I0001 and J0001.</td>
</tr>
<tr>
<td>MAPPINGTABLE</td>
<td>Type the name of a previously defined mapping table for use in mapping between RIDs of data records in the original copy of the area and the corresponding RIDs in the shadow copy. This field supports up to 250 characters.</td>
</tr>
</tbody>
</table>

Table 78 describes the Reorg options.

### Table 78  Reorg options (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REUSE</td>
<td>Type <strong>Y</strong> to reuse and reset data sets managed by DB2. Do not delete and redefine them. SHRLEVEL should be specified as NONE.</td>
</tr>
<tr>
<td>SCOPE</td>
<td>Specify this option to have REORG reorganize all table space partitions specified or only those in REORG pending (REORP or AREO*) status. The default is to reorganize all partitions specified.</td>
</tr>
<tr>
<td>REBALANCE</td>
<td>Specify <strong>Y</strong> to instruct REORG PLUS to create new partition boundaries such that the number of pages are as evenly distributed between partitions as possible. The default is not to change the partition boundaries. This option will be ignored for LOB, XML and XML base objects.</td>
</tr>
<tr>
<td>LOG</td>
<td>Type <strong>Y</strong> to log records during the RELOAD phase of reorganization. If you type <strong>N</strong>, the product does not log records and turns on the copy pending restriction.</td>
</tr>
</tbody>
</table>
Setting up a REORG corrective action

Chapter 8 Analyzing objects by using BMCTRIG

Specify the Copy options.

**Figure 192** shows the Copy options.

**Figure 192  Copy DD options panel**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLOAD</td>
<td>Use the default value C to specify continue after unloading data.</td>
</tr>
<tr>
<td>KEEPDICTIOINARY</td>
<td>Type Y to keep the current compression dictionary</td>
</tr>
<tr>
<td>PREFORMAT</td>
<td>Type Y to preformat the table spaces or index spaces up to the highest allocated RBA (HARBA) with empty pages before reloading the data.</td>
</tr>
<tr>
<td>Resize</td>
<td>Type N, T, I, or A. <strong>Table 69</strong> shows valid values for the <strong>Resize</strong> field. The <strong>Resize</strong> option is valid only for a corrective action that a BMCTRIG service references.</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>Use the default value C to specify continue after unloading data.</td>
</tr>
<tr>
<td>KEEPDICTIOINARY</td>
<td>Type Y to keep the current compression dictionary</td>
</tr>
<tr>
<td>PREFORMAT</td>
<td>Type Y to preformat the table spaces or index spaces up to the highest allocated RBA (HARBA) with empty pages before reloading the data.</td>
</tr>
<tr>
<td>Resize</td>
<td>Type N, T, I, or A. <strong>Table 69</strong> shows valid values for the <strong>Resize</strong> field. The <strong>Resize</strong> option is valid only for a corrective action that a BMCTRIG service references.</td>
</tr>
<tr>
<td>RECOVERYDDN dd1</td>
<td>J (N/J/D N-None, J-JCL, D-Dynamic allocation)</td>
</tr>
<tr>
<td>RECOVERYDDN dd2</td>
<td>J (N/J/D N-None, J-JCL, D-Dynamic allocation)</td>
</tr>
</tbody>
</table>

**NOTE**

If you leave the fields on the Copy Options panel blank, the product uses the values from the POF.

For each option, type J to specify JCL allocation.

If you specify J, the first COPYDDN Options panel appears (Figure 193).
Setting up a REORG corrective action

**Figure 193**  COPYDDN ddname1 JCL panel

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIT</td>
<td>Device type / Unit</td>
</tr>
<tr>
<td>Dname Prefix</td>
<td>(Prefix for output)</td>
</tr>
<tr>
<td>RETPD</td>
<td>1-9999 retention period</td>
</tr>
<tr>
<td>EXPDT</td>
<td>YYYY/DDD or YYDDD Expiration date</td>
</tr>
</tbody>
</table>

---

**B** Specify the unit, output data set name, retention period, or expiration date for ddname1 and press END.

---

**NOTE**

BMC recommends that you specify values for statically allocated image copies in the POF, and specify such values at the action level only when you need overrides.

---

The subsequent panels depend on your selections on the Copy Options panel.

**C** Complete the fields for each options panel, pressing END to display the next panel.

**D** When you press END after completing the last options panel, you return to the BMCTRIG dialog.

**9** Specify the Sort options.

**Figure 194** shows the Sort options.

---

**Figure 194**  Sort options panel

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort JCL</td>
<td>Y/N Put sortwork DD in JCL or N-Set SORTNUM</td>
</tr>
<tr>
<td>SORTNUM</td>
<td>0-99 Number of SORTWK DDs or SORTNUM value</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>Device type for temporary sort data sets</td>
</tr>
<tr>
<td>Parallel Index</td>
<td>Y/N Rebuild indexes in parallel</td>
</tr>
<tr>
<td>WORKDDN1 template</td>
<td>N/D Specify WORKDDN1 template options</td>
</tr>
</tbody>
</table>
Table 79 describes the Sort options.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort JCL</td>
<td>Type Y to generate DD statements for the SORTWK data sets. If you type N, the sort allocates the data sets dynamically. If you specify Y, you must leave the Sort device field blank.</td>
</tr>
<tr>
<td>SORTNUM</td>
<td>Type the number of DD statements to generate or the number of data sets to dynamically allocate for SORTWK.</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>Type the device type for dynamically allocated sort data sets, such as SYSDA or SYSALLDA.</td>
</tr>
<tr>
<td>Parallel index</td>
<td>Type Y to sort the index keys and rebuild more than one index in parallel, as opposed to sequentially. This approach improves performance.</td>
</tr>
<tr>
<td>WORKDDN1 template</td>
<td>Type D to use dynamic allocation for the temporary work file for sort input and to specify Template options. Type N to allocate the temporary work file in the JCL by using options from the JCL Options file (POF).</td>
</tr>
</tbody>
</table>

10 Specify statistics options if you want to collect inline statistics.

Figure 195 shows the Statistics options.

<table>
<thead>
<tr>
<th>Statistics Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATISTICS . . . N</td>
</tr>
<tr>
<td>TABLE . . . . . * N</td>
</tr>
<tr>
<td>SAMPLE . . . . . 25</td>
</tr>
<tr>
<td>INDEX . . . . . * N</td>
</tr>
<tr>
<td>KEYCARD . . . . N</td>
</tr>
<tr>
<td>NUMCOLS . . . . 1</td>
</tr>
<tr>
<td>Auto Decrement . . N</td>
</tr>
<tr>
<td>COUNT . . . . . 10</td>
</tr>
<tr>
<td>REPORT . . . . . N</td>
</tr>
<tr>
<td>UPDATE . . . . A</td>
</tr>
<tr>
<td>HISTORY . . . . A</td>
</tr>
<tr>
<td>FORCEROLLUP . . .</td>
</tr>
</tbody>
</table>

11 Specify Conditional Reorg options.

Figure 196 shows the Conditional Reorg options.
Setting up a REORG corrective action

Figure 196  Conditional Reorg options

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFFPOSLIMIT</td>
<td>10</td>
<td>(0-65535 tablespaces with OFFPOSLIMIT greater)</td>
</tr>
<tr>
<td>INDREFLIMIT</td>
<td>10</td>
<td>(0-65535 tablespaces with INDREFLIMIT greater)</td>
</tr>
<tr>
<td>LEAFDISTLIMIT</td>
<td>200</td>
<td>(0-65535 indexes with LEAFDISTLIMIT greater)</td>
</tr>
<tr>
<td>REPORTONLY</td>
<td>N</td>
<td>(N/Y Report only. No reorganization)</td>
</tr>
</tbody>
</table>

Specify additional options that statistical thresholds must meet before performing reorganization or specify a report and REORG recommendation, as follows:

- If REPORT is Y and the threshold for OFFPOSLIMIT or INDREFLIMIT is exceeded, the product generates a report that recommends reorganization.
- If REPORT is N and the threshold for OFFPOSLIMIT or INDREFLIMIT is exceeded, the product performs reorganization.

12 Press END and you will return to the Edit Action Services panel (Figure 197).

Figure 197  Edit Action Services panel

13 Generate the BMCTRIG service that refers to this corrective action.

See “Generating an automatic reorganization with BMCTRIG” on page 553.
Working with restarts and batch JCL generation

To generate the BMCTRIG service that uses the REORG corrective action

1. Set up a BMCTRIG job as described in “To create an action with BMCTRIG” on page 504.

2. In the JCL Generation options of the BMCTRIG syntax, type the name of the REORG action as the default action.

3. Generate the JCL to run BMCTRIG as described in “To generate JCL to run BMCTRIG” on page 524.

4. Add the job to your schedule.

Working with restarts and batch JCL generation

This section discusses how to restart a BMCTRIG-generated job if you specified Standard JCL N and you are using worklist JCL. This section also provides the steps to use if you specified Standard JCL Y and JCL generation fails.

DASD MANAGER PLUS provides the following ways of restarting a JCL job stream that the BMCTRIG utility generated:

- by main menu option **BMCTRIG Generated Job Restart**
- manually
- in batch
- by CLIST

To run the Restart program online by using ISPF

Run the program from the DASD MANAGER PLUS main menu or from a CLIST outside the product. The external CLIST (RSTRIG) is in the product CLIST library.

To run the Restart program in batch mode

Modify and use the JCL in the Control (CNTL) data set that the product provides. The batch restart member name is BRSTRIG.
Using BMCTRIG-generated job restart

BMCTRIG-Generated Job Restart expedites recovery from failed BMCTRIG jobs by providing both restart and startover capability. Use BMCTRIG Generated Job Restart if you have specified worklist format JCL (Standard JCL N).

**NOTE**

If you have specified Standard JCL Y in a BMCTRIG job, the JCL contains the parameters for restarting the job in comment lines. Restart the generated job as you would any other job.

Instead of restarting the failed job, the DASD MANAGER PLUS Restart program modifies the existing JCL. When you submit the job, it runs in restart or startover mode. The Restart program performs the following tasks:

1. reads the BMCTRIG-generated utility job that failed
2. modifies the generated worklist, execution parameters, and JCL
3. rebuilds the JCL for the Restart job in a separate output file that you specify

Working with restart and startover processing

Restart processing resumes execution of a worklist at a specific command sequence number that you supply. Although the Restart program allows you to restart jobs as often as you need, the program assumes that the point of restart advances. Repeated restarts reduce your ability to restart at a previous restart point.

**NOTE**

To restart or start over again at an earlier point in the worklist, you must use the original job stream or a job stream from a previous restart that contains the restart point in the worklist and JCL. Because restart processing deletes unnecessary worklist commands, BMC recommends using the initial JCL for all restart processing. The initial JCL must always be used for Startover processing.

Startover processing allows you to resubmit a job from the beginning without any changes to the worklist. Table 80 describes the data set dispositions that the Restart program assigns for Startover processing.
Relative generation numbers for GDGs remain unchanged for Startover processing.

### Using the restart program

The Restart program runs against the BMCTRIG-generated job stream, modifying the worklist, execution parameters, and JCL. The program writes the modifications to a separate JCL output file (Figure 198).

![Figure 198 Online restart](image)

If the output data set is sequential and does not exist, the product creates the data set for you. Table 81 lists the data set dispositions that the Restart program assigns to tape and nontape data sets before and beyond the point of restart.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Disposition before restarting</th>
<th>Disposition after restarting</th>
</tr>
</thead>
<tbody>
<tr>
<td>nontape</td>
<td>OLD,KEEP,KEEP</td>
<td>OLD,KEEP,KEEP</td>
</tr>
<tr>
<td>tape</td>
<td>OLD,KEEP,KEEP</td>
<td>unchanged from original JCL</td>
</tr>
</tbody>
</table>
JCL DD processing of GDG data sets

If JCL DD statements contain GDG data set names, these DD statements appear at the end of the job. The product modifies relative generation numbers of GDG data sets with the assumption that the data sets are cataloged. In nontape data sets, the product modifies relative generation numbers in last-in, first-out (LIFO) order; and in tape data sets, it adjusts relative generation numbers according to their position relative to the restart point. That is, the product processes relative generation numbers of tape data sets in LIFO order up to the point of restart. Beyond the restart point, the data sets reflect the next and subsequent generations in the order of occurrence found in the input JCL.

For instance, if the product used only two members of the GDG before the restart, the restart JCL uses DATASET(-1) and DATASET(0) instead of the +1 and +2 generations. Carefully inspect the modified JCL to ensure that it is referencing the proper GDG member for restart.

Restart parameter

The product passes the utility restart information that you supply to the Execution Monitor program by using the RESTARTPARM keyword with the keyword RESTART in the Execution input parameters. Consult the appropriate utility reference manual for restart parameter information. If you do not supply the restart parameter, the product uses a default value of RESTART.

Omitted DD statements

You can omit JCL DD statements for nontape data sets from the restart JCL job stream when the product no longer requires them.

Online processing

When you execute the BMCTRIG Restart option from the DASD MANAGER PLUS main menu or from the RSTTRIG CLIST, the restart program displays an ISPF panel. When you use the program through the panel, the restart program processes the JCL input file. You can edit and submit the output from the panel.
Restarting jobs that BMCTRIG generates

The procedures in this section describe how to restart jobs that BMCTRIG generates, both automatically and manually.

To restart a BMCTRIG-generated job

Use this procedure to restart a job that BMCTRIG generates when you specify Standard JCL N on the BMCTRIG main menu.

1 On the DASD MANAGER PLUS main menu, select BMCTRIG Generated Job Restart.

The BMCTRIG Generated Job Restart panel appears (Figure 199).

Figure 199 BMCTRIG Generated Job Restart panel

2 Specify the restart options, as follows:

A In the Restart Type field, type R or S, as follows:

- R restarts the worklist at a specific command sequence number.
- S restarts the worklist from the beginning (startover).

B If you specified Restart Type R, use the Restart Sequence Number field to type the six-digit sequence number to restart the worklist.

The job’s diagnostic output identifies the worklist sequence number where job failure occurred.
Restarting jobs that BMCTRIG generates

C If you specified **Restart Type R**, use the **Restart Parameter** field to type the optional restart parameters for the utility that is being restarted.

---

**NOTE**
If you have not specified the restart parameters, DASD MANAGER PLUS uses default restart parameters, depending on the utility type.

---

3 Specify the data set names, as follows:

A Starting at the first **Data Set Name** field, type the name of the JCL data set that contains the BMCTRIG-generated job to use as input.

B In the second **Data Set Name** field, type the name of the restart job JCL data set to contain the output of the restart or startover processing.

If you specify a sequential data set name that does not exist, the product allocates a new data set.

4 Specify whether to edit or submit the output JCL, as follows:

A In the **Edit output JCL** field, type **Y** to display the output JCL job stream for review or edit; otherwise, type **N**.

B In the **Submit output JCL** field, type **Y** to submit the JCL automatically; otherwise, type **N**.

5 Press **Enter** to begin the restart or startover processing.

6 Review and submit the output JCL, as follows:

A Review and edit the JCL if you have set **Edit output JCL** to **Y**.

B Submit the restart JCL.

---

**NOTE**
Although you can use the Restart program to restart jobs as often as you need, the program assumes that the point of restart advances. To restart or start over again at an earlier point in the worklist, use either the original job stream or a job stream from a previous restart that contains the desired restart point in the worklist and JCL.
To restart a BMCTRIG-generated job manually

1. Edit the worklist commands in the WORKLIST DD input stream and delete those steps that have run successfully.

2. Edit the AEXIN DD input stream to indicate RESTART and, optionally, insert the RESTARTPARM keyword with the appropriate parameter to pass to the utility.

   See the example in Figure 200. Restart processing follows the algorithms that are described in the “Execution” chapter of the DASD MANAGER PLUS for DB2 Reference Manual.

Figure 200  Sample AEXIN_DD input stream for BMCTRIG JCL restart

```
//AEXIN DD *
ASU
SSID DB25
DASDTRIG
EVENTS LINES 54 STATS
VCAT DB25CAT
UTILITYID TP263010.S088 RESTART
RESTARTPARM RESTART
```

3. Edit all data set dispositions appropriately.

**NOTE**

Starting the job over does not require RESTARTPARM but requires only the keyword STARTOVER.

Using batch processing

Figure 201 shows sample JCL for batch restart processing of a BMCTRIG-generated job. For commented RESTART EXEC parameters, see also Figure 183.

Figure 201  Sample JCL for batch restart processing  (part1 of2)

```
//JOBCARD JOB (XXXX), 'NAME', MSGCLASS=X, CLASS=A, MSGLEVEL=(1,1)
//******************************************************************************
/* Restart Procedures */
/* a) add a RESTART parameter to the Job Card */
/* b) uncomment of a EXEC parameter for RESTART */
/* in the step where you wish to resume */
/* c) Swap the DISP parms by */
/* uncommenting the // DISP=SHR parm and */
/* commenting the // DISP=(NEW,CATLG,CATLG) parm */
```
Parameter list

The parameter list for batch processing of the Restart program differs, depending on whether you need restart or startover processing. Separate each parameter in the list by a space character.

For restart processing, the first parameter is the installation options module name, followed by the six-digit Restart Sequence Number and the Restart parameter required to restart the specific utility. If you do not supply the Restart parameter, the product supplies a default value of RESTART.

For startover processing, the first parameter is the name of the installation options module, followed by the letter S or the word Startover.
Consult the appropriate utility reference manual for Restart parameter information before modifying the sample JCL.

**Required DD statements**

The input data set has the ddname UTILJOB1, and the output data set has the name ddname UTILJOB2. Both of these DD statements can reference OS/390 and z/OS partitioned or sequential data set names. The record format for both data sets is Fixed or Fixed Block with a record length of 80 characters. BMC Software recommends that you do not use the same data set for input and output, because overwriting the input file removes your ability to restart at a previous restart point.

If the product detects a problem, it uses DD statements for SYSPRINT and SYSTERM for error messages.
Changing the secondary quantity

This chapter contains the following topics:

Before you begin ................................................................. 575
Understanding the ALTERSEC utility ................................. 576
Altering the secondary quantity (SQTY) ............................... 579

Before you begin

Before reading this chapter, you should be familiar with creating actions and the job generation function. For more information, see Chapter 4, “Maintaining and generating actions.”

You can run the ALTERSEC utility alone or as a step in a corrective action. In deciding the number of extents, the main consideration is the amount of storage available. If storage is limited, you might want to set an aggressive goal, such as two extents. If you have ample storage and want to change the secondary quantity only for objects that require this maintenance, you might specify 100 extents as the extent limit that initiates the utility.

Also, consider your Service Level Agreement (SLA). Choose an extent limit that your SLA can tolerate. The default extent limit is 10.

Before running ALTERSEC, choose the table spaces and indexes that you want to monitor and a strategy—the number of extents that you want for the objects. Generally, monitor the objects that grow rapidly.
Understanding the ALTERSEC utility

The ALTERSEC utility prevents objects from running out of extents and avoids a resource-unavailable condition when objects grow rapidly. The utility increases the secondary quantity for storage group-defined table spaces and indexes that have more than a specified number of extents. BMC and DB2 utilities use the new secondary quantity the next time it extends the data set or after a reorganization. For more information about ALTER TABLESPACE or INDEX, see the IBM SQL Reference.

The ALTERSEC utility reads and compares data from the DB2 catalog and the Integrated Catalog Facility (ICF) catalog. From the DB2 catalog, for the object that you specify, the utility reads the current SQTY in SYSIBM.SYSTABLEPART or SYSIBM.SYSINDEXPART. From the ICF catalog, the utility looks for the last two extents of the last data set. The ALTERSEC utility then compares the SQTY with the length of the last two extents in the ICF catalog.

ALTERSEC increases the secondary quantity for the storage-group-defined object by the percentage that you specify when all of the following conditions exist:

- The last two extents are the same as the current secondary quantity.
- The number of extents exceeds the number of extents that you specified.

The ALTERSEC utility generates SQL ALTER statements with the new secondary quantity. The BMC or DB2 utility applies the new secondary extent size when extending the relevant data set.

Figure 202 describes the workflow of the ALTERSEC utility.

![Figure 202 Altering the secondary quantity](image)

Table 82 describes what happens over time when each execution of ALTERSEC specifies Percent increase = 100 and Extent limit = 2.
Understanding the ALTERSEC utility

**NOTE**

Until the BMC or DB2 utility has applied the new SQTY twice, it does not increase the SQTY again.

---

### Table 82  How ALTERSEC works

<table>
<thead>
<tr>
<th>Situation or event</th>
<th>Extents in ICF catalog</th>
<th>SQTY in DB2 catalog</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Situation on Day1</td>
<td>5,5,5</td>
<td>5</td>
<td>The last two extents and the SQTY are the same. (This is the condition required for ALTERSEC to change the SQTY.)</td>
</tr>
<tr>
<td>ALTERSEC1</td>
<td>5,5,5</td>
<td>10</td>
<td>ALTERSEC increases the SQTY to 10 (100 percent).</td>
</tr>
<tr>
<td>Extend1</td>
<td>5,5,5,10</td>
<td>10</td>
<td>The new SQTY is used when the data set is extended.</td>
</tr>
<tr>
<td>ALTERSEC2</td>
<td>5,5,5,10</td>
<td>10</td>
<td>No change. The SQTY is not changed because the last two extents in the ICF catalog are not the same.</td>
</tr>
<tr>
<td>Extend2</td>
<td>5,5,5,10,10</td>
<td>10</td>
<td>The SQTY increased by ALTERSEC1 is applied the second time.</td>
</tr>
<tr>
<td>ALTERSEC3</td>
<td>5, 5,5,10,10</td>
<td>20</td>
<td>ALTERSEC increases the SQTY to 20 (100 percent) because the last two extents in the ICF catalog are the same as the current SQTY (10).</td>
</tr>
<tr>
<td>Extend3</td>
<td>5,5,5,10,10,20</td>
<td>20</td>
<td>The SQTY increased by ALTERSEC3 is applied the first time.</td>
</tr>
</tbody>
</table>

To change the secondary quantity, the utility writes one SQL ALTER SECQTY command for each data set that meets the specified criteria. When generating the action that specifies the ALTERSEC utility, the product generates the ALTER command by using the worklist command –SQL, as Figure 203 shows.

**Figure 203  ALTERSEC –SQL command**

```
-SQL  000004 ALTER TABLESPACE ASUDB029.TS29P1       PART 001
     SECQTY 3600                                        12345678
```

In Figure 203, the –SQL command includes the following parameters:

- 000004 is the step sequence number.
- ALTER represents the SQL ALTER SECQTY command.
- TABLESPACE is the object type.

- ASUDB029.TS29P1 is the name of the table space against which you ran ALTERSEC.

- PART 001 specifies the partition number and appears only for partitioned objects.

- SECQTY 3600 is the secondary quantity expressed in kilobytes. (VSAM, not the ALTERSEC utility, rounds up the quantity to a track or cylinder boundary.)

- 12345678 is the worklist hash code that appears in columns 73 through 80.

Figure 204 illustrates how to set up a job to monitor table spaces and indexes for a specified number of extents and then increase the secondary quantity when this threshold, or extent limit, is exceeded.

**Figure 204 Flow diagram for ALTERSEC**

![Flow diagram for ALTERSEC](image-url)
Altering the secondary quantity (SQTY)

Use this procedure to increase the SQTY for storage group-defined table spaces and indexes. In specifying the parameters for the ALTERSEC utility, set a threshold for the number of extents. ALTERSEC changes the SQTY only when that extent limit is exceeded and the last two extents equal the current SQTY.

To alter the secondary quantity

1. Create an action as instructed in “Creating an action” on page 271.

2. On the Action List panel, select the new action by typing E (or S) in the Act field beside the action and pressing Enter.

3. On the Edit Action Services panel in the Act field, type E (or S).

4. Tab to the Service column, and type ALTERSEC.

5. In the Object Name field, type the object name. You can use wildcards.

6. In the Type field, type the two-character abbreviation for the object type that you want to run the utility against:
   - TS (table space)
   - IX (index)
   - IS (index space)
   - TT (table space set)

7. Press Enter.

8. Type E in the Act field to see the Service Syntax List or type I to create new syntax and press Enter.

The Alter Secondary Quantity panel appears (Figure 205).

Figure 205  Alter Secondary Quantity panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Alter Secondary Quantity</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDK</td>
<td></td>
<td>CSR</td>
</tr>
</tbody>
</table>

Service Syntax: ALTERSEC.SKHALTS

Enter data, then press end.          More:
Percent increase  0   (Increase secondary by this percent)
Extent limit ... 1   (Alter objects with extents greater than limit)
*************************************************************************
*************************************************************************
9 **In the Percent increase** field, type the percentage of the current secondary quantity to add to the current secondary quantity.

If the object exceeds the extent limit, the ALTERSEC utility generates SQL to increase the object’s secondary quantity by this percentage.

10 **In the Extent limit** field, type the maximum number of extents to permit without changing the secondary quantity.

The maximum extent limit is 7257.

If this number of extents is exceeded, the utility changes the object’s secondary quantity. For suggestions about specifying the extent limit, see “Before you begin” on page 575.

11 Generate and execute the job as described in “Generating actions” on page 299.
Producing reports

This chapter contains the following topics:

Before you begin ........................................................................................................ 582
Standard reports ...................................................................................................... 583
  Standard reports descriptions ........................................................................... 585
  Organization of report options ...................................................................... 587
  Average length of RID chain report .............................................................. 587
  Event recording report ..................................................................................... 590
  Index analysis report ......................................................................................... 593
  Index clustering analysis report ................................................................. 596
  Index combination statistics report .............................................................. 600
  Index keys analysis report .............................................................................. 604
  Index leaf distribution report ......................................................................... 606
  Indexspace space detail report ...................................................................... 609
  Monthly detail by tablespace report ............................................................. 611
  Page update analysis report ........................................................................... 614
  Statistics exception report ............................................................................ 617
  Table column detail report ............................................................................. 623
  Table space extent summary report .............................................................. 626
  Table space offset RIDS report ....................................................................... 629
  Table space space detail report ..................................................................... 631
  Volume analysis report ................................................................................... 635
  Volume free space detail report ..................................................................... 637
Model JCL for batch generation ............................................................................ 640
QMF reports .......................................................................................................... 641
Customizable reports .......................................................................................... 644
  Understanding customizable reports .............................................................. 644
  Creating customizable reports ...................................................................... 650
  Running customizable reports ...................................................................... 650
  Running a customizable report in batch mode .............................................. 653
  Customizing the report programs ................................................................. 655
  Adding a report to the customizable reports menu ....................................... 656
  Understanding output data sets ...................................................................... 657
  Location return code descriptions ................................................................. 657
Before you begin

This chapter describes the reports that you can produce in the DASD MANAGER PLUS product from the Reports option on the main menu. This chapter also describes querying the DASD MANAGER PLUS tables. “Customizable reports” on page 644 discusses the reports in option 0 on the Report Selection Menu. You can run these reports online or in batch. Before using the steps in this section, gather statistics from BMCSTATS or RUNSTATS. Collect statistics on the same objects on a regular basis.

NOTE

When DASD MANAGER PLUS 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 DASD MANAGER PLUS sorts the data retrieved from the DB2 Catalog queries, the product displays sorted data on panels and in reports in an EBCDIC collating sequence.
Standard reports

DASD MANAGER PLUS provides numerous standard reports, which run online or in batch. Report statistics are from BMCSTATS unless otherwise indicated. You can access DASD MANAGER PLUS reports from the Report Selection Menu (Figure 206).

**Figure 206  Report Selection Menu panel**

<table>
<thead>
<tr>
<th>Option</th>
<th>Report Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Customizable Reports</td>
</tr>
<tr>
<td>1</td>
<td>Event Recording Report</td>
</tr>
<tr>
<td>2</td>
<td>Statistics Exceptions Report</td>
</tr>
<tr>
<td>3</td>
<td>Space Allocation Reports</td>
</tr>
<tr>
<td>4</td>
<td>Reorg Reports</td>
</tr>
<tr>
<td>5</td>
<td>Image Copy Reports</td>
</tr>
<tr>
<td>6</td>
<td>Tablespace Reports</td>
</tr>
<tr>
<td>7</td>
<td>Index Reports</td>
</tr>
<tr>
<td>8</td>
<td>Table Reports</td>
</tr>
<tr>
<td>9</td>
<td>Stogroup Reports</td>
</tr>
</tbody>
</table>

Standard, noncustomizable DASD MANAGER PLUS reports are available under report menu options 1 through 9 (Figure 207).
Figure 207  Standard DASD MANAGER PLUS reports

![Diagram of Standard DASD MANAGER PLUS reports]

- Customizable Reports
- Event Recording Report
- Statistics
- Space Allocation Reports
- Reorg Reports
- Image Copy Reports
- Tablespace Reports
- Index Reports
- Table Reports
- Stogroup Reports

- Tablespace Space Detail
- Page Update Analysis
- Page Update Analysis
- Offset RIDS
- Average Length of RID Chain
- Table Column Detail
- Volume Analysis

- Indexspace Space Detail
- Extent Summary
- Tablespace Extent Summary
- Page Update Analysis
- Index Leaf Distribution
- Average Length of RID Chain
- Volume Freespace Detail

- Volume Analysis
- Index Leaf Distribution
- Space Detail Summary
- Clustering Analysis
- Clustering Analysis
- Volume Freespace Detail

- Tablespace Extent Summary
- Offset RIDS
- Monthly Detail Summary
- Index Analysis
- Index Key Analysis
- Space Detail Summary
**Standard reports descriptions**

Table 83 describes DASD MANAGER PLUS standard reports. Table 86 describes DASD MANAGER PLUS customizable reports.

### Table 83  DASD MANAGER PLUS standard reports (part 1 of 2)

<table>
<thead>
<tr>
<th>Report title</th>
<th>Description</th>
<th>Report short name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Length of RID Chain report(^a)</td>
<td>This report helps to determine whether to reorganize indexes to avoid fragmentation. If a row identifier chain continually grows, you might want to redefine the index for greater uniqueness.</td>
<td>INDXRIDC</td>
</tr>
<tr>
<td>Event Recording report</td>
<td>This report records utility executions if you request event recording on the Action Generation panels. You can display records of specific events or groups of events from the Event Record Log.</td>
<td>N/A</td>
</tr>
<tr>
<td>Index Analysis report(^a)</td>
<td>This report provides a quick reference of index definitions by table.</td>
<td>INDXANAL</td>
</tr>
<tr>
<td>Index Clustering Analysis report(^a)</td>
<td>This report provides comprehensive information about indexes. Use the report as the basis for decisions about fragmentation, clustering, and some space allocation—such as whether to make the secondary allocation larger.</td>
<td>CLUSANAL</td>
</tr>
<tr>
<td>Index Combination Statistics report(^a)</td>
<td>This report provides definitions for all indexes that satisfy the requirements, by table.</td>
<td>INDXSTAT</td>
</tr>
<tr>
<td>Index Keys Analysis report(^a)</td>
<td>This report helps determine the order in which to put columns in an index and whether a column should be part of an index. The report also indicates the number of occurrences of a value in the top 10 columns. By changing the order of columns or dropping a column, you can influence the path chosen by the optimizer.</td>
<td>INDXKEYS</td>
</tr>
<tr>
<td>Index Leaf Distribution report(^a)</td>
<td>This report helps determine when to reorganize indexes. The report indicates the number of pages between leaf pages. A large number indicates fragmentation that degrades performance.</td>
<td>INDXLDST</td>
</tr>
<tr>
<td>Indexspace Space Detail report</td>
<td>This report provides a quick reference of space usage by index space. Statistics on the left are from the Integrated Catalog Facility (ICF) catalog. Statistics on the right are from the DB2 catalog.</td>
<td>IXSPACAS</td>
</tr>
<tr>
<td>Monthly Detail by Tablespace report(^a)</td>
<td>This report provides a quick reference to the condition and activity of table spaces that you can use to identify trends. For this report to provide meaningful trending data, you must run BMCSTATS each month.</td>
<td>TSMONTH</td>
</tr>
</tbody>
</table>
### DASD MANAGER PLUS standard reports (part 2 of 2)

<table>
<thead>
<tr>
<th>Report title</th>
<th>Description</th>
<th>Report short name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Update Analysis report&lt;sup&gt;a&lt;/sup&gt;</td>
<td>This report helps determine the table spaces that need to be image-copied and whether the copy should be full or incremental. The report recommends a full image copy when fifty percent or more of the pages are dirty or modified.</td>
<td>PAGEUPD</td>
</tr>
<tr>
<td>Statistics Exception report</td>
<td>This report lists the exception conditions that the BMCTRIG utility finds. You can select the exceptions to report by object type, by date and time, or by exception type. You can monitor changes in the database by analyzing the exceptions report.</td>
<td>N/A</td>
</tr>
<tr>
<td>Table Column Detail report&lt;sup&gt;a&lt;/sup&gt;</td>
<td>This report displays column definitions for all columns in a table. When used with the index reports, column definitions can be used to identify columns that are possible candidates for an index.</td>
<td>TBCOLDET</td>
</tr>
</tbody>
</table>
| Tablespace Extent Summary report<sup>a</sup> | This report helps you decide whether to change the table space definition in one of the following ways:  
  - increasing the primary allocation to reduce the number of extents  
  - changing the balance of primary and secondary allocations  
  - increasing or decreasing the number of pages used  
                                                                                                                                                                                                                     | TSEXTENT          |
| Tablespace Offset RIDs report<sup>a</sup> | This report helps determine when to reorganize data by showing the number of rows not on their original page.                                                                                                                                                                                                                             | OFSETRID          |
| Tablespace Space Detail report<sup>a</sup> | This report provides a quick reference of space usage by table space. Statistics on the left are from the Integrated Catalog Facility (ICF) catalog. Statistics on the right are from the DB2 catalog.                                                                                                                  | TSSPACAS          |
| Volume Analysis report<sup>a</sup>   | This report helps determine where to place table spaces or index spaces by displaying volumes that have the amount of free space that you specify. The report indicates free cylinders and free tracks and also the number of DB2 and non-DB2 data sets that the volume contains.                                                                                             | VOLANAL           |
| Volume Free Space Detail report<sup>a</sup> | When paired with the Volume Analysis report, this report provides additional information about volumes that are candidates for new table spaces and index spaces.  
                                                                                                                                                                                                                     | VOLFREE           |

<sup>a</sup> To run this report, DASD MANAGER PLUS requires BMCSTATS.
Organization of report options

The standard DASD MANAGER PLUS reports consist of two sets of reports. Options 1 and 2 represent single reports, the Event Recording Report and the Statistics Exception Report (or Exception Report).

DASD MANAGER PLUS organizes the remaining standard reports by administrative function, such as space allocation, reorg, and image copy, and by DB2 object type, table space, index, table, and storage group. This dual organization permits you to access reports either way.

Redundancy of reports

Organization by both administrative function and object type means that the same report might appear under more than one report menu option. For example, the Tablespace Space Detail Report appears under Space Allocation Reports and Tablespace Reports. The product presents the standard reports in alphabetical order and, for efficiency, lists only the first occurrence of each report.

Average length of RID chain report

Use this procedure to create a report that helps to determine whether to reorganize indexes to avoid fragmentation. If a RID chain continually grows, you might want to redefine the index for uniqueness.

To create an average length of RID chain report

1 Access the Average Length of RID Chain panel as follows:

   A On the DASD MANAGER PLUS main menu, select Reports and press Enter.

   B On the Report Selection Menu, select Index Reports and press Enter.

   C On the Index Reports Selection Menu, select Average Length of RID Chains and press Enter.

The Average Length of RID Chain panel appears (Figure 208).
2 Specify one of the following choices:

- In the Database field, type the name of the database to display statistics for all indexes in a specific database that meet the input criteria.

- In the Database.Tablespace name field, type the qualified name of the table space in the field to display statistics for all indexes in a specific table space that meet the input criteria. If you type a name without the first qualifier, the product assumes that you have entered a wildcard.

- In the Creator.Table name field, type the qualified name of the table in the field to display statistics for all indexes in a specific table that meet the input criteria. If you type a name without the first qualifier, the product assumes that you have entered a wildcard.

- In the Creator.Index name field, type the qualified name of the index in the field to display statistics for an index space. If you type a name without the first qualifier, the product assumes that you have entered a wildcard.

3 In the Average RIDS per entry >= field, type a minimum average in the field to display only objects that have an average of more than the specified number of RIDS per entry.

4 In the Report style field, specify the source of the report statistics, as follows:

1. the most recent execution of BMCSTATS (allows you to see the latest execution)
2. the newest and oldest execution (allows you to compare changes between the oldest and newest executions)
3. all executions (allows you to determine if the object has changed over time)
In the **Batch Report** field, type **Y** in the field and press **Enter** to produce the report in batch.

One of the following events happens:

- If you specified **Y** in the **Batch Report** field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified **N** in the **Batch Report** field, the Average Length of RID Chain report appears online (Figure 209).

**Figure 209  Average Length of RID Chain report**

<table>
<thead>
<tr>
<th>INDEX</th>
<th>INDEX NAME</th>
<th>AVERAGE RIDS</th>
<th>BMCSTATS EXECUTION TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB:AEX.AEXT01_D10S01</td>
<td>AEXX02_D10S01T01</td>
<td>5000.0</td>
<td>2006-08-02-12.39</td>
</tr>
<tr>
<td>TB:AEX.AEXT01_D10S01</td>
<td>AEXX03_D10S01T01</td>
<td>32.0</td>
<td>2006-08-02-12.39</td>
</tr>
</tbody>
</table>

The Average Length of RID Chain for Duplicate Keys portion of the report contains the following fields:

- **Index Creator/Index Name** is the fully qualified name of the reported index.
- **Average RIDS per Index Entry** is the average number of row IDs per index.
- **BMCSTATS Execution Time** is when BMCSTATS collected the statistics in the report.
Event recording report

Use this procedure to create a report that shows specific rows from the DASD MANAGER PLUS EVENTS table. The Event Recording report retrieves and displays specific rows from the DASD MANAGER PLUS database EVENTS table, which contains utility generation information. If you enable the Record Events option when generating a job, the product logs the job in the EVENTS table.

To create an event recording report

1. Access the Event Recording Request panel as follows:
   
   **A** On the DASD MANAGER PLUS main menu, select Reports and press Enter.
   
   
   The Event Recording Request panel appears (Figure 210).

Figure 210  Event Recording Request panel

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>Event Recording Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type data and press Enter.</td>
<td></td>
</tr>
<tr>
<td>Display Events on DB2 Objects</td>
<td></td>
</tr>
<tr>
<td>Object Type</td>
<td></td>
</tr>
<tr>
<td>Object Name</td>
<td></td>
</tr>
<tr>
<td>Display Events by Function</td>
<td></td>
</tr>
<tr>
<td>Function Name</td>
<td></td>
</tr>
<tr>
<td>Restrict to Action</td>
<td></td>
</tr>
<tr>
<td>Action Name</td>
<td></td>
</tr>
<tr>
<td>Restrict by Timestamp</td>
<td></td>
</tr>
<tr>
<td>YYYY-MM-DD-HH.MM.SS</td>
<td></td>
</tr>
<tr>
<td>Start Time</td>
<td></td>
</tr>
<tr>
<td>End Time</td>
<td></td>
</tr>
</tbody>
</table>
Specify the selection criteria as follows:

- Use any combination of the following criteria for selecting rows to include in the report:
  - choose objects or use a wildcard pattern to report on a group of related objects
  - specify a utility function
  - select only certain actions (with or without a wildcard)
  - restrict the search to a specific date and time

Fields that you leave blank return all values for those criteria.

- To report on events for a certain type of object, specify the **Object Type**. The **Object Type** can be TS, TB, or IX.

- To report on events for a certain object, specify the **Object Name**. Use a fully qualified name, a nonqualified name, or wildcard characters in this field.

- To report on events by function, type the **Function Name**. Use any of the following utility names or names of services that you defined, but do not use wildcards:
  - ALTERSEC
  - BMCCHECK
  - BMCCPRS
  - BMCCOPY
  - BMCREORG
  - BMCRESTATS
  - BMCTRIG
  - BMCPURS
  - CHEK DA
  - CHEK LOB
  - DSN1COPY
  - FULLCOPY
  - INCRCOPY
  - LOAD
  - MERGECOPY
  - MODIFY
  - QUIESCE
  - REORG
  - REP REC
  - REP SET
  - RUNSTATS
  - STOP
  - STOSPACE
  - SYNC
  - UNLOAD
  - STARTDB
  - STOPDB

- To report on events for a certain action or group of actions, type the **Action Name**. You can use a wildcard.

- To restrict the report to events that fall within a certain time frame, specify **Start Time** and **End Time**.
If you leave the Start field blank, the report includes all rows from the beginning of the EVENTS table. If you leave the End field blank, the report includes all rows to the end of the EVENTS table. The format is YYYY-MM-DD-HH.MM.SS.

C After you specify all of the search criteria, press Enter.

The Event List appears, showing any rows that match your selection criteria (Figure 211).

Figure 211  Event List

<table>
<thead>
<tr>
<th>Act</th>
<th>Object</th>
<th>Name</th>
<th>Typ</th>
<th>Function</th>
<th>RC</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>TS BMCSTATS</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>TS BMCSTATS</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>TS BMCSTATS</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>TS BMCSTATS</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>TS BMCSTATS</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>TS BMCSTATS</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>TS BMCSTATS</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>TS BMCSTATS</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>TS BMCSTATS</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Event List contains the following fields:

- **Act** is the action field where you can enter Z to zoom the object and name. Entering Z displays the long name in a pop-up panel.

- **Object Name** is the two-part qualified name of the object: `databaseName.tableSpaceName` for a table space or `creator.indexName` for an index.

- **Obj Typ** is the type of DB2 object.

- **Function** is the utility or object manipulation that the product runs.

- **RC** is the return code from the utility or SQL.

- **Action** is the action that ran the job.
- **Timestamp** is the DB2 timestamp value showing when the event ran.
- **Elapse Time** is the elapsed CPU time in seconds.

**NOTE**
In this report, the ISPF RIGHT and LEFT commands are unavailable for scrolling right or left, respectively. To scroll right or left, enter GORIGHT or GOLEFT or use the standard function keys.

---

**Index analysis report**

Use this procedure to create a report that provides a quick reference of index definitions by table.

**To create an index analysis report**

1. Access the Index Analysis panel as follows:

   A. On the DASD MANAGER PLUS main menu, select Reports and press Enter.
   
   B. On the Report Selection Menu, select Index Reports and press Enter.
   
   C. On the Index Reports Selection Menu, select Index Analysis (Definitional Aspects) and press Enter.

   The Index Analysis panel appears (Figure 212).

**Figure 212  Index Analysis panel**

```
DEAE ----------------------------- INDEX ANALYSIS -----------------------------
COMMAND ===> 

The fields below determine the list of objects the report will use. Type the full or wildcard name of the objects, then press Enter

Database.................
Database.Tablespace name ....
Creator.Table name ........
Creator.Index name .......

Where
Clusterratio <=............. 96

Report style............... 1 1. Most recent execution
                              2. Newest and Oldest execution
                              3. All executions

Batch Report : N (Y/N)
```
2 Specify one of the following options:

- In the Database field, type the name of the database to display statistics for all indexes in a specific database that meet the input criteria.

- In the Database.Tablespace name field, type the qualified name of the table space to display statistics for all indexes in a specific table space that meet the input criteria. If you type a name without the first qualifier, the product assumes that you are using a wildcard.

- In the Creator.Table name field, type the qualified name of the table to display statistics for all indexes in a specific table that meet the input criteria. If you type a name without the first qualifier, the product assumes that you are using a wildcard.

- In the Creator. Index name field, type the qualified name of the index to display statistics for an index space. If you type a name without the first qualifier, the product assumes that you are using a wildcard.

3 In the Clusterratio <= field, type a percentage of cluster ratio to display only indexes that have a cluster ratio less than that specified.

This report includes indexes that have this cluster ratio or less.

4 In the Report style field, specify the source of the report statistics, in the following order:

1. most recent execution of BMCSTATS
2. newest and oldest execution
3. all executions

5 In the Batch Report field, type Y in the field and press Enter to produce the report in batch.

One of the following events happens:

- If you specified Y in the Batch Report field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified N in the Batch Report field, the Index Analysis report appears (Figure 213).
The Index Analysis report contains the following fields:

- **Ixcreator/Ixname**: is the fully qualified name of the reported index.
- **Part**: is the number of the partition in the index.
- **Ext Type**: is the type of extended index.
- **Clstr**: indicates whether the index is a clustering index (Y or N).
- **Unique**: is the unique rule, indicating whether the index is unique. Valid values are as follows:

<table>
<thead>
<tr>
<th>Unique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>duplicates allowed</td>
</tr>
<tr>
<td>U</td>
<td>unique</td>
</tr>
<tr>
<td>P</td>
<td>unique and primary index</td>
</tr>
<tr>
<td>C</td>
<td>unique and used to enforce a unique constraint</td>
</tr>
<tr>
<td>N</td>
<td>unique and defined with UNIQUE WHERE NOT NULL</td>
</tr>
<tr>
<td>R</td>
<td>unique and used to enforce the uniqueness of a nonprimary parent key</td>
</tr>
<tr>
<td>G</td>
<td>unique and used to enforce the uniqueness of values in the column defined as ROWID GENERATED BY DEFAULT</td>
</tr>
</tbody>
</table>
Index clustering analysis report

- **Pfree** is the percentage of each leaf or nonleaf page that is left as free space.
- **Fpage** is the number of pages that are loaded before a page is left as free space.
- **Leaf** is the number of active leaf pages in the index. The value is -1 before gathering statistics.
- **Nlvl** is the number of levels in the index. If the index is partitioned, this field indicates the maximum number of levels in the index tree for all partitions.
- **Clust Ratio** is the percentage of rows that are in clustering order.
- **Data Repeat Factor** is the number of pages read while following an index key order.
- **BMCSTATS Execution Time** is when BMCSTATS collected the statistics in the report.

Index clustering analysis report

Use this procedure to create a report that provides comprehensive information about indexes. Use the report as the basis for decisions about fragmentation, clustering, and some space allocation, such as whether to make the secondary allocation larger.

To create an index clustering analysis report

1. Access the Index Clustering Analysis panel as follows:
   
   A. On the DASD MANAGER PLUS main menu, select Reports and press Enter.
   
   

   The Index Clustering Analysis panel appears (Figure 214).
2 Specify one of the following options:

- In the **Database** field, type the name of the database to display statistics for all indexes in a specific database that meet the input criteria.

- In the **Database.Tablespace name** field, type the qualified name of the table space to display statistics for all indexes in a specific table space that meet the input criteria. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the **Creator.Table name** field, type the qualified name of the table to display statistics for all indexes in a specific table that meet the input criteria. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the **Creator. Index name** field, type the qualified name of the index to display statistics for an index space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

3 Specify either a cluster ratio or a combination of nearoff and faroff rows, as follows:

- In the **Clusterratio <=** field, type a percentage of cluster ratio to display only indexes that have a cluster ratio less than that specified.
The product includes indexes that have this cluster ratio or less in the report.
Alternatively, specify a combination of nearoff and faroff rows in the following two fields:

- In the **Nearoffpos/Card * 100.0 >=** field, type the minimum percentage of nearoff rows that you want reported. Nearoff rows are those that are less than 16 pages from optimal position.

The product includes indexes that have at least this percentage of nearoff rows and the percentage of faroff rows specified in the next field in the report. Alternatively, you can specify **Clusterratio**.

- In the **Faroffpos/Card * 100.0 >=** field, type the minimum percentage of faroff rows that you want reported. Faroff rows are those that are more than 16 pages from optimal position.

The product includes indexes that have at least this percentage of faroff rows and the percentage of nearoff rows specified in the previous field in the report. Alternatively, you can specify **Clusterratio**.

4 In the **Report style** field, specify the source of the report statistics in the following order:

   1. most recent execution of BMCSTATS
   2. newest and oldest execution
   3. all executions

5 In the **Batch Report** field, type **Y** in the field and press **Enter** to produce the report in batch.

One of the following events happens:

- If you specified **Y** in the **Batch Report** field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified **N** in the **Batch Report** field, the Index Clustering Analysis report appears (Figure 215).
The Index Clustering Analysis report contains the following fields:

- **Ixcreator/Ixname** is the fully qualified name of the reported index.

- **Part** is the partition number.

- **Uniq Rule** is the unique rule, indicating whether the index is unique. Valid values are as follows:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>duplicates allowed</td>
</tr>
<tr>
<td>U</td>
<td>unique</td>
</tr>
<tr>
<td>P</td>
<td>unique and primary index</td>
</tr>
<tr>
<td>C</td>
<td>unique and used to enforce a unique constraint</td>
</tr>
<tr>
<td>N</td>
<td>unique and defined with UNIQUE WHERE NOT NULL</td>
</tr>
<tr>
<td>R</td>
<td>unique and used to enforce the uniqueness of a nonprimary parent key</td>
</tr>
<tr>
<td>G</td>
<td>unique and used to enforce the uniqueness of values in the column defined as ROWID GENERATED BY DEFAULT</td>
</tr>
</tbody>
</table>

- **Clust Ratio** is the percentage of rows that are in clustering order.

- **Data Repeat Factor** is the number of pages read while following an index key order.
- **Near Off** is the number of referenced rows that are near, but not at optimal position, because of an insert into a full page.

- **Far Off** is the number of referenced rows far from optimal position because of an insert into a full page.

- **Number of Rows** is the number of rows in the index partition.

- **BMCSTATS Execution Time** is when BMCSTATS collected the statistics in the report.

## Index combination statistics report

Use this procedure to create a report that provides definitions for all indexes that satisfy the requirements, by table.

### To create an index combination statistics report

1. Access the Index Combination Statistics panel as follows:

   A. On the DASD MANAGER PLUS main menu, select **Reports** and press **Enter**.

   B. On the Report Selection Menu, select **Index Reports** and press **Enter**.

   C. On the Index Reports Selection Menu, select **Index Statistics (1,2, and 3 combined)** and press **Enter**.

   The Index Combination Statistics panel appears (Figure 216).
2 Specify one of the following options:

- In the Database field, type the name of the database to display statistics for all indexes in a specific database that meet the input criteria.

- In the Database.Tablespace name field, type the qualified name of the table space to display statistics for all indexes in a specific table space that meet the input criteria. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the Creator.Table name field, type the qualified name of the table to display statistics for all indexes in a specific table that meet the input criteria. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the Creator. Index name field, type the qualified name of the index to display statistics for an index space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

3 Specify either a cluster ratio or a combination of nearoff and faroff rows, as follows:

- In the Clusterratio <= field, type a percentage of cluster ratio to display only indexes that have a cluster ratio less than that specified.
The product includes indexes that have this cluster ratio or less in the report. Alternatively, specify a combination of nearoff and faroff rows in the following two fields.

- In the Nearoffpos/Card * 100.0 >= field, type the minimum percentage of nearoff rows that you want reported. Nearoff rows are those that are less than 16 pages from optimal position.

The product includes indexes that have at least this percentage of nearoff rows and the percentage of faroff rows specified in the next field in the report. Alternatively, you can specify Clusterratio.

- In the Faroffpos/Card * 100.0 >= field, type the minimum percentage of faroff rows that you want reported. Faroff rows are those that are more than 16 pages from optimal position.

The product includes indexes that have at least this percentage of faroff rows and the percentage of nearoff rows specified in the previous field in the report. Alternatively, you can specify Clusterratio.

4 In the Average RIDS per entry >= field, type a minimum average to display only objects that have an average of more than the specified number of RIDS per entry.

5 In the LEAFDIST >= field, type a minimum leaf distribution to display only objects with a leaf distribution greater than a specified number. (Leaf distribution is the average distance between successive leaf pages.)

6 In the Data set extents >= field, type a minimum number of extents to display only objects that have more than a specified number of extents.

7 In the Report style field, specify the source of the report statistics in the following order:

1. most recent execution of BMCSTATS
2. newest and oldest execution
3. all executions

8 In the Batch Report field, type Y in the field and press Enter to produce the report in batch.

One of the following events happens:

- If you specified Y in the Batch Report field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.
- If you specified N in the Batch Report field, the Index Combination Statistics report appears (Figure 217).

**Figure 217  Index Combination Statistics report**

The Index Combination Statistics report contains the following fields:

- **Ixcreator/Ixname** is the fully qualified name of the reported index.
- **Part** is the partition number.
- **Ext Type** is the type of extended index.
- **Clust Ratio** is the percentage of rows that are in clustering order.
- **Data Repeat Factor** is the number of pages read while following an index key order.
- **Near Off** is the number of referenced rows that are near, but not at optimal position, because of an insert into a full page.
- **Far Off** is the number of referenced rows far from optimal position because of an insert into a full page.
- **Xtnts** is the number of extents that the index is in.
- **Avg Pages Between Leafpages** is the average number of pages between leaf pages.
Index keys analysis report

- Avg RIDS per IX Entry is the average number of row IDs per index.
- BMCSTATS Execution Time is when BMCSTATS collected the statistics in the report.

Index keys analysis report

Use this procedure to create a report that helps determine the order in which to put columns in an index and whether a column should be part of an index.

To create an index keys analysis report

1. Access the Index Keys Analysis panel as follows:

   A. On the DASD MANAGER PLUS main menu, select Reports and press Enter.
   B. On the Report Selection Menu, select Index Reports and press Enter.
   C. On the Index Reports Selection Menu, select Index Key Analysis and press Enter.

   The Index Keys Analysis panel appears (Figure 218).

