DASD MANAGER PLUS for DB2
User Guide

Supporting

Version 11.2 of DASD MANAGER PLUS for DB2
Version 11.2 of Database Performance for DB2

May 2015
Contacting BMC Software

Several methods are available for contacting BMC Software.

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

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>1 713 918 8800</td>
<td>1 713 918 8000</td>
</tr>
<tr>
<td>2103 CITYWEST BLVD</td>
<td>or</td>
<td></td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td>1 800 841 2031</td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

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

© Copyright 2004-2016 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.

Explorer and Together are trademarks or registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

DB2, DB2 Universal Database, DRDA, ECKD, FlashCopy, GDDM, IBM, IBM z, IMS, MVS, OS/390, QMF, RACF, System z, System z10, VTAM, z/Architecture, z/OS, z10, and z9 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

UNIX is the registered trademark of The Open Group in the US and other countries.

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

Restricted rights legend

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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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

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

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

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

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

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

- Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)

- In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.

## Contents

**About this book** 15
- Related publications ................................................................. 15
- Conventions .............................................................................. 16
- Syntax statements ........................................................................ 16
- Syntax diagrams .......................................................................... 17
- Summary of changes .................................................................... 19
  - Version 11.2.00, May 2015 ..................................................... 19
  - Version 11.1.00, June 2013 ...................................................... 23
  - Version 10.1.00 April 2011 ...................................................... 31

**Chapter 1**  Introduction to DASD MANAGER PLUS for DB2 35
- Overview of DASD MANAGER PLUS for DB2 .............................. 35
  - Solution integration ............................................................... 36
  - Product benefits ..................................................................... 37
  - Product features ..................................................................... 39
- System setup .............................................................................. 45
  - DB2 support .......................................................................... 45
  - Features of DB2 that DASD MANAGER PLUS does not currently support ...................................................... 45
  - System requirements ............................................................. 46
  - Software requirements .......................................................... 46
- Interaction with other products from BMC Software ................. 47
  - Administrative products for DB2 ........................................... 47
  - Backup and Recovery products for DB2 ................................ 47
  - Utility products for DB2 ........................................................ 48
  - Associated common components that DASD MANAGER PLUS uses .............................................................. 48
  - Release-level compatibility .................................................... 49

**Chapter 2**  Getting started with the ISPF interface 51
- Overview of the ISPF interface .................................................. 51
  - The DASD MANAGER PLUS Main Menu ................................. 51
  - ISPF panel types ................................................................... 53
- ISPF command descriptions ..................................................... 54
- Panel navigation ....................................................................... 55
- Fast Path Navigation ................................................................ 58
- Space estimation ..................................................................... 58
Help for panels .................................................................59
Wildcard characters ..........................................................59
Long object name fields .....................................................59
Viewing current environment information ..........................60

Chapter 3   Setting up DASD MANAGER PLUS 63

Before you set up the Interactive System Productivity Facility (ISPF) interface ....63
Control of your environment by using options ........................63
  Use of the installation options ........................................64
  Using the user options ..................................................65
  Use of the product options .............................................66
  Using local keywords to override options .......................67
  Putting it all together ....................................................68
Data set allocation ............................................................69
Naming objects .................................................................69
Execution and security setup .............................................70
  Access to DASD MANAGER PLUS plans ......................70
  Execution component plans ........................................71
  Access to the DB2 catalog ............................................72
  Access to the DASD MANAGER PLUS database ............73
  Access to other databases ..........................................73
Setting options and authorizations ....................................73
  Setting general user options ........................................74
  Setting the JCL options for job cards ............................75
  Setting the JCL options for STEPLIBs ............................78
  Setting the JCL options for static data sets ....................79
  Setting the JCL options for tapes ..................................84
  Setting the JCL options for temporary work data sets .......86
  Setting the JCL options for permanent data sets .............88
  Setting the JCL generation data group (GDG) options ......93
  Setting the JCL debugging, display, and Execution options ..95
  Setting the installation options module names for BMC utilities 97
  Setting the non-worklist JCL options ............................98
  Setting the BMCTRIG Control-M options ....................101
  Setting user variables ................................................102
  Creating a user POF ....................................................103
  Updating a user POF ..................................................104
  Using multiple POFs ..................................................106
  Refreshing the initial POF ..........................................107
  Generating POF reports .............................................107
  Reusing a POF in a subsequent installation ...................108
Overriding POF values in SLIBs ................................................................. 109
Adding steps to the JCL .............................................................................. 111
Additional job generation options in DASD MANAGER PLUS .................. 113
Customizing DASD MANAGER PLUS (for experienced users) ................. 114
Generating JCL .......................................................................................... 114
Specifying generation data groups .............................................................. 117
Overriding options ..................................................................................... 124
Overview of product options ........................................................................ 125
Overriding installation options .................................................................... 203
Descriptions of the installation options ......................................................... 206
Post-installation considerations .................................................................... 219
Recommendations for reducing system resource requirements .................. 220
MEMLIMIT system parameter ..................................................................... 222