Figure 218  Index Keys Analysis panel

2. Specify one of the following options:

   - In the Database field, type the name of the database to display statistics for all indexes in a specific database that meet the input criteria.
In the **Database.Tablespace name** field, type the qualified name of the table space to display statistics for all indexes in a specific table space that meet the input criteria. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

In the **Creator.Table name** field, type the qualified name of the table to display statistics for all indexes in a specific table that meet the input criteria. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

In the **Creator. Index name** field, type the qualified name of the index to display statistics for an index space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

3 In the **Batch Report** field, type Y in the field and press **Enter** to produce the report in batch.

One of the following events happens:

- If you specified Y in the **Batch Report** field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified N in the **Batch Report** field, the Index Key Analysis report appears (Figure 219).

### Figure 219  Index Key Analysis report

<table>
<thead>
<tr>
<th>Table Name / Key Column</th>
<th>Index Key Analysis</th>
<th>More Data →</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZU.QZUT01.D11S01</td>
<td>1</td>
<td>A.15.00009-00-00009-00-TIMESTAMP.0</td>
</tr>
</tbody>
</table>
The Index Key Analysis report contains the following fields:

- **Table Name/Index name** is the full name of the table and index.
- **Key Column** is the name of the key column.
- **Ext Type** is the type of extended index.
- **Col/Key Seqno** is the numeric position of the column in the key.
- **Col Ordering/Ordering** identifies columns that are INCLUDE columns.
- **Col Order** is the sequence number of the column in the table.
- **ColCard/KeyCard** is the cardinality of the column.
- **High2Key** is the second highest key value of the column.
- **Low2Key** is the second lowest key value of the column.
- **Coltype/DataType** is the data type of the column.
- **Col Len/Key Len** is the column length.
- **Col Scale** is the scale of decimal data.
- **Column** is the key column for the listed values.
- **Ratio of Rows** is the frequency of the keys in the column.
- **Value/KeyValue** is the value of the column.

**Index leaf distribution report**

Use this procedure to create a report that helps to determine when to reorganize indexes.

**To create an index leaf distribution report**

1. Access the Index Leaf Distribution panel as follows:

   A. On the DASD MANAGER PLUS main menu, select Reports and press Enter.


   C. On the Reorg Reports Selection Menu, select Index Leaf Distribution and press Enter.

The Index Leaf Distribution panel appears (Figure 220).
Figure 220  Index Leaf Distribution panel

DECA ----------------------------- INDEX LEAF DISTRIBUTION -----------------------------
COMMAND ===>

The fields below determine the list of objects the report will use. Type the full or wildcard name of the objects, then press Enter

Database....................
Database.Tablespace name ....
Creator.Table name ..........
Creator.Index name ..........

Where
LEAFDIST >=.................... 100

Report style............... 1 1. Most recent execution
2. Newest and Oldest execution
3. All executions

Batch Report : N (Y/N)

2 Specify one of the following options:

- In the **Database** field, type the name of the database to display statistics for all indexes in a specific database that meet the input criteria.

- In the **Database.Tablespace name** field, type the qualified name of the table space to display statistics for all indexes in a specific table space that meet the input criteria. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the **Creator.Table name** field, type the qualified name of the table to display statistics for all indexes in a specific table that meet the input criteria. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the **Creator. Index name** field, type the qualified name of the index to display statistics for all indexes in a specific index space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

3 In the **LEAFDIST >=** field, type a minimum leaf distribution to display only objects with a leaf distribution greater than a specified number. (Leaf distribution is the average distance between successive leaf pages.)
4 In the **Report style** field, specify the source of the report statistics, in the following order:

1. most recent execution of BMCSTATS
2. newest and oldest execution
3. all executions

5 In the **Batch Report** field, type **Y** in the field and press **Enter** to produce the report in batch.

One of the following events happens:

- If you specified **Y** in the **Batch Report** field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified **N** in the **Batch Report** field, the Index Leaf Distribution report appears (**Figure 221**).

---

**Figure 221  Index Leaf Distribution report**

---

**Table:**

<table>
<thead>
<tr>
<th>INDEX Creator</th>
<th>INDEX NAME</th>
<th>AVG PAGES BETWEEN LEAFPAGES</th>
<th>BMCSTATS EXECUTION TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB:QZU.QZUT01.DE3S01</td>
<td>QZU.QZUT01.DE3S01T01</td>
<td>0</td>
<td>29772931 2006-06-20-16.26</td>
</tr>
<tr>
<td>TB:QZU.QZUT01.DE4S01</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Menu Utilities Compilers Help**

BROWSE SY505227.T104934.RA000.RDAJXS.R0212433 Line 00000000 Col 001 080

Command ===>> Scroll ===>> PAGE

*********************************************************************** Top of Data ***********************************************************************

BMC SOFTWARE, INC.

***** DASD MANAGER PLUS FOR DB2 ***** V9.1.00

Time...: 10:49:35 AM Wednesday, February 14, 2007
DB2 ID...: DECA
DB2 Version: 910
User...: RDAJXS

---

Index Leaf Distribution

---

**Table:**

<table>
<thead>
<tr>
<th>INDEX Creator</th>
<th>INDEX NAME</th>
<th>AVG PAGES BETWEEN LEAFPAGES</th>
<th>BMCSTATS EXECUTION TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB:QZU.QZUT01.DE3S01</td>
<td>QZU.QZUT01.DE3S01T01</td>
<td>0</td>
<td>29772931 2006-06-20-16.26</td>
</tr>
<tr>
<td>TB:QZU.QZUT01.DE4S01</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Index Leaf Distribution report contains the following fields:

- **Index Creator/Index Name** is the fully qualified name of the reported index.
- **Part** is the number of the partition in the index.
- **Avg Pages Between Leafpages** is the average number of pages between leaf pages.
- **BMCSTATS Execution Time** is when BMCSTATS collected the statistics in the report.

**Indexspace space detail report**

Use this procedure to create a report that provides a quick reference of space usage by index space.

**To create an indexspace space detail report**

1. Access the Indexspace Space Detail panel as follows:
   
   **A** On the DASD MANAGER PLUS main menu, select **Reports** and press **Enter**.
   
   **B** On the Report Selection Menu, select **Space Allocation Reports** and press **Enter**.
   
   **C** On the Space Allocation Reports Selection Menu, select **Indexspace Space Detail** and press **Enter**.

   The Indexspace Space Detail panel appears (Figure 222).

**Figure 222 Indexspace Space Detail panel**

```
DECA ----------------------------- INDEXSPACE SPACE DETAIL -----------------------------
COMMAND ===> 

The fields below determine the list of objects the report will use. Type the full or wild card name of the objects, then press Enter

Database....................
Database.Tablespace name ....
Creator.Table name ........
Creator.Index name .........

**** WARNING ****
This report may execute for extended lengths of time.

Batch Report : N (Y/N)
```
### Indexspace space detail report

2. Specify one of the following options:

- In the **Database** field, type the name of the database to display statistics for all indexes in a specific database that meet the input criteria.

- In the **Database.Tablespace name** field, type the qualified name of the table space to display statistics for all indexes in a specific table space that meet the input criteria. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the **Creator.Table name** field, type the qualified name of the table to display statistics for all indexes in a specific table that meet the input criteria. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the **Creator. Index name** field, type the qualified name of the index to display statistics for an index space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

3. In the **Batch Report** field, type **Y** in the field and press **Enter** to produce the report in batch.

One of the following events happens:

- If you specified **Y** in the **Batch Report** field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified **N** in the **Batch Report** field, the Indexspace Space Detail report appears (Figure 223).

---

**Figure 223 Indexspace Space Detail report**

```
<table>
<thead>
<tr>
<th>DSN</th>
<th>OBJ_NAME</th>
<th>OBJ_TYPE</th>
<th>OBJ_SIZE</th>
<th>TRACKS_ALLOC</th>
<th>PARTS</th>
<th>CARD</th>
<th>SEC_ALLOC</th>
<th>SEC_SIZE</th>
<th>COMPRESSION</th>
<th>FREE_SPACE</th>
<th>PERCENT_FREE</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD.MANAGER.PLB.DASD.MANAGER.PLB</td>
<td>OBJ0001</td>
<td>Indexspace</td>
<td>90</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>30</td>
<td>N</td>
<td>10</td>
<td>10</td>
<td>2010-01-01</td>
</tr>
<tr>
<td>QZUD10.QZUX01</td>
<td>Primary</td>
<td>Index</td>
<td>4290</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>36</td>
<td>N</td>
<td>10</td>
<td>10</td>
<td>2010-01-01</td>
</tr>
<tr>
<td>QZUD10.QZUX01</td>
<td>Secondary</td>
<td>Index</td>
<td>1000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>N</td>
<td>10</td>
<td>10</td>
<td>2010-01-01</td>
</tr>
</tbody>
</table>
```
The Indexspace Space Detail report is divided vertically into two parts. The data on the left is from the ICF catalog. The data on the right is from the DB2 catalog. The report contains the following fields:

- **DSN** is the fully qualified data set name of the index space, as follows:

```
Object Type is the index space
Volser volume serial number of the first volume for the object
Devtype device type where the object is stored
Extents number of extents that the index or index partition is in
Kbytes Alloc number of kilobytes allocated for the index
Kbytes Used number of kilobytes that the index uses
Stornam e is the name of the storage group or ICF catalog used for space allocation.
Tracks Alloc number of tracks allocated for the index
Tracks Used number of tracks that the index uses
Alloc Unit allocation unit (T–tracks or C–cylinders)
Prim Alloc primary allocation quantity
Sec Alloc secondary allocation quantity
```

- **Object Name** is the name of the index.
- **Table Name** is the name of the table that the index references.
- **Part** is the number of the partition in the index space.
- **Pagesize** is the page size for indexes (4, 8, 16 and 32-KB).
- **Primary** is the primary allocation quantity.
- **Secondary** is the secondary allocation quantity.
- **Ext Type** is the type of extended index.
- **Card** is the number of rows in the index or index partition.
- **Compress** indicates whether index compression is active.
- **Freespace** is the number of free tracks.
- **Percent Free** is the percentage of the allocated space that is free.
- **Piecesize** is the maximum size of a data set for secondary indexes.
- **Padded** indicates whether values are padded.
- **Dsnum** indicates the data set number.

### Monthly detail by tablespace report

Use this procedure to create a report that provides a quick reference to the condition and activity of table spaces that you can use to identify trends. For this report to provide meaningful trending data, you must run BMCSTATS each month.
To create a monthly detail by tablespace report

1 Access the Monthly Detail by Tablespace panel as follows:

   A On the DASD MANAGER PLUS main menu, select Reports and press Enter.

   B On the Report Selection Menu, select Tablespace Reports and press Enter.

   C On the Tablespace Reports Selection Menu, select Monthly Detail Summary and press Enter.

   The Monthly Detail by Tablespace panel appears (Figure 224).

Figure 224 Monthly Detail by Tablespace panel

```
DECA ----------------------------- MONTHLY DETAIL BY TABLESPACE ------------
COMMAND ===> 

The fields below determine the list of objects the report will use. Type the full or wildcard name of the objects, then press Enter.

Database....................
Database.Tablespace name ....
Creator.Table name ..........

Batch Report : N (Y/N)
```

2 Specify one of the following options:

- In the Database field, type the name of the database to display statistics for all table spaces in a database.

- In the Database.Tablespace name field, type the qualified name of the table space to display statistics for a table space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the Creator.Table name field, type the qualified name of the table to display statistics for the table space associated with the requested table. If you type a name without the first qualifier, the product assumes that you entered a wildcard.
3 In the **Batch Report** field, type **Y** in the field and press **Enter** to produce the report in batch.

One of the following events happens:

- If you specified **Y** in the **Batch Report** field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified **N** in the **Batch Report** field, the Monthly Detail by Tablespace report appears (Figure 225).

**Figure 225  Monthly Detail by Tablespace report**

The Monthly Detail by Tablespace report contains the following fields:

- **DB** is the name of the database that contains the table space.

- **Tsname** is the name of the table space.

- **Part #** is the number of the table space partition.
Page update analysis report

- **Month** is the year and month of the statistics in YYYY-MM format, where YYYY is the year and MM is the month.

- **Card** is the cardinality, or number of rows, in the table space.

- **TS Type** is the type of table space.

- **Far Ind Ref** is the number of indirect references to rows that are 16 or more pages from the original page because of an insert into a full page.

- **Near Ind Ref** is the number of indirect references to rows that are less than 16 pages from the original page because of an insert into a full page.

- **Perc Active** is the percentage of pages that are active.

- **Perc Drop** is the percentage of space occupied by rows of dropped tables.

- **Nactive** is the number of active pages in the table space. The product considers pages with row format active even if they currently contain no rows.

- **Full** is the number of full pages.

- **Dirty** is the number of modified, or dirty, pages in the table space.

- **Space** is the number of allocated tracks.

- **Reorg Space** is the minimum number of tracks required if you reorganized the table space.

- **Pqty** is the primary allocation quantity.

- **Sqty** is the secondary allocation quantity.

- **Xtnts** is the number of current extents.

- **Pct Free** is the percentage of the allocated space that is free.

- **Free Page** is the number of free pages.

**Page update analysis report**

Use this procedure to create a report that helps to determine the table spaces that need to be image-copied and whether the copy should be full or incremental.
To create a page update analysis report

1 Access the Page Update Analysis panel as follows:

   A On the DASD MANAGER PLUS main menu, select Reports and press Enter.


   The Page Update Analysis panel appears (Figure 226).

Figure 226  Page Update Analysis panel

2 Specify one of the following options:

   - In the Database field, type the name of the database to display statistics for a database.

   - In the Database.Tablespace name field, type the qualified name of the table space to display statistics for a table space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

   - In the Creator.Table name field, type the qualified name of the table to display statistics for a table. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

3 In the Percent of dirty pages >= field, type a minimum percentage to display only objects that have more than a certain percentage of modified pages.
4 In the **Report style** field, specify the source of the report statistics, in the following order:

1. most recent execution of BMCSTATS
2. newest and oldest execution
3. all executions

5 In the **Batch Report** field, type **Y** in the field and press **Enter** to produce the report in batch.

One of the following events happens:

- If you specified **Y** in the **Batch Report** field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.
- If you specified **N** in the **Batch Report** field, the Page Update Analysis report appears (Figure 227).

**Figure 227  Page Update Analysis report**

```
BROWSE SY511012.T130724.RA000.RDAJXS.R0362136
Line 00000000 Col 001 132
Command ===>
Scroll ===> PAGE
***************************************************************************** Top of Data ***************
BMC SOFTWARE. INC.
***** DASD MANAGER PLUS FOR DB2 ***** V10.01.00
Time. . .: 01:13:01 PM Wednesday, January 12, 2011
DB2 ID. . .: DEDK
DB2 Version: 910
User. . . .: RDAJXS

<table>
<thead>
<tr>
<th>DATABASE</th>
<th>SPACE</th>
<th>PART</th>
<th>TS</th>
<th>ACTIVE</th>
<th>UPDATED</th>
<th>PERCENT</th>
<th>COPY TYPE</th>
<th>BMCSTATS EXECUTION TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>0</td>
<td>130</td>
<td>128</td>
<td>98</td>
<td>FULL</td>
<td>2010-12-23-12.54</td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>QZUS02A1</td>
<td>0</td>
<td>146</td>
<td>128</td>
<td>87</td>
<td>FULL</td>
<td>2010-12-23-12.54</td>
<td></td>
</tr>
</tbody>
</table>
```

The Page Update Analysis report contains the following fields:

- **Database Name/Table Space Name** is the fully qualified table space name.
Part Nbr is the number of the table space partition.

TS Type is the type of table space.

Active Pages is the number of active pages in the table space. The product considers pages with row format active even if they currently contain no rows.

Updated Pages is the number of modified, or dirty, pages in the table space.

Percent Dirty is the percentage of modified, or dirty, pages in the table space.

Copy Type Recommended is the recommended type of image copy: full or incremental.

BMCSTATS Execution Time is when BMCSTATS collected the statistics in the report.

Statistics exception report

Use this procedure to create a report (the Exception Report) that lists the exception conditions that the BMCTRIG utility finds.

Before you begin

The BMCTRIG utility records the exceptions that it finds in the DASD MANAGER PLUS EXCEPTIONS2 table. You can produce exception reports by using the DASD MANAGER PLUS report function. This function lets you specify a variety of selection criteria for creating the reports.

You can perform any of the following actions:

- choose specific objects or use a wildcard pattern to report on a group of related objects
- specify a utility function
- select only certain actions (with or without a wildcard)
- restrict the search to a specific date and time

The BMCTRIG utility also produces a batch report when the utility encounters exception conditions. See “Setting up a BMCTRIG job” on page 504.
**To create a statistics exception report**

1. Access the Statistics Exception Selection panel as follows:

   **A** On the DASD MANAGER PLUS main menu, select **Reports** and press **Enter**.

   **B** On the Report Selection Menu, select **Statistics Exceptions Report** and press **Enter**.

   The Statistics Exception Selection panel appears (**Figure 228**).

**Figure 228  Statistics Exception Selection panel**

```
DECA ----------------- Statistics Exception Selection -------------------------
COMMAND ===> Type data and press Enter.
Object Type . . . . . (TB,TS,IX)
Object Name . . . .
YYYY-MM-DD-HH.MM
Exception Date . . . > (> or =)
Select Exception type? N (Y/N to see exception types and select one)
```

2. On this panel, specify the selection criteria to use when choosing rows from the exceptions2 table for the report as follows:

   **A** In the **Object Type** field, type **TS** (table space), **TB** (table), or **IX** (index) to indicate the type of object for which you want to display statistics exceptions.

   **B** In the **Object Name** field, type the object name to display exception data for a particular object. The object name can be fully qualified, nonqualified, or a wildcard pattern.

   **C** In the **Exception Date** field, type a greater than or equal to character followed by the starting date to display exception data beginning at that date.

   The exception date format is YYYY-MM-DD-HH.MM where **YYYY** indicates a four-digit year, **MM** a two-digit month, **DD** a two-digit day, **HH** a two-digit hour, and **MM** a two-digit minute.

   **D** In the **Select Exception type** field, type **Y** to display statistics for a single type of exception.

   **E** Press **Enter**.
One of the following events happens:

- If you have specified **Select Exception type = Y**, the Statistics Exceptions Type Selection panel appears and you can select an exception type (Figure 229).

- If you have specified **Select Exception type = N**, DASD MANAGER PLUS displays a Exceptions Selection panel that shows all exceptions found for the specified objects (see Figure 230).

**Figure 229  Statistics Exception Type Selection panel**

```
DEDK ------------------ Statistics Exception Type Selection ------------------
COMMAND ===> 

Type the number of the exception you wish to select and press Enter.

8. Farind     21. Rows/Key     34. Loadcopy   47. Areppend
```

**Figure 230  Exceptions Selection panel**

```
DECA --------------------- Exceptions Selection -------- Row 1 to 14 of 21,878
COMMAND ===> SCROLL ===> PAGE

Please type an option and press Enter.

S = Display exception   D = Delete exception.     Z = Zoom Object Name
MORE DATA ==> 

Act Owner/DB Object Name Type Part Exception Value
-----------------------------------------------
CCB XCPPART IX 1 EXTENTS 1
CCB XCPPART IX 1 EXTENTS 1
CCB XCPPART IX 1 EXTENTS 1
CCB XCPPART IX 1 EXTENTS 1
CCB XCPPART IX 1 EXTENTS 1
CCB XCPPART IX 1 EXTENTS 1
```
The Exception Selection panel contains the following fields:

- **Act** indicates whether to select (S) the panel entry to display the Exceptions Report, delete (D) the entry, or zoom (Z) the entry to see the entire object name.

- **Owner/DB** is the owner (creator) of the index, table, or table space.

- **Object Name** is the DB2 object name.

- **Type** is the DB2 object type: TS (table space), IX (index), or TB (table).

- **Part** is the number of the table space partition or index space partition.

- **Exception** is the type of exception identified by BMCTRIG.

- **Current Value** is the value with the latest timestamp in the DASD MANAGER PLUS tables.

- **Last Update** is the latest timestamp for the current value in the DASD MANAGER PLUS tables.

- **Active** indicates whether the exception is active.

To display the exception details for an item in the list, type S next to the item in the Act field.

The Exceptions panel appears (Figure 231).
Figure 231 Exceptions report for a single exception type

The Exceptions report contains the following fields:

- **Object Name** is the fully qualified object name.
- **Type** is the DB2 object type: TS (table space), IX (index), or TB (table).
- **Action Taken** is the action that the product generated or submitted when the exception occurred.
- **Utility JCL** is the name of the data set that contains the generated JCL.
- **Part** is the number of the table space partition or index space partition.
- **Exception** is the type of exception. Exceptions correspond to BMCTRIG triggers.
- **Current Value** is the value having the latest timestamp in the DASD MANAGER PLUS tables.
- **Compare Value** is the value that the product is comparing to the current value or the portion of the formula for computing the compare value.

**NOTE**

The **Compare Value** is applicable only if a trigger required a previous value for its computation. For example, a trigger that identifies a percentage increase requires a previous value from which to calculate the change.
Trigger Value is the value set as a trigger by the user to indicate that an exception has occurred in the DB2 database.

Figure 232 shows the Exceptions Selection panel that appears when you specify Select Exception Type N on the initial Statistics Exception Selection panel (see Figure 228).

**Figure 232  Exceptions Selection panel**

<table>
<thead>
<tr>
<th>Act Owner/DB Object Name</th>
<th>Type Part</th>
<th>Exception</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUDA1 QZUS01A1</td>
<td>TS</td>
<td>DSEXTENT</td>
<td>1</td>
</tr>
<tr>
<td>QZUDA1 QZUS01A1</td>
<td>TS</td>
<td>EXTENTS</td>
<td>1</td>
</tr>
<tr>
<td>QZUDA1 QZUS01A1</td>
<td>TS</td>
<td>FARIND</td>
<td>0</td>
</tr>
<tr>
<td>QZUDA1 QZUS01A1</td>
<td>TS</td>
<td>PACTHI</td>
<td>94</td>
</tr>
<tr>
<td>QZUDA1 QZUS01A1</td>
<td>TS</td>
<td>REORSPAC</td>
<td>90</td>
</tr>
<tr>
<td>QZUDA1 QZUS01A1</td>
<td>TS</td>
<td>SPACE</td>
<td>90</td>
</tr>
<tr>
<td>QZUDA1 QZUS02A1</td>
<td>TS</td>
<td>TOTALIND</td>
<td>0</td>
</tr>
<tr>
<td>QZUDA1 QZUS02A1</td>
<td>TS</td>
<td>DSEXTENT</td>
<td>1</td>
</tr>
<tr>
<td>QZUDA1 QZUS02A1</td>
<td>TS</td>
<td>EXTENTS</td>
<td>1</td>
</tr>
</tbody>
</table>

For a description of the fields on this panel, see Figure 230. By selecting an entry on this Exceptions Selection panel, you display an Exceptions report that shows the current values, compare values, and threshold values for all exceptions found for the specified object (Figure 233).
Figure 233  Exceptions report for all exception types found for object

For a description of the fields on this panel, see Figure 231.

**NOTE**
In this report, the ISPF RIGHT and LEFT commands are unavailable for scrolling right or left, respectively. To scroll right or left, enter GORIGHT or GOLEFT or use the standard function keys.

**Table column detail report**

Use this procedure to create a report that displays column definitions for all columns in a table.

**To create a table column detail report**

1 Access the Table Column Detail panel as follows:

   **A** On the DASD MANAGER PLUS main menu, select Reports and press Enter.

   **B** On the Report Selection Menu, select Table Reports and press Enter.

   **C** On the Table Reports Selection Menu, select Table Column Detail and press Enter.

   The Table Column Detail panel appears (Figure 234).
Figure 234  Table Column Detail panel

DECA ----------------------------- TABLE COLUMN DETAIL ----------------------------
COMMAND ===>

The fields below determine the list of objects the report will use. Type the full or wild card name of the objects, then press Enter

Database....................
Database.Tablespace name ....
Creator.Table name ..........

**** WARNING ****
This report may produce thousands of lines of output when reporting on a single table. Be very specific when supplying the object names. Since thousands of lines of output may be produced, this report may also execute for extended periods of time.

Batch Report : N (Y/N)

2  Specify one of the following options:

- In the Database field, type the name of the database to display statistics for all tables in a database.

- In the Database.Tablespace name field, type the qualified name of the table space to display statistics for all tables in a table space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the Creator.Table name field, type the qualified name of the table to display statistics for a table. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

3  In the Batch Report field, type Y in the field and press Enter to produce the report in batch.

One of the following events happens:

- If you specified Y in the Batch Report field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified N in the Batch Report field, the Table Column Detail report appears (Figure 235).
The Table Column Detail report contains the following fields:

- **Table Name** is the name of the table that contains the column.
- **Column** is the column name.
- **Part** is the number of the table space partition if the table space is partitioned.
- **ColCard** is the number of rows in the column.
- **High2Key** is the second highest key value of the column (truncated to 8 bytes).
- **Low2Key** is the second lowest key value of the column (truncated to 8 bytes).
- **Coltype** is the data type of the column.
- **Col Len** is the length of the column.
- **Col Scale** is the scale of decimal data.
Table space extent summary report

Use this procedure to create a report that helps you decide whether to change the table space definition.

To create a table space extent summary report

1. Access the Tablespace Extent Summary panel as follows:

   A. On the DASD MANAGER PLUS main menu, select Reports and press Enter.


   The Tablespace Extent Summary panel appears (Figure 236).

Figure 236 Tablespace Extent Summary panel

2. Specify one of the following options:

   - In the Database field, type the name of the database to display statistics for all tables in a database.

   - In the Database.Tablespace name field, type the qualified name of the table space to display statistics for all tables in a table space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.
In the **Creator.Table name** field, type the qualified name of the table to display statistics for a table. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

3 In the **Data set extents >=** field, type a minimum number of extents to display only objects that have more than a specified number of extents.

4 In the **Report style** field, specify the source of the report statistics, in the following order:

   1. the most recent execution of BMCSTATS
   2. the newest and oldest execution
   3. all executions

5 In the **Batch Report** field, type Y in the field and press **Enter** to produce the report in batch.

   One of the following events happens:

   - If you specified Y in the **Batch Report** field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

   - If you specified N in the **Batch Report** field, the Tablespace Extent Summary report appears (Figure 237).
The Tablespace Extent Summary report contains the following fields:

- **Database Name/Table Space Name** is the fully qualified table space name.
- **Part Nbr** is the number of the table space partition.
- **TS Type** is the type of table space.
- **Number of Rows** is the cardinality of the table space or table space partition.
- **Xtnts** is the number of extents that the table space is in.
- **Nactive** is the number of active pages in the table space. The product considers pages with row format active even if they currently contain no rows.
- **Npages** is the number of pages in the table space.
- **Primary (K)** is the primary allocation quantity in kilobytes.
- **Secndry (K)** is the secondary allocation quantity in kilobytes.
- **BMCSTATS Execution Time** is when BMCSTATS collected the statistics in the report.
Table space offset RIDs report

Use this procedure to create a report that helps to determine when to reorganize data by showing the number of rows not on their original page.

To create a table space offset RIDs report

1. Access the Tablespace Offset RIDs panel as follows:

   A. On the DASD MANAGER PLUS main menu, select Reports and press Enter.


   The Tablespace Offset RIDs panel appears (Figure 238).

2. Specify one of the following options:

   - In the Database field, type the name of the database to display statistics for all tables in a database.

   - In the Database.Tablespace name field, type the qualified name of the table space to display statistics for all tables in a table space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.
In the **Creator.Table name** field, type the qualified name of the table to display statistics for a table. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

3 In the **Percent of rows not on page >=** field, type the minimum percentage of rows that you want reported that are not on their original page.

4 In the **Batch Report** field, type Y in the field and press **Enter** to produce the report in batch.

One of the following events happens:

- If you specified Y in the **Batch Report** field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.
- If you specified N in the **Batch Report** field, the Tablespace Offset RIDs report appears (Figure 239).

**Figure 239 Tablespace Offset RIDs report**

The Tablespace Offset RIDs report contains the following fields:

- **Database Name/Table Space Name** is the fully qualified table space name.
- **Part Nbr** is the number of the table space partition.

- **Near Indref** is the number of indirect references to rows that are less than 16 pages from the original page because of an insert into a full page.

- **Far Indref** is the number of indirect references to rows that are 16 or more pages from the original page because of an insert into a full page.

- **Nbr of Rows** is the number of rows in the table space or table space partition.

- **BMCSTATS Execution Time** is when BMCSTATS collected the statistics in the report.

---

**Table space space detail report**

Use this procedure to create a report that provides a quick reference of space usage by table space.

**To create a table space space detail report**

1. Access the Tablespace Space Detail panel as follows:

   A. On the DASD MANAGER PLUS main menu, select **Reports** and press **Enter**.

   B. On the Report Selection Menu, select **Space Allocation Reports** and press **Enter**.

   C. On the Space Allocation Reports Selection Menu, select **Tablespace Space Detail** and press **Enter**.

   D. The Tablespace Space Detail panel appears (Figure 240).
2 Specify one of the following options:

- In the **Database** field, type the name of the database to display statistics for all tables in a database.

- In the **Database.Tablespace name** field, type the qualified name of the table space to display statistics for all tables in a table space. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

- In the **Creator.Table name** field, type the qualified name of the table to display statistics for a table. If you type a name without the first qualifier, the product assumes that you entered a wildcard.

3 In the **Batch Report** field, type **Y** in the field and press **Enter** to produce the report in batch.

One of the following events happens:

- If you specified **Y** in the **Batch Report** field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified **N** in the **Batch Report** field, the Tablespace Space Detail report appears (Figure 241).
The Tablespace Space Detail report is divided vertically into two parts. The data on the left is from the ICF catalog. The data on the right is from the DB2 catalog.

The report contains the following fields:

- **DSN** is the fully qualified data set name.
- **Volser** is the volume serial number of the first volume for the table space.
- **Devtype** is the device type where the table space is stored.
- **Extents** is the number of extents that the table space or table space partition is in.
- **Kbytes Alloc** is the number of kilobytes allocated for the table space.
- **Kbytes Used** is the number of kilobytes that the table space uses.
- **Tracks Alloc** is the number of tracks allocated for the table space.
- **Tracks Used** is the number of tracks that the table space uses.
- **Alloc Unit** is the allocation unit (T for tracks or C for cylinders).
- **Prim Alloc** is the primary allocation quantity.
- **Sec Alloc** is the secondary allocation quantity.
- **Object Name** is the fully qualified name of the table space.
- **Object Type** is table space.
- **Part** is the number of the table space partition.
- **Card** is the number of rows in the table space.
- **Page size** is the page size: 4, 8, 16, or 32 kilobytes.
- **TS Type** is the type of table space.
- **Lobdatatype** is the type of LOB table space.
- **Implicit** indicates whether the database was implicitly created.
- **Storname** is the name of the storage group or ICF catalog used for space allocation.
- **Primary** is the primary allocation quantity.
- **Secondary** is the secondary allocation quantity.
- **Freepage** is the number of free pages.
- **Percent Free** is the percentage of the allocated space that is free.
- **LOB** indicates whether the table space is a LOB table space, base, or auxiliary.
- **Bpool** is the name of the buffer pool.
- **Lockmax** is the maximum number of locks per user to acquire for the table space before escalating.
- **Compress** indicates whether the table space is compressed.
- **Segsize** is the segment size.
- **Dsnum** indicates the data set number.
- **Dssize** indicates the size of the data set.
- **Member Cluster** indicates whether the table space has a member cluster page set structure.
- **Lockrule** is the lock size of the table space.
- **Log** indicates whether changes to the table space are to be logged.
Volume analysis report

Use this procedure to create a report that helps to determine where to place table spaces or index spaces.

To create a volume analysis report

1. Access the Volume Analysis panel as follows:

   A. On the DASD MANAGER PLUS main menu, select Reports and press Enter.


   The Volume Analysis panel appears (Figure 242).

Figure 242  Volume Analysis panel

```
DECA ----------------------------- VOLUME ANALYSIS -----------------------------
COMMAND ===>

The fields below determine the list of objects the report will use. Type the full or wild card name of the objects, then press Enter

Volume name....................
Where
Cylinders needed >=......... 0

Report style.................... 1 1. Most recent execution
                                         2. Newest and Oldest execution
                                         3. All executions

Batch Report : N (Y/N)
```

2. In the Volume name field, specify one or more volume names as follows:

   - Type its serial number to display statistics on one volume, including free cylinders.

   - To display statistics on multiple volumes, type a wildcard pattern.

3. In the Cylinders needed field, type the minimum number of cylinders needed to display volumes that have a specified number of free cylinders. Use this option with the wildcard volume name.
4 In the Report style field, specify the source of the statistics for the report, in the following order:

1. the most recent execution of BMCSTATS
2. the newest and oldest execution
3. all executions

5 In the Batch Report field, type Y in the field and press Enter to produce the report in batch.

One of the following events happens:

- If you specified Y in the Batch Report field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified N in the Batch Report field, the Volume Analysis report appears (Figure 243).

Figure 243  Volume Analysis report
The Volume Analysis report contains the following fields:

- **Volser** is the volume serial number of the DASD volume.
- **DB2 Tracks Alloc** is the number of tracks allocated for DB2 data sets on the volume.
- **DB2 Data Sets** is the number of DB2 data sets on the volume.
- **Non DB2 Data Sets** is the number of non-DB2 data sets on the volume.
- **Total Data Sets** is the total number of data sets on the volume.
- **Free Cyl** is the number of free cylinders on the volume.
- **Free Trk** is the number of free tracks on the volume.
- **Free Extents** is the number of free extents on the volume.
- **BMCSTATS Execution Time** is when BMCSTATS collected the statistics in the report.

**Volume free space detail report**

Use this procedure to create a report that provides additional information about volumes that are candidates for new table spaces and index spaces.

*NOTE*

This report lists only volumes named for the STOGROUP and stored in the SYSIBM.SYSVOLUMES catalog table. It will not provide volume analysis for SMS-managed storage groups defined with an asterisk (*) or SMS constructs.

**To create a volume free space detail report**

1. Access the Volume Free Space Detail panel as follows:

   A. On the DASD MANAGER PLUS main menu, select **Reports** and press **Enter**.

   B. On the Report Selection Menu, select **Space Allocation Reports** and press **Enter**.

   C. On the Space Allocation Reports Selection Menu, select **Volume Free Space Detail (Real Time)** and press **Enter**.

   The Volume Free Space Detail panel appears (Figure 244).
2 Specify one of the following options:

- In the Stogroup name field, type the name of the storage group to display statistics for all volumes in a storage group. You can also type a wildcard pattern.

- In the Volume name field, type its serial number to display statistics for a particular volume. You can also type a wildcard pattern.

3 In the Cylinders needed >= field, type a minimum number of cylinders to display volumes having a specified number of free cylinders.

4 In the Batch Report field, type Y in the field and press Enter to produce the report in batch.

One of the following events happens:

- If you specified Y in the Batch Report field, the product produces JCL in a temporary data set when you exit the report dialog, and a prompt panel gives you the option to edit and submit the JCL.

- If you specified N in the Batch Report field, the Volume Free Space Detail report appears (Figure 245).
Because of rounding, in some cases percentage values might add up to slightly more or less than 100 percent.

The Volume Free Space Detail report contains the following fields:

- **Volume Name** is the volume serial number of the DASD volume.
- **Dev Type** is the type of DASD volume, for example, 3380 or 3390.
- **Alloc Space** is the space that has been allocated on the volume.
- **%** is the percentage of allocated space that has been used.
- **Free Cyls** is the number of unused cylinders on the volume.
- **%** is the percentage of free cylinders that has not been used.
- **Addtl Tracks** is the total number of free tracks.
- **%** is the percentage of free tracks in cylinders that has not been used.
- **Largest Extent** is the largest free extent, calculated on a cylinder boundary.
  - **Cyls** is the number of cylinders in the largest free extent.
  - **Addtl Tracks** is the number of free tracks before or after the last cylinder boundary in the contiguous block of space.
Model JCL for batch generation

DASD MANAGER PLUS provides automatic batch generation of reports. However, you can also modify and use the sample JCL in Figure 246 for batch generation of the reports on the DASD MANAGER PLUS Reports Selection Menu. You can find the JCL in the Control (CNTL) library member ASURBAT. Notes for the JCL also are in the CNTL library in member ASURDOC.

NOTE
Use the same plan as bound for the online batch reports.

Figure 246  JCL for batch generation

```plaintext
//________ JOB (____),'ASURBAT',MSGCLASS=X,CLASS=A,NOTIFY=_______
//*
//ASURBAT EXEC PGM=ASURBAT,PARM='DB2,ASU812DR,DB2VCAT,ASUDOPTs'
//*---------------------------------------------------------------
// PARMS 1.SSID = DB2 SUBSYSTEM ID DEFAULT = DB2
// 2.PLAN = ASU PLAN NAME DEFAULT = ASU812DR
// 3.VCAT = DB2 CAT HIGH LVL QUAL. DEFAULT = DB2VCAT
// 4.DOPTs= DEFAULT OPTIONS MODULE DEFAULT = ASUDOPTs
//*---------------------------------------------------------------
//STEPLIB DD DISP=SHR,DSN=YOUR.DASDMGR.LOAD
// DD DISP=SHR,DSN=YOUR.DB2.DSNLOAD
// DD DISP=SHR,DSN=YOUR.DB2.DSNEXIT
//*
//SYSTERM DD SYSOUT=*,DCB=BLKSIZE=142
//SYSPRINT DD SYSOUT=*,DCB=BLKSIZE=142
//* >>>>>> SEE DOCUMENTATION OF SYSIN PARMS IN MEMBER ASURDOC <<<<<<<<
//*---------------------------------------------------------------
//SYSIN DD *
RUN OFSETRID
DATABASE BMCASU
STYLE 2
NOT_ON_PAGE 5
//SYSUDUMP DD SYSOUT=*  
```
QMF reports

DASD MANAGER PLUS provides sample QMF queries that produce reports from the historical statistics database. You can select one or more different views for the sample reports. Table 84 shows views that the CNTL library member ASURVIEW provides for use by the QMF reporting procedures.

Table 84 Views for QMF reports

<table>
<thead>
<tr>
<th>View</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCQMF.RS_MO_AV_IXPART</td>
<td>This view produces averages for statistics by month on all indexes in the statistics database for the last three months. The view points to the BMCQMF.RS_USER_IXPART view.</td>
</tr>
<tr>
<td>BMCQMF.RS_MO_AV_TSPART</td>
<td>This view produces averages by month over the last three months for all table spaces in the statistics database. The view points to the BMCQMF.RS_USER_TABLEPART view.</td>
</tr>
<tr>
<td>BMCQMF.RS_MO_AV_TSPART12</td>
<td>This view produces averages for statistics by month on all table spaces in the statistics database for the last twelve months. The view points to the BMCQMF.RS_USER_TSPART view.</td>
</tr>
<tr>
<td>BMCQMF.RS_TABLEPART</td>
<td>This view provides the base for BMCQMF views that require the grouping of data by monthly periods (includes YYMM). The view points to the BMCATSnn.RS_TABLEPART table (where nn is the release level).</td>
</tr>
<tr>
<td>BMCQMF.RS_USER_DBIX</td>
<td>This view provides a count of the number of indexes in each database. The view points to the BMCQMF.RS_MO_AV_IXPART view.</td>
</tr>
<tr>
<td>BMCQMF.RS_USER_DBTS</td>
<td>This view creates summary statistics from the average table space statistics for each database in the statistics tables. The view points to the BMCQMF.RS_MO_AV_TSPART view.</td>
</tr>
<tr>
<td>BMCQMF.RS_USER_IXPART</td>
<td>This view selects detail statistics on all indexes for which statistics have been collected within the last three months and groups them by database and month. The view points to the BMCATSnn.RS_INDEXPART table (where nn is the release level).</td>
</tr>
<tr>
<td>BMCQMF.RS__USER_TBCOUNTS</td>
<td>This view creates a count of the number of tables in each table space. The view points to the BMCQMF.RS_USER_TBNAMES view.</td>
</tr>
<tr>
<td>BMCQMF.RS_USER_TBNAMES</td>
<td>This view selects all table names in the statistics tables where statistics have been collected within the last three months. The view points to the BMCATSnn.RS_TABLES table (where nn is the release level).</td>
</tr>
<tr>
<td>BMCQMF.RS_USER_TSPART</td>
<td>This view selects detail information about table spaces from statistics entries less than three months old. The view points to the BMCATSnn.RS_TABLEPART table (where nn is the release level).</td>
</tr>
<tr>
<td>BMCQMF.RS_USER_TSPART12</td>
<td>This view selects detail statistics for all table spaces for which statistics have been collected within the last twelve months and groups them by database and month. The view points to the BMCATSnn.RS_TABLEPART table (where nn is the release level).</td>
</tr>
</tbody>
</table>
DASD MANAGER PLUS provides the sample QMF reports (Table 85) to access useful information in the statistics tables.

### Table 85  Descriptions of QMF reports

<table>
<thead>
<tr>
<th>Report</th>
<th>Query</th>
<th>Form</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBMONTHLY</td>
<td>QDBMONTHLY</td>
<td>FDBMONTHLY</td>
<td>BMCQMF.RS_USER_DBTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBSUMMARY</td>
<td>QDBSUMMARY</td>
<td>FDBSUMMARY</td>
<td>BMCQMF.RS_USER_TBCOUNTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BMCQMF.RS_USER_DBTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BMCQMF.RS_USER_DBIX</td>
</tr>
<tr>
<td>MONTHCHANGE</td>
<td>QMONTHCHANGE</td>
<td>FMONTHCHANGE</td>
<td>BMCQMF.RS_MO_AV_TSPART</td>
</tr>
<tr>
<td>PARTSTATS3</td>
<td>QPARTSTATS3</td>
<td>FPARTSTATS3</td>
<td>BMCQMF.RS_MO_AV_TSPART</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPACESTATS12</td>
<td>QSPACESTATS12</td>
<td>FSPACESTATS12</td>
<td>BMCQMF.RS_MO_AV_TSPART</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSBYMONTH</td>
<td>QTSBYMONTH</td>
<td>FTSBYMONTH</td>
<td>BMCQMF.RS_MO_AV_TSPART</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSBYSET</td>
<td>QTSBYSTATSSET</td>
<td>FTSBYSTATSSET</td>
<td>BMCATSnn.RS_TABLEPART</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Use the following procedure to set up the sample QMF reports feature.

**To implement the sample QMF Report Facility**

1. Build views for the DASD MANAGER PLUS sample QMF reports. Run SPUFI and specify the CNTL data set and member ASURVIEW.

   **NOTE**
   
   If you are installing DASD MANAGER PLUS on multiple DB2 subsystems, run this SPUFI job for each DB2 SSID that uses the DASD MANAGER PLUS QMF reports.

2. Import the DASD MANAGER PLUS QMF Reports into QMF.

   **A** Edit all members of the QMFPROC data set, as follows:
   
   - Change **HLQ** to the high-level qualifier for DASD MANAGER PLUS data sets at your installation.
   
   - Change **U********* to the logon ID of the person running the QMF install.

   **B** From QMF, execute the following command:

   ```sql
   IMPORT PROC FROM 'HLQ.QMFPROC' (M=LOADPROC)
   ```

   **NOTE**
   
   Type the data set name in capital letters. You can execute the IMPORT command from the first QMF panel or from the Query panel.

   **C** Press **PF2** or enter the RUN command to run the procedure which imports all of the BMC-supplied DASD MANAGER PLUS QMF reports.
Customizable reports

DASD MANAGER PLUS provides fully documented, customizable report programs that you can run online or in batch mode. With the report programs, you can produce reports on data in the DASD MANAGER PLUS historical database, the DB2 catalog, or your own databases. You can run the reports as they are, customize them, or use the provided external functions to write your own report programs.

NOTE
When preparing to run a customizable report in batch, you must perform one of the following items:

- Review and edit the SYSEXEC DD statement to provide the correct DSN for the report code.
- Set the Sysexec variable on the DASD MANAGER PLUS JCL Generation Jobcard Options Update panel. Providing this information in your options before generating batch JCL prevents you from having to provide this information each time you generate a batch job for a customized report.

The DASD MANAGER PLUS reports are in REXX, a general-purpose, interpreted language that exists on all OS/390 systems. Because REXX is an interpreted language, the development cycle for REXX programs is short. Specifically, you edit and run instead of edit, compile, link, bind, and run.

Appendix B, “Customizable reports” describes the DASD MANAGER PLUS external functions, formatting, REXX conventions, and other aspects of the reports. The appendix also contains an annotated sample report program.

The following pages describe the DASD MANAGER PLUS report programs, show examples, and provide the steps to run or customize them.

Understanding customizable reports

The DASD MANAGER PLUS customizable reports include space trend reports, space estimation reports, reports on storage limits and utilization, and detailed reports by database and action. You can find the customizable reports in Table 86 in the .EXEC library.
Table 86  DASD MANAGER PLUS customizable reports (part 1 of 2)

<table>
<thead>
<tr>
<th>Report title</th>
<th>Description</th>
<th>EXEC name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Report</td>
<td>This report gives the contents of actions—identifying their services, objects, object type, partition, and any object set. If the service in the action is BMCTRIG, the report also identifies the corrective action.</td>
<td>ASUACTN</td>
</tr>
<tr>
<td>Average Monthly Growth Report(^{a})</td>
<td>This report lists the average monthly growth in cardinality, tracks, and pages for table spaces and indexes for the past twelve months based on the statistics taken.</td>
<td>ASUAMGR</td>
</tr>
<tr>
<td>Corrective Action Cross Reference Report</td>
<td>This report identifies all actions that are associated with a corrective action in the DB2 subsystem.</td>
<td>ASUDFACT</td>
</tr>
<tr>
<td>Data Set List by Database Name Report</td>
<td>This report lists information about the data sets residing on the specified database.</td>
<td>ASUZDSLI</td>
</tr>
<tr>
<td>Database Space Detail Report(^{a})</td>
<td>This report provides detailed information about table space and index space allocation, current space utilization, and reorganized space estimation.</td>
<td>ASUDSPAC</td>
</tr>
<tr>
<td>Database Space Trend Report(^{a})</td>
<td>This report predicts the number of days until a DB2 object will exceed the maximum number of extents, the space limit, or the space on the volume.</td>
<td>ASUSPCTR</td>
</tr>
<tr>
<td>Database Storage Utilization Report</td>
<td>This report lists the space allocated, the space used, the percentage of space used, and the number of extents for each table space and index of a DB2 database; also shows totals by table space, index, and database.</td>
<td>ASUDBSPA</td>
</tr>
<tr>
<td>Exception Report by BMCTRIG Action(^{a})</td>
<td>This report lists exceptions that a BMCTRIG job detected.</td>
<td>ASUEXCPT</td>
</tr>
<tr>
<td>Index Clustering Report(^{a})</td>
<td>This report lists, for clustering indexes, the unique rule (UR) of the index, the percentage of rows in clustering order, and the number of rows not in optimal position.</td>
<td>ASUCRANA</td>
</tr>
<tr>
<td>Modified Table Space Pages Report(^{a})</td>
<td>This report lists the number of rows in a table space, the number of active pages, the number of pages modified since the last copy, the percentage of modified pages, and the recommended type of copy to make.</td>
<td>ASUTSDIR</td>
</tr>
<tr>
<td>Space Estimation Report(^{a})</td>
<td>This report estimates the reorganized space of a DB2 object based on the specified percentage change in the object’s cardinality and the specified percentage free and number of free pages.</td>
<td>ASUPSE</td>
</tr>
<tr>
<td>Space Estimation Trend Report(^{a})</td>
<td>This report uses linear regression of BMCSTATS historical data to estimate the cardinality of an object and then computes the reorganized space of an object.</td>
<td>ASUTRSE</td>
</tr>
<tr>
<td>Space Utilization by Table Report(^{a})</td>
<td>This report lists the number of pages containing table data, the percentage of pages that contain rows, and the table cardinality for the specified DB2 objects.</td>
<td>ASUTBSPA</td>
</tr>
<tr>
<td>Table Space and Index Extents Report(^{a})</td>
<td>This report lists the allocated space, used space, and number of extents for DB2 table spaces and indexes that equal or exceed a specified number of extents.</td>
<td>ASUDBEXT</td>
</tr>
</tbody>
</table>
Understanding customizable reports

### Table 86  DASD MANAGER PLUS customizable reports (part 2 of 2)

<table>
<thead>
<tr>
<th>Report title</th>
<th>Description</th>
<th>EXEC name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Space and Index Storage Limits Report&lt;sup&gt;a&lt;/sup&gt;</td>
<td>This report shows the storage limits for DB2 table spaces or indexes, showing the percentage of the physical object-limit that has been used.</td>
<td>ASUDBLIM</td>
</tr>
<tr>
<td>Tables with More than Average Growth Report&lt;sup&gt;a&lt;/sup&gt;</td>
<td>This report identifies DB2 tables that are growing at more than the average rate, based on the last two sets of statistics taken.</td>
<td>ASUNPGR</td>
</tr>
<tr>
<td>Volume Space Trend Report&lt;sup&gt;a&lt;/sup&gt;</td>
<td>This report predicts the number of days until DASD volumes will run out of space.</td>
<td>ASUVOLTR</td>
</tr>
</tbody>
</table>

<sup>a</sup> To run this report, DASD MANAGER PLUS requires BMCSTATS.

All reports have the same general layout: the title and page number at the top, followed by the product name *DASD MANAGER PLUS*, the EXEC name of the report (from Table 86), the DB2 subsystem ID (SSID), version of DB2, and the date and time. The following example shows a typical header.

```
TABLE SPACE STORAGE LIMITS     PAGE    1
DASD MANAGER PLUS   ASUDBLIM   DBDA  812   22 Aug 2005 17:02:31
```

If you remove the top-of-page formatting (the title, page number, and column headings) other utilities can use the generated report as input.

In addition to providing data on DB2 objects, some statistical reports include summaries by object type, such as table space, index, or database.

### Input parameters

In the online dialog, the **Parms** input line appears above a line showing the required input (see example in Figure 247).

**Figure 247** Parms input line and required input line

```
Parms  DBDA ASU812DR
ssid plan  dbname pctlimit
```

DASD MANAGER PLUS supplies the DB2 subsystem identifier (SSID) and plan name automatically. You supply the rest of the parameters.

For each report, look at the required input line, tab to the third input field, if any, and type any remaining parameters, adding at least one space between them. For example, to complete the input parameters in Figure 247, type a database name and pctlimit. (Pctlimit and the other input parameters are described in Table 87.)
The last page of the generated report shows the parameters that you specified on theParms input line (Figure 248), which identifies the parameters that produced a particular report.

**Figure 248 Input parameters at the end of the report**

<table>
<thead>
<tr>
<th>TABLE SPACE AND INDEX STORAGE LIMITS REPORT</th>
<th>PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS ASUDBLIM DECA 810 22 Aug 2005 10:13:56</td>
<td></td>
</tr>
</tbody>
</table>

Input Parameters

<table>
<thead>
<tr>
<th>ssid plan dbname pctlimit</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECA ASU812DR QZUD2% 10</td>
</tr>
</tbody>
</table>

**Figure 249** shows the same input parameters as they appear in the generated batch JCL. In the batch JCL, DASD MANAGER PLUS automatically supplies the program name, SSID, and plan name before the input parameters that you specify in the report dialog.

---

**NOTE**

When preparing to run a customizable report in batch, you must perform one of the following items:

- Review and edit the SYSEXEC DD statement to provide the correct DSN for the report code.

- Set the Sysexec variable on the DASD MANAGER PLUS JCL Generation Jobcard Options Update panel. Providing this information in your options before generating batch JCL prevents you from having to provide this information each time you generate a batch job for a customized report.