Chapter 4         Maintaining and generating Service Actions 223
Overview of Service Actions ......................................................................... 223
Service Actions ......................................................................................... 224
Services ...................................................................................................... 224
Service Action Generation function .............................................................. 231
Working with Service Actions ...................................................................... 239
Listing Service Actions ............................................................................. 239
Creating a Service Action .......................................................................... 241
Editing Service Action properties ................................................................. 244
Copying a Service Action .......................................................................... 246
Deleting Service Actions ........................................................................... 248
Generating Service Actions ....................................................................... 248
Working with Services ................................................................................ 252
Listing Services .......................................................................................... 252
Editing Service Syntax ............................................................................... 254
Editing Service properties ......................................................................... 255
Example of creating a User Defined service .............................................. 260
Deleting a User Defined Service ................................................................. 264
Editing Services ......................................................................................... 265
Previewing generated syntax ..................................................................... 272
Restarting a Service Action ......................................................................... 274
Understanding return codes for batch programs ........................................ 277
Access to actions, services, and service syntax .......................................... 278

Chapter 5         Using object sets 279
Before you begin using object sets ............................................................. 279
Overview of object sets ............................................................................. 279
Wildcards for object names ................................................................. 280
Patterns for object sets ........................................................................ 281
Specify Object Set panel ..................................................................... 285
Example of defining object sets ......................................................... 288
Working with object sets ..................................................................... 289
Defining and Maintaining object sets .................................................. 290
Creating an Object Set ...................................................................... 291
Creating dynamic SQL Object Sets ..................................................... 294
Editing Object Sets specifications ....................................................... 298
Identifying actions associated with an object set ................................. 300
Renaming object sets ........................................................................ 301
Copying object sets .......................................................................... 302
Deleting object sets .......................................................................... 303
Editing object set properties .............................................................. 304
Viewing object set results ................................................................... 304
Associating object sets with action services ....................................... 306
Associating a new object set with an action service ......................... 306
Associating an existing object set with an action service .................. 307
Associating an existing object set with a new action ......................... 309
Controlling access to Object Sets ......................................................... 311

Chapter 6 Collecting and managing statistics .......................... 313
Overview of collecting and managing statistics .............................. 313
Quick reference of statistics-related tasks ......................................... 314
Tips for collecting statistics .............................................................. 315
Collecting statistics .......................................................................... 317
IBM RUNSTATS utility ....................................................................... 317
BMCSTATS utility ............................................................................. 317
DASD MANAGER PLUS historical database ..................................... 318
BMCSTATS options .......................................................................... 319
Sampling statistics ........................................................................... 332
Minimum size requirements .............................................................. 332
Specifying sampling percentages ..................................................... 333
Restricting sampling by object type ................................................ 333
Sampling table columns .................................................................. 333
Sampling by using a seed value ....................................................... 334
Accessing the statistics display features ......................................... 334
Displaying DB2 object lists ............................................................... 335
Creating a DB2 object list ............................................................... 336
Listing data sets used by a database ............................................... 338
Updating DB2 catalog statistics ....................................................... 339
Browsing and updating statistics ................................................................. 341
DB2 and BMCSTATS statistics ................................................................. 341
DB2 table space statistics ...................................................................... 342
DB2 table statistics .................................................................................. 343
DB2 table space partition statistics ......................................................... 344
DB2 column statistics ............................................................................ 345
DB2 index statistics ................................................................................ 346
DB2 index partition statistics ................................................................. 348
Updating the DB2 catalog interactively .................................................. 348
Table space statistics update ................................................................. 349
Table statistics update .......................................................................... 350
Table space partition table statistics update ........................................ 351
Column statistics update ...................................................................... 352
Index statistics update ......................................................................... 353
Index partition statistics update ........................................................... 355
Managing DB2 and BMC Software statistics ........................................... 355
Having authority to collect statistics ....................................................... 356
Updating DB2 columns by DASD MANAGER PLUS ............................... 357
Updating DB2 real-time statistics ......................................................... 363
Updating the DB2 catalog by using BMCSTATS .................................... 363
Updating the DB2 catalog by using BMCUPRS ...................................... 370
Updating DASD MANAGER PLUS statistics by using BMCCPRS ....... 374
Altering, renaming, or dropping and re-creating objects ....................... 376
Deleting BMC Software statistics .......................................................... 377