---

**Figure 249 Input parameters in generated batch JCL**

```// PARM='ASUDBLIM DBDA ASU812DR QZUD2% 10'```

**Description of input parameters**

Table 87 describes input parameters that are used to produce the various reports.
### Table 87  Input parameters (part 1 of 2)

<table>
<thead>
<tr>
<th>Name of input parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| action                  | name of the action  
Wildcards are allowed. |
| cr                      | clusterratio percentage of rows that are in clustering order |
| days                    | number of days from date of report to compute the reorganized space of a DB2 object (1 to 366) |
| dbname                  | name of the database  
Wildcards are valid. |
| extent                  | number of extents that the object must equal or exceed to include the object in the report  
The maximum allowed is 7257 extents. For non-Large table spaces, the number is figured as follows:  
  - 32 data sets per table space * 7257 extents per data set (119 per volume)  
  - 32 data sets per index * 7257 extents per data set (119 per volume)  
For Large table spaces, the number is figured as follows:  
  - 32 data sets per table space * 7257 extents per data set (119 per volume) (cannot be greater 4 GB)  
  - 128 data sets per index * 7257 extents per data set (119 per volume) (if the index supports Large partitioned table spaces)  
For table spaces: the maximum 7257 extents * the maximum 32 data sets per table space = maximum 232224 extents.  
For indexes: the maximum 7257 extents * the maximum 128 data sets = maximum value far beyond 928896. (If at this level, you should run REORG.) |
| farpct                  | percentage of rows that are far from optimal position because of an insert into a full page |
| ixfpg                   | number of free pages for indexes |
| ixpf                    | percentage of free space for indexes |
| nearpct                 | percentage of rows that are near, but not in, optimal position because of insert into a page |
| observations            | number of the most recent rows from BMCSTATS historical data to use for linear regression analysis |
Understanding customizable reports

Table 87  Input parameters (part 2 of 2)

<table>
<thead>
<tr>
<th>Name of input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pctcard</td>
<td>percentage of cardinality to be added to the most recent cardinality of the object. This cardinality is then used in the computation for the estimated reorganized space of the object.</td>
</tr>
<tr>
<td>pctdirty</td>
<td>percentage of dirty pages that a table space must reach or exceed to be included in the report.</td>
</tr>
<tr>
<td>pctlimit</td>
<td>percentage of maximum space that is used by the table space or index. If the ratio of (used/limit) * 100 is less than pctlimit, then the object is not reported.</td>
</tr>
<tr>
<td>plan</td>
<td>plan to which the DBRM of the REXX SQL plan (ASURXSQL) is bound. To identify the customizable reports plan name, select User Options on the DASD MANAGER PLUS main menu, then select Current Environment information and find the Stats plan.</td>
</tr>
<tr>
<td>qualname</td>
<td>creator or dbname; wildcards are allowed.</td>
</tr>
<tr>
<td>qualttype</td>
<td>literal CREATOR or DBNAME</td>
</tr>
<tr>
<td>report-name</td>
<td>name of the report program, such as ASUNPGR. Table 86 lists all of the report names.</td>
</tr>
<tr>
<td>service</td>
<td>name of the service; wildcards are allowed.</td>
</tr>
<tr>
<td>ssid</td>
<td>DB2 subsystem ID (SSID) where the report is to run</td>
</tr>
<tr>
<td>table-creator</td>
<td>table creator; wildcards are allowed</td>
</tr>
<tr>
<td>tspf</td>
<td>number of free pages for table spaces</td>
</tr>
<tr>
<td>tsnname</td>
<td>name of the table space; wildcards are allowed</td>
</tr>
<tr>
<td>tspf</td>
<td>percentage of free space for table spaces</td>
</tr>
<tr>
<td>vcatname</td>
<td>high-level qualifier of the VSAM catalog. Specify this parameter only for DSNDB% objects.</td>
</tr>
<tr>
<td>volume</td>
<td>the name of the volume, which can contain the %, *, or _ wildcard</td>
</tr>
</tbody>
</table>
Creating customizable reports

Internal KB switch

Two report programs, the Database Detail Space report and the Database Storage Utilization report, contain an internal KB switch. By using this switch (Figure 250), you can change each report’s units from tracks to KB.

Figure 250  Internal KB switch

if (1 = 0) then /* if equal then */
    units = "KB"; /* units in KB */
else /* else if not equal */
    units = "TRKS"; /* units in TRKS */

To report in KB, access the reports, ASUDSPAC and ASUDBSPA respectively, in the .EXEC library, and change the following expression: (1 = 0) to (1 = 1).

Creating customizable reports

The following procedures tell you how to run the customizable reports and how to customize the reports in various ways.

Running customizable reports

Use this procedure to run a customizable report online or in batch mode.

To create customizable reports

1 Access the Customizable Reports Selection List panel as follows:

A On the DASD MANAGER PLUS main menu, select Reports and press Enter.

B On the Report Selection Menu, select Customizable Reports and press Enter.

The customizable reports menu appears (Figure 251).
Running customizable reports

Chapter 10 Producing reports

Type S or / to select report(s) and then press Enter.

The first report panel appears.

Specify online or batch execution mode, as follows:

- For online execution, type S or slash (/) by the Online Report option.
- For batch execution, type S or slash (/) by the Batch Report option.

In theParms input line, type the input parameters, including one space between them, and press Enter.

NOTE

The input parameters appear on the line below theParms input line. For a description and example, see Figure 247. The input parameters are described in Table 87.

The report is submitted for online execution or the JCL is generated for batch execution, depending on the mode of execution specified in step 3.

For batch execution only, perform these additional steps:

A Type S or slash (/) by the Edit JCL option and press Enter.

B Review and edit the generated JCL (Figure 252).
 NOTE

The format of the generated batch JCL varies depending on whether you specify long name parameters. For all customized reports that include long name parameters, the keyword ASUINDD is inserted in the EXEC parameters, and the object specification is included as inline data. For an example of a creator name that does not include a long name, see Figure 252. For an example that shows the generated batch JCL for a long name parameter, see Figure 253.

Figure 252  Generated batch JCL from the online dialog (without long name)

```plaintext
//RDAJXSE JOB (5712), 'REPORT ASUDBEXT',
// CLASS=A, MSGCLASS=X, MSGLEVEL=(1,1),
// NOTIFY=RDAJXS
// *
// ASUNPGR EXEC PGM=IKJEFT01,
// PARM= 'ASUNPGR DEDK ASU912DR TABLE_CREATOR_NAME'
// STEPLIB DD DISP=SHR,
// DD DISP=SHR,
// DSN=CSGI.DB2V91M.DSNLOAD
// DD DISP=SHR,
// DSN=SYS3.DEDK.DSNEXIT
// *--------------------------------------------------------*
// * IF THE SYSEXEC DD STMT CONTAINS THE STRING *SYSEXEC* *
// * EITHER EDIT THE DD STMT TO PROVIDE THE CORRECT DSN *
// * OR SET THE SYSEXEC VARIABLE ON THE DASD MANAGER PLUS *
// * JCL GENERATION JOBCARD OPTIONS UPDATE PANEL *
// *--------------------------------------------------------*
// SYSEXEC DD DISP=SHR, DSN=*SYSEXEC*
// SYSTIN DD *
// SYSTSPRT DD SYSOUT=* 
// SYSPRINT DD SYSOUT=*, 
// DBC=(DSORG=PS, LRECL=142, BLKSIZE=7100, RECFM=VBA) 
// SYSTERM DD SYSOUT=* 
```

Figure 253  Generated batch JCL from the online dialog (with long name) (part 1 of 2)

```plaintext
//RDAJXSD JOB (5712), 'REPORT ASUDBEXT',
// CLASS=A, MSGCLASS=X, MSGLEVEL=(1,1),
// NOTIFY=RDAJXS
// *
// ASUNPGR EXEC PGM=IKJEFT01,
// PARM= 'ASUNPGR DEDK ASU912DR ASUINDD'
// STEPLIB DD DISP=SHR,
// DD DISP=SHR,
// DSN=CSGI.DB2V91M.DSNLOAD
// DD DISP=SHR,
// DSN=SYS3.DEDK.DSNEXIT
// *--------------------------------------------------------*
// * IF THE SYSEXEC DD STMT CONTAINS THE STRING *SYSEXEC* *
// * EITHER EDIT THE DD STMT TO PROVIDE THE CORRECT DSN *
```
Figure 253  Generated batch JCL from the online dialog (with long name) (part 2 of 2)

```
//* OR SET THE SYSEXEC VARIABLE ON THE DASD MANAGER PLUS  *
//* JCL GENERATION JOBCARD OPTIONS UPDATE PANEL  *
//*--------------------------------------------------------*
//SYSEXEC  DD DISP=SHR,DSN=*SYSEXEC*
//SYSTSIN  DD *
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,
//  DCB=(DSORG=PS,LRECL=142,BLKSIZE=7100,RECFM=VBA)
//SYSTERM  DD SYSOUT=*
//ASUINDD DD *
LONG_TABLE_CREATOR_NAME_EXAMPLE_SHOWING_LONG_NAME_STRING
/*
```

C  Type **SUBMIT** on the **COMMAND** line, and press **Enter**.

Alternatively, after reviewing and editing the generated JCL, press **END**, type **S** or slash (**/**) by the **Submit JCL** option, and press **Enter**.

---

**Running a customizable report in batch mode**

Use this procedure to run a customizable report as designed without using the online dialog. To report the latest statistics, run BMCSTATS first. The JCL for running the reports is in member ASURXJCL in the .CNTL library.

**To run a customizable report in batch mode**

1  Access the JCL for the customizable reports as instructed in **“To create customizable reports”** on page 650.

The JCL in **Figure 254** runs IKJEFT01, which interprets the DASD MANAGER PLUS report programs.

---

**Figure 254  Batch JCL to run reports outside the online dialog**

```
//* JOBNAME  JOB (acct), 'ASURXJCL', MSGCLASS=X, CLASS=A, NOTIFY=userid  
//* ASURXJCL EXEC PGM=IKJEFT01, PARM='REPORT SSID PLAN ...'
//*--------------------------------------------------------*
//* PARMS 1.REPORT = NAME OF REPORT TO EXECUTE  
//*  2.SSID = DB2 SUBSYSTEM ID OR DATA SHARING GROUP NAME  
//*  3.PLAN = ASU PLAN NAME  
//* ... REPORT DEPENDENT PARAMETER(S)  
//*--------------------------------------------------------*
//STEPLIB  DD DISP=SHR,DSN=H*******.LOAD  
// DD DISP=SHR,DSN=DB2******.DSNLOAD  
//SYSEXEC  DD DISP=SHR,DSN=H******.EXEC  
//SYSPRINT DD SYSOUT=*
```
2 Edit the Job card by typing a job name, accounting information, and the logon ID to notify.

3 Edit the EXEC statement, as follows:

   A After **PARM=**, type the name of the desired report over **REPORT**.
      The report names are in Table 86.

   B Type the DB2 subsystem ID or data-sharing group name over **SSID**.

   C Type the name of the plan to which the ASURXSQL DBRM is bound over **PLAN**.
      To identify the plan name, select **User Options** on the DASD MANAGER PLUS main menu. Then, select **Current Environment information**, and find the Stats plan.

   D After the plan name, type any input variables over the periods (** . . . **), ending the line with a single quotation mark.

      You can find the report-dependent parameters in this document and in the comments at the top of each report program.

4 Edit the STEPLIB DD statement, as follows:

   A Type the high-level qualifier of the production load library over **HLQ**.

   B Complete the name of the DB2 DSN LOAD library.

5 In the SYSEXEC DD statement, type the high-level qualifier of the **.EXEC** library over **HLQ**.

      This library contains the customizable report programs.

6 Type **SUBMIT** on the **COMMAND** line and press **Enter**.

7 (optional) Save the JCL for future use.
Customizing the report programs

Use this procedure to customize reports. Customization can be as simple as adding your company name to the header or as complex as writing new SQL and changing the report layout. Advanced users can use the ASURXRPT and ASURXSQL functions to write their own report programs.

To customize the report programs

1. Choose the customizable report that is most similar to the report that you want to create.

2. Review Appendix B, “Customizable reports” which includes an annotated example of a customizable report program.

3. Examine the report program and identify the areas of change.

4. Copy and change the program, as follows:
   - To add columns or remove columns, change the SQL and the report layout.
   - To change the report layout, change the column headings, format definitions, and the user-defined variables wherever they are referenced.
   - To report on data in a different table or set of tables, edit the lines that specify or reference the input tables. Figure 255 shows the list of input tables from one of the report programs.

5. If your new program is not in the DSN=HLQ.EXEC library, copy the new program into the library or edit the SYSEXEC DD statement to reference the library that contains the new program.

6. Run the new report program, using the steps in “Running a customizable report in batch mode” on page 653.

Figure 255 Input tables

```
sysdatabase  = "SYSIBM.SYSDATABASE";
sysindexes   = "SYSIBM.SYSINDEXES";
systables    = "SYSIBM.SYSTABLES";
systablespace = "SYSIBM.SYSTABLESPACE";
dmtablepart  = asualias("BMCASU_STABLEPART");
dmindexpart  = asualias("BMCASU_SINDEXPART");
```
Adding a report to the customizable reports menu

Use this procedure to add a report that you have created to the Customizable Reports Menu.

**To add a report to the customizable reports menu**

1. Access member ASURPTS in the .EXEC library.

   ASURPTS contains a REXX program that lists the customizable reports, their input parameters, and descriptions and returns them to the caller by queueing the data to the external data queue.

2. Decide where in the menu to place the new report.

   **TIP**
   
   You can place the new report anywhere in the menu.

3. Block copy a complete set of parameters (:REPORT, :PARMS, and :DESC) to the place you have chosen (see Figure 256 for an example).

**Figure 256  Set of report menu code**

<table>
<thead>
<tr>
<th>:REPORT</th>
<th>ASUAMGR</th>
<th>Average Monthly Growth Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>:PARMS</td>
<td>ssid plan</td>
<td>dbname</td>
</tr>
<tr>
<td>Where:</td>
<td>ssid</td>
<td>- DB2 subsystem id</td>
</tr>
<tr>
<td></td>
<td>plan</td>
<td>- plan name</td>
</tr>
<tr>
<td></td>
<td>dbname</td>
<td>- database name, may be wildcarded</td>
</tr>
<tr>
<td>:DESC</td>
<td>This report lists the average monthly growth in cardinality, tracks, and pages for table spaces and indexes for the past twelve months based on the statistics taken.</td>
<td></td>
</tr>
</tbody>
</table>
4 Edit the copied report name, parameters, and description to represent the new report, being careful to maintain the original spacing of the first three parameters and adding one space between subsequent parameters.

5 Save member ASURPTS.

6 Access and run your new report by using the steps in “Running customizable reports” on page 650.

Understanding output data sets

The report output goes into the following data sets:

- The reports are in the SYSPRINT data set.
- Error messages that the ASURXRPT and ASURXSQL functions issue are in the SYSTERM data set.
- SAY output of return codes and error messages are in the SYSTSPRT data set.

Location return code descriptions

The return codes for customizable reports are in the DASD MANAGER PLUS for DB2 Reference Manual.

Customizable report samples

The following pages provide samples of the current customizable reports.

Action report

The Action Report (Figure 257) identifies the contents of actions, including the services that they contain, the objects that the services run against, the object type, partition, object set name (if any), the owner, and the default corrective action (if any). Only actions that contain the BMCTRIG utility can contain a corrective action. Actions that show dashes are empty (have no steps).
The Action Report contains the following fields:

- **Action** is the action for which you generate the report. The actions in this column include any corrective actions.

- **Service** is the service or services that the action specifies. If no service has been specified, this field contains dashes (–).

- **Object** is the name of the DB2 object that the action service uses.

You can specify a wildcard name in the utility. If you do not specify a utility, this field contains dashes (–). If you do not specify a utility without specifying an object, this field is blank.

- **Type** is the DB2 object type.

- **Part** is the partition of the DB2 object (if specified in the action).
- **Object Set Name** is the object selection filter that is associated with the action service, if any.

- **Owner** is the DB2 object set owner name.

- **Default Corr Action** is the corrective action, if any. Only actions that contain the BMCTRIG utility can specify a corrective action.

**Input parameters**

The Action Report requires one of the following lines of input parameters:

<table>
<thead>
<tr>
<th>ssid plan</th>
<th>action service</th>
</tr>
</thead>
</table>

The report in *Figure 257* results from specifying only the *action* parameter. The report in *Figure 258* results from specifying both parameters. You can use wildcards for both *action* and *service*.
To report all services within the actions specified, omit the *service* parameter. To produce a report of all actions that contain a particular utility, specify the *service* parameter using a wildcard if you like, such as %COPY (Figure 258).

**Figure 258  Action Report: Example 2**

<table>
<thead>
<tr>
<th>ACTION REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS ASUACTN DECA 810 22 Aug 2005 11:05:30</td>
</tr>
<tr>
<td>ACTION</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>OSSQ0299</td>
</tr>
<tr>
<td>OSTB0118</td>
</tr>
<tr>
<td>OSTB0119</td>
</tr>
<tr>
<td>OSTB0120</td>
</tr>
<tr>
<td>OSTB0122</td>
</tr>
<tr>
<td>OSTB0244</td>
</tr>
</tbody>
</table>

**Action Report**

DASD MANAGER PLUS ASUACTN DECA 810 22 Aug 2005 11:05:30

Input Parameters

DECA ASU812DR O* %COPY
ssid plan action service

**Average monthly growth report**

The Average Monthly Growth report (Figure 259) lists the average monthly growth in cardinality, tracks, and pages for table spaces and indexes for the past twelve months, based on the collected statistics. The report also shows subtotals for the average monthly growth by partition.

**Figure 259  Average Monthly Growth report  (part 1 of 2)**

<table>
<thead>
<tr>
<th>AVERAGE MONTHLY GROWTH REPORT</th>
<th>PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS ASUAMGR- DEAE 22 Aug 2005 16:41:12</td>
<td></td>
</tr>
</tbody>
</table>

DBNAME: QZU040

<table>
<thead>
<tr>
<th>TSNAME</th>
<th>PART</th>
<th>MONTH</th>
<th>CARD</th>
<th>%CHANGE</th>
<th>#TRKS</th>
<th>%CHANGE</th>
<th>#PAGES</th>
<th>%CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUS0140</td>
<td>0</td>
<td>2002-05</td>
<td>201890</td>
<td>585</td>
<td>7020</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2002-10</td>
<td>101695</td>
<td>-49</td>
<td>315</td>
<td>-45</td>
<td>3780</td>
<td>-45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2002-11</td>
<td>87167</td>
<td>-13</td>
<td>315</td>
<td>0</td>
<td>3780</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2003-01</td>
<td>101695</td>
<td>17</td>
<td>315</td>
<td>0</td>
<td>3780</td>
<td>0</td>
</tr>
</tbody>
</table>
### Figure 259 Average Monthly Growth report (part 2 of 2)

<table>
<thead>
<tr>
<th>DBName</th>
<th>TSName</th>
<th>2002-05</th>
<th>2002-11</th>
<th>2003-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUS0440</td>
<td>0</td>
<td>60000</td>
<td>225</td>
<td>2700</td>
</tr>
<tr>
<td></td>
<td>2002-11</td>
<td>51429</td>
<td>-13</td>
<td>225</td>
</tr>
<tr>
<td></td>
<td>2003-01</td>
<td>60000</td>
<td>17</td>
<td>225</td>
</tr>
<tr>
<td></td>
<td>AVERAGE MONTHLY TS GROWTH:</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QZUS0540</td>
<td>1</td>
<td>2002-05</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>2002-11</td>
<td>0</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2003-01</td>
<td>0</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>AVERAGE MONTHLY TS GROWTH:</td>
<td>-14</td>
<td>-14</td>
<td>0</td>
</tr>
</tbody>
</table>

### Report fields

The Average Monthly Growth report contains the following fields:

- **DBName** is the name of the database for which the product reports statistics.
- **TSName** is the name of the table space.
Corrective action cross reference report

- **Part** is the number of the partition.

- **Month** is the year and month of the statistics in YYYY-MM format, where YYYY is the year and MM is the month.

- **Card** is the cardinality, or number of rows, in the table space.

- **%Change** is the percentage of change in the cardinality, if any, from the previously reported month.

- **#Trks** is the number of tracks in use that month.

- **%Change** is the percentage of change in the number of tracks, if any, from the previously reported month.

- **#Pages** is the number of pages in use that month.

- **%Change** is the percentage of change in the number of pages, if any, from the previously reported month.

- **Average Monthly TS Growth** is the average growth in the table space over the months reported, expressed in the percentage of change in the cardinality, the number of tracks, and the number of pages.

### Input parameters

The Average Monthly Growth Report requires the following input parameters:

```
ssid plan dbname
```

### Corrective action cross reference report

The Corrective Action Cross Reference report (Figure 260) lists each corrective action in the DB2 subsystem and the actions that specify them.

**Figure 260  Corrective Action Cross Reference report (part 1 of 2)**

```
DEFAULT CORRECTIVE ACTION CROSS REFERENCE PAGE 1

DASD MANAGER PLUS -ASUDFACT- DECA
6 Sep 2005  11:01:06

DEFAULT CORR ACTION ACTION
---------------------------
BC01WB04       BC01JB04
```

---

---
Report fields

The Corrective Action Cross Reference report contains the following fields:

- **Default Corr Action** is the default corrective action.
- **Action** is the BMCTRI action, if any.

Several actions can use the same corrective action. If an action has multiple BMCTRI services, the services can reference the same corrective action.

Input parameters

The corrective action Cross Reference Report requires the following input parameters:

- ssid plan

Data set list by database name report

The Data Set List by database name report (Figure 261) lists the tracks used, extents, allocated units, primary quantity, secondary quantity, and volume for each data set that is in the specified database. The report also shows the total tracks for table spaces, total tracks for indexes, and total tracks used in the database.
Report fields

The Data Set List by database name report contains the following fields:

- **Database Name** is the name of the specified database.
- **Data Set Name** is the name of each data set in the database.
- **Trks** is the number of tracks used by the data set.
- **Ext** is the number of extents on the data set.
- **AU** is the allocation unit for the data set (T – tracks or C – cylinders).
- **Pqty** is the primary quantity allocated to the data set.
- **Sqty** is the secondary quantity allocated to the data set.
- **Volume** is the volume where the data set resides.

Input parameters

The Data Set List by database name report requires the following input parameters:
Database space detail report

The Database Space Detail Report (Figure 262) provides detailed information about table space and index space allocation, current space utilization, and reorganized space estimation.

Figure 262  Database Space Detail Report

<table>
<thead>
<tr>
<th>DATABASE OBJECT</th>
<th>PART</th>
<th>TRKS</th>
<th>PQTY</th>
<th>TRKS</th>
<th>SQTY</th>
<th>TRKS</th>
<th>USED</th>
<th>TRKS</th>
<th>SP</th>
<th>CUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUD10.QZUS0110</td>
<td>0</td>
<td>90</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX01_D10S01T01</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX02_D10S01T01</td>
<td>0</td>
<td>90</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX03_D10S01T01</td>
<td>0</td>
<td>90</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX04_D10S01T01</td>
<td>0</td>
<td>90</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX05_D10S01T01</td>
<td>0</td>
<td>90</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUD10.QZUS0210</td>
<td>0</td>
<td>3</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX01_D10S02T01</td>
<td>0</td>
<td>3</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Input Parameters

DEAE ASU812DR QZUD10*
ssid plan  dbname

Report fields

The Database Space Detail Report contains the following fields:

- **Database Object** is the name of the table space or index in the specified database.
- **Part** is the partition number.
- **Trks Pqty** is the primary quantity allocation in tracks.
NOTE
To change report units to KB, see “Internal KB switch” on page 650.

- **Trks Sqty** is the secondary quantity allocation in tracks.
- **Current**
  - **Trks Used** is the number of tracks that the table space or index uses.
  - **Trks Space** is the number of tracks for the table space or index.
  - **Ext** is the number of extents that the table space or index is in.
  - **% Used** is the percentage of the allocated space that is currently used.
- **Reorganized**
  - **Trks Space** is the number of tracks that would be allocated if you reorganized the object.
  - **Ext** is the number of extents that the object would span if you reorganized it by using the current primary and secondary quantities.
  - **%Space Change** is the percentage of space that you will save by reorganization.

Negative numbers indicate that current space is larger than reorganized space. The formula used to calculate %Space Change is

$$100 \times \frac{\text{Reorganized Space} - \text{Current Space}}{\text{Current Space}}.$$

**Input parameters**

The Database Space Detail Report requires the following input parameters:

```plaintext
ssid plan dbname
```

**Database space trend report**

The Database Space Trend Report (Figure 263) predicts the number of days until a DB2 object exceeds the maximum number of extents, the object’s space limit, or the space on the volume. If running out of space is a frequent problem, run BMCSTATS on a regular basis for points of comparison. The report lists both table space partitions and index space partitions and their partition numbers, the reason for running out of space, and the name of the volume that contains the data set.
### Database Space Trend Report

**DASD MANAGER PLUS** -ASUSPCTR-  **DBDA**

**22 Aug 2005  15:22:02**

<table>
<thead>
<tr>
<th>DATABASE NAME</th>
<th>PART</th>
<th>TYPE</th>
<th>DAYS TO FULL</th>
<th>REASON</th>
<th>VOLUME</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCASU81</td>
<td>0</td>
<td>IP</td>
<td>118</td>
<td>VOLUME</td>
<td>DEV381</td>
</tr>
<tr>
<td>DXRSEVT</td>
<td>0</td>
<td>IP</td>
<td>181</td>
<td>EXTENTS</td>
<td>DEV370</td>
</tr>
<tr>
<td>DXRSEVT</td>
<td>0</td>
<td>IP</td>
<td>293</td>
<td>EXTENTS</td>
<td>DEV190</td>
</tr>
<tr>
<td>BMCSRSEX</td>
<td>0</td>
<td>TP</td>
<td>372</td>
<td>VOLUME</td>
<td>DEV311</td>
</tr>
<tr>
<td>DXRSTP</td>
<td>0</td>
<td>TP</td>
<td>510</td>
<td>EXTENTS</td>
<td>DEV102</td>
</tr>
<tr>
<td>BMCDSYNC</td>
<td>0</td>
<td>TP</td>
<td>613</td>
<td>EXTENTS</td>
<td>DEV145</td>
</tr>
<tr>
<td>BMCRSEX</td>
<td>0</td>
<td>TP</td>
<td>658</td>
<td>EXTENTS</td>
<td>DEV195</td>
</tr>
<tr>
<td>DXRSTP</td>
<td>0</td>
<td>TP</td>
<td>795</td>
<td>EXTENTS</td>
<td>DEV135</td>
</tr>
<tr>
<td>BMCSRSTP</td>
<td>0</td>
<td>TP</td>
<td>1235</td>
<td>EXTENTS</td>
<td>DEV170</td>
</tr>
<tr>
<td>BMCSRSTP</td>
<td>0</td>
<td>TP</td>
<td>1661</td>
<td>VOLUME</td>
<td>DEV165</td>
</tr>
<tr>
<td>BMCRSVOL</td>
<td>0</td>
<td>TP</td>
<td>1780</td>
<td>EXTENTS</td>
<td>DEV335</td>
</tr>
<tr>
<td>BMCRSTP</td>
<td>0</td>
<td>TP</td>
<td>1976</td>
<td>VOLUME</td>
<td>DEV511</td>
</tr>
<tr>
<td>DMCSTB2</td>
<td>0</td>
<td>IP</td>
<td>2052</td>
<td>EXTENTS</td>
<td>DEV120</td>
</tr>
<tr>
<td>BMCDOWNID</td>
<td>0</td>
<td>TP</td>
<td>2191</td>
<td>EXTENTS</td>
<td>DEV160</td>
</tr>
<tr>
<td>DMSYNC</td>
<td>0</td>
<td>IP</td>
<td>2209</td>
<td>EXTENTS</td>
<td>DEV383</td>
</tr>
<tr>
<td>BMCRSST</td>
<td>0</td>
<td>TP</td>
<td>2550</td>
<td>EXTENTS</td>
<td>DEV190</td>
</tr>
<tr>
<td>BMCSST</td>
<td>0</td>
<td>TP</td>
<td>2799</td>
<td>EXTENTS</td>
<td>DEV330</td>
</tr>
<tr>
<td>BMCSRST</td>
<td>0</td>
<td>TP</td>
<td>2855</td>
<td>VOLUME</td>
<td>DEV175</td>
</tr>
<tr>
<td>DMRSCIX</td>
<td>0</td>
<td>IP</td>
<td>3033</td>
<td>EXTENTS</td>
<td>DEV382</td>
</tr>
<tr>
<td>DXRSTB2</td>
<td>0</td>
<td>IP</td>
<td>3271</td>
<td>EXTENTS</td>
<td>DEV102</td>
</tr>
<tr>
<td>BMCRSCSD</td>
<td>0</td>
<td>TP</td>
<td>3374</td>
<td>VOLUME</td>
<td>DEV155</td>
</tr>
<tr>
<td>BMCRSTB</td>
<td>0</td>
<td>TP</td>
<td>3568</td>
<td>EXTENTS</td>
<td>DEV102</td>
</tr>
<tr>
<td>DXUTFILT</td>
<td>0</td>
<td>IP</td>
<td>4073</td>
<td>EXTENTS</td>
<td>DEV140</td>
</tr>
<tr>
<td>DXRSDIX</td>
<td>0</td>
<td>IP</td>
<td>4755</td>
<td>VOLUME</td>
<td>DEV165</td>
</tr>
<tr>
<td>DXWKID</td>
<td>0</td>
<td>IP</td>
<td>4779</td>
<td>EXTENTS</td>
<td>DEV310</td>
</tr>
<tr>
<td>BMCRSTB</td>
<td>0</td>
<td>TP</td>
<td>6588</td>
<td>EXTENTS</td>
<td>DEV150</td>
</tr>
<tr>
<td>BMCRST</td>
<td>0</td>
<td>TP</td>
<td>8947</td>
<td>VOLUME</td>
<td>DEV340</td>
</tr>
</tbody>
</table>

**DASD MANAGER PLUS** -ASUSPCTR-  **DBDA**

**29 Jan 2002  15:22:02**

**Input Parameters**

DBDA ASUB12DR BMCASU81

ssid plan  dbname
Report fields

The Database Space Trend Report contains the following fields:

- **Database** is the name of the database for which the product generates the report.
- **Name** is the index name or the table space name, depending on the object type (Type).
- **Part** is the number of the partition of the table space partition or index space partition.
- **Type** is the kind of DB2 object that is reported: index partition (IP) or table space partition (TP).
- **Days to Full** is the predicted number of days until the DB2 object runs out of space.
- **Reason** is the reason that the object is expected to be full in the number of days shown (Table 88).
- **Volume** is the DASD volume that stores the DB2 object.

**Table 88  Reasons for days to full warning**

<table>
<thead>
<tr>
<th>Reason</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume</td>
<td>The Direct Access Storage Device (DASD) volume is approaching capacity.</td>
</tr>
<tr>
<td>Extents</td>
<td>The volume is approaching the maximum number of extents.</td>
</tr>
<tr>
<td>Max Size</td>
<td>The object is approaching the maximum size that DB2 allows.</td>
</tr>
</tbody>
</table>

Input parameters

The Database Space Trend Report requires the following input parameters:

```
ssid plan   dbname
```

Database storage utilization report

The Database Storage Utilization Report (Figure 264) lists the tracks allocated, tracks used, percentage of space used, number of extents, number of data sets, primary and secondary space allocation, allocation unit, number of volumes, and the name of the first volume for each table space and index of a DB2 database. The report also displays totals by indexes, table spaces, table spaces and indexes, database—and by report if you specify multiple databases.
### Database Storage Utilization Report (part 1 of 2)

<table>
<thead>
<tr>
<th>DATABASE OBJECT</th>
<th>TRKS</th>
<th>TRKS %</th>
<th>PART</th>
<th>SPACE</th>
<th>USED</th>
<th>USED</th>
<th>EXT</th>
<th>DSNS</th>
<th>PQTY</th>
<th>SQTY</th>
<th>U</th>
<th>VOLS</th>
<th>VOLSER</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUD40.QZUS0140</td>
<td>0</td>
<td>315</td>
<td>0</td>
<td>315</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>1</td>
<td>AZU021</td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX01_D40S01T01</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>1</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX01_D40S01T02</td>
<td>0</td>
<td>45</td>
<td>45</td>
<td>100</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>2</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX02_D40S01T01</td>
<td>0</td>
<td>45</td>
<td>45</td>
<td>100</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>1</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX02_D40S01T02</td>
<td>0</td>
<td>45</td>
<td>45</td>
<td>100</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>2</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX03_D40S01T01</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>1</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX03_D40S01T02</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>1</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX04_D40S01T01</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>1</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX04_D40S01T02</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>1</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX05_D40S01T01</td>
<td>0</td>
<td>45</td>
<td>45</td>
<td>100</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>2</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX05_D40S01T02</td>
<td>0</td>
<td>45</td>
<td>45</td>
<td>100</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>2</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX06_D40S01T01</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>T</td>
<td>1</td>
<td>AZU021</td>
<td></td>
</tr>
<tr>
<td>QZU.QZUX06_D40S01T02</td>
<td>0</td>
<td>45</td>
<td>45</td>
<td>100</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>2</td>
<td>AZU021</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

--------------- ---------------
TOTAL - INDEXES 360 357
--------------- ---------------
TOTAL - TABLE SPACE 315 315
--------------- ---------------
TOTAL - TABLE SPACE AND INDEXES 675 672
--------------- ---------------
QZUD40.QZUS0240 | 0 | 15 | 15 | 100 | 1 | 1 | 1 | C | 1 | AZU021 |
QZU.QZUX01_D40S02T01 | 0 | 3 | 2 | 66 | 1 | 1 | 3 | 3 | T | 1 | AZU021 |
QZU.QZUX01_D40S02T02 | 0 | 3 | 2 | 66 | 1 | 1 | 3 | 3 | T | 1 | AZU021 |
--------------- ---------------
TOTAL - INDEXES 6 4
--------------- ---------------
TOTAL - TABLE SPACE 15 15
--------------- ---------------
TOTAL - TABLE SPACE AND INDEXES 21 19
--------------- ---------------
QZUD40.QZUS0340 | 1 | 15 | 15 | 100 | 1 | 1 | 1 | C | 1 | AZU021 |
| 2 | 15 | 15 | 100 | 1 | 1 | 1 | C | 1 | AZU021 |
| 3 | 15 | 15 | 100 | 1 | 1 | 1 | C | 1 | AZU021 |
| 4 | 15 | 15 | 100 | 1 | 1 | 1 | C | 1 | AZU021 |
--------------- ---------------
TOTAL - TABLEPARTS 60 60
--------------- ---------------
QZU.QZUX01_D40S03T01 | 1 | 15 | 15 | 100 | 1 | 1 | 1 | C | 1 | AZU021 |
| 2 | 15 | 15 | 100 | 1 | 1 | 1 | C | 1 | AZU021 |
| 3 | 15 | 15 | 100 | 1 | 1 | 1 | C | 1 | AZU021 |
| 4 | 15 | 15 | 100 | 1 | 1 | 1 | C | 1 | AZU021 |
--------------- ---------------
TOTAL - INDEXPARTS 60 60
Report fields

The Database Storage Utilization Report contains the following fields:

- **Database Object** is the name of the table space or index for which the product generates the report.

- **Part** is the number of the partition in the table space or index.

- **Trks Space** is the number of tracks that are allocated for the object.

- **Trks Used** is the number of tracks that the object uses.

- **% Used** is the percentage of the allocated space that the object uses.

- **Ext** is the number of extents that the object is in.

- **DSNS** is the number of data sets in the object.

- **Pqty** is the primary allocation quantity.

- **Sqty** is the secondary allocation quantity.

- **AU** is the allocation unit (T – tracks or C – cylinders).

- **Vols** is the number of volumes for the object.

- **Volser** is the volume serial number of the first volume for the object.

- **Total – Tableparts** is the total number of tracks allocated and used for the table space partitions.

- **Total – Indexparts** is the total number of tracks allocated and used for the index space partitions.

---

**NOTE**

To change report units to KB, see “Internal KB switch” on page 650.
- **Total – Indexes** is the total number of tracks allocated and used for the indexes.

- **Total – Table Space** is the total number of tracks allocated and used for the table spaces.

- **Total – Table Space and Indexes** is the total number of tracks allocated and used for the table spaces and indexes.

- **Total – Database** is the total number of tracks allocated and used for the database.

**Input parameters**

The Database Storage Utilization Report requires the following input parameters:

```
ssid plan dbname vcatname
```

**Exception report for BMCTRIG action**

The Exception report by BMCTRIG Action (Figure 265) lists exceptions that a BMCTRIG job detected. The report lists the exception name along with the current value and the compare value for each object that is associated with the BMCTRIG action.

**Figure 265 Exception report for BMCTRIG Action**

<table>
<thead>
<tr>
<th>Obj Type</th>
<th>Name</th>
<th>Part</th>
<th>Exception</th>
<th>Current</th>
<th>Compare</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLESPACE</td>
<td>CCBPART.TSPART</td>
<td>1</td>
<td>EXTENTS</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>CCBPART.TSPART</td>
<td>1</td>
<td>SPACE</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>CCBPART.TSPART</td>
<td>3</td>
<td>EXTENTS</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>CCBPART.TSPART</td>
<td>3</td>
<td>SPACE</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>INDEX</td>
<td>CCB.X2PPART</td>
<td>0</td>
<td>EXTENTS</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>INDEX</td>
<td>CCB.X2PPART</td>
<td>0</td>
<td>SPACE</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>INDEX</td>
<td>CCB.XCPPART</td>
<td>1</td>
<td>EXTENTS</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>INDEX</td>
<td>CCB.XCPPART</td>
<td>1</td>
<td>SPACE</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>INDEX</td>
<td>CCB.XCPPART</td>
<td>3</td>
<td>EXTENTS</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>INDEX</td>
<td>CCB.XCPPART</td>
<td>3</td>
<td>SPACE</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

BMCTRIG generates Corrective Action CCBBMCREO
Executed 2006-10-18-09.20.55

<table>
<thead>
<tr>
<th>Seq#</th>
<th>Service Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BMCREORG</td>
</tr>
</tbody>
</table>
**Report fields**

The Exceptions Report for BMCTRIG Action contains the following fields:

- **Obj Type** is the type of object.
- **Name** is the object name.
- **Part** is the partition number.
- **Exception** is the exception name.
- **Current** is the current value.
- **Compare** is the compare value.

**Input parameters**

The Exceptions Report for BMCTRIG Action requires the following input parameters:

- ssid plan action

**Index clustering analysis report**

The Index Clustering Analysis Report (Figure 266) lists, for clustering indexes, the unique rule of the index, the percentage of rows in clustering order, and the number of rows not in optimal position. The product reports only clustering indexes that meet one of the following specifications:

- **CLUSTERRATIO <= CR specified**
- **NEAROFFPOS / CARD * 100 >= NEARPCT specified**
- **FAROFFPOS / CARD * 100 >= FARPCT specified**

**Figure 266  Index Clustering Analysis Report (part 1 of 2)**

<table>
<thead>
<tr>
<th>DB.TS/TABLE/INDEX</th>
<th>PART</th>
<th>CR</th>
<th>DATA REPEAT FACTOR</th>
<th>NEAROFF</th>
<th>FAROFF</th>
<th>CARD</th>
<th>STATSTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUD16.QZUS0116</td>
<td>0</td>
<td>1</td>
<td>787</td>
<td>902</td>
<td>2856</td>
<td>3861</td>
<td>2009-10-01-01.16</td>
</tr>
<tr>
<td>QZU.QZUT01_D16S01</td>
<td>0</td>
<td>1</td>
<td>2561</td>
<td>839</td>
<td>1660</td>
<td>2601</td>
<td>2009-10-01-01.16</td>
</tr>
</tbody>
</table>

**Figure 266  Index Clustering Analysis Report (part 2 of 2)**

<table>
<thead>
<tr>
<th>DB.TS/TABLE/INDEX</th>
<th>PART</th>
<th>CR</th>
<th>DATA REPEAT FACTOR</th>
<th>NEAROFF</th>
<th>FAROFF</th>
<th>CARD</th>
<th>STATSTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUD16.QZUS0216</td>
<td>0</td>
<td>1</td>
<td>902</td>
<td>2856</td>
<td>3861</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.QZUT01_D16S02</td>
<td>0</td>
<td>1</td>
<td>839</td>
<td>1660</td>
<td>2601</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Report fields

The Index Clustering Analysis Report contains the following fields:

- **DB.TS/Table/Index** is the fully qualified table space name, the table name, and the index name that the product is reporting.
- **Part** is the number of the partition in the index.
- **UR** is the unique rule, indicating whether the index is unique. Valid values are as follows:
  - D: duplicates allowed
  - U: unique
  - P: unique and primary index
  - C: unique and used to enforce a unique constraint
  - N: unique and defined with UNIQUE WHERE NOT NULL
  - R: unique and used to enforce the uniqueness of a nonprimary parent key
- **CR** is the clustering ratio, the percentage of rows in clustering order.
- **Data Repeat Factor** is the number of pages read while following an index key order.
- **Nearoff** is the number of referenced rows near, but not in, optimal position because of an insert into a full page.
- **Faroff** is the number of referenced rows far from optimal position because of an insert into a full page.
- **Card** is the number of columns in the referenced table.
- **Statstime** is the date and time when the statistics were taken.

Input parameters

The Index Clustering Report requires the following input parameters:

```
ssid plan   dbname cr nearpct farpct
```
Modified table space pages report

The Modified Table Space Pages Report (Figure 267) lists the number of rows in a table space, the number of active pages, the number of pages modified since the last copy, the percentage of modified pages, and the recommended type of copy to make. The report is sorted by percentage dirty in descending order and prints a line for objects whose percentage dirty is equal to or greater than the \textit{pctdirty} that the input parameters specify.

\textbf{Figure 267  Modified Table Space Pages Report}

\begin{tabular}{|l|l|l|l|l|l|l|l|l|}
\hline
DBNAME & TSNAME & PART & CARD & NACTIVE & DIRTY & DIRTY & COPY & STATTIME \\
\hline\hline
QZUD21 & QZUS0121 & 3 & 3243216 & 73106 & 72790 & 100 & FULL & 2002-03-05-17.02 \\
QZUD21 & QZUS0121 & 4 & 1580688 & 35710 & 35437 & 100 & FULL & 2002-03-05-17.02 \\
QZUD64 & QZUS0264 & 0 & 3854 & 3854 & 3854 & 100 & FULL & 2002-02-24-10.49 \\
QZUD65 & QZUS0165 & 1 & 99448 & 2484 & 2474 & 100 & FULL & 2002-02-24-10.51 \\
QZUD70 & QZUS0170 & 0 & 239232 & 7488 & 7474 & 100 & FULL & 2002-01-22-11.03 \\
QZUD21 & QZUS0121 & 1 & 490248 & 11007 & 10942 & 99 & FULL & 2002-03-05-17.02 \\
QZUD21 & QZUS0121 & 6 & 1907352 & 42893 & 42562 & 99 & FULL & 2002-03-05-17.02 \\
QZUD21 & QZUS0121 & 7 & 2022248 & 45293 & 44862 & 99 & FULL & 2002-03-05-17.02 \\
\hline
\end{tabular}

Input Parameters

\begin{tabular}{l}
DBDA ASU812DR QZUD% 10 \\
ssid plan dbname pctdirty
\end{tabular}

\section*{Report fields}

The Modified Table Space Pages Report contains the following fields:

- \textbf{DBname/TSname} is the fully qualified table space name.
- \textbf{Part} is the number of the partition in the table space.
- \textbf{Card} is the cardinality, or number of rows, in the table space.
- \textbf{Nactive} is the number of active pages in the table space.

The product considers a page that has row format active even if it currently contains no rows. The value is 0 if you have not gathered statistics.
Dirty is the number of modified, or dirty, pages in the table space.

%Dirty is the percentage of modified, or dirty, pages in the table space.

Copy is the recommended type of image copy: full or incremental.

Statstime is the date and time when the product collected statistics.

**Input parameters**

The Modified Table Space Pages Report requires the following input parameters:

```
ssid plan  dbname  pctdirty
```

---

**Space estimation report**

The Space Estimation Report (Figure 268) estimates the reorganized space of a DB2 object based on the specified percentage change in the object’s cardinality and the specified percentage free and number of free pages. The report lets you try “what-if” scenarios by varying the percentage specified for free space and free pages.

**NOTE**

Space estimation is not available for XML objects or for objects that have partial statistics.

---

**Figure 268  Space Estimation Report (part 1 of 2)**
The Space Estimation Report contains the following fields:

- **Database Object** is the fully qualified DB2 object name.
- **Part** is the partition number.
- **Pqty** is the primary allocation quantity.
- **Sqty** is the secondary allocation quantity.
- **AU** is the allocation unit (T – tracks or C – cylinders).
- **% Free** is the percentage of the allocated space that is free.
- **Free Page** is the number of free pages.
- **Computed Card** is the cardinality of the object, taken from the historical database, with the **pctcard** percentage applied.
- **KB** is the estimated number of kilobytes after the reorganization.
- **Trks** is the estimated number of tracks after the reorganization.
- **Cyls** is the estimated number of cylinders after the reorganization.
Ext is the estimated number of extents after the reorganization.

DSN is the estimated number of data sets after the reorganization.

**Input parameters**

The Space Estimation Report requires the following input parameters:

```
ssid plan qualtype quainame pctcard tspf tsfpg ixpf ixfpg
```

The parameters `tspf`, `tsfpg`, `ixpf`, and `ixfpg` are optional. You can leave them all blank, set each to a specific value, or set them to NC (No Change from data in the BMCSTATS database).

**Space estimation trend report**

The Space Estimation Trend Report (Figure 269) uses linear regression of BMCSTATS historical data to estimate the cardinality of an object and then computes the reorganized space of the object.

---

**NOTE**

Space estimation is not available for XML objects or for objects that have partial statistics.

---

**Figure 269  Space Estimation Trend Report**
Report fields

The Space Estimation Trend Report contains the following fields:

- **Database Object** is the fully qualified table space or index name.
- **Part** is the partition number.
- **Pqty** is the primary allocation quantity.
- **Sqty** is the secondary allocation quantity.
- **AU** is the allocation unit (T – tracks or C – cylinders).
- **% Free** is the percentage of the allocated space that is free.
- **Free Page** is the number of free pages.
- **Change in Card per Day** is the average increase or decrease in the number of rows per day over the specified number of days.
- **Projected Card** is the estimated number of rows in the object after the specified number of days, based on the change in cardinality per day.
- **Reorg Space nnn Days from report-date** is the projected statistics for the object if a reorganization is performed on that specified date.
  - **KB** is the estimated number of kilobytes.
  - **Trks** is the estimated number of tracks.
  - **Cyls** is the estimated number of cylinders.
  - **Ext** is the estimated number of extents.
  - **DSN** is the estimated number of data sets.

Input parameters

The Space Estimation Trend Report requires the following input parameters:

<table>
<thead>
<tr>
<th>ssid</th>
<th>plan</th>
<th>qualtype</th>
<th>qualname</th>
<th>observations</th>
<th>days</th>
</tr>
</thead>
</table>

---

DASD MANAGER PLUS for DB2 User Guide
Space utilization by table report

The Space Utilization by Table Report (Figure 270) lists the number of pages containing table data, the percentage of pages that contain rows, and the table cardinality for the specified DB2 objects. If you do not specify a table space name, the product retrieves all table spaces.

**Figure 270  Space Utilization by Table Report**

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSPACE</th>
<th>TYPE</th>
<th>TABLE NAME</th>
<th>NPAGES</th>
<th>PCTPGS</th>
<th>CARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUDAC</td>
<td>QZUS01AC</td>
<td>QZUT01_DACS01</td>
<td>21669</td>
<td>93</td>
<td>65000</td>
<td></td>
</tr>
<tr>
<td>QZUS02AC</td>
<td>QZUT01_DACS02</td>
<td>21669</td>
<td>93</td>
<td>65000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUS03AC</td>
<td>GQZUT01_DACS03</td>
<td>32484</td>
<td>100</td>
<td>65000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUS04AC</td>
<td>QZUT01_DACS04</td>
<td>21667</td>
<td>99</td>
<td>65000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUS05AC</td>
<td>QZUT01_DACS05</td>
<td>21667</td>
<td>99</td>
<td>65000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUS06AC</td>
<td>QZUT01_DACS06</td>
<td>21667</td>
<td>99</td>
<td>65000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUS07AC</td>
<td>QZUT01_DACS07</td>
<td>21667</td>
<td>99</td>
<td>65000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUS08AC</td>
<td>QZUT01_DACS08</td>
<td>21667</td>
<td>99</td>
<td>65000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUS09AC</td>
<td>GQZUT01_DACS09</td>
<td>335818</td>
<td>99</td>
<td>975000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>QZUT01_DAS01</td>
<td>128</td>
<td>100</td>
<td>2036</td>
<td></td>
</tr>
<tr>
<td>QZUS02A1</td>
<td>QZUT01_DAS02</td>
<td>128</td>
<td>100</td>
<td>2036</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUT02_DAS02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUS03A1</td>
<td>QZUT01_DAS03</td>
<td>129</td>
<td>8</td>
<td>2036</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUS04A1</td>
<td>QZUT01_DAS04</td>
<td>79</td>
<td>61</td>
<td>2007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUT02_DAS04</td>
<td>167</td>
<td>86</td>
<td>5003</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Report fields**

The Space Utilization by Table Report contains the following fields:

- **DBname/TSpace** is the fully qualified name of the table space.
- **Type** is the type of table space.
- **Table Name** is the name of the table for which statistics appear.
- **NPages** is the number of pages that contain rows of this table.
- **PctPgs** is the percentage of table space pages that contain rows from this table.
- **Card** is the cardinality, or number of rows, in the table.

**Input parameters**

The Space Utilization by Table Report requires the following input parameters:

```
   ssid plan   dbname tsname
```
The Table Space and Index Storage Limits Report (Figure 271) indicates the percentage of the physical object limit that DB2 table spaces and indexes have used that is equal to or greater than the limit specified. The report prints a line for objects whose %GB is greater than or equal to the \textit{pctlimit}. The report sorts the objects by the largest percentage of possible space used.

Figure 271 Table Space and Index Storage Limits Report

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSNAME</th>
<th>PART</th>
<th>TYPE</th>
<th>USED SPACE</th>
<th>GB</th>
<th>% GB</th>
<th>ROWS</th>
<th>NEW ROWS</th>
<th>DATE STATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUDXM</td>
<td>QZUS29XM</td>
<td>&lt;= 4</td>
<td>G</td>
<td>720</td>
<td>64</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDXM</td>
<td>QZUS26XM</td>
<td>&lt;=256</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDXM</td>
<td>QZUS33XM</td>
<td>&lt;= 4</td>
<td>G</td>
<td>400508</td>
<td>1</td>
<td>38.20</td>
<td>50050</td>
<td>80987</td>
<td>12.17.2010</td>
</tr>
</tbody>
</table>

Input Parameters

DEDK ASUA1DDR QZUDXM 10
ssid plan dbname pctlimit

Report fields

The Table Space and Index Storage Limits Report contains the following fields:

- **DBName/TSName** is the fully qualified table space name.
- **DBName/Creater/IXname** is the fully qualified index name.
- **Part** is the partition number.
- **Type** is the type of table space.
- **Used Space (KB)** is the space that the DB2 object currently uses, in kilobytes.
- **Max GB** is the maximum amount of space to be allocated for the object in gigabytes.
- **% GB** is the percentage of the maximum space that the object currently uses.
- **Rows** is the number of rows in the object.
- **Estimated New Rows** is the estimated number of rows still available for the object.
- **Date Stats** is the date that you last ran BMCSTATS on the object.
**Input parameters**

The Table Space and Index Storage Limits Report requires the following input parameters:

<table>
<thead>
<tr>
<th>ssid</th>
<th>plan</th>
<th>dbname</th>
<th>pctlimit</th>
</tr>
</thead>
</table>

**Tables with more than average growth report**

The Tables with More than Average Growth Report identifies DB2 tables that are growing more than the average based on the last two collections of statistics. The report orders the tables that are growing more than average by descending growth rate and shows the age of the last statistics. If none of the tables had any growth, the report program issues the message *Nothing to report.*

**Figure 272   Tables with More than Average Growth Report**

<table>
<thead>
<tr>
<th>TABLES WITH MORE THAN AVERAGE GROWTH REPORT</th>
<th>PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS -ASUNPGR- DBDA</td>
<td>22 Aug 2005 15:15:16</td>
</tr>
</tbody>
</table>

AVERAGE GROWTH (PAGES/DAY): 70

<table>
<thead>
<tr>
<th>CREATOR</th>
<th>NAME</th>
<th>GROWTH</th>
<th>OBSERVATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASU81D</td>
<td>V81_RS_COLDISTSTAT</td>
<td>440</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>V81_RS_COLSTATS</td>
<td>229</td>
<td>14</td>
</tr>
</tbody>
</table>

**Input Parameters**

DBDA ASU812DR ASU81D

**Report fields**

The Tables with More than Average Growth Report contains the following fields:

- **Average Growth** is the average increase in the size of the tables per day, expressed in number of pages/day.

- **Table Creator/Name** is the fully qualified table name.
The report lists only the tables that have grown more than the value of the **Average Growth** field.

- **Growth (Pages/Day)** is the average number of pages per day that have been added to the table as determined by the last two instances of captured statistics.
- **Observation Age in Days** is the number of days since the last statistics collection. A 0 age indicates that you collected statistics on the current date.

**Input parameters**

The Tables with More than Average Growth Report requires the following input parameters:

```
ssid plan  table-creator
```

### Table space and index extents report

The Table Space and Index Extents Report (Figure 273) lists the allocated space, used space, and number of extents for DB2 table spaces and index spaces that equal or exceed the specified number of extents. The report sorts objects by percentage of space limit in descending order.

**Figure 273 Table Space and Index Extents Report (part 1 of 2)**

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSNAME</th>
<th>PART</th>
<th>PQTY</th>
<th>SQTY</th>
<th>U</th>
<th>SPACE</th>
<th>EXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUD66</td>
<td>QZUS0166</td>
<td>1</td>
<td>252</td>
<td>3</td>
<td>T</td>
<td>564</td>
<td>105</td>
</tr>
<tr>
<td>QZUDB6</td>
<td>QZUS0166</td>
<td>42</td>
<td>164</td>
<td>2</td>
<td>T</td>
<td>350</td>
<td>94</td>
</tr>
<tr>
<td>QZUDB6</td>
<td>QZUS02B6</td>
<td>4</td>
<td>54</td>
<td>6</td>
<td>T</td>
<td>552</td>
<td>84</td>
</tr>
<tr>
<td>QZUDB6</td>
<td>QZUS02B6</td>
<td>4</td>
<td>54</td>
<td>6</td>
<td>T</td>
<td>516</td>
<td>78</td>
</tr>
<tr>
<td>QZUDL5</td>
<td>QZUS02L5</td>
<td>4</td>
<td>54</td>
<td>6</td>
<td>T</td>
<td>516</td>
<td>78</td>
</tr>
<tr>
<td>QZUDL65</td>
<td>QZUS0166</td>
<td>11</td>
<td>396</td>
<td>4</td>
<td>T</td>
<td>672</td>
<td>70</td>
</tr>
<tr>
<td>QZUDL5</td>
<td>QZUS01L5</td>
<td>2</td>
<td>54</td>
<td>6</td>
<td>T</td>
<td>426</td>
<td>63</td>
</tr>
<tr>
<td>QZUDL5</td>
<td>QZUS02L5</td>
<td>2</td>
<td>54</td>
<td>6</td>
<td>T</td>
<td>426</td>
<td>63</td>
</tr>
<tr>
<td>QZUDL5</td>
<td>QZUS02L5</td>
<td>1</td>
<td>27</td>
<td>3</td>
<td>T</td>
<td>207</td>
<td>61</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>CREATOR</th>
<th>IXNAME</th>
<th>PART</th>
<th>PQTY</th>
<th>SQTY</th>
<th>A</th>
<th>U</th>
<th>SPACE</th>
<th>EXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUD42</td>
<td>QZU</td>
<td>QZUX04_D42S01T01</td>
<td>0</td>
<td>76</td>
<td>1</td>
<td>T</td>
<td>194</td>
<td>119</td>
<td></td>
</tr>
<tr>
<td>QZUD70</td>
<td>QZU</td>
<td>QZUX01_D70S01T01</td>
<td>0</td>
<td>72</td>
<td>3</td>
<td>T</td>
<td>330</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>QZUDL3</td>
<td>QZU</td>
<td>QZUX06_DL3S01T01</td>
<td>0</td>
<td>13</td>
<td>3</td>
<td>C</td>
<td>3300</td>
<td>70</td>
<td></td>
</tr>
</tbody>
</table>
Report fields

The Table Space and Index Extents Report contains the following fields:

- **DBName/TSName** is the fully qualified name of the table space.
- **Part** is the partition number.
- **Pqty** is the primary allocation quantity.
- **Sqty** is the secondary allocation quantity.
- **AU** is the allocation unit (T – tracks or C – cylinders).
- **Space** is the allocated space in tracks or cylinders.
- **Ext** is the number of extents for the object.
- **DBName/Creator/IXName** is the fully qualified index name.

Input parameters

The Table Space and Index Extents Report requires the following parameters:

<table>
<thead>
<tr>
<th>ssid</th>
<th>plan</th>
<th>dbname</th>
<th>extents</th>
</tr>
</thead>
</table>

Volume space trend report

The Volume Space Trend Report (Figure 274) uses a linear regression of historical data to predict the number of days until DASD volumes run out of space. The report also indicates the number of free tracks per volume.

Figure 274  Volume Space Trend Report (part 1 of 2)
Report fields

The Volume Space Trend Report contains the following fields:

- **Volume** is the volume serial number of the DASD volume.
- **Days to Full** is the estimated number of days until the volume is full.
- **Free Tracks** is the number of free tracks on the volume.

Input parameters

The Volume Space Trend Report requires the following parameters:

- `ssid plan volume`
Chapter 11 Exporting and deploying product definitions

This chapter contains the following topics:

Overview of exporting object definitions ........................................ 686
   How the Export utility works ....................................................... 686
   Recommendations for setting up connections .............................. 690
   Enterprise list and personal list of connections ......................... 690
   Required authorizations for using Export .................................. 691
   Task summary for exporting object definitions ........................... 693
Preparing your environment for exporting ...................................... 694
Accessing the Export utility .......................................................... 695
Setting up connections .................................................................... 696
   Setting up enterprise connections .............................................. 696
   Setting up destination connections ............................................ 698
Specifying definitions to export ...................................................... 700
   Specifying definitions to export ................................................. 701
   Specifying action definitions ..................................................... 701
   Specifying service definitions ................................................... 702
   Specifying object set definitions ................................................. 703
   Specifying object-action priority definitions ............................ 703
   Specifying exceptions definitions .............................................. 704
Verifying the list of definitions to export ....................................... 706
Selecting the destinations for an export ......................................... 706
Deploying the definitions .............................................................. 707
Reviewing the deployment report ................................................... 708
Overview of exporting object definitions

NOTE
The Export utility is available only to those who have a license and password for Database Performance for DB2 solution.

With the Export utility, you can ensure that your DASD MANAGER PLUS object definitions match on all DB2 subsystems where DASD MANAGER PLUS resides. You can copy definitions from a local “controlling” DASD MANAGER PLUS repository to “destination” DASD MANAGER PLUS repositories on other DB2 subsystems. The subsystems can reside in the same sysplex or across sysplexes.

You can use Export to add or update definitions on the destination subsystems, but not to delete definitions. (For deletions, you must manually delete the definitions by logging into IBM z/OS images and accessing DASD MANAGER PLUS.)

How the Export utility works

When deploying definitions, Export analyzes what type of request to generate (Add or Update) and what definitions to include in the export. A definition is included in the export either explicitly or implicitly.

- A definition is included explicitly when it is selected by the user.
- A definition is included implicitly when it is dependent on a definition that is explicitly selected. Implicitly included objects are referred to as dependent objects.

The following example illustrates a simple export.

EXAMPLE
If you explicitly select object Sets A, B, and C to be exported, Export adds object Sets A, B, and C to the destination so that both the source and destination definitions match. If the definitions already exist on the destination, Export overwrites object Sets A, B, and C with the definitions from the source (if you chose Overwrite existing rows on the Export Deployment panel).
Export lists dependent objects and their parent objects as follows:

- services
  - service syntax
- exception definitions
  - thresholds
  - corrective actions

Export also validates remote object definitions that a selected definition references (called referenced objects). For example, when exporting an action, Export validates that the referenced service is being exported (because you selected it), or the referenced service already exists on the destination.

Table 89 lists the types of definitions and how Export analyzes them.

<table>
<thead>
<tr>
<th>Exported definition</th>
<th>Dependent objects included implicitly</th>
<th>Referenced objects validated by Export</th>
<th>Referenced objects not validated by Export(^a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Service</td>
<td>Service syntax</td>
<td>Object sets</td>
</tr>
<tr>
<td>Corrective action</td>
<td>Exception definition</td>
<td>Corrective action</td>
<td>Object sets</td>
</tr>
<tr>
<td>Exception definition</td>
<td>Corrective action</td>
<td>Action</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Threshold</td>
<td>Object sets</td>
<td></td>
</tr>
<tr>
<td>Object-action priorities</td>
<td>Exception Threshold</td>
<td>Object sets</td>
<td>Action</td>
</tr>
<tr>
<td>Object sets</td>
<td>Service</td>
<td>Service syntax</td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exception thresholds</td>
<td></td>
<td>Exception definition</td>
<td>Object sets</td>
</tr>
</tbody>
</table>

\(^a\) Export cannot validate these relationships because you can specify a wildcard for the object sets or action definitions.
Examples with referenced objects

Actions, object sets, and Exception definitions might reference other object types that you must include in the export unless the referenced objects already exist on the destination. Following are some examples of how Export processes referenced objects.

Example of exporting a definition that has referenced objects

If you export an action that uses services and their associated service syntax definitions, you must explicitly specify their definitions for export, too (unless they already exist on the destination). Otherwise, you will receive an error.

--- EXAMPLE ---
You explicitly select action A for export. Action A uses service A, and syntax A and B; their definitions do not exist on the destination. Thus, you must also select service A for export. (By default, Export automatically includes syntax A and B in the export because service A was explicitly included and service syntax is considered to be a dependent of service.)

Export adds action A, service A, syntax A, and syntax B to the destination so that the source and destination definitions match.

Example of exporting a definition that has referenced objects that are not referentially related to the definition

If all of the following conditions are true, you must explicitly specify the referenced definition in the export:

- You are exporting an object that references another definition.
- The object is not referentially related to that definition.
- The referenced definition does not already exist on the destination.

If you do not specify the referenced definition for export in this case, you will not receive an error. However, the jobs on the destination system will not operate as expected. This issue occurs with object actions, object sets, thresholds, and corrective actions that reference object sets and actions whose names can use wildcards.
How the Export utility works

Chapter 11 Exporting and deploying product definitions

Examples with dependent objects

When an object that you explicitly specify for export has dependent objects (such as a service and its service syntax), Export implicitly includes the dependent objects in the export. A simple example follows:

EXAMPLE

You explicitly specify Threshold A for export. Threshold A has these characteristics:

- is set to a value other than the default
- references an object set
- uses a wildcard reference to all table spaces
- uses a default threshold for these table spaces

The definitions for these thresholds and objects do not exist on the destination. To ensure that threshold A and its referenced object set is set to the value specified, you must export the object set for threshold A as well. Otherwise, all objects will receive the default threshold. If you do not export the object set, you will not receive an error, but you will encounter unexpected results.

EXAMPLE

Assume that you explicitly select service A for export, and service A exists on the source with Syntax A and B. Export includes syntax A and B in the export. If the service already exists on the destination, Export overwrites service A with the definitions from the source (if you chose Overwrite existing rows on the Export Deployment panel).

Definitions on the source can get out of sync with the definitions on the destination. For example, you might delete a syntax on the source. To keep the destination and source definitions in sync, you could explicitly export that service. In that case, Export would delete the additional syntax definition from the destination, and update the service with the dependent syntax definitions.

EXAMPLE

Assume that you explicitly select service A for export, and service A exists on the source with syntax A and B. Export includes syntax A and B definitions in the export. If service A exists with syntax A, B, and C on the destination, Export deletes the syntax C definition from the destination, and updates service A with syntax A and B. The source and destination definitions now match.
Recommendations for setting up connections

BMC recommends that you use one DB2 subsystem as your control or master subsystem. You can copy (export) all definitions from the control subsystem.

BMC also recommends using a primary User Interface Middleware (UIM) Server as the connection repository. The UIM Server is a TCP/IP application that facilitates communication between logical partitions (LPARs), which can span sysplexes. The UIM Server provides the data transport mechanism between the source and destination.

Following these recommendations helps you avoid accidentally overwriting object definitions and connection information.

**NOTE**
To delete definitions from multiple data sources, you must manually delete the definitions from each data source.

Enterprise list and personal list of connections

When you launch Export, you must define at least one host connection. After you define a host connection, you can add and work with DB2 data sources. When you define a host connection, the connection definition remains available each time that you start Export and log in.

Host connections for personal use are managed separately from host connections for the entire enterprise. This separation makes it easier to isolate activities in different environments (such as testing systems versus production systems or different groups of application systems).

Export supports a shared list called an enterprise connection list (ECL) which is used to identify the host connections that you define. The ECL is maintained by one or more administrators and resides on the UIM Server. It contains host definitions and port numbers of one or more UIM Servers. If you have the appropriate security authority, you can add, delete, and edit connection information in the ECL.

All destinations are obtained from your personal connection list (PCL). You can define a connection in your personal list by entering connection information (such as the host name and port number). Also, if a connection has been predefined in the shared ECL, you can add that connection by selecting it from the shared list. After you define a host connection in your personal list, that connection definition remains available each time you log onto Export.
Required authorizations for using Export

Before launching Export, you need to provide the appropriate Resource Access Control Facility (RACF®) authority and System Authorization Facility (SAF) authority to access the ECL.

TCP/IP and UIM Server access

Export uses existing login credentials for the definition phase. Export also prompts you for login credentials when you specify a UIM connection for the primary UIM Server, and for any other UIM Servers that will participate in an export. Export requires a valid RACF or equivalent user ID and password for these credentials. The security administrator for your site sets up the user ID and password.

The RACF security administrator must define an Open Multiple Virtual Storage (OMVS) segment for the UIM Server started task in order to enable TCP/IP access. The security administrator must also assign a user ID with an OMVS segment to the started task procedure name for the UIM Server address space.

The UIM Primary Server is the connection repository where PCLs and ECLs are stored. Communication to the primary UIM Server is through the use of POF values for host name and port number. The security administrator usually specifies this information during installation.

When you launch Export, the JCL Generation component of DASD MANAGER PLUS accesses the POF to retrieve the primary UIM host name and port number. Export then prompts you for a TSO user ID and password and creates a UIM connection using the host and port number specified in the POF.

NOTE
Typically, the security administrator sets the primary server value in the ASU_XP_UIMSRVHOST option before you launch Export. If that option was not set, Export uses the current system where you are logged in as the primary UIM server.

Table 90 lists the POF keywords that are associated with Export.

Table 90  POF keywords for specifying the UIM host definition (part 1 of 2)

<table>
<thead>
<tr>
<th>POF keyword</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASU_XP_LOGD_DATAC=</td>
<td>specifies the SMS data class and the allocation attributes of the Export log file</td>
</tr>
<tr>
<td>ASU_XP_LOGD_MGMTC=</td>
<td>specifies the SMS management class that defines the migration, retention, and backup requirements of the Export log file</td>
</tr>
<tr>
<td>ASU_XP_LOGD_PRIQTY=10</td>
<td>defines the primary allocation for the Export log file</td>
</tr>
</tbody>
</table>
Required authorizations for using Export

TCP/IP and user access

An OMVS segment must be defined in RACF for each Export user ID. An OMVS segment is required to make use of TCP/IP services such as the FTP server on z/OS. The OMVS segment specifies the UIM to be used, the home directory, and the shell program name.

SAF authority to access the ECL

Export uses a shared ECL that resides on the UIM Server. Users who have the appropriate authority can modify information in the ECL. The security administrator sets the authority level (Table 91) that limits your ability to access and edit these connections.

Table 90  POF keywords for specifying the UIM host definition (part 2 of 2)

<table>
<thead>
<tr>
<th>POF keyword</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASU_XP_LOGD_SECQTY=2</td>
<td>defines the secondary allocation for the Export log file</td>
</tr>
<tr>
<td>ASU_XP_LOGD_STORC=</td>
<td>specifies the SMS storage class that defines the processing requirements of the Export log file</td>
</tr>
<tr>
<td>ASU_XP_LOGD_UNIT=SYSDA</td>
<td>specifies the unit for the Export log file</td>
</tr>
<tr>
<td>ASU_XP_LOGDSN=&amp;PREFIX..XPORT.LOG(R)</td>
<td>specifies the Export log file</td>
</tr>
<tr>
<td>ASU_XP_UIMSRVPORT=</td>
<td>specifies the port number of the primary UIM server that contains the host definitions repository for the Export utility</td>
</tr>
<tr>
<td>ASU_XP_UIMSRVHOST=</td>
<td>specifies the host name of the primary UIM server for the Export utility</td>
</tr>
<tr>
<td>ASU_XP_UIMSRVTIMEOUT=300</td>
<td>specifies the UIM timeout parameter that determines how long the Export utility should wait for a response from the UIM server before timing out</td>
</tr>
</tbody>
</table>

Table 91  Authorization to edit the ECL

<table>
<thead>
<tr>
<th>Authority level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>You must set up your own connections and cannot view or edit the ECL.</td>
</tr>
<tr>
<td>READ</td>
<td>You can view and select from the ECL.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>You have full access to the ECL and can view, edit, delete, and add connection definitions.</td>
</tr>
</tbody>
</table>
In addition, the SAF provides an interface to your security product, such as Computer Associates ACF, eTrust CA ACF2 Security for DB2, or eTrust CA Top Secret for DB2. Using security rules, SAF determines who can access z/OS resources, and what type of access approved users have. Through SAF, you can define who can read or maintain the ECL based on

- user ID
- product function or feature

### Task summary for exporting object definitions

Table 92 summarizes the tasks that you must complete to export new and updated definitions (services, service syntax, actions, objects sets, exceptions, thresholds, and corrective actions) to other DB2 connections.

<table>
<thead>
<tr>
<th>Task</th>
<th>Subtasks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare your environment for exporting</td>
<td>Review the task list</td>
<td>“Task summary for exporting object definitions” on page 693</td>
</tr>
<tr>
<td></td>
<td>Review the tasks for preparing your environment</td>
<td>“Preparing your environment for exporting” on page 694</td>
</tr>
<tr>
<td>Access Export</td>
<td>Access the Export utility.</td>
<td>“Accessing the Export utility” on page 695</td>
</tr>
<tr>
<td>Specify and manage connections</td>
<td>Specify and manage the ECL (enterprise definitions) and the PCL (destination definitions)</td>
<td>“Setting up connections” on page 696</td>
</tr>
<tr>
<td>Select definitions to export</td>
<td>Select actions to export</td>
<td>“Specifying action definitions” on page 701</td>
</tr>
<tr>
<td></td>
<td>Select services to export</td>
<td>“Specifying service definitions” on page 702</td>
</tr>
<tr>
<td></td>
<td>Select object sets to export</td>
<td>“Specifying object set definitions” on page 703</td>
</tr>
<tr>
<td></td>
<td>Select object-action priorities</td>
<td>“Specifying object-action priority definitions” on page 703</td>
</tr>
<tr>
<td></td>
<td>Select exceptions to export</td>
<td>“Specifying exceptions definitions” on page 704</td>
</tr>
<tr>
<td>Verify the Export list</td>
<td>Review a summary of definitions to export</td>
<td>“Verifying the list of definitions to export” on page 706</td>
</tr>
<tr>
<td>Select the destination</td>
<td>Select the destination from the PCL that is stored in a UIM file on the mainframe.</td>
<td>“Selecting the destinations for an export” on page 706</td>
</tr>
</tbody>
</table>
Preparing your environment for exporting

Use this task to set up your environment to accommodate exporting definitions.

Before you begin

- Ensure that you have DASD MANAGER PLUS installed on each LPAR and configured for each DB2 subsystem that will participate in the export.

- Verify with the security administrator that TCP/IP access is enabled.

If you need more information about the UIM Server (such as starting and stopping the server, viewing active users, and refreshing the content), see Appendix D, “Using the UIM Server.”

To prepare your environment for exporting

1 Verify that the UIM Server that contains the connection repository (as well as each z/OS image that will participate in the export) is running by checking the JESMSGGLG SYSOUT file for the following messages:

```
BMC3402901 UIM Server, Level v.r.mm mm dd,yy, initialization complete!
BMC340122I Ready for MVS Operator Commands
```

2 (optional) If you security administrator did not specify a primary UIM Server during installation, designate the primary UIM Server now by using the following POF keywords to specify the host name and port number:

- ASU_XP_UIMSRVPORT
- ASU_XP_UIMSRVHOST

3 (optional) Limit update access to the ECL (which contains all the connection information for the enterprise) by specifying the following definition:

```
BBM.SDBA.DNA.ECL
```
Use this profile name with the RESOURCE CLASS of FACILITY to maintain users who can control the ECL.

## Accessing the Export utility

When you launch Export, you must define at least one host connection. Use this procedure to access Export.

### NOTE

To use the Export utility from DASD MANAGER PLUS, you must have a license and password for Database Performance for DB2 solution.

### Before you begin

Ensure that you have completed all tasks in “Preparing your environment for exporting” on page 694.

### To access Export

1. On the DASD MANAGER PLUS main menu, choose Export Definitions and press Enter.

2. Complete the Export Login panel.

   A. Specify the login information including your user ID, password, and group name to log into the UIM primary server to access your connection repository.

   B. (optional) To use these credentials for all logins throughout your session, choose Select to use these credentials for all logins at the bottom of the panel.

   C. Press Enter.

The required steps are automatically selected as indicated by the letter S on the panel. After you press Enter, additional dialogs for the selected steps are displayed in sequential order. When you complete a step, the S is replaced by an asterisk.

The Export Definitions dialog (the main Export dialog) is displayed. You can perform any of the following actions:

- Set up connection information and provide access to the ECL (as described in “Setting up connections” on page 696).
Setting up connections

- Select definitions to be exported (as described in “Specifying definitions to export” on page 700).
- Select destinations for the exported definitions (as described in “Selecting the destinations for an export” on page 706).
- Execute the request to deploy definitions (as described in “Deploying the definitions” on page 707).

Setting up connections

The Manage destinations panels help you set up and manage connection information for your export sessions. From this panel you can perform the following actions:

- set up and manage enterprise connections (ECL)
- set up and manage destination connections (PCL)

Setting up enterprise connections

Use this procedure to set up and manage enterprise connections. Enterprise connections (ECL) comprise the following information:

- UIM host name
- UIM server port number

DASD MANAGER PLUS needs the UIM host name and port number to contact the UIM server that is running on the destination LPAR.

To set up enterprise connections

1. From the Export Definitions main menu, choose Manage destinations and press Enter.

**NOTE**

The following panel is presented only if you have READ authority. To define, edit, or remove a connection, you must have UPDATE authority. If you do not have UPDATE or READ authority, you will only be able to manage destination definitions.
2 From the Manage List Type menu, type 2 to choose Manage enterprise definitions and press Enter.

The Enterprise Destinations List panel lists the available UIM shared connections. From this panel, you can select one or more connections to add to your PCL.

3 From the Enterprise Destinations List panel, perform one of the following actions to update connection information in your ECL:

<table>
<thead>
<tr>
<th>If you enter this in the Act column</th>
<th>Complete this action</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Note: You must have UPDATE authority to see and use this option.</td>
</tr>
<tr>
<td></td>
<td>1. From the Define Connection panel, add a host name, port number, display name, and description to the ECL and press Enter.</td>
</tr>
<tr>
<td></td>
<td>2. Press END to save the connection information.</td>
</tr>
<tr>
<td>E</td>
<td>Note: You must have UPDATE authority to see and use this option.</td>
</tr>
<tr>
<td></td>
<td>1. From the Edit Connection panel, change the display name and description.</td>
</tr>
<tr>
<td></td>
<td>2. Press END to save the updated connection information.</td>
</tr>
<tr>
<td></td>
<td>Note: To edit connection information such as the host and port number, specify I to insert the new definition and D to remove the definition from the Enterprise Destinations List.</td>
</tr>
<tr>
<td>D</td>
<td>Note: You must have UPDATE authority to see and use this option.</td>
</tr>
<tr>
<td></td>
<td>1. From the Delete Confirmation panel, confirm the deletion of the host by specifying option I.</td>
</tr>
<tr>
<td></td>
<td>2. Press Enter.</td>
</tr>
<tr>
<td>S</td>
<td>When you select a host from the Enterprise Definitions List and press Enter, Export displays the following messages in the upper right of the panel to indicate whether the definition was added to your Destinations List or whether it already exists in your Destinations List:</td>
</tr>
<tr>
<td></td>
<td>- Insert successful</td>
</tr>
<tr>
<td></td>
<td>- Insert/Update failed</td>
</tr>
</tbody>
</table>
Setting up destination connections

Use this procedure to set up and manage destination connections. Destination connections (PCLs) comprise the following information:

- UIM host name
- UIM port number
- DB2 subsystem identifier (DB2 SSID)

DASD MANAGER PLUS needs the UIM host name and port number to contact the UIM server that is running on the destination LPAR. The DB2 SSID is the destination DB2 subsystem that runs on the LPAR that receives the export request.

To set up destination connections

1. From the Export Definitions main menu, choose Manage destinations and press Enter.

2. From the Manage List Type menu, type 1 to choose Manage destination definitions and press Enter.

3. (If no previous connections exist) From the Define Connection panel, add a host name, port number, display name, and description to the ECL and press END to save the connection information.

The Destination List panel lists the available UIM shared connections. From this panel, you can select one or more connections to add to your PCL.
From the Destination List panel, perform one of the following actions:

<table>
<thead>
<tr>
<th>If you enter this in the Act column</th>
<th>Location</th>
<th>Complete this action</th>
</tr>
</thead>
</table>
| I                                    | Host name| 1. From the Add Host Definition Method panel, choose one of the following options:  
   - Choose 1 to select an available connection from the Enterprise List.  
   - Choose 2 to enter host information manually.  
   2. If you chose 1 in Step 1, from the Enterprise Destinations List, select a destination and press Enter. Then, press END to save the connection information.  
   3. If you chose 2 in Step 1, from the Define Connection panel, add a host name, port number, display name, and description to the PCL and press Enter. Then, press END to save the connection information. |
| I                                    | SSID     | 1. From the Export Destination Subsystems panel, choose one of the following options:  
   - Choose 1 to select an available connection from the subsystem discovery list.  
   - Choose 2 to enter subsystem information manually.  
   2. If you chose 1 in Step 1, from the Subsystem Discovery List, type S in the Act column to select a subsystem that you want to add to your defined connections and press Enter. Then, press END to save the connection information.  
   3. If you chose 2 in Step 1, from the Export Destinations Add Subsystem panel, add a SSID and DB2 location name to your defined connections and press Enter. Then, press END to save the connection information. |
| S                                    | SSID     | 1. From the Export Destination Subsystems panel, choose one of the following options:  
   - Choose 1 to select an available connection from the subsystem discovery list.  
   - Choose 2 to enter subsystem information manually.  
   2. If you chose 1 in Step 1, from the Subsystem Discovery List, type S in the Act column to select a subsystem that you want to add to your defined connections and press Enter. Then, press END to save the connection information.  
   3. If you chose 2 in Step 1, from the Export Destinations Add Subsystem panel, add a SSID and DB2 location name to your defined connections and press Enter. Then, press END to save the connection information. |
### Specifying definitions to export

From the Export Definitions dialog, you can specify the following types of definitions to export:

- actions
- services
- object sets
- object-action priority
- exceptions

You can choose to export all definitions or selected definitions. For more information, see “Specifying definitions to export” on page 701.

<table>
<thead>
<tr>
<th>If you enter this in the Act column</th>
<th>Location</th>
<th>Complete this action</th>
</tr>
</thead>
</table>
| E                                  | host name| 1. From the Edit Connection panel, change the display name and description.  
2. Press END to save the updated connection information. |
| D                                  | host name| 1. From the Delete Confirmation panel, confirm the deletion of the host by specifying option 1.  
2. Press Enter.  
**Note:** Removing a host from the Destination List removes all corresponding DB2 subsystems for that host. |
| D                                  | SSID     | 1. From the Subsystem Delete Confirmation panel, confirm the deletion of the SSID by specifying option 1.  
2. Press Enter. |
| A                                  | host     | **Note:** You must have UPDATE authority to see and use this option.  
When you type A in the Act field next to a host from the Destination List and press Enter, Export displays the following messages in the upper right of the panel to indicate whether the definition was added to the Enterprise Destinations List or whether it already exists in that list:  
- Insert successful  
- Insert/Update failed |
Specifying definitions to export

Use the procedures in this section to specify the types of definitions to export:

<table>
<thead>
<tr>
<th>To specify the definition type</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>“Specifying action definitions” on page 701</td>
</tr>
<tr>
<td>service</td>
<td>“Specifying service definitions” on page 702</td>
</tr>
<tr>
<td>object set</td>
<td>“Specifying object set definitions” on page 703</td>
</tr>
<tr>
<td>object-action priorities</td>
<td>“Specifying object-action priority definitions” on page 703</td>
</tr>
<tr>
<td>exception</td>
<td>“Specifying exceptions definitions” on page 704</td>
</tr>
</tbody>
</table>

Specifying action definitions

Use this procedure to specify which actions to export.

---

**NOTE**

If an action is exported without its parent service object, its parent object must be present on the destination. For more information, see “How the Export utility works” on page 686.

---

**To specify action definitions to export**

1. From the Export Definitions main menu, choose **Specify definitions** and press **Enter**.

2. From the Export Definitions panel, select individual actions to export by typing **S** in the **Act** column next to the **Actions**. Then press **Enter**.

---

**NOTE**

Typing **A** in the **Act** column next to **Actions** selects all actions.

---

3. In the **Act** column on the Action List panel, perform one of the following actions:
   - Type **S** to select individual actions.
   - Type **A** in the *first* **Act** column to export all definitions for the listed items.
   - Type **U** and press **Enter** to deselect an action that you have selected.

4. Press **END** to save the specifications.
Specifying service definitions

Use this procedure to specify which service definitions to export.

---

**NOTE**
If service syntax is exported without its parent service object, and the service has a ddname associated with it, the corresponding service definition must be present on the destination. For more information, see “How the Export utility works” on page 686.

---

**To specify service definitions to export**

1. From the Export Definitions main menu, choose **Specify definitions** and press **Enter**.
2. From the Export Definitions panel, type **S** in the **Act** column next to the **Services** type to select individual services to export. Then press **Enter**.

---

**NOTE**
Typing **A** in the **Act** column next to Services selects all services.

---

3. In the **Act** column on the Service List panel, perform one of the following actions:
   - Type **S** to select individual services.
   - Type **A** in the first **Act** column to export all definitions for the listed items.
   - Type **U** and press **Enter** to deselect a Service you have selected.
   - Type **S** to select a service, press **Enter**, and then **L** to selectively export syntax only.
4. Press **END** to save the specifications.
5. If you chose **L** in step 3 on page 702, perform one of the following actions in the **Act** column on the Service Syntax List panel, and press **END** to save the specifications.
   - Type **A** in the first **Act** column to select all syntax.
   - Type **S** to select individual syntaxes.
   - Type **U** and press **Enter** to deselect a syntax you have selected.
Specifying object set definitions

Use this procedure to specify which object set definitions to export.

**To specify object set definitions to export**

1. From the Export Definitions main menu, choose Specify definitions and press **Enter**.

2. From the Export Definitions panel, type **S** in the **Act** column next to the **Object Set** type to select individual object sets to export. Then press **Enter**.

   **NOTE**
   
   Typing **A** in the **Act** column next to Object Sets selects all object sets.

3. In the **Act** column on the Object Set List panel, perform one of the following actions:
   - Type **S** to select individual object sets.
   - Type **A** in the *first* **Act** column to export all definitions for the listed items.
   - Type **U** and press **Enter** to deselect an object set you have selected.

4. Press **END** to save the specifications.

Specifying object-action priority definitions

Use this procedure to specify which object-action priority definitions to export.

**To specify object-action priority definitions to export**

1. From the Export Definitions main menu, choose Specify definitions and press **Enter**.

2. From the Export Definitions panel, type **S** in the **Act** column next to the **Object-Action Priorities** type to select individual object-action priorities to export. Then press **Enter**.

   **NOTE**
   
   Typing **A** in the **Act** column next to Object-Action Priorities selects all object-action priorities.
Specifying exceptions definitions

3 In the Act column on the Object-Action Priorities panel, perform one of the following actions:

- Type S to select individual object-action priorities.
- Type A in the first Act column to export all definitions for the listed items.
- Type U and press Enter to deselect an object-action priority you have selected.
- Type Z (for zoom) to expand the object name pattern.

4 Press END to save the specifications.

Specifying exceptions definitions

Use this procedure to specify which exception definitions to export:

NOTE
If a Threshold is exported without its parent Exception Definition Threshold, its parent object must be present on the destination. For more information, see “How the Export utility works” on page 686.

To specify exceptions definitions to export

1 From the Export Definitions main menu, choose Specify definitions and press Enter.

2 From the Export Definitions panel, type S in the Act column next to the Exceptions type to select individual exceptions to export. Then press Enter.

NOTE
Typing A in the Act column next to Exceptions selects all exceptions.

3 In the Act column on the Exception Definitions List panel, perform one of the following actions.

- Type S to select individual exceptions.
- Type A in the first Act column to export all definitions for the listed items.
- Type U and press Enter to deselect an object-action priority you have selected.
- Type C to selectively export only corrective actions.
- Type T to selectively export only thresholds.

4 Press END to save the specifications.
5 If you chose C in step 3, perform one of the following actions in the Act column on the Corrective Actions panel and press END to save the specifications.

- Type S to select individual corrective actions.
- Type A in the first Act column to export all definitions for the listed items.
- Type U and press Enter to deselect a corrective action you have selected.
- Type Z (for zoom) to expand the object name pattern.

**NOTE**

If a corrective action is exported without its parent action object, its parent object must be present on the destination.

Also, if a corrective action is exported without its parent exception definition object, its parent must be present on the destination.

For more information, see “How the Export utility works” on page 686.

6 If you chose T in step 3, perform one of the following actions in the Act column on the Exception Thresholds panel and press END to save the specifications.

- Type A in the first Act column to select all thresholds.
- Type S to select individual thresholds.
- Type U and press Enter to deselect a threshold you have selected.
- Type Z (for zoom) to expand the object name pattern.

**NOTE**

You cannot use this panel to select an exception and selectively specify some of its thresholds. Selecting an exception automatically selects all of its associated thresholds.
Verifying the list of definitions to export

Use this procedure to review the list of definitions to export and to remove any definitions that you do not want to export.

To review the list of definitions to export

1. From the Export Definitions main menu, choose View Export List and press Enter.

   Due to the dependencies that some definitions have upon the existence of other definitions, this list is ordered as follows:

   - services
   - service syntax
   - actions
   - object sets
   - object-action priorities
   - exceptions
   - thresholds
   - corrective actions

2. From the View Export List panel, review the list of definitions to ensure that the list reflects what you want to export. Perform one of the following actions:

   - Type X in the Act column next to any definitions that you want to remove from the export list and press END.
   - Press END to accept this list and return to the Export Definitions main menu.

Selecting the destinations for an export

After specifying object definitions, you need to select where to deploy the specifications. Use this procedure to specify the destination DB2 subsystems for the export.

To specify the destination subsystems

1. From the Export Definitions main menu, choose Select destinations and press Enter.

2. From the Export Destination List panel, type S in the Act column next to the destinations to which you want to export definitions.
Deploying definitions is the final procedure in exporting definitions to destination DB2 subsystems. After analyzing the export data, Export executes it and displays the deployment status.

To deploy the definitions

1. From the Export Definitions main menu, choose Execute Request and press Enter.

2. From the Export Destination Login panel, specify a login to each destination.

3. Press Enter.

4. From the Export Deployment panel, set the options as follows:

   A. To update existing definitions on the destination connections, choose the Overwrite existing rows option.

      If chosen, this option updates duplicate items. If you do not choose this option, Export rejects any attempts to export definitions that already exist on the destination connections.

   B. To permit Export to change the default service syntax on the destination, choose the Allow default service syntax changes option.

      This option is available only if you are explicitly exporting service syntax definitions. If you are implicitly exporting service syntax definitions, this option is not applicable. The default syntax currently specified for the service becomes the default syntax on the destination for that service.

   C. Choose Run Deployment.

   D. To view the log information after execution, choose View Log.
### Reviewing the deployment report

Export generates a Deployment Report, which lists statuses after comparing local export specifications with the exported destinations. The report uses the following statuses:

<table>
<thead>
<tr>
<th>Status</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added</td>
<td>Indicates that the definition was added.</td>
</tr>
<tr>
<td>Updated</td>
<td>Indicates that the definition was updated.</td>
</tr>
<tr>
<td>Duplicate</td>
<td>Indicates that the definition was not added because a duplicate exists on the destination and you did not specify to overwrite the definition.</td>
</tr>
<tr>
<td>RIError</td>
<td>Displays information about the referential integrity failure.</td>
</tr>
<tr>
<td>Not Added</td>
<td>Indicates that the definition was not added because it violated referential integrity on the DASD MANAGER PLUS repository.</td>
</tr>
<tr>
<td>Not Updated</td>
<td>Indicates that the definition was not updated due to an error.</td>
</tr>
</tbody>
</table>

For more information about any errors you encounter when running Export, see the messages in the BMC Documentation Center.
Using the skeleton library compiler

This appendix presents the following topics:

Overview ................................................................. 709
Compiling SLIBs ........................................................ 710
Changing SLIBs ......................................................... 711
  Testing changes using ISPF file tailoring .......................... 712
  Compiling changed SLIBs ........................................ 713
Processing SLIBs ....................................................... 713
  Generating the SLIB report ...................................... 714

Overview

BMC has improved the performance of JCL construction by using the BMC skeleton library (SLIB) compiler. The SLIB compiler is a tool that is supplied with the Administrative Products for DB2. The use of compiled SLIBs with JCL Generation’s runtime unit, which was designed to handle the compiled SLIBs, can eliminate the ISPF file tailoring process. Figure 275 illustrates the processing flow of the SLIB compiler.
This appendix describes the compiler, how to test SLIBs before compiling them, and the compiler’s associated runtime unit.

**Compiling SLIBs**

The SLIB compiler is written in REXX and runs as a batch TSO job. Each SLIB is compiled, assembled, and linked into a separately executable load module. The separation of load modules provides maximum flexibility and ease of use because no other dependencies between SLIBs or other object modules exist. You can change one SLIB without having to recompile or relink any other SLIBs or modules.
The SLIB compiler examines each line of the interpretive SLIB language and translates it into assembler source code with commands and instructions that directly interact with the runtime unit. The runtime unit processes the requests and builds the JCL. Like other compilers, the SLIB compiler translates a higher-level language into assembler instructions. Like compilers for C, C++, COBOL, and other languages, the SLIB compiler has its own runtime unit.

**NOTE**
The SLIB compiler does not support all options provided with ISPF file tailoring, but it does support all features that the Administrative products currently use.

The SLIB compiler assumes that the SLIB adheres to standard ISPF file tailoring rules and constructs. Because the compiler does little verification of SLIB syntax, BMC recommends that you verify SLIB changes by using standard ISPF file tailoring before you compile the SLIBs. For more information, see “Testing changes using ISPF file tailoring” on page 712.

## Changing SLIBs

To change an SLIB, code the changes to your SLIB source. Before you compile the SLIB, you should use JCL Generation to test the changes. Use ISPF standard file tailoring. Testing the changes ensures that the SLIB is coded correctly and that no ISPF-related errors exist.

You must recompile an SLIB each time that you change its source. Sample JCL for the SLIB compiler is in member AJXCOMPS in the `HLQ.DBCNTL` data set that is supplied by BMC 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.
Testing changes using ISPF file tailoring

The runtime unit attempts to process compiled SLIBs first. If it cannot process a compiled SLIB, it reverts to standard ISPF file tailoring. (If standard file tailoring is required, the SLIB that is being processed must exist in the ISPSLIB data set.)

Any of the following criteria force the runtime unit to use standard ISPF file tailoring for an SLIB:

- A compiled SLIB cannot be loaded from either STEPLIB or ISPLLIB.

    Remove or rename the compiled version of the SLIB in the STEPLIB or ISPLLIB. Removing the compiled version forces the runtime unit to process the uncompiled SLIB, using standard ISPF file tailoring.

- The first four bytes of the compiled SLIB contain binary zeros.

    Insert a new line 1 in the SLIB. Beginning in column 1, enter the following option:

    )CM NO-COMPILE

    Then, compile the SLIB. The compiler recognizes this SLIB as noncompilable and builds a load module with binary zeros in the first four bytes. The binary zeros force the runtime unit to process this SLIB by using standard ISPF file tailoring.

- The ddname $USESTFT is allocated as DUMMY to either the TSO session or the batch job that is being executed.

    Add the ddname $USESTFT to the JCL stream for batch jobs, or allocate it to your TSO session with the TSO ALLOC command. This option forces all SLIBs to be processed by standard ISPF file tailoring.

    For example, in Batch Execution JCL Generation, add the following JCL:

    //$USESTFT DD DUMMY

    The option remains in effect for the duration of the batch job. In foreground processing, issue the following command before entering the product:

    TSO ALLOC FI($USESTFT) DA(‘NULLFILE’) SHR

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

    TSO FREE FI($USESTFT)
Compiling changed SLIBs

After you successfully test the SLIB using standard file tailoring, compile the SLIB into your production BMC &HLQ.UDBLINK library. Ensure that you remove the )CM NO-COMPILE option if you used it. Before you test the compiled SLIB, turn off any other options that you used, such as $USESTFT.

BMC strongly recommends that you process all SLIBs as compiled SLIBs, because the runtime performance can be adversely affected by processing noncompiled SLIBs. You can use the report that the runtime unit generates to verify that you are running with compiled SLIBs and that the compile date on the SLIB is what you expect. For more information about the runtime report, see “Generating the SLIB report” on page 714.

Processing SLIBs

The runtime unit replaces the ISPF file tailoring interface. In most cases, interfaces to the runtime unit are the same as interfaces to ISPF file tailoring. To improve runtime performance, the runtime unit offers special logic that is designed specifically for JCL Generation.

The runtime unit handles mixed mode processing of compiled and noncompiled SLIBs. However, mixed mode processing is not recommended because it compromises the improved performance that the runtime unit was designed to provide.

All JCL processing flows through the runtime unit. The runtime unit

- resolves all variables
- provides numeric data padding
- handles I/O
- processes standard file tailoring requests, if necessary
- builds a report about the JCL generation process
Generating the SLIB report

The reporting feature of the runtime unit helps you determine which SLIBs were processed, how they were processed, and when they were last assembled. To use the SLIB reporting feature in Batch JCL Generation, perform the following steps:

1. Add the ddname JGENSRPT to your batch JCL stream in the step that executes AJXBMAIN with a DCB of the following parameters.

   \[\text{LRECL}=80, \text{RECFM}=FB, \text{BLKSIZE}=6160, \text{DSORG}=PS\]

2. Resubmit your job.

An example follows:

```plaintext
//JGENSRPT DD SYSOUT=*,
// DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB,DSORG=PS)
```

If you need to produce this report in the foreground, you can use the TSO ALLOC command to allocate the ddname to any data set with RECFM=FB and LRECL=80. An example follows:

```
TSO ALLOC FI(JGENSRPT) DA('dataSet.name') SHR
```

In this example, `dataSet.name` is an existing sequential data set of RECFM=FB and LRECL=80.

Figure 276 shows a sample runtime report.

**Figure 276  Sample runtime report (part 1 of 2)**

<table>
<thead>
<tr>
<th>Skelname</th>
<th>Usage</th>
<th>Compile Type</th>
<th>Compile Date</th>
<th>Compile Time</th>
<th>Usage Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJX$ACMX</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.07</td>
<td>1</td>
</tr>
<tr>
<td>AJXJOB0</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.42</td>
<td>1</td>
</tr>
<tr>
<td>AJX#USRV</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.39</td>
<td>1</td>
</tr>
<tr>
<td>AJXJOB5</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.37</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTEP1</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>09.50</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTEP7</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.00</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTEPU</td>
<td></td>
<td>Compiled</td>
<td>01/29/2013</td>
<td>13.50</td>
<td>1</td>
</tr>
<tr>
<td>AJXSYSX$</td>
<td></td>
<td>Compiled</td>
<td>01/29/2013</td>
<td>17.09</td>
<td>1</td>
</tr>
<tr>
<td>AJXSYSMD</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>11.05</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTK0</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>11.01</td>
<td>1</td>
</tr>
<tr>
<td>AJXSYSTS</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.47</td>
<td>1</td>
</tr>
<tr>
<td>AJXISPFM</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>09.51</td>
<td>1</td>
</tr>
<tr>
<td>AJXCLIBU</td>
<td></td>
<td>Compiled</td>
<td>01/29/2013</td>
<td>17.09</td>
<td>1</td>
</tr>
</tbody>
</table>
The report summary at the end of Figure 276 provides the information shown in Table 93.

Table 93  Runtime report statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of FTINCLs</td>
<td>the number of file tailoring FTINCL requests</td>
</tr>
<tr>
<td>Number of )IMs</td>
<td>the number of imbeds that are encountered when FTINCLs are processed</td>
</tr>
<tr>
<td>SLIBs processed</td>
<td>the number of SLIBs</td>
</tr>
<tr>
<td>Number of JCLRECs</td>
<td>the number of JCL records</td>
</tr>
<tr>
<td>Runtime units lastcc</td>
<td>the last condition code encountered</td>
</tr>
<tr>
<td>Runtime units maxrc</td>
<td>the highest return code encountered</td>
</tr>
</tbody>
</table>
Generating the SLIB report
Customizable reports

This appendix contains the following topics:

Introduction ................................................................. 718
Customizable report programs ........................................ 719
  Report program structure ........................................... 720
  Sample report program .............................................. 720
  Sample report .......................................................... 721
Report program customization ......................................... 722
  Report layout ............................................................ 722
  Variables ................................................................. 724
  SQL ....................................................................... 727
  JCL ....................................................................... 729
ASURXSQNL external function ........................................ 730
  ASURXSQNL return codes .......................................... 731
  SQL support ............................................................ 733
ASURXRPT external function .......................................... 733
ASURXLSPL external function ........................................ 734
ASURXSE external function .......................................... 735
ASURXLOCL external function ....................................... 739
Notes on REXX ............................................................. 740
  Comments ............................................................... 740
  REXX delimiters ...................................................... 740
  Case .................................................................... 740
  Continuation characters ........................................... 740
  Concatenation operator ........................................... 741
Annotated customizable report ....................................... 741
  Sample program ....................................................... 741
  JCL and execution ................................................... 754
  Sample report ........................................................ 755
  Complete sample program ......................................... 756
This appendix describes the report programs and emphasizes customization. You can customize the reports by changing the report layout, user-defined variables, Structured Query Language (SQL), and processing logic, or you can use the external functions provided to write your own report programs.

The DASD MANAGER PLUS report programs are written in REXX, a general-purpose, interpreted language that exists on all OS/390 systems. Because REXX is an interpreted language, the development cycle for REXX programs is short. Specifically, you edit and run instead of edit, compile, link, bind, and run.

DASD MANAGER PLUS provides the following features that are not available in REXX on OS/390:

- dynamic SQL
- report page layout
- real-time listing of volume space
- real-time access to the Integrated Catalog Facility (ICF) catalog

DASD MANAGER PLUS provides these features through the external functions listed in Table 94.
This section describes the DASD MANAGER PLUS report programs, using a sample program to illustrate report layout, variables, and SQL. This section also describes the external functions that the programs call and provides brief notes on REXX as it applies to the report programs. The customizable report programs are in the HLQ.CLIST library. (For more information, see Chapter 10, “Producing reports.”)

### Table 94  External functions not available in REXX

<table>
<thead>
<tr>
<th>External function</th>
<th>Description</th>
</tr>
</thead>
</table>
| ASURXSQL (REXX SQL)     | ASURXSQL runs the following database attachment and SQL statements:  
  - CONNECT  
  - DISCONNECT  
  - OPEN cursor  
  - FETCH cursor  
  - CLOSE cursor  
  - SELECT  
  - COMMIT  
  - ROLLBACK  
  
  ASURXSQL functionality goes beyond reporting. Using ASURXSQL, you can run any SQL statement including the following statements:  
  - CREATE  
  - INSERT  
  - UPDATE  
  - DELETE  
  - GRANT  
  
  The ability to modify the SQL or substitute new SQL gives the report programs their great flexibility. |
| ASURXRPT (REXX Report)  | ASURXRPT handles a variety of reporting tasks, such as initializing the report, formatting and writing it, creating page breaks, and releasing resources at the end of processing. |
| ASURXLSP (REXX LSpace)  | ASURXLSP reads the volume table of contents of direct access storage devices (DASD VTOCs) to collect real-time information about space on volumes. |
| ASURXSE (REXX Space Estimation) | ASURXSE returns the reorganized space for the specified partitioned table space, nonpartitioned table space, table or index. |
| ASURXLOC (REXX Locate)  | ASURXLOC is a real-time data collector that accesses the ICF catalog to find the volumes where data sets are stored and the number of extents that the data sets are in. |
Report program structure

DASD MANAGER PLUS customizable report programs share the same basic structure, as Figure 277 shows.

**Figure 277  Basic report program structure**

```
Define report format
CONNECT to ssid
INIT report
OPEN cursor

DO UNTIL END OF DATA
    FETCH
    WRITE a report line
END

CLOSE cursor
EXIT
```

Sample report program

The DASD MANAGER PLUS sample program in Figure 278 is based on one of the report programs. The adaptation produces a report (Figure 279) that lists the number of free tracks on DASD volumes that have names beginning with the letters *DEV*.

To emphasize the basic structure, the sample program does not include error checking, and it shows calls to ASURXRPT and ASURXSQL in boldface type. The complete report programs include error checking and error reporting.

For more information about the DASD MANAGER PLUS external functions, see the following sections:

- “ASURXSQL external function” on page 730
- “ASURXRPT external function” on page 733
- “ASURXLSL external function” on page 734
- “ASURXSE external function” on page 735
- “ASURXLOC external function” on page 739

**Figure 278  Sample REXX report program (part 1 of 2)**

```
01 /* rexx *******************************************************************/
02 /* detail-line format definition */
03 format. = ;
04 format.1 = " @<<<<<< @>>>>>>>>";*
05 format.2 = " vl.volid vl.freetrk";
06 format.3 = " . ";
07 /* end of format definition */
```
The sample report program (Figure 278) produces the report in Figure 279. The report programs use the data in the DASD MANAGER PLUS historical database, for example, from ATSVnn.RS_VOLUMES. However, you can also specify input from the DB2 catalog or your own databases.

**Figure 279  Volume free space report (part 1 of 2)**

<table>
<thead>
<tr>
<th>VOLUME</th>
<th>FREE TRK</th>
</tr>
</thead>
<tbody>
<tr>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>----</td>
<td>-------</td>
</tr>
</tbody>
</table>

**Sample report**
Report program customization

You can customize a report program by changing the report layout, user-defined variables, SQL, or processing logic. (To run the report programs as they are, select option 0 on the reports menu.)

Report layout

Figure 280 contains the layout for the sample report in Figure 279. The report layout consists of detail-line format definitions (lines 3 through 6) and top-of-page format definitions (lines 8 through 14), which are optional.

In defining formats, you can perform the following actions:
- define as many lines as you like
- mix text and variables (line 9)
- include counters in comments across the top (such as 123456789012345678 . . .) to help space titles and headings

The input that created the report requires the following input parameters:

<table>
<thead>
<tr>
<th>ssid</th>
<th>plan</th>
<th>volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV115</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>DEV100</td>
<td>106</td>
<td></td>
</tr>
<tr>
<td>DEV591</td>
<td>165</td>
<td></td>
</tr>
<tr>
<td>DEV055</td>
<td>436</td>
<td></td>
</tr>
<tr>
<td>DEV020</td>
<td>782</td>
<td></td>
</tr>
<tr>
<td>DEV592</td>
<td>794</td>
<td></td>
</tr>
<tr>
<td>DEV070</td>
<td>919</td>
<td></td>
</tr>
<tr>
<td>DEV075</td>
<td>956</td>
<td></td>
</tr>
<tr>
<td>DEV060</td>
<td>1059</td>
<td></td>
</tr>
<tr>
<td>DEV095</td>
<td>1118</td>
<td></td>
</tr>
<tr>
<td>DEV320</td>
<td>1223</td>
<td></td>
</tr>
<tr>
<td>DEV035</td>
<td>1244</td>
<td></td>
</tr>
</tbody>
</table>

The example uses the following values for the input parameters: DBDA ASU620DC DEV%. See also “Report-dependent variables” on page 725.
You can place format definitions almost anywhere in the code. If you remove the top-of-page formatting, which consist of the title, page number, and column headings (any format definition that contains the word *top* as the second node), other utilities can use the generated report as input.

The product stores format definitions during report initialization (line 18 on page 720), and the WRITE instruction (line 38 on the same page) uses them.

**Figure 280  Sample report layout**

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>02</td>
<td>/* detail-line format definition */</td>
</tr>
<tr>
<td>03</td>
<td>format. = ;</td>
</tr>
<tr>
<td>04</td>
<td>format.1 = &quot; @&lt;&lt;&lt;&lt;&lt;      @&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&quot;;</td>
</tr>
<tr>
<td>05</td>
<td>format.2 = &quot; vl.valid   vl.freetrk&quot;;</td>
</tr>
<tr>
<td>06</td>
<td>format.3 = &quot; . &quot;;</td>
</tr>
<tr>
<td>07</td>
<td>/* end of format definition */</td>
</tr>
<tr>
<td>08</td>
<td>/* top of page format definition */</td>
</tr>
<tr>
<td>09</td>
<td>format.top.1 = &quot; &quot;;</td>
</tr>
<tr>
<td>11</td>
<td>format.top.4 = &quot; &quot;;</td>
</tr>
<tr>
<td>12</td>
<td>format.top.5 = &quot; VOLUME FREE TRK&quot;;</td>
</tr>
<tr>
<td>13</td>
<td>format.top.6 = &quot; ------ ---------&quot;;</td>
</tr>
<tr>
<td>14</td>
<td>format.top.7 = &quot; . &quot;; /* end of format definition */</td>
</tr>
</tbody>
</table>

**Format names**

In the sample report layout, line 3 specifies the word *format* as the *stem*, or format name. The stem identifies both the top-of-page and detail-line format definitions, as Table 95 shows.

**Table 95  Format names**

<table>
<thead>
<tr>
<th>Type of format</th>
<th>Format of the format name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail line</td>
<td>stem.(\text{number})</td>
<td>format.1</td>
</tr>
<tr>
<td>Top of page</td>
<td>stem.top.(\text{number})</td>
<td>format.top.1</td>
</tr>
</tbody>
</table>

**NOTE**

The second node in the top-of-page format name must be the word *top*.

The last line of each format definition ends with a single period (line 6 and line 14).

**Reserved word**

Do not use the reserved word *suppress* as a format name.
Format definitions

Each format definition contains one or more lines, called field lines. If the definition extends to a second line, place the || concatenation operator at the end of the first line to append the next line.

Field Lines

A field line can contain any of the following items:

- fixed text (for example, lines 12 and 13)
- field holders for variable text (line 4)
- a combination of fixed text and field holders (line 9)

Field Holder

A field holder is an expression that begins with the @ symbol. A field line that contains a field holder must be followed immediately by a value line that specifies the variable that belongs in each field holder as in the following example:

```
04 format.1 = “@<<<<   @>>>>>>>”; (field line)
05 format.2 = “vl.volid  vl.freetrk”; (value line)
```

In a field holder, the characters that follow the @ indicate field justification. The number of characters (including the @) indicate the field length. For example, @>>> indicates a four-character, right-justified field. Left angle brackets (<) indicate a left-justified field, and vertical bars after an @ symbol (@||) indicate a centered field.

Value Line

The value line specifies the variables that plug into the field holders—one variable for each field holder.

Line 4 contains two field holders. Line 5, the corresponding value line, contains the variables, vl.volid and vl.freetrk, which represent the volume ID and the number of free tracks.

Variables

You can edit or add variables. Some report programs have more variables than others, but all report programs use variables in the ways that Table 96 shows. (Numbers in parentheses represent line numbers in the sample program on Figure 278.)

Table 96  How to use variables (part 1 of 2)

<table>
<thead>
<tr>
<th>Variable use</th>
<th>Examples</th>
<th>Where used in program</th>
</tr>
</thead>
<tbody>
<tr>
<td>input table-name</td>
<td>dmttable</td>
<td>setting input table name (16),</td>
</tr>
<tr>
<td></td>
<td>tablepart</td>
<td>SQL (26, 29)</td>
</tr>
<tr>
<td></td>
<td>dbname</td>
<td></td>
</tr>
<tr>
<td>report title</td>
<td>title</td>
<td>optional report title (20)</td>
</tr>
<tr>
<td>page number</td>
<td>rpt.page#</td>
<td>page number (10)</td>
</tr>
</tbody>
</table>
The *asualias* REXX EXEC returns the corresponding synonym name to the *input-table-name* variable. In the following example, the Database Space Trend Report identifies the table partition tables and index partition tables with the code. The SQL plan that the reports use must use DYNAMICRULES(BIND) to bind to the tables through the synonym and qualifier on the plan.

```plaintext
tablepart = asualias('BMCASU_STABLEPART')
indexpart = asualias('BMCASU_SINDEXPART')
```

### Input table-name

The *asualias* REXX EXEC returns the corresponding synonym name to the *input-table-name* variable. In the following example, the Database Space Trend Report identifies the table partition tables and index partition tables with the code. The SQL plan that the reports use must use DYNAMICRULES(BIND) to bind to the tables through the synonym and qualifier on the plan.

```plaintext
Table96 How to use variables (part 2 of 2)
```

<table>
<thead>
<tr>
<th>Variable use</th>
<th>Examples</th>
<th>Where used in program</th>
</tr>
</thead>
<tbody>
<tr>
<td>line number</td>
<td><code>rpt.line#</code></td>
<td>line number (not shown)</td>
</tr>
<tr>
<td>report-dependent</td>
<td><code>volume</code></td>
<td>title’s input identifier (20), SQL(27)</td>
</tr>
<tr>
<td>parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>format identifier</td>
<td><code>format</code></td>
<td>format definitions (3-6, 8-14), report initialization (18), write (38)</td>
</tr>
<tr>
<td>value line fields</td>
<td><code>vl.volid</code></td>
<td>value line (5)</td>
</tr>
<tr>
<td>REXX EXEC PARMS</td>
<td><code>ssid, plan and dbname, tbname, volume, workid</code></td>
<td>parse arg (15), CONNECT (17)</td>
</tr>
</tbody>
</table>

### Page number and line number

The ASURXRPT external function sets both the page number variable (`rpt.page#`) and the line number variable (`rpt.line#`).

### Report-dependent variables

The report-dependent variables, or parameters, are in the online dialog under the PARMS input line as in the following example. (You can also find these parameters in the comments under USAGE NOTES near the beginning of each report program.)

* USAGE NOTES
  *Parms
  * ssid plan workid utility
The PARMS input line automatically supplies the current SSID and report plan (Figure 280). Some of the report-dependent variables permit wildcards. For example, in the PARM for the Action Report, you can use wildcards for the action name and the utility name. Table 97 shows the PARM input of an Action Report. The report-dependent variables (V% and %COPY) specify all actions that begin with the letter V and contain any kind of COPY utility step.

### Table 97  Wildcards in report-dependent variables

<table>
<thead>
<tr>
<th>report-name</th>
<th>ssid</th>
<th>plan</th>
<th>wkid</th>
<th>utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>// PARM='ASUWKID DBDA ASU620DC V% %COPY'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Value-line fields**

The fields in the value line contain literals or variables. In detail lines, the value-line fields represent input from table columns, such as VOLID and FREETRK, or they represent values derived in the report program. The field names contain the cursor name and the name of the table column, using the following format:

`cursorname.columnname`

The first format produces names such as `vl.volid` and `vl.freetrk`, as follows:

```
12 format.1 = " @<<<<<< @>>>>>>>>"; 2 field holders
13 format.2 = " vl.volid    vl.freetrk"; 2 value-line fields
```

The cursor name, such as `VL`, is declared in the OPEN cursor statement, as in lines 22 and 23. The column names repeat immediately below OPEN cursor (lines 24 and 25). The column names in the format definition must match the column names in the SQL or a derived value, as follows:

```
22 rc = asurxsql("OPEN ", /* open cursor */
23          "VL CURSOR FOR SELECT ",
24          "VOLID ",
25          "FREETRK ",
```

### REXX EXEC PARM

The REXX EXEC PARM in the JCL passes information to the report programs. Table 97 provides an example of input parameters. Figure 279 describes the PARM values.
The ability to modify the SQL or substitute new SQL gives you flexibility in creating reports. Using ASURXSQL, you can run any SQL statement, including CREATE, INSERT, UPDATE, DELETE, and GRANT.

**SQL as argument**

The report programs place the SQL in parentheses to represent a single argument, as Figure 281 shows.

**Figure 281  SQL as argument**

```
22  rc = asurxsql("OPEN ", /* open cursor */
23         "VL CURSOR FOR SELECT ",
24           "VOLID ",
25           "FREETRK ",
26           "FROM " dmtable "A ",
27           "WHERE VOLID LIKE "volume" ",
28           " AND TIMESTMP = (SELECT MAX(TIMESTMP)" ,
29           " FROM " dmtable "B ",
30           " WHERE B.VOLID = A.VOLID ) ",
31           "ORDER BY FREETRK ASC ",
32           "FOR FETCH ONLY ");
```

The argument spans several lines to make it easier to read. However, the report programs treat the entire argument as a single line. Consequently, error messages refer to the entire argument by its first line number. For example, if a REXX syntax error occurs anywhere in the SQL in Figure 281, the error message refers to line 22.

**DECLARE and OPEN cursor**

The report programs combine the DECLARE cursor and OPEN cursor statements, as in the following example:

```
OPEN cursor-name CURSOR FOR SELECT
```

For an example, see OPEN VL CURSOR in lines 22 and 23 of the sample program.

**Variables in the SQL**

Variables in the SQL vary by program. In unmodified report programs, at a minimum, the variables represent the input table name, `dmtable`, and EXEC PARM input data, such as `volume` (see Figure 281).
Retrieving the most recent data

Fetching the most recent data from the DASD MANAGER PLUS historical database requires the MAX function, MAX(TIMESTAMP), within a subselect (for example, lines 28 through 30), as in the following example:

```
"AND TIMESTAMP = (SELECT MAX(TIMESTAMP) " ,
  " FROM " dmtable " B" ,
  " WHERE B.VOLID = A.VOLID)" .
```

Quotation marks

When SQL requires single quotation marks within REXX code, you must enclose the single quotation mark with double quotation marks to indicate that the single quotation mark is a literal, for example:

```
"'"variable"'"
```

The first set of double quotation marks can enclose code other than the single quotation mark, like this:

```
""........................""variable""",
```

In the following example, the first set of double quotation marks encloses the words WHERE VOLID LIKE in addition to the first single quotation mark for the variable 'volume'.

```
27 "WHERE VOLID LIKE '"volume"'" .
```
The JCL (Figure 282) in the HLQ.CNTL member ASURXJCL executes the program IKJEFT01, which interprets the DASD MANAGER PLUS report programs. Ordinarily, you use the online dialog instead to run the reports.

**Figure 282  JCL to Run Customized REXX Reports**

```
//JOBNAME JOB (acct), 'ASURXJCL', MSGCLASS=X, CLASS=A, NOTIFY=userid
//ASURXJCL EXEC PGM=IKJEFT01, PARM='REPORT SSID PLAN ...

//*--------------------------------------------------------------------
//* PARMS
//* 1.REPORT = NAME OF REPORT TO EXECUTE
//* 2.SSID = DB2 SUBSYSTEM ID OR DATA SHARING GROUP NAME
//* 3.PLAN = ASU PLAN NAME
//* ... REPORT DEPENDENT PARAMETER(S)
//*--------------------------------------------------------------------

//STEPLIB DD DISP=SHR, DSN=HLQ.ASU810.D71.LOAD
// DD DISP=SHR, DSN=DB2*****.DSNLOAD
//SYSEXEC DD DISP=SHR, DSN=HLQ.ASU810.CLIST
//SYSPRINT DD SYSOUT=* 
//SYSTEM DD SYSOUT=* 
//SYSTSPRT DD SYSOUT=* 
//SYSTSIN DD DUMMY
```

**The EXEC PARM**

The REXX EXEC PARM passes the information in Table 98 to the program.

**Table 98  Information passed by the EXEC PARM**

<table>
<thead>
<tr>
<th>PARM</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>report-name</td>
<td>name of the CLIST library member that contains the report program</td>
<td>ASUSPCTR</td>
</tr>
<tr>
<td>ssid</td>
<td>DB2 subsystem on which the report will run</td>
<td>DBDA</td>
</tr>
<tr>
<td>plan</td>
<td>plan to which the ASURXSQL plan DBRM is bound</td>
<td>ASU620DC</td>
</tr>
<tr>
<td>report-parms</td>
<td>report-dependent parameters</td>
<td>dbname-wildcard</td>
</tr>
</tbody>
</table>

The online dialog automatically supplies the SSID and plan name. If you use the JCL in Figure 282, you must supply all parameters shown in Table 98. To find the plan, select **User Options** on the DASD MANAGER PLUS main menu, then select **Current Environment Information** and look for the Statistics plan.
**DD statements**

Table 99 describes the Data Definition (DD) statements in the JCL that execute the customizable report programs.

<table>
<thead>
<tr>
<th>ddname</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSEXEC</td>
<td>the partitioned data set (PDS) library that contains the report program; the EXEC PARM references the member name</td>
</tr>
<tr>
<td>SYSTSPRT</td>
<td>the data set that contains SAY output of return codes and error messages</td>
</tr>
<tr>
<td>SYSTSIN</td>
<td>a DD dummy</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>the data set that contains the report</td>
</tr>
<tr>
<td>SYSTERM</td>
<td>the data set that contains error messages that are issued by the ASURXRPT and ASURXSQ external function and the C runtime library</td>
</tr>
</tbody>
</table>

**ASURXSQ external function**

Table 100 describes the SQL statements and commands that the ASURXSQ external function provides. In each example in the table, the argument in parentheses is an SQL statement or command, such as CONNECT or DISCONNECT. When ASURXSQ runs, it sets the variable `rc` to the return code of the command that ran. The value in the `rc` variable determines subsequent processing. Using this external ASURXSQ function, you can perform any SQL statement.

<table>
<thead>
<tr>
<th>SQL statement</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT</td>
<td><code>rc = asurxsq(&quot;CONNECT&quot; ssid plan)</code></td>
<td>Connect to the specified DB2 subsystem and open the specified plan. This is the plan to which the ASURXSQ plan DBRM is bound.</td>
</tr>
<tr>
<td>DISCONNECT</td>
<td><code>rc = asurxsq(&quot;DISCONNECT&quot;)</code></td>
<td>Commit, free all storage that ASURXSQ obtains, close the plan, disconnect from the DB2 subsystem, delete ASURXSQ from memory.</td>
</tr>
<tr>
<td>OPEN CURSOR</td>
<td><code>rc = asurxsq(&quot;OPEN cursor_name CURSOR FOR&quot; sql_statement)</code></td>
<td>Prepare the SQL SELECT statement and open the specified cursor.</td>
</tr>
</tbody>
</table>
Table 100  ASURXSQL external function (part 2 of 2)

<table>
<thead>
<tr>
<th>SQL statement</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FETCH CURSOR</td>
<td>rc = asurxsql(&quot;FETCH cursor_name&quot;)</td>
<td>Fetch the first or next row. Return SQLCODE = 100 when all rows have been fetched. The first form of the function puts the data into variables whose names are cursor_name.column_name. It also puts values into variables like cursor_name.n, where n is the column number of the query. The second form puts values into the specified variables. The maximum length of a variable name is 250 characters. Variable names can contain @, #, $, ¢, ., !, ?, _, .</td>
</tr>
<tr>
<td></td>
<td>rc = asurxsql(&quot;FETCH cursor_name INTO v1, v2, . . . vn&quot;)</td>
<td></td>
</tr>
<tr>
<td>CLOSE CURSOR</td>
<td>rc = asurxsql(&quot;CLOSE cursor_name&quot;)</td>
<td>Close the specified cursor.</td>
</tr>
<tr>
<td>EXECUTE</td>
<td>rc = asurxsql(sql_statement)</td>
<td>Prepare and run the SQL statement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the statement is SELECT, return no more than one row. The function sets variables as it does for FETCH. If the SELECT statement has no INTO clause, the stem name is always SELECT.</td>
</tr>
<tr>
<td>COMMIT</td>
<td>rc = asurxsql(&quot;COMMIT&quot;)</td>
<td>Commit uncommitted work, and close all open cursors.</td>
</tr>
<tr>
<td>ROLLBACK</td>
<td>rc = asurxsql(&quot;ROLLBACK&quot;)</td>
<td>Back out relational database changes made since the last commit.</td>
</tr>
</tbody>
</table>

ASURXSQL return codes

If ASURXSQL successfully asks DB2 for information, ASURXSQL produces return code 0. If ASURXSQL encountered non-SQL errors, ASURXSQL returns one of the error messages that Table 101 shows.

Table 101  ASURXSQL return codes (part 1 of 2)

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>no command in the argument string</td>
</tr>
<tr>
<td>31</td>
<td>already disconnected</td>
</tr>
<tr>
<td>32</td>
<td>already connected</td>
</tr>
<tr>
<td>33</td>
<td>cursor already open</td>
</tr>
<tr>
<td>34</td>
<td>too many open cursors</td>
</tr>
<tr>
<td>35</td>
<td>unknown cursor</td>
</tr>
<tr>
<td>36</td>
<td>out of memory</td>
</tr>
</tbody>
</table>
Printing error messages

The following code formats and prints SQL error messages to the SYSTERM data set:

```sql
if sqlcode != 0 then do i = 1 to 1000 by 80
  m = substr(sqlerrm, i+1, 79)
  if m = ' ' then leave
  say m
end
```

Call attachment return codes

If the CONNECT or DISCONNECT command fails, the DB2 Call Attachment Facility (CAF) sets other return codes, which are documented in the *IBM DB2 Messages and Codes* manual. The `CAFREASON` variable contains the CAF reason code for CAF failures as follows:

```
Connect failed, rc = 8, cafreason = 00F30082
```

You can use the variable `SQLSTATE` in addition to `SQLCODE`. `SQLSTATE` indicates warnings as well as errors as follows:

```
ASURXSQL return code = 37 near line 137
DSNT408I SQLCODE = -991, ERROR: CALL ATTACH WAS UNABLE TO ESTABLISH AN
  IMPLICIT CONNECT OR OPEN TO DB2.  RC1=000C  RC2=00F30006
DSNT418I SQLSTATE = 57015 SQLSTATE RETURN CODE
```

If the product issues a warning without an error, ASURXSQL will return `rc=0` and `SQLCODE=0` so that the row values will be available.

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>SQL failure; the variable <code>SQLCODE</code> will be set and the variable <code>SQLERRM</code> will contain a printable error message (see next topic)</td>
</tr>
<tr>
<td>38</td>
<td>argument string length greater than 32,767 characters</td>
</tr>
<tr>
<td>39</td>
<td>an ‘INTO’ was expected but not found</td>
</tr>
</tbody>
</table>

Table 101  ASURXSQL return codes (part 2 of 2)
SQL support

The current version of the DASD MANAGER PLUS ASURXSQL external function does not support the following items:

- more than three open cursors at the same time
- parameter markers
- the DESCRIBE command
- mixed case commands

Commands must be entirely lowercase or uppercase. For example, `connect` and `CONNECT` work, but `coNNect` does not work.

ASURXRPT external function

The ASURXRPT external function performs a variety of reporting tasks, as follows:

- initialization
- formatting
- writing
- page eject
- termination

Table 102 describes ASURXRPT functionality.

**Table 102  ASURXRPT external function**

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>init</td>
<td><code>rc = asurxrpt(&quot;init 30 suppress format1 format2 format3 . . &quot;);</code></td>
<td>After <code>init</code>, you can optionally specify the number of lines per page. The default is 66. <code>Suppress</code> is optional and prevents printing a column value when the value does not change from one row to the next. The last items in the argument string are the names of the stems that contain the format definitions. You can specify any number of formats.</td>
</tr>
<tr>
<td>write</td>
<td><code>rc = asurxrpt(&quot;write format1 &quot;);</code></td>
<td>Format and write a detail line to SYSPRINT. The second item in the argument string is the name of the stem that contains the format definition.</td>
</tr>
<tr>
<td>eject</td>
<td><code>rc = asurxrpt(&quot;eject&quot;);</code></td>
<td>Force a new top-of-page. The top-of-page format is not written until the next &quot;write&quot; command is issued. Use this feature to change the report title at a variable control break.</td>
</tr>
<tr>
<td>term</td>
<td><code>rc = asurxrpt(&quot;term&quot;);</code></td>
<td>Release resources obtained during initialization and writing.</td>
</tr>
</tbody>
</table>
ASURXRPT produces the return codes that Table 103 shows.

### Table 103  ASURXRPT return codes

<table>
<thead>
<tr>
<th>Return codes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>successful completion</td>
</tr>
<tr>
<td>40</td>
<td>syntax error in report format definition</td>
</tr>
<tr>
<td>41</td>
<td>error obtaining a REXX variable</td>
</tr>
<tr>
<td>44</td>
<td>general error; see message in <code>RPTERM</code></td>
</tr>
<tr>
<td>48</td>
<td>error loading module IRXEXCOM</td>
</tr>
</tbody>
</table>

When ASURXRPT fails, the variable `RPTCODE` is set, and the variable `RPTERM` contains a printable diagnostic message.

## ASURXLSP external function

The ASURXLSP external function collects real-time space data for the volume specified, which does not use wildcards during processing. The return value is a string of words that are separated by blanks. ASURXLSP returns the volume information in the following order:

- return code from ASURXLSP (or 4 if the volume is not found)
- volume serial number
- device type (14 = 3380, 15=3390)
- cylinders on the volume less alternate
- tracks per cylinder
- number of free extents
- total number of free cylinders
- total number of additional free tracks
- number of cylinders in the largest free extent
- number of additional tracks in the largest free extent
- count of Format 0 records
- VTOC Index Record (VIR) count
- fragmentation index

ASURXLSP produces the return codes that Table 104 shows.
The ASURXSE external function returns the reorganized space for the partitioned table space, nonpartitioned table space, table, or index that you specify. The product returns the results for each type of DB2 object in the variable SERELST as blank-separated values. Table 105 describes ASURXSE functionality.

### Table 104  ASURXSP return codes

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>successful completion</td>
</tr>
<tr>
<td>4</td>
<td>volume not found</td>
</tr>
</tbody>
</table>

### ASURXSE external function

Estimate the reorganized space for a partitioned table space.
SERELST = TP rpp pages space trks cyls extents dsns

Where:
- **TP** = the literal ‘TP’
- **rpp** = rows per page
- **pages** = number of pages in partition
- **space** = space in KB
- **trks** = space in tracks
- **cyls** = space in cylinders
- **extents** = number of extents
- **dsns** = number of data sets

#### Example

```
rc = asurxse("TP" db2ver card pqty sqty partitions pctfree freepage pgsiz dssize rowavg maxrows compress devtype alocunit type segsize);
```

Where:
- **TP** = the literal ‘TP’
- **db2ver** = DB2 Version
- **card** = number of rows in partition
- **pqty** = primary quantity in allocation units
- **sqty** = secondary quantity in allocation units
- **partitions** = number of partitions
- **pctfree** = percentage of free space
- **freepage** = free pages
- **pgsize** = size of page (4, 8, 16, or 32)
- **dssize** = maximum size of a data set in kilobytes
- **rowavg** = average row length
- **maxrows** = maximum rows per page for table space
- **compress** = data compression
- **devtype** = device type
- **alocunit** = allocation unit
- **type** = type of table space (L, I, K, R, or 1 where 1 indicates blank or not specified)\(^a\)

**Note:**
- **segsize** = number of pages in the segment\(^b\)
### Table 105 ASURXSE external function (part 2 of 4)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB non-partitioned table</td>
<td>rc = asurxse(&quot;TB&quot; db2ver card pctfree freepage pgsise rowavg compress segsize maxrows);</td>
<td>Estimate the reorganized space for a nonpartitioned table.</td>
</tr>
<tr>
<td></td>
<td>Where:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TB = the literal ‘TB’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>db2ver = DB2 Version</td>
<td></td>
</tr>
<tr>
<td></td>
<td>card = number of rows in table</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pctfree = percentage of free space</td>
<td></td>
</tr>
<tr>
<td></td>
<td>freepage = free pages</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pgsise = size of page (4, 8, 16, or 32)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rowavg = average row length</td>
<td></td>
</tr>
<tr>
<td></td>
<td>compress = data compression</td>
<td></td>
</tr>
<tr>
<td></td>
<td>segsize = segment size</td>
<td></td>
</tr>
<tr>
<td></td>
<td>maxrows = maximum rows per page for table space</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SERESULT = TB rpp pages</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Where:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TB = the literal ‘TB’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rpp = number of rows per page</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pages = number of pages in the table</td>
<td></td>
</tr>
<tr>
<td>TS non-partitioned table space</td>
<td>rc = asurxse(&quot;TS&quot; db2ver pages pqty sqty freepage pgsise compress segsize devtype alocunit type);</td>
<td>Estimate the reorganized space for a nonpartitioned table space.</td>
</tr>
<tr>
<td></td>
<td>Where:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TS = the literal ‘TS’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>db2ver = DB2 Version</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pages = total pages for all tables</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pqty = primary quantity in allocation units</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sqty = secondary quantity in allocation units</td>
<td></td>
</tr>
<tr>
<td></td>
<td>freepage = free pages</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pgsise = size of page (4, 8, 16, or 32)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>compress = data compression</td>
<td></td>
</tr>
<tr>
<td></td>
<td>segsize = segment size</td>
<td></td>
</tr>
<tr>
<td></td>
<td>devtype = device type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>alocunit = allocation unit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>type = type of table space (L, I, K, or blank)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SERESULT = TS pages space trks cyls extents dsns</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Where:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TS = the literal ‘TS’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pages = total pages for all tables</td>
<td></td>
</tr>
<tr>
<td></td>
<td>space = space in KB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>trks = space in tracks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cyls = space in cylinders</td>
<td></td>
</tr>
<tr>
<td></td>
<td>extents = number of extents</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dsns = number of data sets</td>
<td></td>
</tr>
</tbody>
</table>
Table 105  **ASURXSE external function (part 3 of 4)**

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS partition-by-growth table</td>
<td>rc = asurxse(‘TS’ db2ver pages pqty sqty freepage pgsise compress segsize devtype alocunit type rpp dssize card);</td>
<td>Estimate the reorganized space for a partition-by-growth table space. SERESULT = TS parts pages space trks cyls extents dsns pages2 space2 trks2 cyls2 extents2 dsns2 card card2</td>
</tr>
<tr>
<td>space</td>
<td>Where:</td>
<td>Where:</td>
</tr>
<tr>
<td></td>
<td>TS = the literal ‘TS’</td>
<td>TS = the literal ‘TS’</td>
</tr>
<tr>
<td></td>
<td>db2ver = DB2 Version</td>
<td>parts = total number of partitions</td>
</tr>
<tr>
<td></td>
<td>pages = total pages for all tables</td>
<td>pages = total pages per partition (excluding the last partition)</td>
</tr>
<tr>
<td></td>
<td>pqty = primary quantity in allocation units</td>
<td>space = KBs per partition (excluding the last partition)</td>
</tr>
<tr>
<td></td>
<td>sqty = secondary quantity in allocation units</td>
<td>trks = tracks per partition (excluding the last partition)</td>
</tr>
<tr>
<td></td>
<td>freepage = free pages</td>
<td>cyls = cylinders per partition (excluding the last partition)</td>
</tr>
<tr>
<td></td>
<td>pgsise = size of page (4, 8, 16, or 32)</td>
<td>extents = extents per partition (excluding the last partition)</td>
</tr>
<tr>
<td></td>
<td>compress = data compression</td>
<td>dsns = data sets per partition (excluding the last partition)</td>
</tr>
<tr>
<td></td>
<td>segsize = segment size</td>
<td>pages2 = pages in the last partition</td>
</tr>
<tr>
<td></td>
<td>devtype = device type</td>
<td>space2 = KBs in the last partition</td>
</tr>
<tr>
<td></td>
<td>alocunit = allocation unit</td>
<td>trks2 = tracks in the last partition</td>
</tr>
<tr>
<td></td>
<td>type = type of table space (G)</td>
<td>cyls2 = cylinders in the last partition</td>
</tr>
<tr>
<td></td>
<td>rpp= rows per page</td>
<td>extents2 = extents in the last partition</td>
</tr>
<tr>
<td></td>
<td>dssize =maximum size of a data set in kilobytes</td>
<td>dsns2 = data sets in the last partition</td>
</tr>
<tr>
<td></td>
<td>card= number of rows in table</td>
<td>card = rows per partition (excluding the last partition)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>card2 = rows in the last partition</td>
</tr>
</tbody>
</table>
ASURXSE external function

Table 105  ASURXSE external function (part 4 of 4)

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP index</td>
<td>rc = asurxse(&quot;IP&quot; db2ver card rpk pqty sqty parts avgkeylength nlavgkeylen pctfree freepage piecesize uniquerule indextype devtype allocunit typets pgsize);</td>
<td>Estimate the reorganized space for an index. SERESULT= IP pages space trks cyls nlevels extents dsns</td>
</tr>
</tbody>
</table>

   Where:
   IP = the literal 'IP'
db2ver = DB2 Version
card = number of rows in table
rpk = rows per key
pqty = primary quantity in allocation units
sqty = secondary quantity in allocation units
parts = number of partitions
avgkeylength = average length of key
nlavgkeylen = average length for nonleaf page key
subpages= number of subpages
pctfree = percentage of free space
freepage = free pages
piecesize = file size for ix
uniquerule = unique rule
indextype = index type
devtype = device type
allocunit = allocation unit
typets = type of table space (G, R, L, 1 (1 indicates blank or not specified))c
pgsize = size of the leaf pages in the index (4, 8, 16, or 32)c

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ASURXLOC external function

The ASURXLOC external function returns the list of volumes where a data set resides. The syntax is ASURXLOC(stemname,datasetname) where stemname is the name of a REXX stem and datasetname is the name of the data set. ASURXLOC returns values in the following REXX variables:

- stemname.0 = number of volumes on which datasetname resides
- stemname.n = nthDeviceType volumeSerialNumber volumeSequenceNumber

Figure 283 shows a sample REXX program that uses the ASURXLOC external function.

Figure 283  Sample REXX program using ASURXLOC

```rehxx
/* ASURXLOC Example */
arg DatasetName
say 'Dataset Name is ' || DatasetName
VolList.0 = 0
rc = asurxloc('VolList.',DatasetName)
if ( rc = 0 ) then do
  do i = 1 to VolList.0
    say 'Volume #' || i || ' ' || VolList.i
  end
end
else do
  if ( rc = 4 ) then do
    say 'Could not open ICF catalog'
  end
  if ( rc = 8 ) then do
    say 'Data set not cataloged'
  end
end
exit rc
```

To set the returned data into individual variables, use the following REXX statements:

```rehxx
devtype = word(VolList.i,1)
volserno = word(VolList.i,2)
volseqno = word(VolList.i,3)
```

ASURXLOC produces the return codes that Table 107 shows.
Notes on REXX

Although this appendix does not discuss REXX in great detail, you might find the following information useful. (For SQL within REXX code, see “Quotation marks” on page 728.)

Comments

Every REXX program must begin with a comment that contains the word REXX. REXX comments begin with the characters /* and end with */ as follows:

```
01 /* rexx ***************************************************************************/

```

REXX delimiters

REXX uses the semicolon (;) as a clause delimiter that you can omit if the end of a line implies a delimiter.

Case

Uppercase and lowercase are not significant outside quoted strings.

Continuation characters

To continue a REXX statement, use a trailing comma, as follows:

```
"WHERE DBNAME" egorlike "' dbnamex' ",
" AND TIMESTMP = (SELECT MAX(TIMESTMP) ").
```

---

**Table 107  ASURXLOC return codes**

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>successful completion</td>
</tr>
<tr>
<td>4</td>
<td>could not open the ICF catalog</td>
</tr>
<tr>
<td>8</td>
<td>data set not cataloged</td>
</tr>
</tbody>
</table>
Concatenation operator

To concatenate strings, use the | | concatenation operator. The column headings require two lines. The concatenation operator is at the end of the first line as follows:

```
ipfd.top.9 = " DBNAME CREATOR IXNAME           PART USED SPACE"| |
       "        GB   % GB DATE STATS";
```

Annotated customizable report

This section describes a simple REXX program that creates a report on space utilization for DB2 table spaces. The JCL that ran the report program and the report that the program produced appear at the end of the section.

Sample program

The annotation is in the following sections:

- “Specify the DB2 table and report title” on page 742
- “Define report formats” on page 743
- “Connect to the DB2 subsystem and open the plan” on page 746
- “Initialize the report” on page 747
- “Begin reading input and get database names” on page 747
- “Open cursor for SELECT” on page 749
- “Fetch and write the table space data” on page 750
- “The report error procedure” on page 751
- “The CAF error procedure” on page 752
- “The SQL error procedure” on page 752
- “The cleanup procedure” on page 754

The complete sample program is on page 756.
Specify the DB2 table and report title

Lines 1 through 22 of Figure 284 document the program and specify the name of the DB2 table that contains the input data and the report title.

Figure 284 Specifying the DB2 object and report title

```
01 /* rexx ***************************************************************************/
02 /*                                                                    */
03 /*  SYNOPSIS                                                          */
04 /*      TSSPACE - Space utilization for DB2 tablespaces                */
05 /*                                                                    */
06 /*  DESCRIPTION                                                       */
07 /*      A report by tablespace of space allocated and number of       */
08 /*      extents.                                                      */
09 /*                                                                    */
10 /*  USAGE NOTES                                                       */
11 /*     Parms                                                          */
12 /*      ssid plan tbname-prefix dbname                                 */
13 /*                                                                    */
14 /*                                                                      */
15 /* The following statement should specify the table name of the       */
16 /* DASD MANAGER PLUS tablepart table                                   */
17 /*                                                                      */
18 /* dmtablepart = asualias('BMCASU_STABLEPART');                        */
19 /*                                                                      */
20 /* title = "DASD SPACE FOR TABLESPACE";                               */
21 /*                                                                      */
22 /*                                                                      */
```

lines 1-17 These lines are comments.

line 1 This line identifies REXX as the programming language. The word REXX is required in the first line of each REXX program.

lines 3-4 These lines identify the name of the report and its subject: space utilization for DB2 table spaces.

lines 6-8 These lines provide a brief description of the report.

lines 10-12 These lines identify the parameters passed to the program by the EXEC PARM in the JCL that creates the report as in the following example. (The complete JCL is in Figure 297.)

```
//REXX EXEC PGM=IKJEFT01,PARM='TSSPACE ssid plan dbname-wildcard'
```
Table 108 describes the information passed by the EXEC PARM.

Table 108  Information passed by the EXEC PARM

<table>
<thead>
<tr>
<th>PARM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSSPACE</td>
<td>name of the CLIST member that contains the REXX program</td>
</tr>
<tr>
<td>ssid</td>
<td>DB2 subsystem on which the report runs</td>
</tr>
<tr>
<td>plan</td>
<td>plan to which the ASURXSQL DBRM is bound</td>
</tr>
<tr>
<td>dbname-wildcard</td>
<td>name of the databases against which to run the report; you can use a wildcard pattern</td>
</tr>
</tbody>
</table>

lines 16-17  These lines are comments.

line 19  This line assigns the name of the DASD MANAGER PLUS table partition table to the variable dmtablepart. The assignment uses the following asualias REXX EXEC. The ASUALIAS table is in the HLQ CLIST library as in the following example:

"BMCASU_STABLEPART                   BMCASU_STABLEPART ".

line 21  This line assigns the report title, DASD SPACE FOR TABLESPACE, to the variable title.

Define report formats

Lines 23 through 38 of Figure 285 define the report formats for detail lines and top-of-page headings. Top-of-page formatting is not required. If you remove the top-of-page formatting—the title, page number, and column headings—other utilities can process the output more easily.

In defining formats, you can perform the following tasks:

- define as many lines as you want
- mix text and variables (line 33 in Figure 285)
- include counters in comments across the top (such as 123456789012345678 . . .) to help space titles and headings

Format definitions can be placed almost anywhere in the code. The ASURXRPT WRITE statement runs them later.

Figure 285  Defining report formats (part 1 of 2)

23 /* tablepart format definition */
24
25 tpfd. = ;
26 tpfd.1 = " @<<<<<<<<<<<<<<<<< @>>> @>>>>>>>>  @>>>";
27 tpfd.2 = " tsobj tp.partition tp.space tp.extents";
This line is a comment.

These lines define the format for the table partition detail lines.

This line is a null assignment to stem variable `tpfd`, identifying the report format. Lines 26 through 28 and lines 32 through 38 add suffixes to the stem. (Line 48, shown later, refers to the stem variable during report initialization.)

This line uses four field holders to define the length, justification, and spacing of fields in the detail line (Table 109). Line 26 assigns the detail line format definition to variable `tpfd.1`. The line contains spaces that appear in the output.

Table 109  Detail line format definitions

<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
<th>Justification</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>@&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;</td>
<td>left</td>
<td>17 (1@ + 16 &lt;)</td>
</tr>
<tr>
<td>2</td>
<td>@&gt;&gt;&gt;</td>
<td>right</td>
<td>4 (1@ + 3 &gt;)</td>
</tr>
<tr>
<td>3</td>
<td>@&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>right</td>
<td>9 (1@ + 8 &gt;)</td>
</tr>
<tr>
<td>4</td>
<td>@&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>right</td>
<td>4 (1@ + 3 &gt;)</td>
</tr>
</tbody>
</table>

For information about this type of format definition, see “Report layout” on page 722.

Line 27 identifies the four variables to plug into the format definitions in line 26 and assigns them to variable `tpfd.2`. These detail-line variables correspond to the column headings defined in line 36. Table 110 describes the variables in the detail line. The last three variables use the following format:

cursorname.columnname
The cursor TP is declared in the OPEN CURSOR statement in line 67. (See Figure 289.) Table 110 describes the variables in the detail line.

Table 110  Variables in the detail line format definitions

<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
<th>Variable content</th>
<th>Source of variable</th>
<th>Resolution of variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>@&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;&lt;</td>
<td>tsobj</td>
<td>lines 92 through 94</td>
<td>dbname.tsname</td>
</tr>
<tr>
<td>2</td>
<td>@&gt;&gt;&gt;</td>
<td>tp.partition</td>
<td>cursorname.column-name</td>
<td>partition number</td>
</tr>
<tr>
<td>3</td>
<td>@&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>tp.space</td>
<td>cursorname.column-name</td>
<td>space</td>
</tr>
<tr>
<td>4</td>
<td>@&gt;&gt;&gt;</td>
<td>tp.extents</td>
<td>cursorname.column-name</td>
<td>number of extents</td>
</tr>
</tbody>
</table>

line 28 This line ends the format definition of the detail line by assigning a single period to variable tpfd.3.

lines 30 This line is a comment.

lines 32-38 These lines define the format for the top-of-page. This definition includes the report title, page number, and column headings.

line 32 This line assigns a space to variable tpfd.top.1. The variable creates a blank line at the top of each page.

NOTE The word top must be the second node of each top-of-page format name.

line 33 This line defines the length, justification, and spacing of the report title and page number and assigns them to the variable tpfd.top.2. Line 33 contains a mixture of variables and constants (Table 111). Spaces separate the field definitions that appear in the output.

Table 111  Top of page format definitions (second line)

<table>
<thead>
<tr>
<th>Field</th>
<th>Definition</th>
<th>Alignment</th>
<th>Length</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>@</td>
<td>center</td>
<td>36 (1@ + 35 vertical lines)</td>
<td>variable title</td>
</tr>
<tr>
<td>2</td>
<td>PAGE</td>
<td>left</td>
<td>4</td>
<td>constant</td>
</tr>
<tr>
<td>3</td>
<td>@&lt;&lt;&lt;&lt;</td>
<td>left</td>
<td>5 (1@ + 4 &lt;)</td>
<td>variable rpt.page#</td>
</tr>
</tbody>
</table>

For information about top-of-page-processing, see “ASURXRPT external function” on page 733.
line 34  This line assigns the report title defined in line 21 to variable `tpfd.top.3`, and assigns the page number to variable `rpt.page#`, which is an internal variable set in ASURXRPT.

line 35  This line assigns a space to variable `tpfd.top.4`. The variable creates a blank line under the report title.

line 36  This line specifies the column headings and assigns them to `tpfd.top.5`.

line 37  This line specifies dashed lines under the column headings and assigns them to variable `tpfd.top.6`.

line 38  This line ends the format definition by assigning a single period in quotes to `tpfd.top.7`.

Connect to the DB2 subsystem and open the plan

Lines 41 through 46 of Figure 286 obtain the DB2 subsystem ID, plan name, and database name, then connect to the subsystem and open the plan.

**Figure 286  Connecting to the DB2 subsystem and opening the plan**

```
41 parse upper arg ssid plan dbnamex .; /* get input data               */
42
43 rc = asurxsql("CONNECT" ssid plan); /* connect and open plan       */
44
45 if (rc ≠ 0) then                   /* if connect or open plan error */
46  signal caferror;                  /* go output caf error message  */
47```

line 41  This line obtains the SSID, plan name, and database names—information passed to the program by the PARM of the EXEC statement.

- After retrieving the information, this instruction converts the information to uppercase before saving it into the variables `ssid`, `plan`, and `dbnamex`.
- The period at the end of the instruction discards information (such as the sequence number) that follows the SSID, plan, and database name.

line 43  This line issues the SQL CONNECT instruction, using the SSID and plan name obtained in line 41, and assigns the return code to the `rc` variable.

line 45-46 If the CONNECT or open fails—if the return code is not 0—these lines trap the CAF error condition (`caferror`) and issue a CAF error message (lines 118 through 126 on page 752).
Initialize the report

Lines 48 through 51 of Figure 287 initialize the report, defining the page length, the stem variable associated with the format definitions, and whether to skip column values that repeat from one row to the next.

Figure 287  Initializing the report

```
48  rc = asurxrpt("INIT 60 suppress tpfd"); /* initialize report */
49
50  if (rc ≠ 0) then /* if an error */
51  signal rpterror; /* go output report error msg */
52
```

line 48  This line runs external function ASURXRPT with an init argument string. The return code from the function is assigned to the rc variable. The product sends the following arguments to ASURXRPT:

<table>
<thead>
<tr>
<th>init</th>
<th>initializes the report</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>defines the page length as 60 lines. The default is 66.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>suppress</th>
<th>a reserved word, specifies not to print a column value when the value does not change from one row to the next. When used, the reserved word applies to all columns. You cannot specify specific columns.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpfd</td>
<td>names the stem variable that identifies the format definition. (See lines 25 through 38 on page 743.)</td>
</tr>
</tbody>
</table>

lines 50-51  If the return code is not 0—if initialization fails—these lines trap the rpterror error condition and issue a report error message (lines 111 through 115 on page 751).

Begin reading input and get database names

Lines 53 through 58 of Figure 288 read and process the input data from the EXEC PARM (Figure 295). The input contains one or more database names, which can contain wildcards.

Figure 288  Getting the input and testing for wildcards

```
53  dbnamex = translate(dbnamex, ".", "."); /* change * to % in dbnamex */
54  eqorlike = "="; /* default to equal */
55
56  if (0 < pos("%", dbnamex)) | 
57     if wildcard in dbnamex */
58  eqorlike = "LIKE"; /* use like */
59
```
lines 53-54 These lines substitute the % sign for any asterisks (*) encountered in input database names and set the variable eqorlike to equal (=).

line 53 This line changes any * in a database name to a %.

line 54 This line assigns an equal sign (=) to the variable eqorlike. The equal sign tells REXX to interpret the database name exactly as it is written.

lines 56-57 These lines test for other wildcards in the database names. If they find a wildcard, they reset the value of eqorlike (line 54) to ”LIKE” (line 58).

**NOTE**
The | logical operator represents OR.

line 56 This line searches the database name in the input for the percent (%) wildcard, using the pos function. The function returns one of the following:

- the position of the percent within the string being searched
- 0 if no percent sign is found

If no percent sign is found (0 not < 0), go to the next line. If a percent sign is found, go to line 58.

line 57 This line searches the database name in the input stream for the underscore (_) wildcard, using the pos function. The function returns one of the following:

- the position of the underscore within the string being searched
- 0 if no underscore is found

If it does not find an underscore (0 not < 0), go to the next line. If it does find an underscore, go to line 58.

line 58 If the product finds a wildcard (*, %, or _) in the database name that it is reading, line 58 assigns the value ”LIKE” to the variable eqorlike (that is, eqorlike =”LIKE”) in the subsequent SELECT statement (line 63 on page 749).
Open cursor for SELECT

Lines 60 through 79 of Figure 289 prepare to obtain table partition data by opening a cursor for SELECT FOR FETCH ONLY processing.

Figure 289  Fetching the data

```plaintext
60 /* obtain tablepart data */
61
62 rc = asurxsql("OPEN ", /* open cursor */
63   "TP CURSOR FOR SELECT ",
64   "A.DBNAME ",
65   "A.TSNAME ",
66   "A.PARTITION ",
67   "A.SPACED ",
68   "A.EXTENTS ",
69   "FROM " dmtablepart "A ",
70   "WHERE A.DBNAME" eqorlike "**dbnamex**" ",
71   " AND TIMESTMP = (SELECT MAX(TIMESTMP) ",
72   " FROM " dmtablepart "B ",
73   " WHERE B.DBNAME = A.DBNAME ",
74   " AND B.TSNAME = A.TSNAME ",
75   " AND B.PARTITION = A.PARTITION) ",
76   "ORDER BY DBNAME, ",
77   " TSNAME, ",
78   " PARTITION ",
79   "FOR FETCH ONLY ");
80
81 if (rc ≠ 0) then /* if open failed */
82  signal error; /* go output sql error message */
83```

line 60  This line is a comment.

lines 62-79  These lines open a cursor by using the ASURXSQL function. They name the cursor TP. They also prepare the SELECT statement; and assign the return code to rc. The SELECT statement uses previously defined variables: dmtablepart (line 19), eqorlike (line 54 or 58), and dbnamex (line 41).

**NOTE**

A maximum of three cursors can be open at the same time. Cursor length is up to 80 characters. Columns can be referenced by name or by number.

lines 81-82  If the OPEN fails— if the return code is not zero—these lines trap the SQL error condition (sqlerror) (lines 135 through 141 on page 752).
# Fetch and write the table space data

Lines 84 through 102 of Figure 290 prepare the data, write it, close the cursor, and execute the cleanup procedure.

**Figure 290  Preparing and writing the data**

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>84</td>
<td>do</td>
<td>do forever</td>
</tr>
<tr>
<td>85</td>
<td>rc</td>
<td>rc = asurxsql(&quot;FETCH TP &quot;);  /* fetch next row */</td>
</tr>
<tr>
<td>86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>if</td>
<td>if (rc ¬= 0) then /* if fetch failed */</td>
</tr>
<tr>
<td>88</td>
<td></td>
<td>signal sqlerror; /* go output sql error message */</td>
</tr>
<tr>
<td>89</td>
<td>if</td>
<td>if (sqlcode = 100) then /* if end of data */</td>
</tr>
<tr>
<td>90</td>
<td></td>
<td>leave; /* leave forever loop */</td>
</tr>
<tr>
<td>91</td>
<td></td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>dbn</td>
<td>dbn = strip(tp.dbname, 'T'); /* remove trailing blanks */</td>
</tr>
<tr>
<td>93</td>
<td>tsn</td>
<td>tsn = strip(tp.tsname, 'T');</td>
</tr>
<tr>
<td>94</td>
<td>tsobj</td>
<td>tsobj = dbn&quot;.tsn; /* dbname.tsname */</td>
</tr>
<tr>
<td>95</td>
<td>rc</td>
<td>rc = asurxrpt(&quot;WRITE tpfd&quot;); /* output tablepart data */</td>
</tr>
<tr>
<td>96</td>
<td>if</td>
<td>if (rc ¬= 0) then /* if an error */</td>
</tr>
<tr>
<td>97</td>
<td></td>
<td>signal rpterror; /* go output report error message */</td>
</tr>
<tr>
<td>98</td>
<td></td>
<td>end /* end do forever */</td>
</tr>
<tr>
<td>99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>rcx</td>
<td>rcx = asurxsql(&quot;CLOSE TP&quot;); /* close cursor */</td>
</tr>
<tr>
<td>101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>signal</td>
<td>signal cleanup; /* cleanup */</td>
</tr>
</tbody>
</table>

**lines 84-98** These lines are a do-forever loop to fetch the table partition rows.

**line 85** This line fetches the first row or next row by using the ASURXSQL function. The line places the data in the TP variable; assigns the return code to the variable rc.

**lines 87-88** If the FETCH fails (if the return code is not 0), these lines trap the SQL error condition (sqlerror) and execute the SQL error procedure (lines 135 through 144 on page 752).

**line 89-90** These lines exit the loop if all rows have been fetched (if the return SQLCODE = 100).

**lines 92-93** These lines remove trailing blanks from the database name and table space name by using the strip function with the T (Trailing) argument. The lines assign the resulting names to the dbn and tsn variables, respectively.

**line 94** This line creates the qualified table space name by placing a period (.) between the trimmed database name (line 92) and the trimmed table space name (line 93). This line assigns the resulting table space name to the variable tsobj.

**line 95** This line executes the ASURXRPT WRITE function, identifies the stem variable that identifies the report formats, and assigns the resulting return code to the variable rc.
If the return code is not 0, this line traps the report error condition and output the report error message (lines 111 through 115).

This line ends the do-forever loop begun in line 84.

This line closes the cursor opened in lines 62 and 63.

This line executes the cleanup procedure.

**The report error procedure**

Lines 105 through 115 of Figure 291 executes the Report error procedure, which issues an error code and error message from the external function ASURXRPT, then releases resources and calls the cleanup procedure.

**Figure 291 The report error procedure**

```plaintext
104 /*-------------------------------------------------------------------*/
105  rpterror - output error code and message from asurxrpt
106 /*-------------------------------------------------------------------*/
107 rpterror:
108  say "ASURXRPT return code = "rptcode; /* error code            */
109  say rpterrm;                      /* output report error message     */
110  signal cleanup;                   /* cleanup                         */
111
```

These lines are comments.

This line marks the beginning of the report error procedure, which runs during report initialization (line 48 on page 747) or while it writes the data (line 95 on page 750).

This line outputs the report return code to the SYSTSPRT DD data set, using the format "ASURXRPT error code = rptcode". The ASURXRPT error code is anything other than zero (0).

This line outputs a printable diagnostic message to the SYSTSPRT DD data set by using the variable rpterrm. This variable is set internally in ASURXRPT.

This line directs REXX to go to the cleanup label and begin executing the cleanup procedure (lines 144 through 147 on page 754).
The CAF error procedure

Lines 118 through 126 of Figure 292 are the CAF error procedure, which the product runs when encountering a CAF processing error, for example in lines 45 and 46. The procedure returns the CAF error code and message.

Figure 292 Reporting CAF error code and message

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>117</td>
<td>/<em>--------------------------------</em>/</td>
</tr>
<tr>
<td>118</td>
<td>/<em>--------------------------------</em>/</td>
</tr>
<tr>
<td>119</td>
<td>caferror - output caf error msg</td>
</tr>
<tr>
<td>120</td>
<td>/<em>--------------------------------</em>/</td>
</tr>
<tr>
<td>121</td>
<td>caferror:</td>
</tr>
<tr>
<td>122</td>
<td>say 'Connect failed, rc = 'rc'. cafreason = 'cafreason;</td>
</tr>
<tr>
<td>123</td>
<td>exit 8;</td>
</tr>
<tr>
<td>124</td>
<td>/<em>--------------------------------</em>/</td>
</tr>
<tr>
<td>125</td>
<td>sqlerror - output sql error codes and messages</td>
</tr>
<tr>
<td>126</td>
<td>/<em>--------------------------------</em>/</td>
</tr>
<tr>
<td>127</td>
<td>sqlerror:</td>
</tr>
<tr>
<td>128</td>
<td>say &quot;ASURXSQL return code = &quot;rc;</td>
</tr>
</tbody>
</table>

These lines are comments.

- **Line 124**: This line marks the beginning of the CAF error procedure, which the product runs if the CONNECT fails (line 46 on page 746).
- **Line 125**: This line returns the CAF error message and CAFREASON to the SYSTSPRT DD. For more information on CAF error messages, see “Call attachment return codes” on page 732.
- **Line 126**: This line exits the program and sets the return code to 8.

The SQL error procedure

Lines 129 through 142 of Figure 293 are the SQL error procedure, which the product runs each time it encounters an SQL processing error. The procedure returns the SQL error codes and messages.

Figure 293 The SQL error procedure (part 1 of 2)

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>/<em>--------------------------------</em>/</td>
</tr>
<tr>
<td>129</td>
<td>/<em>--------------------------------</em>/</td>
</tr>
<tr>
<td>130</td>
<td>sqlerror - output sql error codes and messages</td>
</tr>
<tr>
<td>131</td>
<td>/<em>--------------------------------</em>/</td>
</tr>
<tr>
<td>132</td>
<td>sqlerror:</td>
</tr>
<tr>
<td>133</td>
<td>say &quot;ASURXSQL return code = &quot;rc;</td>
</tr>
<tr>
<td>134</td>
<td>/<em>--------------------------------</em>/</td>
</tr>
<tr>
<td>135</td>
<td>sqlerror:</td>
</tr>
<tr>
<td>136</td>
<td>say &quot;ASURXSQL return code = &quot;rc;</td>
</tr>
</tbody>
</table>
These lines are comments.

This line marks the beginning of the SQL error procedure, which runs if any of the following error conditions occur:

- OPEN cursor fails (lines 81 and 82 on page 749)
- SQL errors (lines 81 and 82)
- FETCH row fails (line 88 on page 750)

This line returns the SQL error return code in the format

\[ \text{ASURXSQL return code} = \text{rc} \]

These lines check for an SQL error, and if one is present, divide the message into 80-byte segments and output the SQL error message.

If the product finds a nonzero SQL code, this line increments the variable \( i \) (loop counter) by 80.

This line returns a substring of the SQL error message (sqlerrm) text. This instruction divides the message into 80-byte segments and assigns the resulting message to the variable \( m \).

This line leaves the procedure if no message is left (if a null is returned).

This line returns the ASURXSQL error message to the SYSTSPRT DD. For more information about ASURXSQL error messages, see “ASURXSQL external function” on page 730.

This line ends SQL error processing.
The cleanup procedure

Lines 144 through 147 of Figure 294 are the cleanup procedure to terminate the report, disconnect from DB2, and issue a return code.

Figure 294 Invoking the cleanup procedure

```
144 cleanup:
145  rcx = asurxrpt("TERM");     /* terminate report */
146  rcx = asurxsql("DISCONNECT");  /* disconnect from db2 */
147  exit rc;
```

- line 144 This line marks the beginning of the cleanup procedure.
- line 145 This line releases resources obtained during initialization and writing.
- line 146 This line frees all storage that ASURXSQL obtains, closes the plan, disconnects from the DB2 subsystem, and deletes ASURXSQL from memory.
- line 147 This line exits the program and sets the return code.

JCL and execution

The JCL in Figure 295 executes program IKJEFT01 to interpret the REXX program which is used to create reports. The lines are numbered for easy reference.

Figure 295 JCL for customized REXX report

```
01 //JOBNAME   JOB (acct),'ASURXJCL',MSGCLASS=X,CLASS=A,NOTIFY=userid
02 //ASURXJCL EXEC PGM=IKJEFT01,PARM='REPORT SSID PLAN ...'
03 // *--------------------------------------------------------------------
04 // PARM 1.REPORT = NAME OF REPORT TO EXECUTE
05 // 2.SSID = DB2 SUBSYSTEM ID OR DATA SHARING GROUP NAME
06 // 3.PLAN = ASU PLAN NAME
07 // ... REPORT DEPENDENT PARAMETER(S)
08 // *--------------------------------------------------------------------
09 // STEPLIB DD DISP=SHR,DSN=HLQ.ASU810.D71.LOAD
10 // SYSEXEC DD DISP=SHR,DSN=DB2*****.DSNLOAD
11 // SYSEXEC DD DISP=SHR,DSN=HLQ.ASU810.CLIST
12 // SYSPRINT DD SYSOUT=*
13 // SYSTEM DD SYSOUT=*
14 // SYSTSPRT DD SYSOUT=*
15 // SYSTSIN DD DUMMY
```
line 2  This line runs the IKJEFT01 program to run the REXX program and supplies PARM information (Table 108). In the annotated program, the report-dependent parameter was a database name that contained a wildcard.

lines 3-8  These lines are comments.

lines 4-6  These lines identify the required step libraries.

line 11  This line specifies the partitioned data set (PDS) library that contains the REXX report program. The member name of the report program (TSSPACE) is the REPORT parameter in line 2.

lines 12-13  These lines identify the data sets that contain the report and the error messages that the ASURXRPT and ASURXSQ SQL functions generate.

line 12  This line identifies the data set that contains the report.

line 13  This line identifies the data set that contains error messages from the functions ASURXRPT and ASURXSQ SQL and the SAS/C runtime library.

line 14  This line identifies the data set that contains SAY output of return codes and error messages (lines 113, 114, 125, 136, and 141).

Sample report

The annotated report program is shown in Figure 296.

Figure 296  A sample customizable report  (part 1 of 2)
Complete sample program

The complete report program annotated in this appendix is in Figure 297.

Figure 297 Complete sample program (part 1 of 4)

/* rexx *******************************************************************/
/* SYNOPSIS */
/* TSPACE - Space utilization for DB2 tablespaces */
/* DESCRIPTION */
/* A report by tablespace of space allocated and number of */
/* extents. */
/* USAGE NOTES */
/*Parms */
/* ssid plan tbname-prefix dbname */
/* */
/* The following statement should specify the table name of the */
/* DASD MANAGER PLUS tablepart table */

dmtablepart = asualias('BMCASU_TABLEPART');

title = "DASD SPACE FOR TABLESPACE":

/* tablepart format definition */

tpfd. =
tpfd.1 = " @<<<<<<<<<<<<<<<<< @>>> @>>>>>>>>  @>>>";
tpfd.2 = " tsobj tp.partition tp.space tp.extents";
tpfd.3 = ". ";
/* end of format definition */
/* tablepart top of page format definition */
tpfd.top.1 = ":";

Figure 296 A sample customizable report (part 2 of 2)
Figure 297  Complete sample program (part 2 of 4)

tpf.d.top.2 = " @|@@@@|@@@@|@@@@|@@@@|@@@@|@@@@|@@@@|@@@@|@@@@|@@@@|@@@@|PAGE@@@@@@@@@@@@@@":
tpf.d.top.3 = " title rpt.page# ":
tpf.d.top.4 = " ":
tpf.d.top.5 = " TABLESPACE PART SPACE EXT":
tpf.d.top.6 = " ----------------- ----- ---------  ----":
tpf.d.top.7 = " . ": /* end of format definition */

parse upper arg ssid plan dbnamex . /* get input data */

rc = asurxsql("CONNECT" ssid plan); /* connect and open plan */

if (rc ¬= 0) then /* if connect or open plan error */
signal caferror; /* go output caf error message */

rc = asurxrpt("INIT 60 suppress tpfd"); /* initialize report */

if (rc ¬= 0) then /* if an error */
signal rpterror; /* go output report error msg */

dbnamex = translate(dbnamex, ",", ":"); /* change * to % in dbname */
eqorlike = ":-": /* default to equal */

if (0 < pos("", dbnamex) | , /* if wildcard in dbname */
0 < pos("_", dbnamex)) then

eqorlike = ":LIKE": /* use like */

/* obtain tablepart data */

rc = asurxsql("OPEN " , /* open cursor */
"TP CURSOR FOR SELECT ",
"A.DBNAME ,",
"A.TSNAME ,",
"A.PARTITION ,",
"A.SPACE ,",
"A.EXTENTS ,", "FROM " dmttablepart "A ",
"WHERE A.DBNAME" eqorlike ""dbnamex"" ,
" AND TIMESTMP = (SELECT MAX(TIMESTMP) ) ,
" FROM " dmttablepart "B ",
" WHERE B.DBNAME = A.DBNAME ",
" AND B.TSNAME = A.TSNAME ",
" AND B.PARTITION = A.PARTITION) .
"ORDER BY DBNAME ,",
" TSNAME ,",
" PARTITION ",
"FOR FETCH ONLY ");

if (rc ¬= 0) then /* if open failed */
signal error; /* go output sql error message */

do forever /* loop for all tablepart rows */
rc = asurxsql("FETCH TP"); /* fetch next row */
if (rc != 0) then /* if fetch failed */
signal sqlerror; /* go output sql error message */
if (sqlcode = 100) then /* if end of data */
leave; /* leave forever loop */
dbn = strip(tp.dbname, 'T'); /* remove trailing blanks */
ts = strip(tp.tname, 'T');
tsobj = dbn"."tsn; /* dbname.tname */
rc = asurxrpt("WRITE tpfd"); /* output tablepart data */
if (rc != 0) then /* if an error */
signal rterror; /* go output report error message */
end /* end do forever */
rcx = asurxsql("CLOSE TP"); /* close cursor */
signal cleanup; /* cleanup */

/*---------------------------------------------*/

rterror - output error code and message from asurxrpt

---------------------------------------------*/

rterror:
say "ASURXRPT return code = "rptcode; /* error code */
say rpterrm; /* output report error message */
signal cleanup; /* cleanup */

/*---------------------------------------------*/

cafererror - output caf error msg

---------------------------------------------*/

ciafererror:
say 'Connect failed, rc = 'rc', cafreason = 'cafreason;
exit 8;

/*---------------------------------------------*/

sqlerror - output sql error codes and messages

---------------------------------------------*/

sqlerror:
say "ASURXSQL return code = "rc;
if sqlcode != 0 then do i = 1 to 1000 by 80
### Figure 297  Complete sample program (part 4 of 4)

```sql
m = substr(sqlerrm,i+1,79)
if m = ' ' then leave
say m
end

cleanup:
rcx = asurxrpt("TERM"); /* terminate report */
rcx = asurxsql("DISCONNECT"); /* disconnect from db2 */
exit rc;
```
Example of utility automation

This appendix contains the following topics:

Overview ................................................................. 761
   BMCSTATS utility ...................................................... 761
   About this example .................................................. 762
   Modifying the sample files ......................................... 762
   Adding corrective action definitions ............................... 763
   Modifying thresholds and corrective actions .................... 763
   Modifying the default corrective actions ......................... 764
   Running the BMCSTATS utility ................................... 764
   Modifying the BMCTRIG JCL ...................................... 765
   Executing or scheduling the BMCTRIG job .................... 766
   Scheduling the BMCTRIG generated jobs ....................... 766

Overview

Running reorganizations and copies on a scheduled basis instead of waiting until an object exceeds site guidelines can be costly in terms of wasted CPU cycles and work hours. DASD MANAGER PLUS allows you to test for user-defined thresholds and, if exceptions occur, to take user-defined corrective action.

BMCSTATS utility

The BMCSTATS utility collects the same statistics as RUNSTATS (plus some additional statistics) and writes the data to the DB2 catalog and to a historical database. The historical database allows you to display statistics for the first, last, and previous times that you ran BMCSTATS on an object, analyze trends via reports and graphs, and perform “what if” space estimation. You can use the additional statistics that BMCSTATS collects to determine when to run maintenance utilities on the physical objects.
For example, BMCSTATS calculates the REORGNLEVELS (the number of index levels that will be necessary if you reorganize the object). By comparing REORGNLEVELS with the current levels, you can determine whether reorganizing the index space will reduce the number of levels that the index requires.

Also, BMCSTATS calculates REORGSPACE, which identifies the amount of space an object will consume if free space is reestablished (that is, if the object is redefined and reorganized). When the REORGSPACE of an object is greater than the current allocation, the reserved free space in an object has been used. Inserting new rows into that object causes page splits, fragmentation, and rapid disorganization. Resizing and reorganizing the object reestablishes the free space (as determined by the FREEPAGE and PCTFREE parameters on the table space or index space partition). Keeping the object tuned in this manner helps to ensure that application performance is not degraded. You can set this ratio as a threshold in the BMCTRIG utility.

About this example

In this example, BMCTRIG evaluates thresholds and exceptions for all objects based on BMCSTATS data and generates corrective actions for the selected objects. A corrective action is a copy, a reorganization, or a statistics-collection job step.

To set up the automation, you modify the corrective actions and optionally modify the threshold values. You then run BMCTRIG, which checks for objects that meet or exceed the designated thresholds and identifies those objects as exceptions. BMCTRIG automatically generates copy, reorganization, or statistics job steps for each object and action that was identified. BMCTRIG balances the workload across the number of jobs that the NUMJOBS option specifies. If you choose to assign priorities to objects, actions, or thresholds, BMCTRIG generates work in priority order. You can then use a scheduler to run the generated jobs, or you can run them manually.

Modifying the sample files

BMC provides sample jobs in the .CNTL data set for this example. To modify the samples, you will need to know the following information:

- DASD MANAGER PLUS installed library names
- name of your ISPF library
- DB2 subsystem ID or group attach name
- DB2 load and exit library names
- a PDS to hold the output JCL that BMCTRIG generates
- a PDS to hold the worklists that BMCTRIG generates
Adding corrective action definitions

The ASUINSCA member inserts rows into the DASD MANAGER PLUS control tables to identify corrective actions for taking copies, statistics, and reorganizations. This member uses the default actions which are provided during installation:

- BMC_IBMCOPY_DEFAULT
- BMC_BMCSTATS_DEFAULT
- BMC_BMCREORG_DEFAULT

The ASUINSCA member associates installed default actions to the default exception thresholds that were added during product installation.

To add corrective action definitions

1. Edit the ASUINSCA member to execute it on your subsystem by using DSNTEP2. As an alternative, you can use the SQL with any other software product that executes SQL.

2. Modify the creator name used with the DASD MANAGER PLUS tables to match your site’s installation.

3. Execute the SQL to add the rows.

Modifying thresholds and corrective actions

Optionally, you can set different thresholds for each object (by name, pattern, or object set) in your subsystem. You can also assign a priority to specific thresholds to indicate the severity of the condition. When encountering an exception to any of these thresholds, DASD MANAGER PLUS automatically creates maintenance jobs to perform the corrective actions that you specify, based on the condition of your DB2 objects.

Using the DASD MANAGER PLUS online interface, view and modify (if needed) the installed thresholds and corrective actions. For more information about modifying thresholds and corrective actions, see Chapter 8, “Analyzing objects by using BMCTRIG.”
Modifying the default corrective actions

1 (Optional) Using the DASD MANAGER PLUS online interface, update the default actions named in “Adding corrective action definitions” on page 763 to select utility options for your site.

Typically, you will add a service syntax (a named set of syntax options that are stored in the DASD MANAGER PLUS control tables) for each action to select your syntax options and data set allocation information. If you do not set any options, DASD MANAGER PLUS uses internal defaults and the Product Options File (POF), which was created during installation, to provide default syntax and JCL generation defaults.

2 Add syntax to BMC_BMCSTATS_DEFAULT for the BMCSTATS service to update the DB2 catalog (since that is not the default).

3 (Optional) Add other services (utilities or job steps) to the actions.

For example, you could add a QUIESCE or a BMCCOPY before the BMCREORG in the BMC_BMCREORG_DEFAULT.

For more information about maintaining actions, see Chapter 4, “Maintaining and generating actions.”

Running the BMCSTATS utility

In this procedure, you will run the BMCSTATS utility to populate the DASD MANAGER PLUS database with statistics. To do so, you will use the ASUSSYST sample JCL. This procedure collects current statistics for evaluation by BMCTTRIG.

To populate the DASD MANAGER PLUS database with statistics

1 Modify the members to include the DB2 and DASD MANAGER PLUS data set names that are installed at your site.

2 Modify the jobcard to meet your installation standards.

3 Change the parameters for ASUSMAIN to name your DB2 subsystem and the DASD MANAGER PLUS installation options module.

4 If you do not want to run against all objects on the subsystem at once, change the %.% pattern in the SYSIN DD syntax to limit the selected objects.
Modifying the BMCTRIG JCL

The ASUTSYST member in the CNTL library is a sample BMCTRIG job that evaluates all objects in your DB2 subsystem. ASUTSYST then applies the system-defined thresholds and corrective actions to generate jobs that perform those actions.

To modify the BMCTRIG JCL

1. Modify the members to include the DB2 and DASD MANAGER PLUS data set names that are installed at your site.

2. Modify the jobcard to meet your installation standards.

3. Change the parameters for ASUSTRIG to name your DB2 subsystem and the DASD MANAGER PLUS installation options module.

4. If you do not want to run against all objects on the subsystem at once, change the %.% pattern in the SYSIN DD syntax to limit the selected objects.

                        NOTE

If there are more than 100,000 objects or partitions in your subsystem, BMC recommends not using %.%.

Alternatively, you can define object sets if you require more complex selection criteria at your site. For more information on creating object sets, see Chapter 5, “Using object sets.”

For more information about setting up BMCTRIG jobs, see the DASD MANAGER PLUS for DB2 Reference Manual.
Executing or scheduling the BMCTRIG job

You can execute or schedule the BMCTRIG job based on your site’s requirements. BMCTRIG checks for objects that meet or exceed the thresholds that you have defined, and generates a copy, reorganization, or statistics job step for each identified object and action. BMCTRIG balances the workload across the number of jobs that the NUMJOBS option specifies. If you choose to assign priorities to objects, actions, or thresholds, BMCTRIG generates work in priority order.

Scheduling the BMCTRIG generated jobs

Schedule the BMCTRIG generated jobs based on your site’s requirements. To schedule the execution of the generated JCL, use a scheduler or the BMCTRIG AUTOSUBMIT option to submit the generated JCL automatically.

For more information about the additional options for executing the corrective actions that BMCTRIG identified, see Chapter 8, “Analyzing objects by using BMCTRIG.”


Using the UIM Server

This appendix contains the following topics:

Overview ................................................................. 767
Starting and stopping the UIM server ................................ 767
Verifying that the UIM server is running .............................. 768
Accessing the BMC UIM server Commands web page .......... 769
Viewing active users ....................................................... 771
Changing the security authorization timeout feature temporarily . 771
Enabling or disabling the overall tracing option temporarily . 772
Enabling or disabling specific tracing options temporarily ........ 773
Refreshing the UIM server content ..................................... 774
Resolving UIM Server problems ....................................... 775

Overview

You can control the operation of the UIM server through a web browser. If the UIM server is not running, you cannot run the Export utility. From a web browser, you can start and stop the UIM server, view the active users, change the security authorization feature, and refresh the content of the UIM server. You can also change the status of the response logs and the internal trace.

Starting and stopping the UIM server

Before you use Export, the UIM server must be running. If the UIM server is not running, you cannot run Export. In addition, you cannot access any UIM commands through a web browser.
Verifying that the UIM server is running

To start and stop the UIM server, you must issue IBM MVS™ operator commands on the host that the UIM server is installed on.

**WARNING**

To avoid data loss, notify active users when you must stop the UIM server.

**To start the UIM server**

To start the UIM server, issue the following MVS operator command:

*S uimServerName*

*uimServerName* is the name of the UIM server started task.

**To stop the UIM server**

To stop the UIM server, issue the following MVS operator command:

*P uimServerName*

*uimServerName* is the name of the UIM server started task.

**Verifying that the UIM server is running**

You can verify whether the UIM server is running by reviewing the JESMSGLG SYSOUT file.

**To verify that the UIM server is running**

To verify that the UIM server is running, review the JESMSGLG SYSOUT file for the following messages:

- BMC340290I UIM server, Level V.R.MM MM,DD,YY, initialization complete!
- BMC340122I Ready for MVS Operator Commands
Accessing the BMC UIM server Commands web page

All UIM server operation tasks are performed from the BMC UIM server Commands web page. The UIM server must be started before you can access this web page.

Authorization settings

Before you can access the BMC UIM server Commands web page, the variable ALLOW_NETCMD must be set in the UIM startup member. For information about this variable, see the information about enabling the network browser command interface in your configuration documentation.

Table 112  ALLOW_NETCMD variable settings

<table>
<thead>
<tr>
<th>ALLOW_NETCMD setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>The BMC UIM server Commands web page is disabled.</td>
</tr>
<tr>
<td>YES</td>
<td>The BMC UIM server Commands web page is enabled.</td>
</tr>
<tr>
<td>AUTH</td>
<td>You must log into the UIM server Logon web page to verify that you are allowed to access the BMC UIM server Commands web page. If the logon is successful, the user can display the BMC UIM server Commands web page.</td>
</tr>
</tbody>
</table>

If the AUTH setting has been set in the UIM startup member, and you are not authorized to access the BMC UIM server Commands web page, you will see the following message in your browser when you try to access the web page:

403 - Access to URN not allowed

If you see this message, you must follow the steps in “To access the BMC UIM server Commands web page using authorization” on page 770.

To access the BMC UIM server Commands web page

From a supported web browser, enter the following URL for the BMC UIM server Commands web page:

http://uimServerHostName:uimPortNumber/htpcmd.html
The variables in the URL are defined as follows:

- `uimServerHostName` is the name of the host computer on which the UIM server is installed.

- `uimPortNumber` is the port number that is assigned to the UIM server. The default is 9999.

For example: `http://syso:9999/htpcmd.html`

**To access the BMC UIM server Commands web page using authorization**

1. From a supported web browser, enter the following URL for the UIM server Logon web page:
   
   `http://uimServerHostName:uimPortNumber/UIMLogon`

   The variables in the URL are defined as follows:

   - `uimServerHostName` is the name of the host computer on which the UIM server is installed.

   - `uimPortNumber` is the port number that is assigned to the UIM server. The default is 9999.

   For example: `http://syso:9999/UIMLogon`

2. Enter your user ID and password.

3. Optional. Enter a group identification and account.

   **NOTE**

   You can change the password on the UIM host by entering a new password in the `New Password` field.

4. Click OK.

   The message Logon successful is displayed in your web browser. The authentication for the UIM server is stored in a cookie in your web browser.

5. Enter the following URL for the BMC UIM server Commands web page:

   `http://uimServerHostName:uimPortNumber/htpcmd.html`
To stop access to the BMC UIM server Commands web page, enter the following URL:

http://uimServerHostName:uimPortNumber/UIMLogoff

**Viewing active users**

Active users are any users who are logged on to the UIM server.

**To view the names of active users**

1. Access the BMC UIM server Commands web page as described in “Accessing the BMC UIM server Commands web page” on page 769.

2. Click Display Active Users.

   The UIM server Active User Display page is displayed, and the names of the active users are listed on the page.

3. Return to the BMC UIM server Commands web page by clicking Back to UIM server Commands.

**Changing the security authorization timeout feature temporarily**

You can change the security authorization timeout feature temporarily for servers that communicate with the UIM server. The change remains effective until the UIM server is stopped.

**NOTE**

Performing this task changes the timeout feature temporarily for all servers that communicate with the UIM server.

**To change the timeout feature temporarily**

1. Access the BMC UIM server Commands web page as described in “Accessing the BMC UIM server Commands web page” on page 769.
Enabling or disabling the overall tracing option temporarily

2 Click Display Active Users.

The UIM server Active User Display page is displayed.

3 In the Inactivity Time Out box, type a value (in minutes or seconds) representing the amount of time that the server can remain inactive without timing out.

4 Choose a unit of time for the server timeout feature by selecting the option button for Seconds or Minutes.

5 Apply your changes by clicking Change.

6 Return to the BMC UIM server Commands web page by clicking Back to UIM server Commands.

Enabling or disabling the overall tracing option temporarily

You can enable or disable the overall tracing option temporarily for the UIM server. The change remains effective until the UIM server is stopped or until you change it.

Trace data is written to the TRCLOGDD. To view trace data, review the contents of the TRCLOGDD.

To enable or disable the overall tracing option temporarily

1 Access the BMC UIM server Commands web page as described in “Accessing the BMC UIM server Commands web page” on page 769.

2 On the BMC HTTP Server Commands Web page, click Internal Trace - Display Trace Status/Modification.

The Trace Indicator Status page is displayed. The status of the tracing option is shown in the Overall Trace Indicators section.

3 In the Overall Trace Indicators section, select either Active or Inactive.

- If the overall tracing option is disabled and you want to enable it, click Inactive in the Overall Trace column.

Inactive changes to Active, indicating that the overall tracing option has been enabled for the UIM server.
Enabling or disabling specific tracing options temporarily

- If the overall tracing option is enabled and you want to disable it, click **Active** in the Overall Trace column.

  **Active** changes to **Inactive**, indicating that the overall tracing option has been disabled for the UIM server.

4 Return to the BMC HTTP Server Commands Web page by clicking **Back to HTTP Server Commands**.

---

**Enabling or disabling specific tracing options temporarily**

You can enable or disable specific tracing options temporarily for the UIM server. The change remains effective until the UIM server is stopped or until you change it.

Trace data is written to the TRCLOGDD. To view trace data, review the contents of the TRCLOGDD.

**To enable or disable specific tracing options temporarily**

1 Access the BMC UIM server Commands web page as described in “Accessing the BMC UIM server Commands web page” on page 769.

2 On the BMC HTTP Server Commands Web page, click **Internal Trace - Display Trace Status/Modification**.

   The Trace Indicator Status page is displayed. The status of the tracing option is shown in the Overall Trace Indicators section.

3 In the Overall Trace Indicators section, ensure that the tracing option is set to **Active**.

   If the overall tracing option is set to Inactive, click **Inactive** in the Overall Trace column. **Inactive** changes to **Active**, indicating that the overall tracing option has been enabled for the UIM server.

   **NOTE**

   The overall trace option must be Active before you can enable or disable specific tracing options.

4 In the Trace Components Indicators section, find the category for the specific tracing option that you want to enable or disable:

   - trace components
   - trace actions
5 To enable or disable specific tracing options, perform one of the following tasks:

- If tracing is disabled for a specific option and you want to enable it, click **Off** in the **Status** column.

  **Off** changes to **On**, indicating that tracing has been enabled for that option.

- If tracing is enabled for a specific option and you want to disable it, click **On** in the **Status** column.

  **On** changes to **Off**, indicating that tracing has been disabled for that option.

6 Return to the BMC HTTP Server Commands Web page by clicking **Back to HTTP Server Commands**.

---

**Refreshing the UIM server content**

You can refresh the content of the UIM server without shutting it down. You may need to refresh the content of the UIM server when a new version is installed.

**To refresh UIM server content**

1 Access the BMC UIM server Commands web page as described in “Accessing the BMC UIM server Commands web page” on page 769.

2 Click **Display/Refresh Contents Directory**.

   The MVS Content Index page is displayed.

3 On the MVS Content Index page, click **Refresh Content Index**.

4 Return to the BMC UIM server Commands web page by clicking **Back to UIM server Commands**.
Resolving UIM Server problems

This section discusses problems that pertain to the UIM Server and how to resolve them.

If you contact BMC Software Customer Support, you might be asked to enable or disable overall tracing options or just specific tracing options. For information about enabling and disabling tracing options, see, “Enabling or disabling specific tracing options temporarily” on page 773.

The UIM server does not start because the OMVS segment is not defined

If the started task is missing a RACF OMVS segment, one of the following messages is displayed in the job log output when the UIM server start command is issued:

- message IEF695I, stating that the job name is assigned to the specified user
- message ICH408I, stating that the OMVS segment is not defined

In addition, message LSCX902 is displayed in the SYSTERM DD output and states that an MVS initialization error occurred.

To solve this problem, include a RACF OMVS segment for the UIM server address space.

To determine whether a started task is missing a RACF OMVS segment

1. Check the JES Message log for the following messages:

   - IEF695I START RGSTEMP WITH JOBNAME RGSTEMP IS ASSIGNED TO USER ++++++++  
   - ICH408I JOB(RGSTEMP ) STEP(RGSTEMP ) CL(PROCESS ) OMVS SEGMENT NOT DEFINED

2. Check the SYSTERM DD output for the following message:

   LSCX902 ***** WARNING ***** ERRNO = EMVSINITIAL  
   Generated in PFSCTL called from line . . .

   If you find the preceding message, one of the following conditions exists:

   - No RACF rule has assigned a user to the started task.
   - A RACF rule has assigned a user to the started task, but the user does not have an associated OMVS segment.
Resolving UIM Server problems

The UIM server uses TCP/IP. TCP/IP requires UNIX system services, and a RACF OMVS segment must exist for the UIM server address space. To define an OMVS segment for a user of the started tasks, contact your security administrator.

**UIM server does not start and displays message LSCX902 in SYSTERM DD**

If the TCP/IP started task is not named TCPIP, when you issued the command to start the UIM server, message LSCX902 is displayed in the SYSTERM DD output and states that an error occurred.

To solve this problem, change the name of the TCP/IP started task name to TCPIP. The UIM server must be able to find the TCP/IP address space and tries to find the default name TCPIP, but it cannot because one of the following conditions exists:

- No TCP/IP address space with the started task name TCPIP exists because the address space has been given another name.
- Multiple TCP/IP address spaces exist, but none are named TCPIP.
- Multiple TCP/IP address spaces exist, but you do not want to use the started task named TCPIP.

To change the name of the TCP/IP started task name to TCPIP, perform one of the following tasks:

- Determine the name of the TCPIP file (TCPIP.DATA). This file is used by TCP/IP client address spaces to determine local TCP/IP configuration information.

  The local TCP/IP configuration information includes the name of the TCP/IP address space that is specified by the TCPIPUSERID parameter. The UIM server procedure must be updated to include a SYSTCPD DD statement that specifies the appropriate TCPIP.DATA file.

- Specify the TCP/IP address space name in a parameter to the UIM server by using the procedure parameter ENV as follows:

```bash
// ENV="'=TCPIP_MACH=startedTaskName"
```

**UIM server displays message LSCX902 in SYSTEM DD**

When Interlink or CA TCPAccess is installed instead of IBM TCP/IP Stack, and the command to start the UIM server is issued, message LSCX902 is displayed in the SYSTERM DD output and states that an error occurred.
To solve this problem, insert the TCPAccess LINKLIB as the first data set in the UIM server STEPLIB DD. The TCPAccess LINKLIB must be the first data set in the STEPLIB concatenation. Inserting the LINKLIB first provides the correct socket API interface modules.

**WARNING**

Insert only the LINKLIB library, not the LOAD library, because TCPAccess uses SAS/C. Using TCPAccess SAS/C may cause an incompatibility in the run-time modules.

---

**UIM server displays message LSCX904 in SYSTERM DD**

When the TCPIP PROFILE file specifies a reserved port number for a specific started task name that does not match the UIM server started task name, and the command to start the UIM server is issued, the message LSCX904 is displayed and states that an access error occurred.

This error occurs because the port number that is specified for the UIM server in the TCPIP PROFILE file is reserved for a job name that the UIM server is not executing under.

To solve this problem, specify another port number for the UIM server, or correct the started task name.

**To specify a different port number in the TCIP PROFILE file**

1. Review the SYSTERM DD output for the following message:

   | LSCX904 ***** WARNING ***** ERRNO = EACCES |
   | Generated in BIND called from line |
   | Unexpected failure in bind, reason code 744C7246 |

   If the message is in the SYSTEM DD output, the wrong name job name was used in the TCPIP PROFILE file.

2. Select another port number for the UIM server.

   Here is a sample entry in the TCPIP PROFILE file:

   ```
   8300 TCP STFTUIM ; BMC UIM server
   ```

   This entry reserves the port number 8300 for the job named STFTUIM that is associated with the UIM server, and any other job trying to access that port number receives the EACCES error in the SYSTERM DD output.
Obtaining enhanced database performance evaluation using SQL Performance for DB2

**NOTE**

To use this feature, you must have version 6.2.00 or later of the SQL Performance for DB2 product.

This appendix contains the following topics:

- Overview .......................................................... 779
- Reorganizations based on the candidate table only .......... 780
- Reorganizations based on the candidate table and BMCTRIG evaluations .................................................. 781
- How Performance Advisor works ........................................ 781
- Preparing to use REORG Advisor .................................. 782
- Populating the REORG candidate table ......................... 783
- Making BIND modifications ........................................ 783
- Obtaining sample JCL files ........................................ 784
- Setting up automated reorganizations ............................... 785

**Overview**

By combining native DASD MANAGER PLUS functionality with a feature in the SQL Performance for DB2 solution, you can obtain more robust database performance evaluations. This feature, Performance Advisor, is available in version 6.2.00 or later of SQL Performance for DB2.
Reorganizations based on the candidate table only

DASD MANAGER PLUS tests for user-defined thresholds and, if exceptions occur, performs user-defined corrective actions (such as reorganizations). For example, with the BMCTRIG utility of DASD MANAGER PLUS, you can monitor changes in your database statistics. BMCTRIG reports significant changes (exceptions) and automatically creates maintenance jobs (such as reorganizations) based on objects that exceed the thresholds that you defined. BMCTRIG then performs the actions that you specify based on the condition of your DB2 objects.

With this feature, you can now use the Performance Advisor tool from SQL Performance to generate specific reorganization recommendations based on performance metrics. From Performance Advisor, you run the REORG Advisor (REXX EXEC) tool to generate the metrics. The resulting REORG candidate table includes a column that identifies objects whose performance might improve if a reorganization is run. DASD MANAGER PLUS can then use this table to help determine whether to run a reorganization.

BMCTRIG can generate reorganization jobs based on recommendations from the REORG candidate table only, or on recommendations from the candidate table and BMCTRIG’s evaluations.

Reorganizations based on the candidate table only

In this scenario, BMCTRIG generates a reorganization job based only on the recommended objects list from the REORG candidate table. The following steps represent a high-level summary of the process. For detailed instructions, see “Setting up automated reorganizations” on page 785.

1. You create the REORG candidate table.

2. You create an object set that identifies objects from the Performance Advisor REORG candidate table whose performance might improve if a reorganization is performed on them.

3. Change and execute the corrective actions.

4. Change the sample BMCTRIG JCL.

5. Schedule or manually run the BMCTRIG TRIGFORC job.

   TRIGFORC runs BMCTRIG against the object sets and automatically generates reorganization job steps for each object that was identified by the object set. BMCTRIG balances the workload across the number of jobs that the NUMJOBS option specifies.

6. Schedule the generated jobs or run them manually.
Reorganizations based on the candidate table and BMCTRIG evaluations

In this scenario, BMCTRIG generates a reorganization job only if the objects meet both criteria (recommended by the REORG candidate table and triggered by BMCTRIG-specified exception thresholds). The following steps represent a high-level summary of the process. For detailed instructions, see “Setting up automated reorganizations” on page 785.

1. Create the REORG candidate table.

2. Create an object set that identifies objects from the Performance Advisor REORG candidate table whose performance might improve if a reorganization is performed on them.

3. Create exceptions threshold values.

4. Define corrective actions and execute the job.

5. *(optional)* Change the corrective actions.

6. Change the sample BMCTRIG JCL.

7. Schedule or manually run the BMCTRIG TRIGEXCP job.

   TRIGEXCP runs BMCSTATS against the object set, uses exceptions to further evaluate each object identified by the object set to determine whether they exceed the thresholds values. It then automatically generates reorganization job steps for each object that meets the criteria of both the object set and the threshold definitions. BMCTRIG balances the workload across the number of jobs that the NUMJOBS option specifies.

8. Schedule the generated jobs or run them manually.

How Performance Advisor works

The Performance Advisor component of the SQL Performance solution provides the following features:

- identifies performance trends
- compares recent performance against a baseline
- identifies usage patterns
- generates recommendations for improved performance
Performance Advisor enables you to implement a database that can contain types of data such as the following:

- errors, exceptions, SQL text, and index usage patterns
- daily snapshots of real-time statistics (RTS) collected by DB2
- results of analyses that provide recommendations to improve your system and application performance

Performance Advisor’s REORG Advisor component provides the data that DASD MANAGER PLUS uses to help determine whether an object is a REORG candidate. REORG Advisor is a REXX EXEC program (included as LLQSAMP member IODADREO where LLQ is DB, XX, BB, and UBB). REORG Advisor identifies potential candidate objects for reorganization based on two criteria:

- the level of disorganization of the object
- the level of performance degradation for the object since the last REORG was run

REORG Advisor uses data from the daily object statistics table (DAILY_OBJ_STATISTICS). REORG Advisor maintains this data in the REORG candidate table which DASD MANAGER PLUS then accesses to help determine when to run a reorganization.

If you have version 6.2.00 or later of SQL Performance and you want to use the feature, complete the procedures provided in the following sections:

- “Preparing to use REORG Advisor”
- “Setting up automated reorganizations” on page 785

## Preparing to use REORG Advisor

This section explains how to prepare to use the REORG Advisor in your environment. Complete the procedures in this section to

- populate the REORG candidate table
- make BIND modifications if your SQL Performance solution uses a centralized database to collect data for multiple subsystems (using DDF) (page 783)
- obtain sample JCL files (page 784)
Populating the REORG candidate table

Generally, you need to perform the following steps to populate the REORG candidate table. For more information about a particular step, see the *SQL Performance for DB2 User Guide*.

**To populate the REORG candidate table**

1. Create the required tables at your centralized database location, or on each localized database location.

2. Ensure that all Data Collectors are configured properly.

3. Set up automation to capture an archive based on the previous day’s data.

4. Submit a daily job to maintain the object statistics table.

5. Configure REORG Advisor thresholds and options.

6. Submit a daily job to execute REORG Advisor, which populates the REORG candidate table.

**Making BIND modifications**

Complete this procedure *if your SQL Performance solution uses a centralized database to collect data for multiple subsystems (using DDF)*. The BIND modifications enable DASD MANAGER PLUS to retrieve data from the REORG candidate table.

**To make BIND modifications (for centralized databases only)**

1. On each subsystem where your DASD MANAGER PLUS tables exist, copy the BIND statement for the ASUQDYN package.

2. In the BIND statement, change `#AFDSSID` to the subsystem ID, connection ID, or group attach name of the subsystem where the REORG candidate table is installed.

   An example follows:

   ```
   BIND PACKAGE(#AFDSSID.ASU930_D_MAIN) MEMBER(ASUQDYN) +
   OWNER(owner) QUALIFIER(DFD411D) +
   ISOLATION(CS) CURRENTDATA(NO) +
   DYNAMICRULES(BIND) +
   VALIDATE(BIND) ENCODING(EBCDIC) RELEASE(COMMIT)
   ```
3 Run the package BIND.

4 To change the BIND statements for the DZ and DS plans, incorporate a wildcard as a high-level qualifier for the package list, as shown in the following example:

```
BIND PLAN(ASU930DZ)        OWNER(owner)     QUALIFIER(DFD430D)+
NODEFER(PREPARE)          VALIDATE(BIND)    ISOLATION(CS)+
CACHESIZE(1024)           CURRENTDATA(YES) DEGREE(1)+
SQLRULES(DB2)              ACQUIRE(USE)     RELEASE(COMMIT)+
EXPLAIN(NO)               DYNAMICRULES(BIND) REOPT(NONE)+
KEEPDYNAMIC(NO)           IMMEDWRITE(NO)    DBPROTOCOL(DRDA)+
ENCODING(37)              DISCONNECT(EXPLICIT)+
PKLIST(BMU430_D_MAIN.*,+  *.ASU930_D_MAIN.*,+
   ACS930_D_MAIN.*+      ATS011_D_MAIN.*+)
}

BIND PLAN(ASU930DS)        OWNER(owner)     QUALIFIER(DFD430D)+
NODEFER(PREPARE)          VALIDATE(BIND)    ISOLATION(CS)+
CACHESIZE(1024)           CURRENTDATA(YES) DEGREE(1)+
SQLRULES(DB2)              ACQUIRE(USE)     RELEASE(COMMIT)+
EXPLAIN(NO)               DYNAMICRULES(BIND) REOPT(NONE)+
KEEPDYNAMIC(NO)           IMMEDWRITE(NO)    DBPROTOCOL(DRDA)+
ENCODING(37)              DISCONNECT(EXPLICIT)+
PKLIST(BMU430_D_MAIN.*,+  *.ASU930_D_MAIN.*,+
   ACS930_D_MAIN.*+      ATS011_D_MAIN.*+)
}
```

5 Run the plan BIND.

**Obtaining sample JCL files**

SQL Performance for DB2 provides sample files that you can modify to enable the feature to work in your environment.

**To obtain the sample files**

1 Before proceeding, collect the following information:

- DASD MANAGER PLUS installed library names
- ISPF library name
Setting up automated reorganizations

After enabling REORG Advisor, complete whichever procedure you prefer for automating your reorganizations:

- “To automate reorganizations based only on REORG candidate table recommendations” on page 786
- “To automate reorganizations based on REORG candidate table recommendations and BMCTRIG evaluations” on page 788

**NOTE**
The sample files use the pound sign (#) to identify lines that require changes.
Before you begin

Complete the procedures in “Preparing to use REORG Advisor” on page 782.

To automate reorganizations based only on REORG candidate table recommendations

1 Ensure that the `reorg.analyze_physical_organization` configuration option in member IODPAREO (the REORG Advisor) is set to 1.

By using this default setting, the REORG Advisor reviews both physical and performance criteria.

2 Create the object set PUBLIC.AFD_REORG_CAND by modifying the SQL in the INSOBJJS member of the sample library as detailed in the instructions within the sample file. Following is a summary of the steps to edit this file.

**NOTE**
Object sets specify DB2 objects to include or exclude from action generation. Object sets use name patterns, dynamic SQL, and other methods to identify DB2 objects for processing. This step creates an object set that DASD MANAGER PLUS will use to query the REORG candidate table by populating the DASD MANAGER PLUS control tables.

A Edit the member to execute it on your subsystem using DSNTEP2. As an alternative, you can use the SQL with any other software product that executes SQL.

B In the INSERT statements, change the creator name for the DASD MANAGER PLUS tables to match your site’s installation. Also, change the subsystem ID where those tables are located.

C In the INSERT statements, change the creator and table name for the REORG candidate table. Also, change the subsystem ID where that table is located.

D Submit the job.

**NOTE**
For more information about maintaining object sets, see Chapter 5, “Using object sets.”
3 Change and execute the default corrective action:

A *(optional)* Using the DASD MANAGER PLUS online interface, update the sample action named BMC_BMCREORG_DEFAULT (installed with DASD MANAGER PLUS) to select utility options for your site.

**NOTE**
Typically, you add a service syntax (a named set of syntax options that are stored in the DASD MANAGER PLUS control tables) for each action to select your syntax options and data set allocation information. If you do not set any options, DASD MANAGER PLUS uses internal defaults and the product options file (POF), which was created during installation, to provide default syntax and JCL generation defaults.

B *(optional)* Add other services such as utilities or job steps to the actions.

For example, you could add a QUIESCE or a BMCCOPY before BMREORG in the BMC_BMCREORG_DEFAULT action.

**NOTE**
For more information about maintaining actions, see Chapter 4, “Maintaining and generating actions.”

4 Change the sample BMCTRIG JCL as detailed in the instructions within the sample file. Following is a summary of the steps to edit this file.

**NOTE**
TRIGFORC contains a sample job that runs BMCTRIG against the object sets (defined in step 2 on page 786) to generate reorganization actions based on the REORG candidate table. TRIGFORC uses PUBLIC.AFD_REORG_CAND to identify objects, and the default corrective action BMC_BMCREORG_DEFAULT to generate BMC REORG PLUS for DB2 jobs.

A Change the member to include the DB2 and DASD MANAGER PLUS data set names that are installed at your site.

B Change the job card to meet your installation standards.

C Change the parameters for ASUSTRIG to specify your DB2 subsystem and the DASD MANAGER PLUS installation options module.

D Change the BMCTRIG command syntax to identify the data sets that will be used to generate the BMC REORG PLUS jobs.

E Change the job card specifications for the generated jobs.
Setting up automated reorganizations

NOTE
For more information about setting up BMCTRIG jobs, see the DASD MANAGER PLUS for DB2 Reference Manual.

5 Execute or schedule TRIGFORC based on your site’s requirements.

BMCTRIG generates a reorganization job step for each object that the object set identifies. BMCTRIG balances the workload across the number of jobs that the NUMJOBS option specifies.

6 Schedule the BMCTRIG generated jobs based on your site’s requirements.

Use a scheduler or the BMCTRIG AUTOSUBMIT option to submit the generated JCL automatically.

NOTE
For more information about the additional options for executing the corrective actions that BMCTRIG identified, see Chapter 8, “Analyzing objects by using BMCTRIG.”

To automate reorganizations based on REORG candidate table recommendations and BMCTRIG evaluations

1 Ensure that the reorg.analyze_physical_organization configuration option in member IODPAREO (the REORG Advisor) is set to 0.

By default, the REORG Advisor reviews both physical and performance criteria. Setting the configuration option to 0 ensures that REORG Advisor reviews only performance criteria. This allows the DASD MANAGER PLUS BMCTRIG function to handle all real-time statistics (RTS) and other organization related evaluation.

2 Create the object set PUBLIC.AFD_REORG_CAND by using the SQL in theINSOBJJS member as detailed in the instructions within the sample file. Following is a summary of the steps to edit this file.

NOTE
Object sets specify DB2 objects to include or exclude from action generation. Object sets use name patterns, dynamic SQL, and other methods to identify DB2 objects for processing. This step creates an object set that DASD MANAGER PLUS will use to query the REORG candidate table by populating the DASD MANAGER PLUS control tables.
A Edit the INSOBJ$ member to execute it on your subsystem by using DSNTEP2. As an alternative, you can use the SQL with any other software product that executes SQL.

B In the INSERT statements, change the creator name for the DASD MANAGER PLUS tables to match your site’s installation. Also, change the subsystem ID where these tables are located.

C In the INSERT statements, change the creator and table name for the REORG candidate table. Also, change the subsystem ID where that table is located.

D Submit the job.

NOTE
For more information about maintaining object sets, see Chapter 5, “Using object sets.”

3 Change and execute the INSTHRSH member to insert thresholds that BMCTRIG will use to further evaluate the object set as detailed in the instructions within the sample file. Following is a summary of the steps to edit this file.

A Change the job card to meet your installation standards.

B Change #ASUSSID in SYSTSIN to the subsystem ID where your DASD MANAGER PLUS tables are located and where this job will be executed.

C Change the creator name used with the DASD MANAGER PLUS tables to match your site’s installation.

D Submit the job.

E (optional) Using the DASD MANAGER PLUS interface, change the threshold values, if needed.

The following table lists the thresholds and their values. For more information about these thresholds, see Chapter 8, “Analyzing objects by using BMCTRIG.”

<table>
<thead>
<tr>
<th>Threshold name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPNDINS</td>
<td>10</td>
</tr>
<tr>
<td>DSEXTENT</td>
<td>50</td>
</tr>
<tr>
<td>EXTENTS</td>
<td>50</td>
</tr>
<tr>
<td>FARIND</td>
<td>15</td>
</tr>
<tr>
<td>LEAFFOFF</td>
<td>10</td>
</tr>
<tr>
<td>LEAFTOFF</td>
<td>30</td>
</tr>
</tbody>
</table>
Setting up automated reorganizations

4 Change and execute the INSCACT member as detailed in the instructions within the sample file. Following is a summary of the steps to edit this file.

**NOTE**
The INSCACT member inserts rows into the DASD MANAGER PLUS control tables. These rows identify corrective actions for taking reorganizations using the default action, BMC_BMCREORG_DEFAULT, provided during installation.

The INSCACT member associates the BMC_BMCREORG_DEFAULT action to the exception thresholds that were added in step 3.

<table>
<thead>
<tr>
<th>Threshold name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSEUDODL</td>
<td>10</td>
</tr>
<tr>
<td>REORMDEL</td>
<td>1</td>
</tr>
<tr>
<td>REORMODS</td>
<td>10</td>
</tr>
<tr>
<td>REORPEND</td>
<td>1</td>
</tr>
<tr>
<td>REORSPAC</td>
<td>30</td>
</tr>
<tr>
<td>SPACE</td>
<td>30</td>
</tr>
<tr>
<td>TOTALIND</td>
<td>30</td>
</tr>
<tr>
<td>UNCLUST</td>
<td>10</td>
</tr>
</tbody>
</table>

A Change the job card to meet your installation standards.

B Change ASUSSID in SYSTSIN to the subsystem ID where your DASD MANAGER tables are located and where this job will be executed.

C Change the creator name used with the DASD MANAGER PLUS tables to match your site’s installation.

D Submit the job.

5 Change the default corrective action:

A *(optional)* Using the DASD MANAGER PLUS online interface, update the action named BMC_BMCREORG_DEFAULT (installed with DASD MANAGER PLUS) to select utility options for your site.

**NOTE**
Typically, you add a service syntax (a named set of syntax options that are stored in the DASD MANAGER PLUS control tables) for each action to select your syntax options and data set allocation information. If you do not set any options, DASD MANAGER PLUS uses internal defaults and the product options file (POF), which was created during installation, to provide default syntax and JCL generation defaults.
B (optional) Add other services such as utilities or job steps to the actions.

For example, you could add a QUIESCE or a BMCCOPY before BMCREORG in the BMC_BMCREORG_DEFAULT action.

---

**NOTE**

For more information about maintaining actions, see Chapter 4, “Maintaining and generating actions.”

---

6 Change the sample BMCTRIG JCL as detailed in the instructions within the sample file. Following is a summary of the steps to edit this file.

---

**NOTE**

TRIGEXCP contains a sample job that

- runs BMCSTATS against the object set (defined in step 2 on page 788)
- runs BMCTRIG to further evaluate the objects that are using the exceptions (defined in step 3 on page 789)
- generates reorganization jobs for objects that meet both the Performance Advisor and the BMCTRIG criteria

TRIGEXCP uses PUBLIC.AFD_REORG_CAND to identify objects, and the default corrective action BMC_BMCREORG_DEFAULT to generate BMC REORG PLUS for DB2 jobs.

---

A Change the member to include the DB2 and DASD MANAGER PLUS data set names that are installed at your site.

B Change the job card to meet your installation standards.

C Change the parameters for ASUSMAIN and ASUSTRIG to specify your DB2 subsystem and the DASD MANAGER PLUS installation options module.

D Change the BMCTRIG command syntax to identify the data sets that will be used to generate the BMC REORG PLUS jobs.

E Change the job card specifications for the generated jobs.

---

**NOTE**

For more information about setting up BMCTRIG jobs, see the DASD MANAGER PLUS for DB2 Reference Manual.
7 Execute or schedule TRIGEXCP based on your site’s requirements.

First, BMCTRIG checks for objects that meet or exceed the thresholds that you have defined in the object set. BMCTRIG then generates a reorganization job step for each identified object. BMCTRIG balances the workload across the number of jobs that the NUMJOBS option specifies.

8 Schedule the BMCTRIG generated jobs based on your site’s requirements.

To schedule the execution of the generated JCL, use a scheduler or the BMCTRIG AUTOSUBMIT option to submit the generated JCL automatically.

**NOTE**

For more information about the additional options for executing the corrective actions that BMCTRIG identified, see Chapter 8, “Analyzing objects by using BMCTRIG.”
Glossary

A

ACM
The code used to identify the CHANGE MANAGER product in BMC Software database naming conventions.

ACT
The code used to identify the CATALOG MANAGER product in BMC Software database naming conventions.

action
In DASD MANAGER PLUS, an action is the fundamental work unit for generating utilities. An action names a set of services. After you create an action, you can copy, edit, or delete it. See also corrective action.

action name
An action name is a string of up to eight alphanumeric characters excluding percent (%), asterisk (*), underscore (_), and space. When specifying an action name, you can use a wildcard pattern to display a group of similar names.

action owner
Authorization ID of the creator of the action.

action POF
A POF that can be written from the ISPF variables that are set in the product or edited. An action POF can be used to reset all of the options that will be used in the current session to create JCL.

action status
The Execution status of the action. DASD MANAGER PLUS sets the action status to N (not started) when the worklist is created. The Execution program updates the action status to S (started) if the worklist has started but not completed, or to R (rerunable) if the worklist has completed. If you submit a worklist for UNDO, the status becomes U. You can start over or restart a worklist that has a action status of S. You must rebuild a worklist having status U before the worklist can be resubmitted.

action type
The type of action: U for utility.
Administrative Products for DB2
A collection of products from BMC Software that includes ALTER, CATALOG MANAGER, and CHANGE MANAGER as well as 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 for OS/390 system. When all Administrative Products for DB2 are installed properly, they can access some of each other’s functionality. In addition, these products can access the IBM and BMC Software utilities. See also Utilities for DB2.

AEXPRINT
The name of the file of messages output by the Execution program. This file contains all output from the Execution process that includes DB2 for OS/390 utility messages, BMC Software utility messages, dynamic SQL messages, IDCAMS messages, and any other messages generated by the actions of the worklist. AEXPRINT is frequently referred to as the Worklist Execution Log.

AJX variables
A group of variables (JOB, STEP, and DD) that are used during 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 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 for your information in HLQ.SLIB($AJXDOCV).

AJXPOFVL
The ddname of the product options file (POF) validation report that the Batch JCL Generation component uses.

allocation unit
The unit used for space estimation calculations, and primary and secondary quantities. Possible values are K (kilobytes), T (tracks), or C (cylinders). The default value comes from the installation standards.

ALLOCUNIT
An attribute of a table partition or index partition indicating the allocation unit: C (cylinders) or T (tracks); reported by BMCSTATS and RUNSTATS. If the table space or index partition is defined as DEFINE NO, ALLOCUNIT is set to blank.

ALTDD
A BMCSTATS parameter that specifies whether to use an alternate data set, such as the name of a DSN1COPY data set, against which to collect statistics.

ALTER for DB2
One of the Administrative Products for DB2 developed by BMC Software. ALTER automates, copies, and migrates data structure changes within a single DB2 subsystem. This product supports restoration of data, dependencies, and authorizations. See also CHANGE MANAGER for DB2.
ALU
The code used to identify the ALTER product in BMC Software database naming conventions.

ARU
The code used to identify the REORG PLUS product in BMC Software database naming conventions.

ASCII
American National Standard Code for Information Interchange. One of three encoding schemes used for information interchange. ASCII uses a 7-bit coded character set to represent 128 letters, numbers, symbols, and control characters. See also EBCDIC and Unicode.

ASU
The code used to identify the DASD MANAGER PLUS product in BMC Software database naming conventions.

automation modes
Levels of automation that the Database Performance for DB2 solution provides to suit different user requirements. You choose the automation mode at installation, and you can change the mode to match your site’s changing needs. See also full automation mode and standard JCL mode.

automation spawner
The processing component of the Database Performance for DB2 solution that manages registered candidates, object processing, JCL generation, and job management. The automation spawner consists of the automation spawner address space and one or more automation spawner subtasks. The automation spawner address space functions as a control point for automation spawners that correspond to discrete instances of DB2. The automation spawner is the component responsible for all of the Database Performance for DB2 solution utility automation processing.

AutoSub
A BMCTRIG parameter that automatically submits utility JCL on an exception. The data set named in the Util DSN field will be submitted without modification.

B
BMCCOPY
A command that runs the BMC Software COPY PLUS utility to create an image copy. The short form of the command is BMCI.

BMCCPRS
A DASD MANAGER PLUS utility command that copies statistics from the DB2 catalog to the DASD MANAGER PLUS historical database. The short form of the command is BMCC.
BMCLOAD
A command that runs the BMC Software LOADPLUS utility to load DB2 tables. The short form of the command is BMCL.

BMCREORG
A command that runs the BMC Software REORG PLUS utility to perform a reorganization. The short form of the command is BMCR.

BMCSTATS
A DASD MANAGER PLUS utility command that runs the statistics collecting function. The short form of the command is BMCS. BMCSTATS is a functional replacement for the IBM RUNSTATS utility.

BMCTRIG
A DASD MANAGER PLUS program that evaluates exception thresholds and optionally generates corrective actions. Thresholds are user-defined limits that enable automatic utility generation. Thresholds are based on information from the statistics tables in the DASD MANAGER PLUS database, IDF catalog information, DB2 status, RTS data, and user-defined exceptions through REXX programs. Percentage increase thresholds are based on a comparison between the most current statistics run and the previous statistics run. Thresholds are based on a statistical limit. The short form of the command is BMCT.

BMCUNLOD
A command that runs the BMC Software UNLOAD PLUS utility to unload data from a full image copy or DSN1COPY of one or more tables in a table space. The short form of the command is BMCUN.

BMCUPRS
A DASD MANAGER PLUS utility command that updates the DB2 catalog statistics with BMCSTATS statistics (using SQL UPDATE). The short form of the command is BMCU.

box
A DASD MANAGER PLUS graphic display option that specifies whether to place a box around the legend of statistical graphs: KBOX (box) or NKBO (no box).

browse function
A DASD MANAGER PLUS function that enables you to display statistics from the DB2 catalog. This function is available on the DB2 Object List panel. Compare with display function.

Buffers
A BMCSTATS parameter that specifies the number of four-kilobyte I/O buffers each task in a multitasking job uses for reading data (2 through 999).

build JCL
A DASD MANAGER PLUS job generation option that instructs JCL Generation to build the JCL from the worklist.
build worklist
A DASD MANAGER PLUS job generation option that builds (or rebuilds) the worklist using the information specified for a particular action.

Catalog Indirection
An optional method of implementing the BMC Software Administrative Products for DB2 that allows them to access the DB2 catalog indirectly when making information queries. Catalog Indirection is accomplished by using synonyms that point either to a copy of the DB2 catalog or to user-created views of the catalog. Benefits include reducing catalog contention and providing an additional level of security for sensitive catalog data.

CATALOG MANAGER for DB2
One of the Administrative Products for DB2 developed by BMC Software. CATALOG MANAGER is a tool 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 and re-create dropped structures and data.

CHANGE MANAGER for DB2
One of the Administrative Products for DB2 developed by BMC Software. As a robust extension of the ALTER product, CHANGE MANAGER automates data structure changes across multiple DB2 subsystems and between DB2 and CASE 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 MANAGER can efficiently and effectively manage the DB2 structure change control process.

char H/W
A DASD MANAGER PLUS graphic display option for statistical graphs that specifies the character height relative to its width. For example, if you specify 200, the height will be multiplied by twice the width.

char size
A DASD MANAGER PLUS graphic display option that specifies the character size multiplier for statistical graphs. This multiplier is divided by 100 and the spacing sizes are multiplied by this factor.

checkpoint
A point at which information about the status of a job and the system can be recorded so that the service can later be restarted. See also sync point.

CHEK DA
A command that runs the IBM CHECKDATA utility. The short form of the command is CKDA.
CHEK LOB
A command that runs the IBM CHECK LOB utility. The short form of the command is CHEK LOB.

CLUSTERING
An attribute of an index indicating whether CLUSTER was specified when the index was created; reported by BMCSTATS and RUNSTATS.

color
A DASD MANAGER PLUS graphic display option that specifies the color of headers, titles, labels, legends and boxes. Specify the color by number according to the Color Key provided on the panel.

columns
A DASD MANAGER PLUS printing option that specifies the number of columns per row.

This is also a parameter used by BMCSTATS, BMCCPRS, and BMCUPRS to name the columns for which statistics are to be collected, copied, or updated. You can type column names in the format COL1, COL2, and so on; type ALL for all columns; or leave blank for none.

copies
A DASD MANAGER PLUS printing option that specifies the number of image copies to make.

COPY PLUS for DB2
One of the Backup and Recovery products from BMC Software. COPY PLUS produces up to four image COPYs or DSN1COPYs (full or incremental) in a single pass. COPY PLUS significantly reduces elapsed times, CPU cycles, and EXCPs.

CopyPend
An exception you can set using BMCTRIG to determine whether the copy pending flag is on.

corrective action
An action specifying a corrective action (such as REORG, COPY, and so on) to perform based the exceptions that are initiated by a BMCTRIG job. A corrective action is predefined, specifies no objects, and can be initiated when specific objects meet certain criteria. A skeleton specifies only utilities and commands and can be designed to suit multiple objects and situations. See also action.

D

DASD MANAGER PLUS database
The database provided and maintained by DASD MANAGER PLUS. The database is created during installation and named BMCASU. The DASD MANAGER PLUS database contains statistics tables (RS_%), utility job tables (UT_), worklist tables (WL_), and an action table (DO_WORKIDS).
DB2 catalog
   Tables maintained by DB2 that contain descriptions of DB2 objects such as tables, views, and indexes. With the DASD MANAGER PLUS user options, you can specify the location of the default DB2 subsystem.

DBNAME
   A statistic collected by BMCSTATS and RUNSTATS that specifies a database name.

default options module
   See installation options module.

default value
   A predetermined value, attribute, or option that is assumed when no other is explicitly specified.

DeleteAge
   A parameter used by BMCSTATS to delete statistics that have reached a certain age or by BMCTRIG to delete exceptions that have reached a certain age. Both deletions are made in the DASD MANAGER PLUS database.

device name
   A DASD MANAGER PLUS printing option that specifies the LUNAME of the printer.

device type
   The type of disk device used for DB2 data set allocation, such as 3330, 3340, 3350, 3375, 3380, and 3390 or generic types TAPE, CART.

display function
   A DASD MANAGER PLUS function that enables display of statistics from the DASD MANAGER PLUS database (BMCSTATS). This function is available on the DB2 Object List panel. See also browse function.

DOPTs
   See installation options module.

DSETPASS
   The data set password (if any) for the data sets making up the table space. This Virtual Storage Access Method (VSAM) master-level password is passed to VSAM when the data sets are used by DB2. If you leave this field blank, no password is assumed.

DSN1COPY
   A utility command that runs the IBM DSN1COPY utility. The short form of the command is DSN1.
edit JCL
A DASD MANAGER PLUS job generation option that displays the generated JCL for viewing or editing.

edit worklist
A DASD MANAGER PLUS job generation option that displays the generated worklist for viewing or editing.

ENCODING
An attribute of a table space that indicates the encoding scheme: E (EBCDIC), A (ASCII) or U (Unicode); BMCSTATS and RUNSTATS report the encoding scheme.

entry field delimiter
A user option that enables you to specify the highlighting to be used for the user input fields of DASD MANAGER PLUS panels (underscore, reverse video, blink, or no highlighting).

estimate source
A DASD MANAGER PLUS user option that enables you to specify the source to be used for estimating the size of the utility work data sets. Valid entries are N (no estimation), B (current BMCSTATS statistics), C (DB2 catalog statistics), or O (object sampling). See also space estimation function.

exception
A statistical value in the DASD MANAGER PLUS historical database that meets or exceeds a user-specified threshold value that has been set with BMCTRIG. Such exceptions are stored in the exceptions table, and they can be examined through ISPF dialogs (Exceptions report). A predefined corrective action can be generated automatically based on the identification of an exception.

exceptions report
A list of exceptions from the DASD MANAGER PLUS historical database. Each exception line includes the object name, type of exception, and a timestamp.

EXCEPTIONS2 table
A DASD MANAGER PLUS statistics table that stores exceptions identified by BMCTRIG. This table is named BMCATSnn.RS_EXCEPTIONS2.

events log
A collection of records that describe the events occurring during utility execution and their sequence. If you enable the Record Events option on the Action Generation panel, these records will be stored in the DASD MANAGER PLUS EVENTS table. Event records are useful for recovery in case of failure or for general analysis purposes.
EVENTS table
A DASD MANAGER PLUS statistics table that sequentially stores the events recorded during utility execution. One row is recorded for each utility execution event. This table is named BMCASUmEVENTS.

Execution
A BMC Software program that uses the worklist and generated JCL to run utility jobs. DASD MANAGER PLUS monitors JCL execution from the specified worklist and control cards.

F
FULLCOPY
A utility command that runs a full image copy using the IBM COPY utility. The short form of the command is FULL.

G
generation data group (GDG)
A collection of data sets kept in chronological order. Each data set is a generation data set.

H
historical database
A statistics database created by DASD MANAGER PLUS that contains the first, last, and current statistics collected by BMCSTATS for an object. You can display information from the historical database. BMCTRIG reads the historical database for statistics used during exception threshold evaluation.

I
Imagecopy
A BMCSTATS parameter that specifies whether to use the latest image copy as input for collecting statistics.

INCRCOPY
A utility command that runs an incremental image copy using the IBM COPY utility. The short form of the command is INCR.

Index
A set of pointers that are logically ordered by the values of a key. Indexes can provide faster access to data and can enforce uniqueness on the rows in a table.

The DASD MANAGER PLUS BMCSTATS utility provides an Index parameter that specifies whether to collect statistics on all indexes in a table space as well as the table space itself.

The BMCTRIG utility provides an Index exception that specifies whether to update the statistics on all indexes in a table space as well as the table space itself.
Index Cardinality
A DASD MANAGER PLUS statistical graph that plots the FULLKEY and FIRSTKEY cardinality of an index over time.

Index Pages
A DASD MANAGER PLUS statistical graph that plots the relationship over time between active pages, allocated pages, and pages required if the index were reorganized.

Index Partition Cardinality
A DASD MANAGER PLUS statistical graph that plots the number of rows of an index partition over time. The graph plots the number of NEAROFFPOS and FAROFFPOS references as well as the number of keys in relation to the number of rows.

Index Partition Extents
A DASD MANAGER PLUS statistical graph that plots the number of extents in an index partition over time. The maximum extents per data set allowed by VSAM is 7257.

Index Partition Leaf Distribution
A DASD MANAGER PLUS statistical graph that plots the leaf distribution in an index partition over time.

Index Partition Page Group
A DASD MANAGER PLUS statistical graph that plots page group information, specifically the distribution of data in an index partition. (To display this graph, BMCSTATS must have been run with a PAGEGROUP specification greater than zero.) With dual vertical axes, this graph shows how rows and keys are distributed in the data set, and how leaf and full pages are distributed based on the last BMCSTATS values.

Index Partition Pages
A DASD MANAGER PLUS statistical graph that plots the relationship between the number of allocated pages and the number of active pages in an index partition over time.

ISPF skeletons
DD statement templates used by JCL Generation. Descriptions of these can be found in HLQ.SLIB($AJXDOC).

Installation options module
A module that contains the option values that control the default operating environment.

J

JCL DSN
The name of the data set to contain the utility generation JCL. The data set must exist and can be partitioned or sequential. If partitioned, a member name must be specified. Symbolic variables can be used.
JCL Generation (JCLGEN)
A BMC Software program that enables you to generate from a worklist the JCL necessary to run a utility job. When you choose to build JCL, DASD MANAGER PLUS calls JCL Generation and passes the JCL to the worklist containing the control statements. DASD MANAGER PLUS resolves all data set names entered with symbolic variables on the interface panels. JCL Generation resolves all data sets passed from the option panels and the unload data sets used by the Execution facility. The generated JCL includes DD statements for all data sets needed by Execution, as well as the EXEC statement for the program and any necessary control parameters.

JCL variable display
A user option that includes debugging comments within any generated JCL. All AJX-type variables are displayed as /* comments in the JCL to assist in diagnosing JCL Generation problems.

JCLGEN
See JCL Generation.

job
A unit of work defined by an action and worklist to generate utilities. DASD MANAGER PLUS uses the information you supply for an action to generate the worklist, which provides JCL Generation with the information necessary to build the JCL to run a job.

job profile
The documentation describing a specific action. You can document the job title, comment, and scheduling information. Once the job has been generated, the job profile also provides information on who generated the job, when the job was generated, what return codes were issued, which worklist and JCL data sets were used, and the current job status.

job status
See Action Status.

JOB table
A DASD MANAGER PLUS utility definition table that stores utility jobs generated and submitted through DASD MANAGER PLUS. This table is named BMCASUmm.UT_JOB.

justify
A graphic display option that specifies the alignment of the legend on statistical graphs: C (center), R (right), L (left), T (top), or B (bottom).

K
key length
An information field showing the length of the index key as stored, specifically, the number of bytes (1 – 254).
Leaf distribution
An exception you can set using the DASD MANAGER PLUS BMCTRIG utility to monitor the leaf distribution in an index. An increase over time implies several leaf page splits and might indicate that the index should be reorganized.

length
Information fields showing the average, minimum, and maximum lengths of the columns in a table.

Level
An exception you can set using BMCTRIG to indicate that the number of index levels has increased.

LEVELINC
The level increase; an exception you can set using BMCTRIG to monitor increases in the number of index levels.

LEVELMIN
The level minimum; an exception you can set using BMCTRIG to monitor the minimum number of levels required if the index were reorganized.

LEVELS
A statistic indicating the number of levels in the index, which is a function of the number of table rows and the size of the index key. Inserted or changed keys can cause splits that take one full index page and create two pages that are half full. Eventually these splits cause an index tree split, creating another level of index.

With BMCTRIG, you can set the Levels exception to monitor the number of index levels.

lines
A DASD MANAGER PLUS graphic display option that specifies whether to draw lines on statistical graphs: LINE (lines) or NOLI (no lines).

LOAD
A command that runs the IBM LOAD utility. The short form of the command is LOAD.

LOADPLUS for DB2
One of the Utilities for DB2 from BMC Software. LOADPLUS loads tables two to four times faster than the DB2 LOAD utility. The product significantly reduces elapsed times, CPU cycles, and EXCPs to increase the availability of DB2 data and lower the cost for loading data.

LVL
The number of levels in an index.
M

margin
A DASD MANAGER PLUS graphic display option that specifies the placement of the legend on statistical graphs: B (bottom) or T (top) for horizontal orientation; R (right) or L (left) for vertical orientation.

markers
A DASD MANAGER PLUS graphic display option that specifies whether to use place markers on statistical graphs: MARK (markers) or NOMA (no markers).

max row length
An information field showing the maximum possible row length (in bytes) in the table. The number comes from the column values specified on the Table Column List.

MAXROWS
An attribute of a table space indicating the maximum number of rows per page (0 - 255); reported by BMCSTATS and RUNSTATS.

menu
In DASD MANAGER PLUS, a menu is a list of action options. You select an action by typing its corresponding number in the option input field and pressing Enter. A menu can contain other fields that you can use to qualify the action.

MERGECOPY
A command that runs the IBM MERGECOPY utility. The short form of the command is MERG.

MODIFY
A command that runs the IBM MODIFY utility. The short form of the command is MODI.

Most Frequent Value display
A BMCSTATS column statistics display that shows the ten most frequent values found in the column and their sequence.

N

NonUniform
An exception you can set using BMCTRIG to monitor the nonuniformity of an index based on the values that might appear in the SYSFIELDS catalog table. There are up to 10 distinct values.

NumIncremt
An exception you can set using BMCTRIG to monitor the number of incremental copies since the last full copy.
Object
A DASD MANAGER PLUS field identifying the qualified or unqualified name for an object. This can be an input field or an information field. For table spaces and tables space sets, the format is DBNAME.TSNAME. For indexes, the format is IXCREATOR.IXNAME. Wildcard patterns are generally accepted to specify a group of similar objects.

object set
An extension of the DB2 object wildcarding facility in DASD MANAGER PLUS.

OPNDB2ID
The DB2 authorization ID to update BMCSTATS tables. With the installation option of OPNDB2ID = Y, users with STATS authority can collect statistics even if their logon ID does not have RACF authority to read the data set.

order
A DASD MANAGER PLUS graphic display option that specifies the order of the legend: KNOR (normal; left to right for horizontal and top to bottom for vertical) or KREV (reversed).

orient
A DASD MANAGER PLUS graphic display option that specifies the orientation of the legend on statistical graphs: H (horizontal) or V (vertical).

PageGroup
A feature of BMCSTATS that allows you to review the statistics on a specified grouping of pages to uncover additional information on hot spots in the data. When you generate control statements for the BMCSTATS utility, you determine the number of pages to group (0 through 99999). This facility and the graphic displays can help you locate areas of concentrated activity within a table space.

PARTITION
A BMCSTATS and RUNSTATS statistic indicating the partition number.

PartLvl
A BMCTRIG parameter that specifies whether to generate a utility job for each partition in exception or to generate a job at the table space or index level.

PctActivHi
An exception you can set using BMCTRIG to monitor the percentage of active pages. Use this parameter to specify the maximum percentage of active pages allowed.

PctActivLo
An exception you can set using BMCTRIG to monitor the percentage of active pages. Use this parameter to specify the minimum percentage of active pages allowed.
PCTFREE
   An attribute indicating the percentage of each page to leave free when the table space or partition is loaded or reorganized; reported by BMCSTATS and RUNSTATS. Valid values are 0 through 99. The default value is 5 for table spaces, 10 for indexes.

PGSIZE
   An attribute indicating the unit of storage within a table space or index (4 KB, 8 KB, 16 KB, or 32 KB); reported by BMCSTATS and RUNSTATS.

product option file (POF)
   A file that contains options you can specify to generate the JCL for individual data sets.

PQTY
   An attribute indicating the primary space allocation (in tracks); reported by BMCSTATS.

primary index
   An index that enforces the uniqueness of a primary key.

primary key
   A unique, nonnull key that is part of the definition of a table. A table cannot be defined as a parent unless it has a primary key.

PRIQTY
   The primary space quantity (expressed in Alloc Unit) allocation VSAM value from the ICF catalog. This value must be at least 1. The default value comes from the installation standards. If you set Priqty to the same value as shown in Estimated Space, the number of rows will fit in the primary quantity. If the Alloc Unit is K and the primary quantity is less than 1 track, Priqty will be rounded to 1 for space estimation purposes.

Q
QMF
   Query Management Facility. DASD MANAGER PLUS provides QMF procedures for accessing the statistics tables to obtain useful information. DASD MANAGER PLUS provides sample QMF queries (in the CNTL library member ASURVIEW) that are used for the sample QMF reports.

QUIESCE
   A command that runs QUIESCE utility. The short form of the command is QUI.

R
Recall
   A BMCSTATS parameter that enables you to recall archived data sets when collecting statistics.

REORG
   A command that runs the IBM REORG utility. The short form of the command is REOR.
REORG PLUS for DB2
One of the Utilities for DB2 from BMC Software. REORG PLUS reorganizes DB2 tables two to ten times faster than the DB2 REORG utility. REORG PLUS eliminates the need for executing separate RUNSTATS and COPY utilities. The product significantly reduces elapsed times, CPU cycles, and EXCPs and takes advantage of multiple CPUs, multitasking, and parallel processing.

Report
A parameter used by BMCSTATS to print a report on the statistics collected, by BMCTRIG to print a report on the initiated exceptions, by BMCCPRS to print a report on the statistics copied, and by BMCUPRS to print a report on the statistics updated.

REP REC
A command that runs the IBM REPORT utility with RECOVERY option. The short form of the command is REPR.

REP SET
A command that runs the IBM REPORT utility with TABLESPACESET option. The short form of the command is REPS.

Restart option
A DASD MANAGER PLUS option that causes JCL Generation to generate a RESTART keyword in the AEXIN(SYSIN) parameters of the JCL job stream. The restart option lets you restart the job from the point at which the job failed. This option is available on the DASD MANAGER PLUS job generation panel.

restart parameter
An option that enables you to pass parameters to utilities being restarted and thereby ensure proper restart based on the utility and objects being processed. Restart parameters can be generated in RESTART JCL by entering the parameters on the DASD MANAGER PLUS job generation panel or by editing the worklist. The format is RESTARTPARM RESTARTPARMstring where RESTARTPARMstring is a list of lineoneparms.

rows
A DASD MANAGER PLUS printing option that specifies the number of rows per page.

rows/key
The number of rows per index key (1 through 999999). Might be fractional. For a unique index, this value should be 1. This is one of the parameters you can adjust when estimating space requirements for an object. With BMCTRIG, you can set the Rows/Key exception to specify the maximum number of rows per key to allow.

RUNSTATS
A command that runs the IBM RUNSTATS utility. The short form of the command is RUNS.
Sample
A BMCSTATS and RUNSTATS feature that you can use for random sampling of objects when collecting statistics. You can use sampling on all objects, on table spaces only, or on indexes only. Sampling is much faster on large objects. If the object has fewer than 1000 pages, sampling is not performed. Regular statistics will be collected even if sampling is requested. Sampling on table columns estimates values using probability. If you want more detail, do not use sampling.

SAMPLING
A historical parameter indicating whether sampling was used when collecting the statistics; reported by BMCSTATS.

Save
A BMCTRIG parameter that specifies whether to save exceptions in the DASD MANAGER PLUS exceptions table.

save last used
A DASD MANAGER PLUS user option that specifies whether to use the last values specified for the options as the default values.

SAVESTATS
A historical attribute indicating whether the statistics collected by the BMCSTATS utility were saved in the DASD MANAGER PLUS statistics database.

SaveStats
A BMCSTATS parameter that you can use to save the statistics collected in the DASD MANAGER PLUS statistics database.

security exit
A user exit that enables the establishment of some form of system security. DASD MANAGER PLUS provides the following user exits to allow different forms of security:

- Front End Security Exit (limits access to actions)
- Execution Manager Security Exit (provides installation security checking and option enforcement)
- Execution Manager Unload Exit (provides testing and modification of each row of unloaded data)

SEGSIZE
An attribute indicating the number of pages in the segment of a table space; reported by BMCSTATS and RUNSTATS. For space estimation, if the table space is to be segmented, specify the number of pages to be assigned to each segment. Use zero or a value from 4 to 64 in multiples of 4. Zero means the table space is not to be segmented. The default value comes from the installation standards.
selection list
A list of related items that you can select for further action. The actions (line commands) you can specify in the Act field are typically displayed across the top of the panel.

SEQTY
The secondary space quantity (expressed in tracks) used for allocating the DB2-defined data set (0 through 131,068). The default value comes from the installation standards. If the Alloc Unit is K and the primary quantity is less than 1 track, Seqty will be rounded to 1 for space estimation purposes.

service
A user-specified command or utility that makes up an action. You can add, delete duplicate, and modify services to customize a particular job.

space
The primary and secondary quantities used for data set allocation. In DASD MANAGER PLUS, you can specify a default space value when setting user options, or you can estimate the space value using the Estimate Source option when generating JCL. You can also set a Space exception using BMCTRIG to specify a threshold for the percentage of increase in the number of tracks used by an object.

SpaceOnly
A BMCSTATS parameter that you can use to collect only space information (from the VSAM catalog) when collecting statistics. This option is very fast if you need only size and extents information.

Simple Space Estimation (SSE)
A COMMAND line tool that allows you to estimate simple space for table spaces or index objects, giving you “what if” capability. Unlike DASD MANAGER PLUS statistics, you do not need to run BMCSTATS before using SSE. See also space estimation function.

space estimation function
A DASD MANAGER PLUS function that you can use to estimate space requirements for existing table spaces and indexes. Estimates are based on statistics from the DASD MANAGER PLUS database. See also Simple Space Estimation (SSE).

spawner
See automation spawner.

SQUY
An attribute indicating the secondary space allocation (in tracks); reported by BMCSTATS.

stack tapes
A DASD MANAGER PLUS user option that specifies whether to stack image copy tapes.
Standard JCL mode
The basic automation mode. In standard JCL mode, the Database Performance for DB2 solution generates stand-alone JCL and places it in a partitioned data set (PDS) that you specify in the POF. In standard JCL mode, the Database Performance for DB2 solution does not manage the jobs. You must submit them to your scheduler. See also full automation mode.

START DB
A command that runs the DB2 START DATABASE utility. The short form of the command is STA DB.

Start Over option
A RESTART option provided on the DASD MANAGER PLUS Action Generation panel that reprocesses a worklist that did not complete from the beginning.

STATAUTH
The statistics authorization indicator. If the indicator is set to Y (the default), DASD MANAGER PLUS checks users’ authorization to run BMCSTATS and requires the same authorization as for RUNSTATS.

status codes
See Action Status.

STEP table
A DASD MANAGER PLUS utility definition table that stores services generated through DASD MANAGER PLUS. This table is named BMCASUnn.UT_STEP.

stogroup allocation
A volume placement parameter specifying the name of the storage group to use. (Leave the other fields blank.) If you are using VCAT allocation, enter a wildcard in the Stogroup field to list available volumes within the storage group for the specified VCAT. The default value for this field comes from the installation standards.

STOP
A utility command that stops the BMC Software Execution program.

STOP DB
A utility command that runs the DB2 STOP DATABASE command. The short form of the command is STO DB.

STORNAME
An attribute of a table space partition or index partition indicating the name of the storage group used for allocation; reported by BMCSTATS and RUNSTATS. See also STORTYPE.

STORTYPE
An attribute of a table space partition or index partition indicating the type of storage allocation. E indicates explicit; if E, STORNAME names an integrated catalog facility catalog. I indicates implicit; if I, STORNAME designates a storage group; reported by BMCSTATS and RUNSTATS.
STOSPACE
A command that runs the IBM STOSPACE utility. The short form of the command is STOS.

submit job
A DASD MANAGER PLUS job generation option that specifies whether to submit the current job for execution.

symbolic variable
A user interface variable that has its value set interactively for the current user and session at the time of JCL generation. JCL Generation uses symbolic variables to perform ISPF file-tailoring services. On DASD MANAGER PLUS panels, a symbolic variable should be preceded with an ampersand (&). In the JCL of the installation options module, a symbolic variable should be preceded with two ampersands (&&). Examples of these variables include: &SSID, &MEMBER, &OBJTYP, and &TBNAME. See also JCL Generation.

Note: Symbolic variables should not be confused with global job variables (AJX-type), which have their values set for all users and all sessions.

SYNC
A utility command that runs a checkpoint to use for restart processing.

csync point
A completion flag set during the execution of a worklist. The Execution program writes sync points to the SYNC table when the program encounters –SYNC or –STOP commands in the worklist input stream. All SQL statements between sync points run as a single DB2 transaction. If a worklist is halted before completion for any reason, sync points allow the user to begin processing the worklist from the last sync point. Sync point information is stored in the SYNC table.

SYNC table
A DASD MANAGER PLUS utility definition table containing sync point information from the execution of a worklist. This table is named BMCASU\nn.UT_SYNC.

system trigger
Thresholds and corrective actions in the central repository that are available to all BMCTRIG jobs as a central point for administering exception thresholds and corrective actions. See also threshold and corrective action.

T

Table Average Row Length
A DASD MANAGER PLUS statistical graph that plots the average row length in a table over time.

Table Cardinality
A DASD MANAGER PLUS statistical graph that plots the number of rows in a table over time. The graph also plots the number of indirect references in relation to the number of rows.
Table Pages
A DASD MANAGER PLUS statistical graph that plots the number of pages in a table over time.

Table Percent Pages
A DASD MANAGER PLUS statistical graph that plots the percentage of pages in a table over time.

Tables
A BMCSTATS parameter that displays a panel from which to select tables and columns for collecting statistics.

Table Space Pages
A DASD MANAGER PLUS statistical graph that plots the relationship over time of page statistics. The statistics plotted include allocated pages, active pages, and pages required if the table space were reorganized.

Table Space Partition Cardinality
A DASD MANAGER PLUS statistical graph that plots the number of rows in a table space over time. The graph also plots the number of NEARINDREF and FARINDREF in relation to the number of rows in the partition over time.

Table Space Partition Extents
A DASD MANAGER PLUS statistical graph that plots the number of extents in a table space over time. The maximum number of extents per data set allowed by VSAM is 7257.

Table Space Partition Page Group
A DASD MANAGER PLUS statistical graph that plots the distribution of data in the partition. The graph shows the number of rows, number of dirty pages, and number of full pages in relation to the number of pages in the data set.

Table Space Partition Pages
A DASD MANAGER PLUS statistical graph that plots the relationship over time of page statistics. The statistics plotted include allocated pages, active pages, pages required if the table space were reorganized, dirty pages, and full pages.

Table Space Partition Percent Active/Drop
A DASD MANAGER PLUS statistical graph that plots the percentage of active and dropped pages in the table space partition over time.

table space set
Two or more table spaces that are related by referential integrity (RI) constraints. Note that the table spaces can be in different databases.

Tasks
A BMCSTATS parameter that specifies the level of multitasking to use for processing partitioned objects (1 through 16). If you use this option, specify a Buffers value that is at least this large.
three-part name
The full name of a table, view, or alias. The three-part name consists of a location name, authorization ID, and an object name separated by periods.

threshold
A specified value at which the product must perform a specified task, such as issue a message or generate a corrective action. See also corrective action and system trigger.

TOTALIND
The sum of FARIND and NEARIND; an exception you can set using BMCTRIG to monitor the percentage of rows of a table space that are not in their original page.

TOTALOFF
The sum of FAROFF and NEAROFF; an exception you can set using BMCTRIG to monitor the percentage of rows of an index that are not in optimal position.

TSO submit exit
A DASD MANAGER PLUS user option that specifies whether to use a TSO submit exit to generate job statements (default = N).

TYPE
An attribute of a table space indicating the type of table space: L (large), K (large with cluster member), I (nonlarge with cluster member), or blank (other); reported by BMCSTATS and RUNSTATS.

Type
A DASD MANAGER PLUS input or information field identifying the type of object: TS (table space), TT (table space set), IX (index), SG (storage group), or VL (volume).

TYPE TS
An attribute of a table space reported for indexes: L (large), K (large with cluster member), I (nonlarge with cluster member), or blank (other); reported by BMCSTATS.

U
UNIQUE
A field specifying the uniqueness of an index. Following are permissible values:

- D indicates duplicates are allowed.
- U indicates the index is unique.
- P indicates the index is a primary index (as well as unique).
- C indicates the index is unique and used to enforce UNIQUE constraint.
- N indicates the index is unique and defined with UNIQUE WHERE NOT NULL.
- R indicates the index is unique and used to enforce the uniqueness of a nonprimary parent key.
**UNIQUERULE**
An attribute of an index indicating the uniqueness of the index: D indicates that duplicates are allowed, U indicates the index is unique; reported by BMCSTATS and RUNSTATS. See also UNIQUE.

**UNLOAD**
A command that runs the IBM UNLOAD utility. The short form of the command is UNLD.

**UNLOAD PLUS for DB2**
One of the Utilities for DB2 from BMC Software. UNLOAD PLUS simultaneously unloads data from one or more DB2 tables in the same table space. UNLOAD PLUS unloads data four to eight times faster than conventional SQL–based applications such as DSNTIAUL, QMF, or DSNTEP2. UNLOAD PLUS significantly reduces elapsed times, CPU cycles, and EXCPs, increasing the availability of DB2 data and lowering costs for unloading data.

**UPDATEDB2**
A historical parameter indicating whether the DB2 catalog was updated with the statistics collected by the BMCSTATS utility.

**UTILCODE**
A historical parameter indicating the utility code for the generation of statistics; reported by BMCSTATS.

**Utilities for DB2**
A collection of BMC Software products that includes LOADPLUS, UNLOAD PLUS, and REORG PLUS. These utilities are enhanced replacements for similar IBM utilities and are accessible through the utility generation function of DASD MANAGER PLUS. For more information on a specific product, refer to the appropriate BMC Software manual.

**V**

**variable**
See symbolic variable and global job variable.

**VCAT allocation**
A volume placement parameter that specifies the data set high-level qualifier appropriate for the DB2 subsystem. Use this field for nonstogroup allocation only.

**VCATNAME**
An attribute of a table space partition or index partition indicating the name of the ICF used for space allocation; reported by BMCSTATS and RUNSTATS.

**VCATPASS**
If you are using VCAT allocation for volume placement, this is the password associated with the VSAM password for the ICF/VSAM catalog.
volume
A single unit of storage that is referenced by a serial number and is used for data set allocation by DASD MANAGER PLUS.

W

wildcard
A method of specifying a particular character pattern for matching. The DASD MANAGER PLUS wildcard characters are the percent (%), the underscore (_), and asterisk (*):

- Use % as a trailing wildcard to represent any group of characters.
- Use _ to represent a single character.
- Use * as a trailing wildcard to represent any group of characters. This character is valid on some but not all panels. See the online Help to determine valid characters for each panel.

Many of the input fields in DASD MANAGER PLUS allow you to enter a wildcard pattern. For example, in the **Action Name** field of the Action panel, you can enter a wildcard pattern containing * to display a narrower selection list of actions that match the specified pattern.

Other examples are as follows:

- **Table space**: `databaseName.TS%`
- **Stogroup**: `SYS%`
- **Volume**: `V%`

**WORKIDS table**
A DASD MANAGER PLUS object definition table that stores actions and task IDs. This table is named `BMCASU\text{n}.DO_WORKIDS`.

**worklist**
A file that is created by the job generation functions of DASD MANAGER PLUS. The worklist provides the BMC Software JCL Generation program and Execution program with a control statement for the tasks to be performed. Once created, a worklist can be edited and reused.

**worklist DSN**
The name of the data set to contain the utility worklist. The data set must exist and can be partitioned or sequential. If partitioned, a member name must be specified. Symbolic variables can be used.

**worklist execution log**
*See AEXPRINT.*
Index

Symbols
$AJXDOC member 138

Numerics
2MEGSQL POF keyword 160

A
accessing
actions 307
functions 72
object sets 340
other databases 92
plans 89
product database 92
statistics function 360
utility generation 270
ACM_AMS POF keyword 160
ACM_BASDIAG POF keyword 160
ACM_BRPTDIAG POF keyword 160
ACM_BRPTDSN POF keyword 161
ACM_CDLDSN POF keyword 161
ACM_CDLPS POF keyword 161
ACM_CDLSS POF keyword 161
ACM_CDLU POF keyword 161
ACM_CMPDIAG POF keyword 161
ACM_CPLCDLO POF keyword 161
ACM_CPLDIAG POF keyword 162
ACM_CPLWDSN POF keyword 162
ACM_CPLWDSNO POF keyword 162
ACM_DBRM1 POF keyword 162
ACM_DBRM2 POF keyword 162
ACM_DBRM3 POF keyword 162
ACM_DYN SORTW_NUM POF keyword 162
ACM_DYN SORTW_UNIT POF keyword 163
ACM_GLID POF keyword 163
ACM_IBMR_Map_REQ POF keyword 163
ACM_IMPDIAG POF keyword 163
ACM_JDSN POF keyword 163
ACM_JDSNB POF keyword 164
ACM_JDSNCG POF keyword 164
ACM_JDSNBR POF keyword 164
ACM_JDSCNC POF keyword 164
ACM_JDSCNCPL POF keyword 164
ACM_JDSCNCPO POF keyword 165
ACM_JDSCNE POF keyword 165
ACM_JDNSI POF keyword 165
ACM_PARALLEL_MAXINIT POF keyword 165
ACM_PARALLEL_MININIT POF keyword 165
ACM_PARALLEL_WORKLST POF keyword 166
ACM_PARALLEL_XIMGRP POF keyword 166
ACM_PARALLEL_XIMPROC POF keyword 166
ACM_PARALLEL_XIMSTRT POF keyword 166
ACM_PARALLEL_XIMTRCE POF keyword 166
ACM_PIC POF keyword 166
ACM_SDSN POF keyword 167
ACM_SDSNE POF keyword 167
ACM_WDSN POF keyword 167
ACM_WLORDER POF keyword 167
ACM_WLORDERMSG POF keyword 168
ACM_WLPS POF keyword 168
ACM_WLSS POF keyword 168
ACM_WLUL POF keyword 168
ACM_WLUL POF keyword 168
action definitions, specifying to Export 701
action generation
build/edit an action 279
creating object sets 336
definition 260
action POF, associating with actions 280
Action Report 657
actions
Action List panel 274
associating a POF 280
controlling access to 307
corrective, for REORG 556
creating 271
creating object set separately 322, 324
deleting 278
editing 274
eliminating duplicates 463, 464
generating 299
identifying associations with object sets 330
JCL Options file (POF) 280
managing 260
path for object sets 335
POF, defining 149
reports 657
adding a customizable report to the menu 656
ADDOLOAD1 POF keyword 168
ADDOLOAD2 POF keyword 168
Administrative products for DB2 62
advantages of using object sets 311
AEXAUNLD package 90
AEXESTDL package 90
AEXIN input stream, user options 84
AEXSQLIO package 90
AExvrDA Execution Monitor Entry plan 90
AExvrDM Execution Monitor plan 90
AExvrHA Execution Monitor Entry plan 90
AExvrHM Execution Monitor plan 90
AJXIN input stream 84, 137
AJXPODAT edit macro 126
AJXPOFER message file 129
AJXPOFIN input stream 84, 137
AJXPOFVL message file 129
allocating data sets 88
ALLOW_NETCMD 769
ALTER
  administrative product 62
  ALU code to identify the product 794
  data set sizing options 100
  fast path navigation commands 75
ALTER plans
  AEXvrHA 90
  AEXvrHM 90
altering and re-creating objects 404
ALTERSEC
  altering SQTY 579
  command description (ALT2) 257
  considerations 575
  definition 576
  flow diagram 578
  how it works 576
  introduction 576
  SQL command 577
  workflow 576
ampersands in installation options module 232
analyzing
  column statistics 417
  column value statistics 419
  index partition page group statistics 424
  index partition statistics 422
  index statistics 420
  statistics 408
  storage group statistics 425
  table space partition page group statistics 414
  table space partition statistics 411
  table space statistics 409
  table statistics 416
  volume statistics 427
Append insert trigger 477
Application ID 85
APPNDINS exception 477, 551
ARCH_DATACLASS POF keyword 169
ARCH_DATACLASS_ALT POF keyword 169
ARCH_EXPDT POF keyword 169
ARCH_MGMTCLASS POF keyword 169
ARCH_MGMTCLASS_ALT POF keyword 169
ARCH_PREFIX POF keyword 169
ARCH_PRIQTY POF keyword 169
ARCH_RETPD POF keyword 169
ARCH_SECQTY POF keyword 169
ARCH_STACK POF keyword 170
ARCH_STORCLASS POF keyword 170
ARCH_STORCLASS_ALT POF keyword 170
ARCH_THRESH POF keyword 170
ARCH_UNIT POF keyword 170
ARCH_UNIT_ALT POF keyword 170
AREOPENP exception 477, 550, 552
ARERPEND exception 477, 551, 552
ASCII definition 795
associating object sets
  by browsing 337
  by name 338
ASU_XP_UIMSRVHOST POF keyword 692
ASU_XP_UIMSRVRPORT POF keyword 692
ASUDOPTS installation listing 232
ASURXLOC external function 739
ASURXLSP external function 734
ASURXRPT external function 733
ASURXSE external function 735
ASURXSQL external function 730
ASUvrDJ BMCTRIG Utility Job Generation plan 89
ASUvrDR Report Display plan 89
ASUvrDS Statistics Collection plan 89
ASUvrDZ Browse Statistics plan 89
ATBWORKAREA installation option 234
authorization
  actions 307
  collecting statistics 383
  controlled through security exits 91
  DASD MANAGER PLUS database 92
  for Export 691
  object sets 340
  setting 83
  to access the BMC UIM server Commands web page 769
  to other databases 92
Average Length of RID Chain Report 587
Average Monthly Growth Report 660
B
backup and recovery products for DB2 62
batch
  programs, return codes 306
  report generation 640
  running customizable report 653
BINDFAIL POF keyword 171
BLRP_DATACLASS POF keyword 172
BLRP_DATACLASS_ALT POF keyword 172
BLRP_EXPDT POF keyword 172
BLRP_MGMTCLASS POF keyword 172
BLRP_MGMTCLASS_ALT POF keyword 172
BLRP_PREFIX POF keyword 172
BLRP_PRIQTY POF keyword 172
BLRP_RETPD POF keyword 172
BLRP_SECQTY POF keyword 173
BLRP_STACK POF keyword 173
BLRP_STORCLASS POF keyword 173
BLRP_STORCLASS_ALT POF keyword 173
BLRP_THRESH POF keyword 173
BLRP_UNIT POF keyword 173
BLRP_UNIT_ALT POF keyword 173
BMC Common Statistics component (ATS) 64
BMC Software, contacting 2
BMC UIM Server Commands web page 769
BMC_CHECK_LOAD POF keyword 174
BMC_CHECK_OPTS POF keyword 174
BMC_COPY_LOAD POF keyword 174
BMC_COPY_OPTS POF keyword 174
BMC_LOAD_LOAD POF keyword 174
BMC_LOAD_OPTS POF keyword 175
BMC_RECOVER_LOAD POF keyword 175
BMC_RECOVER_OPTS POF keyword 175
BMC_REORG_LOAD POF keyword 175
BMC_REORG_OPTS POF keyword 176
BMC_REORG_XBMID POF keyword 176
BMC_UNLOAD_LOAD POF keyword 176
BMC_UNLOAD_OPTS POF keyword 176
BMCCOPY
  command description (BMCC) 257
  Copy Options panel 144
description 795
grouped services 466
  specifying GDGs on the COPYDDN options panel 144
WEEKLYREORG service 254
BMCCOPYI (BMCC) command description 257
BMCCPRS
  authorization to run 383
command description 257
definition 795
grouped services 466
  updating database 401
BMCDATA2 CLIST 85
BMCLOAD (BMCL) command description 257
BMCMOD command description 257
BMCREORG
  command description (BMCR) 257
definition 796
grouped services 466
  mapping table variables for REORG 557
  specifying GDGs on the COPYDDN options panel 144
WEEKLYREORG service 254
BMCMSTATS
  See also statistics
  accessing 360
  Age trigger 482
  authorization to run 383
  benefits 346
  browsing 366
  command description (BMCS) 257
database tables 347
deleting statistics 404
displaying for analysis 409
grouped services 466
  identifying trigger values 496
options 349
  sample JCL 396
  sampling index parameter 354
  sampling tablespace parameter 353
  table columns updated 385
  table space/index columns updated 388
  updating DB2 catalog 390
WEEKLYREORG service 254
BMCSYNC installation option 234
BMCTRIG
  and commands 455
  and utilities 455
  bypassing evaluations 461
  command description (BMCT) 257
  evaluating exceptions, overview 472
  exceptions 461
  exceptions job 504
  excluding items 555
  flow diagrams 472
  generated standard JCL example 533
  information sources 230
  interim worklist 528
  introduction 454
  JCL generation options 555
  job card exit in interim worklist 531
  job generation options 555
  main panel 487, 506
  manually restarting job 571
  order of using information 230
  processing options, introduction 471
  Rebind option 513
  REORG corrective action 556
  REORG job 553
  Resize Down option 515
  restart/start over a job 569
  setting thresholds 476
  syntax options 507
  taking corrective action 496
  thresholds 476
  worklist JCL 538
BMCULOAD (BMCLUD) command description 257
BMCUPRS
  authorization to run 383
command description (BMCU) 257
definition 796
grouped services 466
  updating DB2 catalog 398
BMCUZOOM ISPF command 71
  -BR14 command 528
browsing
  column statistics 372
  DB2 catalog and BMCSTATS 366
  index partition statistics 374
  index statistics 373
  object sets 337
  table space partition statistics 369
  table space statistics 367
  table statistics 368
BSTATAGE exception 482
building an action 279
Bypass Exceptions option 509

C
CAFREASON variable 732
call attachment return codes 732
CANCEL ISPF command 71
CARD exception 477
Card trigger 477
CAT_LOAD POF keyword 176
CATALOG MANAGER
  ACT code to identify the product 793
  fast path navigation commands 75
Catalog Upd Age trigger 482
CHANGE MANAGER
  ACM code to identify the product 793
  administrative product 62
  data set sizing options 100
  fast path navigation commands 75
CHANGE MANAGER plans
  AEXvrmHA 90
  AEXvrmHM 90
changes to product 28
CHECK+_LOAD POF keyword 177, 213
CHECKDOPT POF keyword 177
CHEK DA (CKDA) command description 257
CHEK LOB (CKEK) command description 257
CHGMAN_LOAD POF keyword 177
cleanup, data set 101
CLEANUP_RC POF keyword 177
CLOSE cursor command in ASURXSQL 731
CNTL_DATACLASS POF keyword 177
CNTL_EXPDT POF keyword 177
CNTL_MGMTCLASS POF keyword 177
CNTL_PREFIX POF keyword 178
CNTL_PRIQTY POF keyword 178
CNTL RETPD POF keyword 178
CNTL_SECQTY POF keyword 178
CNTL_STORCLASS POF keyword 178
CNTL_UNIT POF keyword 178
CNTLMOUT_DSN POF keyword 178
CNTLMMSCH_DSN POF keyword 178
column group statistics
  Select COLGROUP field 356
columns
  analyze statistics 417
  analyze value statistics 419
  browse statistics in DB2 catalog 372
  format of worklists 266
  renaming using RENAME COLUMN 404
  sampling table 360
  update statistics in DB2 catalog 379
  updating the column names 404
commands
  ALTERSEC 257
  BMCCOPY 257
  BMCCOPYI 257
  BMCCPRS 257
  BMCMOD 257
  BMCREORG 257
  BMCSTATS 257
  BMCTRIG 257
  BMCUNLOD 257
  BMCUPRS 257
  -BR14 528
  CHEK DA 257
  CHEK LOB 257
  DB2 object list 363
  descriptions, utilities 257
  DSN1COPY 257
  fast path navigation 75
  FULLCOPY 257
  INCRCOPY 257
  ISPF 71
  LOAD 258
  MERGCOPY 258
  MODICOPY 258
  -NEWJ 528
  QUIESCE 258
  RECOVER 258
  REORG 258
  REP REC 258
  REP SET 258
  REPAIR 258
  RESIZE 258
  RUNSTATS 258
  SQL ALTER SECQTY 577
  START DB 258
  STOP 258
  STOP DB 258
  STOSPACE 258
  submitted through BMCTRIG 455
  SYNC 258
  -UDSJ 528
  use in utility jobs 256
  worklist 264
comment lines in worklists 267
COMMIT command in ASURXSQL 731
compatibility with DB2 utilities 64
compiling SLIBs 713
confirmation panel, definition 70
correction
   Enterprise List 690
   personal list 690
contention issues 249
conventions, documentation 28
copy
   actions 273
   object sets 331
   services 285
COPY exceptions 461
copy exceptions 483
Copy Pend trigger 484
COPY PLUS options, unsupported 65
COPY#IIC exception 485
COPY+_LOAD POF keyword 179
COPYAGE exception 484
COPYDOPT POF keyword 179
COPYMODS exception 484
COPYPEND exception 484
corrective actions
   Action report 657
   adding 499
   ALTERSEC utility 575
   automating 470
   Cross Reference Report 662
   defining system-level 497
   definition 56, 798
   deleting 500
   editing 499
   Exceptions Report by BMCTRIG Action 671
   for exceptions 460
   generating 472
   in BMCTRIG 496
   managing 496
   setting in BMCTRIG syntax 496
   setting up for REORG 556
   specifying default actions 465
   using BMCTRIG 454
   using priorities to limit 457
   when to use 462
CPLAN installation option 234
CPYEXP_DATACLASS POF keyword 179
CPYEXP_EXPDT POF keyword 179
CPYEXP_MGMTCLASS POF keyword 179
CPYEXP_PREFIX POF keyword 179
CPYEXP_RETPD POF keyword 179
CPYEXP_STORCLASS POF keyword 179
CPYEXP_SUPPRESS_SUFF POF keyword 180
CPYEXP_UNIT POF keyword 180
creating
   actions 271, 279
   DB2 object lists 361
   object sets 310
   object sets outside actions 322, 324
   services 279
   statistics graphs 428
CSTATAGE exception 482
current environment, displaying 77
customer support 3
customizable reports
   ASURXLOC external function 739
   ASURXLOC return codes 739
   ASURXLSP external function 734
   ASURXLSP return codes 734
   ASURXRPT external function 733
   ASURXRPT return codes 734
   ASURXSE external function 735
   ASURXSE return codes 738
   ASURXSQL external function 730, 731
   ASURXSQL return codes 731
   call attachment return codes 732
   complete sample program 756
   customization 722
   definitions of external functions 719
   descriptions 645
   EXEC parms 729, 742, 743
   field holders 724
   field justification 724
   field lines 724
   format definition 743
   formatting 722, 723
   JCL DD statements 730
   line numbers 725
   MAX(TIMESTMP) 728
   number of cursors supported 733
   overview 59, 644
   page numbers 725
   plan name 729
   plan variable 724, 727
   printing SQL error messages 732
   program structure 720
   quotation marks in SQL 728
   report layout 722
   report title 724
   reporting tasks 733
   reserved word 723
   retrieving the most recent data 728
   REXX notes 740
   rpt.line# variable 724
   rpt.page# variable 724
   rpterms variable 734
   running 650
   sample program 720, 741
   sample report 721
   SQL considerations 727
   SQL support and limitations 733
   ssid variable 724, 727
   SYSTERM data set 732
   timestamp use 728
   value lines 724
   variables 724, 727
customizing installation 83
report programs 655

D

D2U (DB2 Utilities Common Code) 64
DASD MANAGER PLUS plans
AEXvrmDA 90
AEXvrmDM 90
ASUvrmDJ 89
ASUvrmDR 89
ASUvrmDS 89
ASUvrmDZ 89
DASD_LOAD POF keyword 180
DASDDOPT POF keyword 180
Data Set List by Database Name Report 663
data sets
$AJXDOC 138
allocation 88
cleanup 101
DATAWORK 105
deleting automatically 101
diagnostic output 88
discard 102
error 102
image copy 147
JCL 88
listing by database 364
LOGSORT 105
map 102
names specified for standard JCL 519
names specified for worklist JCL 541
permanent work 108
prefixes 109
punch 102
RECV 140
report 88
resolving names 141
sample VSAM object 100
setting default processing options 108
sizing 100, 101, 137, 139
SORTOnnn 102
SORTOUT 102, 108
SORTPnnn 102
SORTWORK 105
SUTnnn 102
SYSC 140
SYSCOOPY 108
SYSDISC 108
SYSERR 108
SYSMAP 108
SYSPUNCH 108
SYSREC 108
SYSUT 108
SYSUTnnn 102
DASD MANAGER PLUS plans
AEXvrmDA 90
AEXvrmDM 90
ASUvrmDJ 89
ASUvrmDR 89
ASUvrmDS 89
ASUvrmDZ 89
DASD_LOAD POF keyword 180
DASDDOPT POF keyword 180
Data Set List by Database Name Report 663
data sets
$AJXDOC 138
allocation 88
cleanup 101
DATAWORK 105
deleting automatically 101
diagnostic output 88
discard 102
error 102
image copy 147
JCL 88
listing by database 364
LOGSORT 105
map 102
names specified for standard JCL 519
names specified for worklist JCL 541
permanent work 108
prefixes 109
punch 102
RECV 140
report 88
resolving names 141
sample VSAM object 100
setting default processing options 108
sizing 100, 101, 137, 139
SORTOnnn 102
SORTOUT 102, 108
SORTPnnn 102
SORTWORK 105
SUTnnn 102
SYSC 140
SYSCOOPY 108
SYSDISC 108
SYSERR 108
SYSMAP 108
SYSPUNCH 108
SYSREC 108
SYSUT 108
SYSUTnnn 102

DASD MANAGER PLUS for DB2 User Guide

unload 102
work 102
worklist 88
WRKnnn 102
DATA_PACKER_LOAD POF keyword 180
database
listing data sets 364
statistics tables 347
updating using BMCCPRS 401
Database Space
  Detail Report 665
  Trend Report 666
Database Storage Utilization Report 668
DATACLAS installation option 234
Dataset Extents trigger 477
DATASETSIZING keywords 100
DATASETSIZING POF keyword 181
DATAWK_NBR POF keyword 181
DATAWORK data set 105
DATE installation option 235
DB2 catalog
  accessing 91
  authorization 90
  updating index sampled statistics 244
DB2 catalog, browsing
  and BMCTSTATS 366
  column statistics 372
  index partition statistics 374
  index statistics 373
  table space partition statistics 369
  table space statistics 367
  table statistics 368
DB2 catalog, updating
  column statistics 379
  index partition statistics 382
  index statistics 380
  interactively 375
  overview 365
  table columns 385
  table space partition statistics 378
  table space statistics 375
  table space/index columns 388
  table statistics 376
  using BMCTSTATS 390
  using BMCUPRS 398
DB2 objects
  analyzing trends 408
  creating a list 361
  creating a statistics graph 431
  customizable reports 645
  deleting BMC Software statistics 405
displaying 361
displaying DB2 catalog and BMCTSTATS on the same panel 366
including or excluding from service generation 310
listing 361

822
DB2 objects (continued)

- monitoring with BMCTRIG 458
- running BMCSTATS to populate 343
- simulating changes 436
- simulating changes to 436
- statistics tools 342
- updating the DB2 catalog 375
- wildcard facility 310

DB2 RTS option 508

DB2 Solution Common Code (SCC) 64

DB2 utilities
  - ALTERSEC 577
  - RUNSTATS 139

DB2, supported versions 60

DB2CAT installation option 235

DB2EXIT POF keyword 181

DB2LOAD POF keyword 181

DB2ENTRY installation option 235

DB2WAIT installation option 235

DD statements
  - adding for BMC required services 291
  - DD Statement List panel, services 290
  - DD Statement text field, DD Statement panel 296
  - for report programs 730
  - JCL DD statement dispositions for restarts 567
  - JCL DD statement dispositions for startover 567
  - JCL Generation component 137
  - JCL processing 568
  - omitting in JCL 568
  - preparing a customizable report in batch 644
  - processing of GDG data sets 568
  - required for batch processing 573
  - types 295
  - user-defined programs 258
  - Utility Copy 147
  - Utility Dual Copy 147
  - Utility Dual Recovery Site Image Copy 147
  - Utility Recovery Site Image Copy 147

DEADLINE Shrlevel option 559

Debugging, setting default processing options 114

DEF_GDG_BASE POF keyword 182

DEF_GDG_LIMIT POF keyword 182

DEF_GDG_NOSCR POF keyword 182

DEF_GDG2_LIMIT POF keyword 182

default options. See installation options
defaults, setting user options
  - BMCTRIG Control-M options 122
  - data sets 108
  - debugging 114
  - general 83, 93
  - generation data groups 112
  - JCL jobcard 95
  - JCL static data sets 99
  - JCL STEPLIBs 97
  - non-worklist JCL 119
  - online reorgs 118
  - product options file 125

SHRLEVEL CHANGE 118

SORTWORK data sets 105

tapes 104

user variables 124

utility module names 117

defining user options 82

definitions
  - action generation 260
  - action POF 149
  - Action Report 645
  - ALTERSEC 257, 576
  - Append inserts trigger 477
  - ASCII 795
  - Average Length of RID Chain Report 585
  - Average Monthly Growth Report 645
  - BMCCOPY 257
  - BMCCOPYI 257
  - BMCCPRS 257
  - BMCMOD 257
  - BMCLOAD 257
  - BMCTRIG 257
  - BMCREORG 257
  - BMCSSTATS 257, 346
  - BMCSSTATS Age trigger 482
  - BMCTRIIG 257
  - BCMUNLOAD 257
  - BCMUPRS 257
  - Card trigger 477
  - Catalog Upd Age trigger 482
  - CHEK DA 257
  - CHEK LOB 257
  - Copy Pend trigger 484
  - Copy Pend trigger 484
  - Corrective Action Cross Reference Report 645
  - Data Set List by Database Name Report 645
  - Database Space Detail Report 645
  - Database Space Trend Report 645
  - Database Storage Utilization Report 645
  - Dataset Extents trigger 477
  - Dirty trigger 484
  - DSEXENTS 550
  - DSN1COPY 257
  - DtyIncremt trigger 484
  - Event Recording Report 585
  - Exceptions Report by BMCTRIG Action 645
  - Extents trigger 477
  - FARINDREF 552
  - FAROFTPOS 552
  - Faroff trigger 478
  - FULLCOPY 257
  - Full Copy Age trigger 484
  - INCRCOPY 257
  - Index Analysis Report 585
  - Index Clustering Analysis Report 585
  - Index Clustering Report 645
  - Index Combination Statistics Report 585
  - Index Key Analysis Report 585
  - Index Leaf Distribution Report 585
definitions (continued)

Indexspace Space Detail Report 585
LEAFDIST 550
Leafdist trigger 478
LEVELMIN 551
Levels trigger 479
LOAD 258
Load after copy trigger 484
Load after stats trigger 482
LOB Freespace trigger 479
LOB ORGRATIO trigger 479
Mass del reorg trigger 479
Mass del stats trigger 482
MERGCOPY 258
MODICOPY 258
Modified Table Space Pages Report 645
Mods since copy trigger 484
Mods since reorg trigger 479
Monthly Detail by Tablespace Report 585
NEARINDREF 552
NEAROFFPOS 552
Nostats trigger 482
NumIncremt trigger 485
object set 310
Page Update Analysis Report 586
Pct Dropped Rows trigger 479
PctActivHi trigger 480
PctActivLo trigger 480
PCTCLUS 552
PctClus trigger 480
processing options for BMCTRIG 471
Pseudo Deleted Key trigger 480
QUIESCE 258
Rebd after copy trigger 485
Rebd after stats trigger 483
RECOVER 258
REORDER 551
REORG 258
Reorg after copy trigger 485
Reorg after stats trigger 483
Reorg Pend trigger 480
Reorgspace trigger 481
REORINS 551
REORLEAF 551
REORMODS 551
REORPEND 552
REORSPACE 551, 552
REP REC 258
REP SET 258
REPAIR 258
RESIZE 258
RUNSTATS 258, 346
Space Estimation Report 645
Space Estimation Trend Report 645
Space trigger 481
Space Utilization by Table Report 645
START DB 258

Statistics Exception Report 586
STOP 258
STOP DB 258
STOSPACE 258
SYNC 258
Table Column Detail Report 586
Table Space and Index Extents Report 645
Table Space and Index Storage Limits Report 646
Tables with More than Average Growth Report 646
Tablespace Extent Summary Report 586
Tablespace Offset RIDs Report 586
Tablespace Space Detail Report 586
Totaloff trigger 481
Unclust inserts trigger 481
user options 84
utility commands 257
utility worklist 263
Volume Analysis Report 586
Volume Freespace Detail Report 586
Volume Space Trend Report 646
DELAY Shrlevel option 559
DELETEAGE option 509
deleting
actions 278
data sets automatically 101
object sets 333
dependent objects, Export examples 689
deploying object definitions. See Export 685
destination connections, setting up in Export 698
detect inline opts option 511
DIAG_MSGCLASS POF keyword 182
diagnostic output
data sets 88
DIRTY exception 484
Dirty trigger 484
DIRTYIC exception 484
DISC_DATACLASS POF keyword 182
DISC_DATACLASS_ALT POF keyword 182
DISC_EXPDT POF keyword 182
DISC_MGMTCLASS POF keyword 183
DISC_MGMTCLASS_ALT POF keyword 183
DISC_PREFIX POF keyword 183
DISC_PRIQTY POF keyword 183
DISC_RETPD POF keyword 183
DISC_SECQTY POF keyword 183
DISC_STORCLASS POF keyword 183
DISC_STORCLASS_ALT POF keyword 183
DISC_THRESH POF keyword 183
DISC_UNIT POF keyword 184
DISC_UNIT_ALT POF keyword 184
discard data set 102
DISP_ALLOW_POPUP POF keyword 184
DISP_AUTO_TAB POF keyword 184
DISP_LOCATION POF keyword 184
DISP_OMIT_CHAR POF keyword 185
DISP_STATS POF keyword 185
DISP_VAR_DBUG POF keyword 185
displaying
  current environment 77
  product fixes 79
  statistics for analysis 409
distribution statistics
  columns 356
  on columns 347
  on columns at the partition level 347
DPNAM installation option 235
DRAIN Shrlevel option 559
DRAIN_WAIT Shrlevel option 559
dropping and re-creating objects 404
DROPR_NOIC POF keyword 185
DSEXTENT exception 477
DSEXTENTS exception 550, 552
DSNCHECK44 POF keyword 185
DSNTIAD_PLAN POF keyword 185
DynIncement trigger 484
dynamic allocation
  diagnostic output data set 88
  of output for UNLOAD PLUS 64
  temporary work files 563

E
EAP installation option 235
edit list, definition 70
editing
  action services 279
  actions 274, 279
  corrective action 499
  exception definition for a threshold 493
  information-only fields 69
  input fields 69
  JCL generation options 147
  object sets 327, 334
  object-action priority 503
  program details 292
  secondary quantity 579
  services 279
  threshold 492
EIP installation option 236
electronic documentation 27
ELIMDUPACTS option 463
ELIMDUPOPTS option 464
encoding schemes 795
END ISPF command 71
ENTER ISPF command 71
Enterprise Connection List (ECL) 690
enterprise connections, setting up in Export 696
environment, displaying current 77
EPP installation option 236
ERR_DATACLASS POF keyword 186
ERR_DATACLASS_ALT POF keyword 186
ERR_EXPDT POF keyword 186
ERR_MGMTCLASS POF keyword 186
ERR_MGMTCLASS_ALT POF keyword 186
ERR_PREFIX POF keyword 186
ERR_PRIQTY POF keyword 186
ERR_RETPD POF keyword 186
ERR_SECQTY POF keyword 186
ERR_STORCLASS POF keyword 187
ERR_STORCLASS_ALT POF keyword 187
ERR_THRESH POF keyword 187
ERR_UNIT POF keyword 187
ERR_UNIT_ALT POF keyword 187
error data set 102
error messages
  ASURXLOC 739
  ASURXLSP 734
  ASURXRPFT 734
  ASURXSE 738
  ASURXSQL 731
evaluating BMCTRIG exceptions, overview 472
Event Recording Report 590
exception definitions, specifying to Export 704
exceptions
  APPNDINS 477
  AREOPENED 477
  ARERPENDED 477
  BSTATAGE 482
  CARD 477
copy 483
  COPY#IIC 485
  COPYAGE 484
  COPYMODS 484
  COPYPEND 484
  CSTATAGE 482
  DIRTY 484
  DIRTYIC 484
  DSEXTENT 477
  EXTENTS 477
  FICAGE 484
  IXDIRTY 484
  IXICAGE 484
  LEAFDIST 478
  LEAFFOFF 478
  LEVELINC 478
  LEVELMIN 479
  LEVELS 479
  LOADCOPY 484
  LOADSTAT 482
  LOBFRSPC 479
  LOBORGR 479
  MINSPACE 481
  NONUNIFORM 482, 483
  NORTSCPY 484
  NORTSSTAT 482
  NOSTATS 483
  PACTHI 480
exceptions (continued)

POF keywords 691
preparing your environment 694
referenced objects examples 688
report, reviewing statuses 708
SAF authority 692
selecting destinations to export 706
service definitions 702
setting the primary UIM server host name 692
setting the primary UIM server port 692
setting up destination connections 698
setting up enterprise connections 696
TCP/IP access 691
UIM Server access 691
verifying the list to export 706
viewing status of export job 707
exporting
action definitions 701
exception definitions 704
object definitions 686
object set definitions 703
object-action definitions 703
service definitions 702
EXTENTS exception 477
Extents trigger 477
external functions
ASURXLOC 739
ASUR XLSP 734
ASURXRPT 733
ASURXSE 735
ASURXSQL 730
definitions 719
F

F2 (SPLIT) 72
F9 (SWAP) 72
FARIND exception 477
FARINDREF exception 552
FAROFF exception 478
Faroff trigger 478
FAROFFPOS exception 552
fast path navigation commands 75
FASTSWITCH Shrlevel option 560
FCPY_DATACLASS POF keyword 187
FCPY_EXPDT POF keyword 188
FCPY_MGMTCLASS POF keyword 188
FCPY_PREFIX POF keyword 188
FCPY_PRIQTY POF keyword 188
FCPY_RETPD POF keyword 188
FCPY_SECQTY POF keyword 188
FCPY_STACK POF keyword 188
FCPY_STORCLASS POF keyword 188
FCPY_SUPPRESS_SUFF POF keyword 110, 189
FCPY_UNIT POF keyword 189

excluding objects
from sets 327
from trigger job 555
EXEC_LOAD POF keyword 187
EXECUTE command in ASURXSQL 731
Execution plans, names and descriptions 90
execution, worklist
leanup job step for data sets 101
deleting data sets automatically 101
Export
accessing 695
action definitions 701
allowing default service syntax changes 707
ASU_XP_UIMSRVHOST POF keyword 692
ASU_XP_UIMSRVPORT POF keyword 692
authorization 691
connection recommendations 690
definition validation table 687
dependent objects examples 689
deploying the definitions 707
terprise connection list (ECL) 690
exception definitions 704
how it works 686
launching 695
object set definitions 703
object-action priority definitions 703
overview 60
overwriting existing rows 707
personal connection list (PCL) 690
features
analyzing trends 59
collecting and managing statistics 54
customizing reports 59
deploying object definitions 60
estimating space 60
generating utilities 53
setting and analyzing thresholds 55
unsupported 61

FICAGE exception 484
field
holder 724
information-only 69
input 69
justification in customizable reports 724
line 724

file structure in worklists 266
file tailoring 138, 712
FILT_DATACLASS POF keyword 189
FILT_EXPDT POF keyword 189
FILT_MGMTCLASS POF keyword 189
FILT_PREFIX POF keyword 189
FILT_PRIQTY POF keyword 189
FILT_RETPD POF keyword 189
FILT_SECQTY POF keyword 190
FILT_STORCLASS POF keyword 190
FILT_UNIT POF keyword 190
FKKEY0MSG option 510
flow diagrams
ALTERSEC 578
evaluate BMCTRIG exceptions 473
evaluate exceptions and generate utilities in BMCTRIG 472
evaluate UTA exceptions in BMCTRIG 475
first-time utility job 261
generate BMCTRIG jobs without evaluating exceptions 474
generating standard JCL 527
generating worklist JCL 539
regenerated utility job 262
FREQVAL installation option 236
Full Copy Age trigger 484
FULLCOPY (FULL) command description 257
functions on main menu 72

G
GDG (generation data group)
and user options 140
setting default processing options 112
specifying 110
specifying in copy DD options panel 143
symbolic variable 110, 140
using for data sets 110
GDG_MODELP OOF keyword 190
GDGDEF installation option 236

GDGLIM installation option 236
Generate priority option 512
generating
JCL for GDGs 140
standard JCL, flow diagrams 527
trigger job with exceptions 504
utility job 299
worklist JCL, flow diagrams 539
generation data group. See GDG (generation data group).

graphs
creating 428
display options 431
set options 431
statistics 428
types 428

Group objects option 513

GUIDEDK objects option 513

grouped services
examples 467
general 466
not logged objects 467
object-action priorities 464
processing 468
relative to actions 520
guidelines for triggering REORG 549

H
HASHFAIL POF keyword 190
HASHWARNRC POF keyword 190
HELP ISPFP command 71
Help, online 27
historical database 347
HLQ.UDBCNTL members
product options file 149

I
identifying actions associated with object sets 330
Image Copy Age trigger 484
image copy data sets 147
INCLUDE_SYSPRIN2 POF keyword 190
including objects in sets 327
INCRCOPY (INCR) command description 257
index
analyze partition page group statistics 424
analyze partition statistics 422
analyze statistics 420
browse partition statistics in DB2 catalog 374
browse statistics in DB2 catalog 373
graph summary 428
reorganizing 549
sampled statistics, updating the DB2 catalog 244
simulating changes 437
space estimation 444

Index 827
update partition statistics in DB2 catalog 382
update statistics in DB2 catalog 380
Index Analysis Report 593
Index Clustering Analysis Report 596
Index Clustering Report 672
index columns, updated in the DB2 catalog 388
Index Combination Statistics Report 600
Index Key Analysis Report 604
Index Leaf Distribution Report 606
Indexes option 509
Indexspace Space Detail Report 609
INFILE 295
information sources for utilities 231
initial POF, refreshing 128
input parameters for customizable reports 648
input streams 84
installation options
SYSCPREF 242
installation options
ATBWORKAREA 234
BMCSYNC 234
CPLAN 234
customizing product installation 83
DATACLASS 234
DATE 235
DB2CAT 235
DB2NTRY 235
DB2WAIT 235
DPNAM 235
EAP 235
EIP 236
EPP 236
FREQVAL 236
GDGDEF 236
GDGLIM 236
graphic explanation 148
ISPTLIB 236
JC1, JC2, JC3, JC4, JC5 236
JCLCLEAN 237
JDSNE 237
JPLAN 237
listing 232
LOCATION 237
MGMTCLAS 237
notation 232
OPNDB2ID 237
OPTIMIZECOMMIT 238
OSAUTHCHK 238
overriding default values 148
PODFS 85, 238
PRODUCT 238
QUIESCETIMEVAL 238
RECVSS 239
RECVUNIT 239
RPLAN 240
SL1 240
SL2 240
SL3 240
SL4 240
SL5 240
SPLAN 241
STATAUTH 241
STORCLAS 241
SWPS 241
SWSS 241
SWU 241
SYSCMAX 242
SYSRMAX 243
SYSRMAXU 243
SYSPREF 243
SYSPS 243
SYSSR 243
SYSTRIGS 244
SYSTYPE 244
SZU 244
TAPE1, TAPE2, TAPE3 244
UPDCATIXS 244
UPDNUCAT 245
used in utilities 230
user options 84
WDC 245
WDSN 245
WMC 245
WPS 246
WSC 246
WSS 246
WU 246
ZPLAN 246
installation options module
ASUDOFTS 232
using ampersands 232
installation, post-installation considerations 250
interface
commands 71
field types 69
panel types 70
interim worklist 528
internal KB switch 650
IOALOAD1 POF keyword 191
IOALOAD2 POF keyword 191
ISPF
commands, common 71
file tailoring, used by JCL Generation 138
ISPF (continued)
  file tailoring, used to test changes 712
  profile, storing values in 84
ISPTLIB installation option 236
IX Copy option 511
IX Escalate option 511
IXDIRTY exception 484
IXICAGE exception 484

J

JC1 installation option 236
JC2 installation option 236
JC3 installation option 236
JC4 installation option 236
JC5 installation option 236
JCL
  data set, allocating 88
  DD processing, GDG data sets 568
  DD statement, restarts 567
  DD statement, startovers 567
  flow diagram for standard 527
  flow diagram for worklist 539
  for sample BMCTSTATS job 396
  generated standard, example 533
  job statement, defaults 95
  standard benefits 526
  worklist benefits 538
  worklist format 538
JCL generation
  and restarts 565
  BMCTSTATS utility 139
  data set names required for standard 519
  data set names required for worklist 541
  GDGs 140
  ISPF file tailoring 138
  options for standard 527
  options for worklist 540
  options in BMCTRIG 555
  overview 268
  product options 85
  RUNSTATS utility 139
  user options 85
  VSAM object sampling 139
  worklist 267
JCL jobcard, setting default processing options 95
JCL static data sets, setting default processing options 99
JCL STEPLIBs, setting default processing options 97
JCLCLEAN installation option 237
JCLCLEANUP keyword 101
JCLCLEANUP POF keyword 191
JCLLIB POF keyword 192
JDSNE installation option 237
JESS POF keyword 192
  job card exit in interim worklist 531
  JOB_INCLUDE_MEMBER POF keyword 192

JOBCARD1 POF keyword 192
JOBCARD2 POF keyword 192
JOBCARD3 POF keyword 192
JOBCARD4 POF keyword 192
JOBCARD5 POF keyword 192
JPLAN installation option 237

K

KEEPDICTIONARY Reorg option 561
keywords
  ALTERSEC 257
  BMCCOPY 257
  BMCCOPYI 257
  BMCCPRS 257
  BMCLOAD 257
  BMCMOD 257
  BMCREORG 257
  BMCSSTATS 257, 349
  BMCTRIG 257
  BMCUUNL0D 257
  BMCUUPS 257
  CHEK DA 257
  CHEK LOB 257
  DATASETSIZING B 100
  DATASETSIZING C 100
  DATASETSIZING O 100
  DSN1COPY 257
  FULLCOPY 257
  INCRCOPY 257
  LOAD 258
  LOCKSIZE 249
  MERGCOPY 258
  MODICOPY 258
  QUIESC 258
  RECOVER 258
  REORG 258
  REP REC 258
  REP SET 258
  REPAIR 258
  RESIZE 258
  RUNSTATS 258
  START DB 258
  STOP 258
  STOP DB 258
  STOSPACE 258
  SYNC 258
  use in utility jobs 256

L

layout of reports 646
LEAFDIST exception 478, 550
Leafdist trigger 478
LEAFFOFF exception 478
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

LEFT ISPF command 71
LEVELINC exception 478
LEVELMIN exception 479, 551
LEVELS exception 479
Levels trigger 479
list of utilities and commands 256
LISTDEF_DSN POF keyword 192
listing data sets by database 364
LL_CLIB POF keyword 192
LL_CLIB2 POF keyword 192
LL_CLIB3 POF keyword 192
LL_CLIB4 POF keyword 192
LL_CLIB5 POF keyword 192
LL_LINK POF keyword 192
LL_LINK2 POF keyword 192
LL_LINK3 POF keyword 192
LL_LINK4 POF keyword 192
LL_LINK5 POF keyword 192
LL_MLIB POF keyword 193
LL_MLIB2 POF keyword 193
LL_MLIB3 POF keyword 193
LL_MLIB4 POF keyword 193
LL_MLIB5 POF keyword 193
LL_PLIB POF keyword 193
LL_PLIB2 POF keyword 193
LL_PLIB3 POF keyword 193
LL_PLIB4 POF keyword 193
LL_PLIB5 POF keyword 193
LL_SLIB POF keyword 193
LL_SLIB2 POF keyword 193
LL_SLIB3 POF keyword 193
LL_SLIB4 POF keyword 193
LL_SLIB5 POF keyword 193
LL_TLIB POF keyword 193
LL_TLIB2 POF keyword 193
LL_TLIB3 POF keyword 193
LL_TLIB4 POF keyword 193
LL_TLIB5 POF keyword 193
LL_XML POF keyword 193
LL_XML2 POF keyword 193
LL_XML3 POF keyword 193
LL_XML4 POF keyword 193
LL_XML5 POF keyword 193
LLQ POF keyword 194
Load after copy trigger 484
Load after stats trigger 482
LOAD command description 258
LOAD+_LOAD POF keyword 194
LOADCOPY exception 484
LOADDOPT POF keyword 194
LOADSTAT exception 482
LOB Freespace trigger 479
LOB ORGRATIO trigger 479
LOB SYSREC data set 108
LOBFRSPC exception 479, 553
LOBORGR exception 479
LOCATION installation option 237
LOCKSIZE keyword 249
LOG Reorg option 560
LOGSORT data set 105
LOGWK_NBR POF keyword 194
LOGWK_UNIT POF keyword 194
long object name fields 76
LONGLOG Shrlevel option 559
loop counter 753

M
main menu
description 68
fields 69
functions 72
managing
actions 260
statistics 383
map data set 102
MAP_DATACLASS POF keyword 194
MAP_DATACLASS_ALT POF keyword 194
MAP_EXPDT POF keyword 195
MAP_MGMTCLASS POF keyword 195
MAP_MGMTCLASS_ALT POF keyword 195
MAP_PREFIX POF keyword 195
MAP_PRIQTY POF keyword 195
MAP_RETPD POF keyword 195
MAP_SECQTY POF keyword 195
MAP_STORCLASS POF keyword 195
MAP_STORCLASS_ALT POF keyword 195
MAP_THRESH POF keyword 196
MAP_UNIT POF keyword 196
MAP_UNIT_ALT POF keyword 196
MAPPINGTABLE Shrlevel option 560
Mass del reorg trigger 479
Mass del stats trigger 482
Max reorg parts option 513
MAX_CYL POF keyword 196
MAX_PRIQTY POF keyword 196
MAX_SECQTY POF keyword 196
MAX_UNITCNT POF keyword 196
Maximum qty option 515
MAXRO Shrlevel option 559
MEMLIMIT POF keyword 196
MEMLIMIT system parameter 250
memory requirements 250
MERGCOPY (MERG) command description 258
message files 129
MGMTCLAS installation option 237
MINSPACE exception 481
MODICOPY (MODI) command description 258
Modified Table Space Pages Report 678
Mods since copy trigger 484
Mods since reorg trigger 479
Monthly Detail by Tablespace Report 611
MSGLEVEL option 510
multiple-line commands in worklists 267

N
naming objects 88
navigating the panels 72
NEARINDREF exception 552
NEAROFFPOS exception 552
-NEWJ command 528
NEXT ISPF command 71
NONUNIFORM exception 482, 483
non-worklist JCL
  overriding defaults 119
  setting default processing options 119
NORTSCPY exception 484
NORTSSTAT exception 482
NOSTATS exception 483
Nostats trigger 482
NOTIFY option 510
NumIncremt trigger 485

O
object list
  commands 363
  creating DB2 361
  source column 363
  status field 363
  using DB2 361
object set definitions, specifying to Export 703
object set patterns
  resulting object sets 313
  types 312
object sets
  action services path 335
  act-serv 335
  advantages 311
  associating by browsing 337
  associating by name 338
  browsing 337
  controlling access to 340
  copying 331
  creating for actions 336
  creating outside of actions 322, 324
  definition 310
  deleting 333
  example 316
  identifying associated actions 330
  include/exclude objects 327
  renaming 334
  specification panel 316
  using wildcard 310
  viewing results 339
  ways to create 310
object-action definitions, specifying to Export 703
objects
  adding a new priority 502
  altering and re-creating 404
  BMCSTATS for 409
  changing the pattern 282
  collecting statistics on when changed 248
  creating a DB2 list 361
  deleting priorities 503
  displaying DB2 catalog and BMCSTATS statistics for 366
  displaying DB2 lists 361
  dropping and re-creating 404
  editing priority 503
  establishing priorities for when a corrective action is triggered 464
  evaluating 470
  grouping 284
  maximum number of, specifying 519
  minimum object size, specifying 358
  naming 88
  preventing from running out of extents 576
  restricting sampling by type 360
  sampling large 358
  security 88
  simulating changes to 436
  sizing 358
  specifying which ones to monitor 504
  utilities operating concurrently on 342
  viewing exceptions by 547
  VSAM 139
  VSAM sampling 139
  wildcard characters, using 76
online Help 27, 75
online reorgs
  overriding defaults 118
  setting default processing options 118
OPEN cursor command in ASURXSQ 731
OPNDB2ID installation option 237
OPTIMIZECOMMIT installation option 238
options
  BMCSTATS 349
  BMCTRIG Control-M 122
  BMCTRIG processing, introduction 471
  changing 82
  data sets 108
  debugging and display 114
  defining user 82
  ELIMDUPACTS 463
  ELIMDUPOPTS 464
  generating standard JCL 527
  generating worklist JCL 540
  generation data group 112
  graphic display 431
  how they are used by products 87
  JCL jobcards 95
  JCL static data sets 99
options (continued)
JCL STEPLIBs 97
non-worklist JCL 119
online reorg 118
panel display 94
printing graphs 431
product options file 125
refreshing user 84
saving user 84
setting 82
setting general 93
SHRLEVEL CHANGE 118
SORTWORK data sets 105
system information 94
tapes 104
user variables 124
user, defined 84
utility generation and processing 94
utility module names 117
order of use for utility information sources 231
organization of product panels 74
ORGRATIO exception 553
ORTPARM_DSN POF keyword 197
OSAUTHCHK installation option 238
OUTFILE 295
output data sets for reports 657

P
packages
AEXAUNLD 90
AEXSQLIO 90
PACTHI exception 480
PACTLO exception 480
panels
Action List 274
BMCSSTATS 349
BMCTRIG 487, 506
common elements 70
main menu 68
navigating 72
object set specification 316
organization 74
types 70
Parallel index Sort option 563
PARTLVL option 512
PCPY1_DATACLASS POF keyword 197
PCPY1_DATACLASS_ALT POF keyword 197
PCPY1_EXPDT POF keyword 197
PCPY1_MGMTCLASS POF keyword 197
PCPY1_MGMTCLASS_ALT POF keyword 197
PCPY1_PREFIX POF keyword 197
PCPY1_PRIQTY POF keyword 197
PCPY1 RETPD POF keyword 197
PCPY1_SECQTY POF keyword 198
PCPY1_STACK POF keyword 198
PCPY1_STORCLASS POF keyword 198
PCPY1_ STORCLASS ALT POF keyword 198
PCPY1 SUPPRESS_SUFP POF keyword 110, 198
PCPY1_ THRESH POF keyword 198
PCPY1_UNIT POF keyword 198
PCPY1 UNIT ALT POF keyword 198
PCPY2_DATACLASS POF keyword 199
PCPY2_DATACLASS ALT POF keyword 199
PCPY2_EXPDT POF keyword 199
PCPY2_MGMTCLASS POF keyword 199
PCPY2_MGMTCLASS ALT POF keyword 199
PCPY2_PREFIX POF keyword 199
PCPY2_PRIQTY POF keyword 199
PCPY2 RETPD POF keyword 199
PCPY2_SECQTY POF keyword 199
PCPY2_STACK POF keyword 200
PCPY2_STORCLASS POF keyword 200
PCPY2_ STORCLASS ALT POF keyword 200
PCPY2 SUPPRESS SUFP POF keyword 110, 200
PCPY2_ THRESH POF keyword 200
PCPY2_UNIT POF keyword 200
PCPY2_UNIT ALT POF keyword 200
Pct Dropped Rows trigger 479
PctActivHi trigger 480
PctActivLo trigger 480
PCTCLUS exception 480, 552
PctClus trigger 480
PCTDROP exception 479
percentages, specifying sample 359
Performance Advisor, how it works 781
permanent work data sets 108
Personal Connection List (PCL) 690
PFSHOW ISPF command 71
PGMLIB 295
plan names, finding 77, 729
plans
DASD MANAGER PLUS 89
Execution component 90
POF (product options file)
action definition 149
adding steps to JCL 133
AJXPODAT macro 126
associating with actions 280
creating a user POF 125
description 149
generating reports 129
initial 85
initializing 85
ISPF profile variables 85
keyword descriptions 160
keywords for Export 691
keywords, list of 149
overriding values in SLIBs 131
POF Validation Report 129
populating 85
refreshing the initial POF 128
reusing in a subsequent installation 129
POF (product options file) (continued)
  sample file 149
  setting default processing options 125
  specifying additional options 136
  updating a user POF 126
  using multiple POFs 127
  Variables Initialized with Default report 129
POFDATE POF keyword 200
POFDS installation option 85, 149, 238
post-installation 250
Pqty % if max option 515
PRE_JOBSTEP_INCLUDED POF keyword 201
prefixes 109
PREFORMAT Reorg option 561
prerequisites 27
PREV ISPF command 71
priorities
  accessing system object-action 501
  adding a new object-action 502
  defining system-level object-action 500
  deleting object-action 503
  editing object-action 503
  limiting corrective actions 457
  object-action 464
  ordering objects when generating maintenance jobs 457
  threshold 460, 464
PROC_BMCCHECK_NAME POF keyword 201
PROC_BMCCHECK_STEP POF keyword 201
PROC_BMCCOPY_NAME POF keyword 201
PROC_BMCCOPY_STEP POF keyword 201
PROC_BMCPRS_NAME POF keyword 201
PROC_BMCPRS_STEP POF keyword 202
PROC_BMCLOAD_NAME POF keyword 202
PROC_BMCLOAD_STEP POF keyword 202
PROC_BMCRECOVER_NAME POF keyword 202
PROC_BMCRECOVER_STEP POF keyword 202
PROC_BMCREORG_NAME POF keyword 202
PROC_BMCREORG_STEP POF keyword 202
PROC_BMCRESTATS_NAME POF keyword 203
PROC_BMCRESTATS_STEP POF keyword 203
PROC_BMCMUTIL_NAME POF keyword 203
PROC_BMCMUTIL_STEP POF keyword 203
PROC_BMCUTIL_NAME POF keyword 204
PROC_BMCUTIL_STEP POF keyword 204
PROC_BMCUNLOAD_NAME POF keyword 204
PROC_BMCUNLOAD_STEP POF keyword 204
PROC_BMCUPRS_NAME POF keyword 205
PROC_BMCUPRS_STEP POF keyword 205
PROC_DSN1COPY_NAME POF keyword 205
PROC_DSN1COPY_STEP POF keyword 205
PROC_DSNUTILB_NAME POF keyword 205
PROC_DSNUTILB_STEP POF keyword 205
PROC_GEN_SET_VAR POF keyword 205
PROC_IDCAM_NAME POF keyword 206
PROC_IDCAM_STEP POF keyword 206
PROC_IEFBR14_NAME POF keyword 206
PROC_IEFBR14_STEP POF keyword 206
PROC_TSO_NAME POF keyword 206
PROC_TSO_STEP POF keyword 206
PROC_USE POF keyword 207
PROC_USER_DEF_STEP POF keyword 207
PROC_USER_DEFINED POF keyword 207
product changes 28
product fixes, displaying 79
product functionality 50
PRODUCT installation option 238
product integration
  administrative products for DB2 62
  release level compatibility 64
product options file. See POF
product options file. See POF (product options file)
product support 3
Pseudo Deleted Key trigger 480
PSEUDODL exception 480
publications, related 27
punch data set 102
PUNCH_DATACLASS POF keyword 207
PUNCH_EXPDT POF keyword 207
PUNCH_MGMTCLASS POF keyword 207
PUNCH_PREFIX POF keyword 207
PUNCH_PRIQTY POF keyword 208
PUNCH_SECQTY POF keyword 208
PUNCH_STORCLASS POF keyword 208
PUNCH_UNIT POF keyword 208
Q
QMF reports 641
QUIESCE (QUI) command description 258
QUIESCEINTERVAL installation option 238
R
RCPY1_DATACLASS POF keyword 208
RCPY1_DATACLASS_ALT POF keyword 208
RCPY1_EXPDT POF keyword 208
RCPY1_MGMTCLASS POF keyword 209
RCPY1_MGMTCLASS_ALT POF keyword 209
RCPY1_PREFIX POF keyword 209
RCPY1_PRIQTY POF keyword 209
RCPY1_RETPD POF keyword 209
RCPY1_SECQTY POF keyword 209
RCPY1_STACK POF keyword 209
RCPY1_STORCLASS POF keyword 209
RCPY1_STORCLASS_ALT POF keyword 209
RCPY1_SUPPRESS_SUFF POF keyword 110, 210
RCPY1_THRESH POF keyword 210
RCPY1_UNIT POF keyword 210
RCPY1_UNIT_ALT POF keyword 210
RCPY2_DATACLASS POF keyword 210
RCPY2_DATACLASS_ALT POF keyword 210
RCPY2_EXPDT POF keyword 210
RCPY2_MGMTCLASS POF keyword 210
RCPY2_MGMTCLASS_ALT POF keyword 211
RCPY2_PREFIX POF keyword 211
RCPY2_PRIQTY POF keyword 211
RCPY2_RETDP POF keyword 211
RCPY2_SECQTY POF keyword 211
RCPY2_STACK POF keyword 211
RCPY2_STORCLASS POF keyword 211
RCPY2_STORCLASS_ALT POF keyword 211
RCPY2_SUPPRESS_SUFF POF keyword 110, 211
RCPY2_THRESH POF keyword 212
RCPY2_UNIT POF keyword 212
RCPY2_UNIT_ALT POF keyword 212
real-time statistics (RTS)
  SYSIBM.SYSINDEXSPACESTATS 390
  SYSIBM.SYSTABLESPACESTATS 389
  updating 389
REBALANCE Reorg option 560
Rebd after copy trigger 485
Rebd after stats trigger 483
REBDSTAT exception 483
Rebind option 513, 514
Rebind Version option 514
REBINDFAIL POF keyword 212
REBINDRC POF keyword 212
RECOVER (RECV) command description 258
RECOVERDOPT POF keyword 212
RECV data set 140
RECVMAX installation option 239
RECVMAXU installation option 239
RECVPREF installation option 239
RECVPS installation option 239
RECVSS installation option 239
RECVUNIT installation option 239
referenced objects, Export examples 688
refreshing
  content, UIM Server 774
  initial POF 128
  user options 84
  user options in your profile 84
REGION POF keyword 213
related publications 27
Remove dup actions option 511
renaming a column 404
renaming object sets 334
REORCOPY exception 485
REORDEL exception 551, 553
REORDELS exception 477
REORDSPC exception 479, 553
REORG
  command description (REOR) 258
  corrective action, setting up 557
  guidelines for triggering 549
  with resize option 516, 518, 522, 523, 524
REORG Advisor
  automating reorganizations based on REORG
  candidate table recommendations and BMCTRIG
  evaluations 788
  automating reorganizations based only on REORG
  candidate table recommendations 786
  populating the REORG candidate table 783
  preparing to use 782
  setting up automated reorganizations 785
Reorg after copy trigger 485
Reorg after stats trigger 483
REORG candidate table
  making BIND modifications to retrieve data from 783
REORG candidate table, populating 783
REORG exceptions 461
Reorg options
  KEEPIEXTIONARY 561
  LOG 560
  PREFORMAT 561
  REBALANCE 560
  Resize 561
  Restart 561
  REUSE 560
  SCOPE 560
  UNLOAD 561
Reorg Pend trigger 480
REORG PLUS parameters, unsupported 65
REORG+_LOAD POF keyword 213
REORG_MAPTAB POF keyword 213
reorganization exceptions 476
REORGDOPT POF keyword 213
REORGINS exception 478
REORGGLOB exception 480
Reorgspace trigger 481
REORINS exception 551, 553
REORLEAF exception 480, 551
REORMDEL exception 479, 551, 553
REORMODS exception 479, 551, 553
REORPEND exception 480, 551, 553
REORSPACE exception 481, 551, 552
REORSTAT exception 483
REPORT 295
report data set 88
report layout for customizable report 722
REPORT option 509
reports
  action 657
  adding a customizable report to the menu 656
  Average Length of RID Chain 587
  Average Monthly Growth 660
  batch customizable 653
  batch generation 640
  Corrective Action Cross Reference 662
  customizable 645
Index Analysis 593

Index Clustering 672

Index Clustering Analysis 596

Index Combination Statistics 600

Index Key Analysis 604

Index Leaf Distribution 606

Indexspace Space Detail 609

input parameters for customizable 648

internal KB switch 650

layout 646

locating 584

menu 583

Modified Table Space Pages 674

Monthly Detail by Tablespace 611

output 657

QMF descriptions 642

redundancy 587

running customizable 650

Space Estimation 675

Space Estimation Trend 677

Space Utilization by Table 679

standard 585

Statistics Exception 617

Table Column Detail 623

Table Level Space Utilization 679

Table Space and Index Extents 682

Table Space and Index Storage Limits 680

tables with more than average growth 681

Tablespace Extent Summary 626

Tablespace Offset RIDs 629

Tablespace Space Detail 631

Update Analysis 614

Volume Analysis 635

Volume Free Space Detail 637

Volume Space Trend 683

REPT_DATACLASS POF keyword 213

REPT_DATACLASS_ALT POF keyword 213

REPT_EXPDT POF keyword 214

REPT_MGMTCLASS POF keyword 214

REPT_MGMTCLASS_ALT POF keyword 214

REPT_PREFIX POF keyword 214

REPT_PRIQTY POF keyword 214

REPT_RETPD POF keyword 214

REPT_SECQTY POF keyword 214

REPT_STORCLASS POF keyword 214

REPT_STORCLASS_ALT POF keyword 214

REPT_THRESH POF keyword 215

REPT_UNIT POF keyword 215

REPT_UNIT_ALT POF keyword 215

requirements

authorization to collect statistics 383

estimating space 436

memory 250

reducing system resource 247

system 61

resize

allowing objects to be resized down 515

exit, providing 517

jobs option 515

limitations 517

space considerations 517

values and BMCTRIG 516

with REORG 516, 518, 522, 523, 524

Resize % pqty options 515

Resize % sqty option 515

RESIZE command description 258

Resize Down option 515

Resize jobs option 515

Resize Reorg option 516

Restart program 567

Restart Reorg option 516

restarting

BMCTRIG-generated job 569

BMCTRIG-generated job, manually 571

with batch JCL generation 565

restricting access

other databases 92

plans 89

product database 92

RETRY Shrlevel option 559

RETRY_DELAY Shrlevel option 560

return codes

ASURXLOC 739

ASURXLSLP 734

ASURXRPT 734

ASURXSE 738

ASURXSQL 731

batch programs 306

REUSE Reorg option 560

REXX

comments 740

concatenation operator 741

continuation characters 740

creating user-defined exceptions 494

customizable reports, listing 656

delimiters 740

EXEC PARM 726, 729

in customizable reports 59, 740

JCL to run customized report 729

quotation marks, SQL 728

sample program 720

sample program using ASURXLOC 739
REXX (continued)
  upper and lower case 740
  variables 739
RIGHT ISPF command 71
ROLLBACK command in ASURSQL 731
Round qty by option 515
ROWID SYSREC data set 108, 111
ROWS/KEYS exception 483
RPLAN installation option 240
rpt.page# variable in customizable reports 724
rptcode variable 734
RUNSTATS
  (RUNS) command description 258
  definition 346
  grouped services 466
  utility, JCL generation 139
RUNTIME_HLQ POF keyword 215

S
SAF authority for Export 692
sample
  annotated program 741
  complete program 756
  customizable report 721
  customizable report program 720
  files for Performance Advisor 784
sampling
  in BMCSTATS 353
  restricting by object type 360
  table columns 360
Save option 509
saving user options in your profile 84
SCC (DB2 Solution Common Code) 64
SCHED_TRIG_CTLM_JOBS POF keyword 215
SCOPE Reorg option 560
secondary quantity. See ALTERNSEC
security
  accessing plans 89
  BMC object 88
  controlling access 307, 340
  other databases 92
  product database 92
  setting authorizations 83
selection list, definition 70
service definitions, specifying to Export 702
service syntax, controlling access to 307
services
  BMCTRIG information sources 230
  command descriptions 257
  controlling access to 307
  copying 285
  creating 279
  non-BMCTRIG information sources 231
  rebind option 513
  submitted through BMCTRIG 455
  use in utility jobs 256
  using installation options 230
setting up
  customizable report 650
  QMF reports feature 643
  REORG corrective action 556
  trigger job for REORG 553
SHRLEVEL CHANGE 118
Shrlevel options
  DEADLINE 559
  DELAY 559
  DRAIN 559
  DRAIN_WAIT 559
  FASTSWITCH 560
  LONGLOG 559
  MAPPINGTABLE 560
  MAXRO 559
  RETRY 559
  RETRY_DELAY 560
  TIMEOUT 559
Simple Space Estimation (SSE) feature 72
simulating changes to DB2 objects 436
sizing objects 358
skeleton library compiler. See SLIB (skeleton library)
  changing an ISPF skeleton 711
  compiling 710, 713
  ISPF file tailoring 712
  processing 713
  runtime report 714
  runtime report summary 715
  runtime unit 713
testing changes 712
SMS (Storage Management Subsystem) 107, 111, 112
Sort JCL Sort option 563
Sort options
  Parallel index 563
  Sort JCL 563
  SORTDEVT 563
  SORTNUM 563
  WORKDDN1 template 563
SORTDEVT Sort option 563
SORTNUM Sort option 563
SORTOnnn data set 102
SORTOUT data set 102, 108
SORTPnnn data set 102
SORTWK_NBR POF keyword 215
SORTWK_PRIQTY POF keyword 215
SORTWK_SECQTY POF keyword 216
SORTWK_UNIT POF keyword 216
SORTWORK data set 105
space estimation
  index 444
  introduction 436
  overview 60
  resize 517
  simulating changes to DB2 objects 436
  table space 437
Space Estimation Report 675
Space Estimation Trend Report 677
SPACE exceptions 481
Space trigger 481
Space Utilization by Table Report 679
specifying
  data set names for standard JCL 519
  data set names for worklist JCL 541
  data set sizing 101
  exclusions in trigger job 555
  GDGs in copy DD options panel 143
  JCL generation options in BMCTRIG 555
  job generation options in BMCTRIG 555
  sampling percentages 359
  trigger values 496
SPLAN installation option 241
SPLIT ISPF command 72
SQL considerations, customizable reports 727
SQL Performance for DB2
  making BIND modifications to retrieve data from the
  REORG candidate table 783
  Performance Advisor tool 780
  providing more robust evaluations 779
  sample file 784
SQLCODE variable 732
SQLEXP_LOAD POF keyword 216
SQLSTATE variable 732
SRTOUT_DATACLASS POF keyword 216
SRTOUT_DATACLASS_ALT POF keyword 216
SRTOUT_EXPDT POF keyword 216
SRTOUT_MGMTCLASS POF keyword 216
SRTOUT_MGMTCLASS_ALT POF keyword 216
SRTOUT_PREFIX POF keyword 217
SRTOUT_PRIQTY POF keyword 217
SRTOUT_RETPD POF keyword 217
SRTOUT_SECQTY POF keyword 217
SRTOUT_STORCLASS POF keyword 217
SRTOUT_STORCLASS_ALT POF keyword 217
SRTOUT_THRESH POF keyword 217
SRTOUT_UNIT POF keyword 217
SRTOUT_UNIT_ALT POF keyword 218
SSE ISPF command 72
START DB (STA DB) command description 258
starting
  DASD MANAGER PLUS 67
  Export 695
  UIM Server 775
starting over, BMCTRIG-generated job 569
STATAUTH installation option 241
statistical trends 407–46
statistics
  accessing function 360
  analyzing 408
  analyzing column 417
  analyzing index 420
  analyzing index partition 422
  analyzing index partition page group 424
  analyzing table 416
  analyzing table space 375, 409
  analyzing table space partition 411
  analyzing table space partition page group 414
  authorization required 383
  benefits of utility 346
  BMCCPRS 256, 257
  browsing column 372
  browsing DB2 and BMCSTATS 366
  browsing index 373
  browsing index partition 374
  browsing table 368
  browsing table space 367
  browsing table space partition 369
  collecting frequency 236
  column value analysis 419
  database tables 347
  deleting 405
  deleting after altering or dropping objects 404
  Exception Report 617
  graphs 428
  introduction 342
  managing 383
  not collected for some objects 344
  overview 54
  RUNSTATS, definition 346
  storage group analysis 425
  table space/index columns updated 388
  tips for collecting 344
  updating column 376
  updating DB2 catalog interactively 375
  updating DB2 catalog, overview 365
  updating index 380
  updating index partition 382
  updating table 376
  updating table space partition 378
  using BMCSTATS to update DB2 catalog 390
  using BMCUPRS to update DB2 catalog 398
  volume analysis 427
Statistics Exception Report 617
statistics exceptions 482
STATMDEL exceptions 482
STATMODS exception 482
STATS exceptions 461
STEP_INCLUDE_MEMBER POF keyword 218
STEPLIB libraries, setting default values 95
STOP command description 258
STOP DB (STO DB) command description 258
stopping the UIM Server 768
STOPWAIT POF keyword 218
STOPWTSECS POF keyword 218
storage group statistics analysis 425
STORCLAS installation option 241
STORAGE (STOS) command description 258
submitting a utility job 299
summary of changes 28
support, customer 3
supported DB2 versions 60
SUPPRESS_COMMENTS POF keyword 218
SUTnnn data set 102
SWAP ISPF command 72
SWPS installation option 241
SWSS installation option 241
SWU installation option 241
symbolic variables
GDG 110
supported 259
SYNC command description 258
SYNCDELETE POF keyword 218
synonyms, finding 77, 729
syntax options
Bypass Exceptions 509
DB2 RTS 508
DELETEAGE 509
Detect inline opts 511
FKEYOMSG 510
Generate priority 512
Group objects 513
Indexes 509
IX Copy 511
IX Escalate 511
Max reorg parts 513
Maximum qty 515
MSGLEVEL 510
NOTIFY 510
PARTLVL 512
Pqty % if max 515
Rebind 514
Rebind Version 514
Remove dup actions 511
REPORT 509
Resize % qty 515
Resize % sqty 515
Resize Down 515
Resize jobs 515
Round qty by 515
Save 509
System Triggers 508
syntax options, BMCTRIG 507
SYSC data set 140
SYSCMAX installation option 242
SYSCMAXU installation option 242
SYSCOPY data set 108
SYSCPREF installation option 242
SYSCPS installation option 242
SYCSS installation option 242
SYSCUNIT installation option 242
SYSDISC data set 108
SYSERR data set 108
SYSEXEC POF keyword 218
SYSMAP data set 108
SYSPUNCH data set 108
SYSREC data set 108
SYSREC data sets
LOB 108
ROWID 108
SYSRMAX installation option 243
SYSRMAXU installation option 243
SYSRPREF installation option 243
SYSRPS installation option 243
SYSRSS installation option 243
SYSRUNIT installation option 243
System Authorization Facility (SAF) for Export 692
system requirements 61
system resource requirements, reducing 247
System Triggers option 508
SYSTEM_MLIB POF keyword 219
SYSTRIGS installation option 244
SYSTYPE installation option 244
SYST data set 108
SYSUT_DATACLASS POF keyword 219
SYSUT_DATACLASS_ALT POF keyword 219
SYSUT_EXPDT POF keyword 219
SYSUT_MGMTCLASS POF keyword 219
SYSUT_MGMTCLASS_ALT POF keyword 219
SYSUT_PREF POF keyword 219
SYSUT_PRIQTY POF keyword 219
SYSUT_RETPD POF keyword 219
SYSUT_SECQTY POF keyword 220
SYSUT_STORCLASS POF keyword 220
SYSUT_STORCLASS_ALT POF keyword 220
SYSUT_THRESH POF keyword 220
SYSUT_UNIT POF keyword 220
SYSUT_UNIT_ALT POF keyword 220
SYSUTnnn data set 102
SZDEVT installation option 244
SZDEVT installation option 244
SYSUT POF keyword 220

T

table
analyze statistics 416
browse statistics in DB2 catalog 368
columns updated in DB2 catalog 385
graph summary 429
update statistics in DB2 catalog 376
Table Column Detail Report 623
TABLE option, collecting frequency statistics 236
table space
analyze partition page group statistics 414
analyze partition statistics 411
table space (continued)
analyze statistics 409
browse partition statistics in DB2 catalog 369
browse statistics in DB2 catalog 367
graph summary 430
reorganizing 552
simulating changes 437
space estimation 437
update partition statistics in DB2 catalog 378
update statistics in DB2 catalog 375
updated in DB2 catalog 388
Table Space and Index Extents Report 682
Table Space and Index Storage Limits Report 680
Tables with More than Average Growth Report 681
Tablespace Extent Summary Report 626
Tablespace Offset RIDs Report 629
Tablespace Space Detail Report 631
TAPE_EXPDT POF keyword 220
TAPE_RETPD POF keyword 221
TAPE_VOLCNT POF keyword 221
TAPE1 installation option 244
TAPE1 POF keyword 221
TAPE2 installation option 244
TAPE2 POF keyword 221
TAPE3 installation option 244
TAPE3 POF keyword 221
tapes
setting default processing options 104
using 110
TCP/IP access to Export 691
technical support 3
TEMPLATE POF keyword 221
temporary work data sets 105
TEMPUNIT POF keyword 221
thresholds
accessing system 489
deleting 492
ing 492
for alternate unit 111
for exceptions 476
in BMCTRIG command syntax 457
installation of a set of common ones 457
object-action priorities, precedence of 464
priorities 460
reorganization guidelines for 549
setting 457
setting exception 476
setting in the BMCTRIG syntax 485
system-level 457, 488
user-defined 456
viewing the long description of the exception 493
TIMEOUT Shrlevel option 559
TIMEPARAM POF keyword 221
tips
collecting statistics 344
column analysis 419
column value analysis 420
index analysis 422
index partition analysis 424
storage group analysis 426
table analysis 417
table space analysis 411
table space partition analysis 413
table space partition page group analysis 415
volume analysis 427
TOTALIND exception 481
TOTALOFF exception 481
Totaloff trigger 481
tracing
TRCLOGDD 772
tracing, TRCLOGDD 773
TRCLOGDD 772, 773
trend analysis, overview 59
triggers
automating REORG job 553
detecting exceptions job 504
determining values 496
evaluating exceptions, overview 472
rebind 513
resizing 515
viewing exceptions 525
TRIGNOCOPY POF keyword 461
TRIGNOREORG keyword 461
TRIGNOSTATS keyword 461
troubleshooting
Export statuses 708
IBM TCP/IP 776
reversed port number 777
started task 775, 776
TRTCH POF keyword 221
Truncation location field 76
TSO exit, to set default option 95
TSOEXIT POF keyword 222
TSOSUBEXIT POF keyword 222
types of graphs 428
U
-UDSJ command 528
UIM Server
access to Export 691
accessing BMC UIM Server Commands web page 769
accessing the server commands web page 769
authorization to access the BMC UIM server
commands web page 769
changing the security authorization timeout feature
temporarily 771
enabling or disabling specific tracing options
temporarily 773
enabling or disabling the tracing option temporarily
772
refreshing content 774
refreshing the content 774
UIM Server (continued)
starting 775
starting and stopping 767
stopping 768
troubleshooting problems 775
verifying that the server is running 768
viewing active users 771

ULLQ POF keyword 222
UNCLUST exception 481, 553
Unclust inserts trigger 481
Unicode
data in SYSIN streams 61
unit name, ROWID SYSREC data set 111
UNLD_FREF_DATACLASS POF keyword 222
UNLD_FREF_DIRBLOCK POF keyword 222
UNLD_FREF_MGMTCLASS POF keyword 222
UNLD_FREF_PREFIX POF keyword 222
UNLD_FREF_PRIQTY POF keyword 222
UNLD_FREF_SECQTY POF keyword 223
UNLD_FREF_STORCLASS POF keyword 223
UNLD_FREF_UNIT POF keyword 223
UNLD1_DATACLASS POF keyword 223
UNLD1_DATACLASS_ALT POF keyword 223
UNLD1_EXPDT POF keyword 223
UNLD1_MGMTCLASS POF keyword 223
UNLD1_MGMTCLASS_ALT POF keyword 224
UNLD1_PFX POF keyword 224
UNLD1_RETPD POF keyword 224
UNLD1_SECQTY POF keyword 224
UNLD1_STACK POF keyword 224
UNLD1_STORCLASS POF keyword 224
UNLD1_STORCLASS_ALT POF keyword 224
UNLD1_SUPPRESS_SUFF POF keyword 110, 225
UNLD1_THRESH POF keyword 225
UNLD1_UNIT POF keyword 225
UNLD1_UNIT_ALT POF keyword 225
UNLD2_DATACLASS POF keyword 225
UNLD2_DATACLASS_ALT POF keyword 225
UNLD2_EXPDT POF keyword 225
UNLD2_MGMTCLASS POF keyword 225
UNLD2_MGMTCLASS_ALT POF keyword 226
UNLD2_PFX POF keyword 226
UNLD2_RETPD POF keyword 226
UNLD2_SECQTY POF keyword 226
UNLD2_STACK POF keyword 226
UNLD2_STORCLASS POF keyword 226
UNLD2_STORCLASS_ALT POF keyword 226
UNLD2_SUPPRESS_SUFF POF keyword 110, 227
UNLD2_THRESH POF keyword 227
UNLD2_UNIT POF keyword 227
UNLD2_UNIT_ALT POF keyword 227
UNLD3_DATACLASS POF keyword 227
UNLD3_EXPDT POF keyword 227
UNLD3_MGMTCLASS POF keyword 227
UNLD3_PREFIX POF keyword 228
UNLD3_RETPD POF keyword 228
UNLD3_STORCLASS POF keyword 228
UNLD3_SUPPRESS_SUFF POF keyword 228
UNLD3_UNIT POF keyword 228
UNLD4_DATACLASS POF keyword 228
UNLD4_MGMTCLASS POF keyword 228
UNLD4_PFX POF keyword 228
UNLD4_STORCLASS POF keyword 229
UNLD4_SUPPRESS_SUFF POF keyword 229
UNLD4_UNIT POF keyword 229
unload data set used by utilities 102
UNLOAD PLUS unsupported options 65
UNLOAD Reorg option 561
UNLOAD+ LOAD POF keyword 229
UNLOADDOPT POF keyword 229
unsupported
COPY PLUS options 65
DB2 versions 60
features of DB2 61
REORG PLUS parameters 65
UNLOAD PLUS options 65
UP ISPF command 72
Update Analysis Report 614
updating
column statistics 379
database using BMCCPRS 401
DB2 catalog interactively 375
index partition statistics 382
index statistics 380
table space partition statistics 378
table space statistics 375
table statistics 376
UPDCATIXS installation option 244
UPDNUCAT installation option 245
User Interface Middleware (UIM) Server 64
user options
and GDGs 140
definition 84
general 83
JCL Generation 85
panel display 94
refreshing in profile 84
saving in profile 84
setting general 93, 94
SORTWORK data sets 105
system information 94
utility generation and processing 94
user POF
creating 125
updating directly 126
updating in options panels 126
using multiple 127
user variables
character 124
numeric 124
setting default processing options 124
user-defined programs 258
user-defined services
  example of creating 288
    JCL elements for generating services 258
    symbolic variables used in syntax 259
using
  ampersands in installation options module 232
  BMCTRICG, introduction 454
  BMCUPRS instead of BMCSTATS 398
  DB2 object lists 361
    stand-alone path 321
  wildcards in object sets 310
utility automation, example of 761, 767, 779
utility generation
  accessing 270
  and processing options 94
  command descriptions 257
  copying actions 273
  copying services 285
  first job, flow diagrams 261
  generate/submit a job 299
  introduction 254
  list of utilities and commands 256
  overview 53
  regenerated job, flow diagram 262
  sample JCL for BMCSTATS job 396
  worklist 263
utility module names, setting default processing options
  117
utility products for DB2 63

V
value line 724
variable option notation 232
variables
  in customizable reports 724
  in the SQL 727
  job sequence 531
  JOBSEQ 531
  modifying 138
  option notation 232
  REORG mapping table 557
  report-dependent 725
  resolution of &GDG and &GDGDU 147
  REXX 739
  symbolic, supported 259

W
WDC installation option 245
WDSN installation option 245
wildcard
  characters 76
  expansion in worklists 265
    using in object set 310
WMC installation option 245
work data sets
  permanent 108
  temporary 105
  used by utilities 102
WORK_DATACLASS POF keyword 230
WORK_MGMTCLASS POF keyword 230
WORK_STORCLASS POF keyword 230
WORKDDN1 template Sort option 563
worklist
  column format 266
  commands 264
  comment lines 267
  definition 263
  exceptions 525
  file structure 266
  interim, commands 528
  interim, structure 528
  JCL 538
  JCL generation 267
  multiple-line commands 267
  wildcard expansion 265
worklists
  data set 88
  workload balancing 468
  WPS installation option 246
  WRKnnn data set 102
  WSC installation option 246
  WSS installation option 246
  WU installation option 246

Z
zaps, displaying 79
zoom, ISPF command (BMCZOOM) 71
ZPLAN installation option 246