Chapter 7  Analyzing statistical trends .................................................... 379
Before you begin analyzing statistical trends .......................................... 379
Analysis of BMC Software statistics ....................................................... 379
Analyzing BMC Software statistics ....................................................... 380
BMC Software statistics for DB2 objects ................................................. 380
Table space statistics ............................................................................ 381
Table space partition statistics ............................................................... 382
Table space partition pagegroup statistics ............................................. 385
Table statistics ....................................................................................... 386
Column statistics .................................................................................. 388
Column value statistics ........................................................................ 390
Index statistics ..................................................................................... 390
Index partition statistics ...................................................................... 392
Index partition pagegroup statistics ...................................................... 394
Storage group statistics ....................................................................... 395
Volume statistics .................................................................................. 396
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphing statistics</td>
<td>397</td>
</tr>
<tr>
<td>Creating a statistics graph</td>
<td>397</td>
</tr>
<tr>
<td>Setting graphic display options</td>
<td>400</td>
</tr>
<tr>
<td>Estimating space requirements by using statistics</td>
<td>405</td>
</tr>
<tr>
<td>Simulating changes to DB2 objects</td>
<td>405</td>
</tr>
<tr>
<td>Estimating space requirements for a table space</td>
<td>406</td>
</tr>
<tr>
<td>Estimating space requirements for an index</td>
<td>412</td>
</tr>
<tr>
<td>Estimating space requirement based on user-specified values</td>
<td>416</td>
</tr>
<tr>
<td>Estimating space requirements for a table space</td>
<td>417</td>
</tr>
</tbody>
</table>

**Chapter 8  Analyzing objects by using BMCTRIG**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before you begin to use BMCTRIG to analyze objects</td>
<td>425</td>
</tr>
<tr>
<td>Overview of the main features of BMCTRIG</td>
<td>427</td>
</tr>
<tr>
<td>System triggers</td>
<td>428</td>
</tr>
<tr>
<td>Exception conditions</td>
<td>429</td>
</tr>
<tr>
<td>Suppressing categories of exceptions from analysis</td>
<td>432</td>
</tr>
<tr>
<td>Job generation</td>
<td>432</td>
</tr>
<tr>
<td>Methods of running BMCTRIG</td>
<td>436</td>
</tr>
<tr>
<td>Setting up BMCTRIG</td>
<td>440</td>
</tr>
<tr>
<td>Selecting objects to evaluate</td>
<td>440</td>
</tr>
<tr>
<td>Automating corrective actions</td>
<td>440</td>
</tr>
<tr>
<td>BMCTRIG processing options</td>
<td>440</td>
</tr>
<tr>
<td>Sample workflows</td>
<td>441</td>
</tr>
<tr>
<td>Setting exception thresholds</td>
<td>445</td>
</tr>
<tr>
<td>Shared BMCTRIG exceptions</td>
<td>446</td>
</tr>
<tr>
<td>Reorganization-related exceptions</td>
<td>449</td>
</tr>
<tr>
<td>Copy-related exceptions</td>
<td>455</td>
</tr>
<tr>
<td>Thresholds in BMCTRIG syntax</td>
<td>457</td>
</tr>
<tr>
<td>System-level thresholds</td>
<td>459</td>
</tr>
<tr>
<td>Creating user-defined exceptions</td>
<td>466</td>
</tr>
<tr>
<td>Identifying threshold values</td>
<td>469</td>
</tr>
<tr>
<td>Managing corrective actions</td>
<td>469</td>
</tr>
<tr>
<td>Setting a corrective action in BMCTRIG syntax</td>
<td>470</td>
</tr>
<tr>
<td>Defining system-level corrective actions</td>
<td>470</td>
</tr>
<tr>
<td>Defining system-level object-action priorities</td>
<td>473</td>
</tr>
<tr>
<td>Setting up a BMCTRIG job</td>
<td>476</td>
</tr>
<tr>
<td>Creating an action with BMCTRIG</td>
<td>477</td>
</tr>
<tr>
<td>Specifying BMCTRIG syntax options</td>
<td>479</td>
</tr>
<tr>
<td>Generating a BMCTRIG job</td>
<td>495</td>
</tr>
<tr>
<td>Using worklist format or standard JCL format</td>
<td>497</td>
</tr>
<tr>
<td>Viewing exceptions that BMCTRIG produces</td>
<td>514</td>
</tr>
</tbody>
</table>
Working with reorganization guidelines for thresholds ............................................ 517
Generating an automatic reorganization with BMCTRIG ..........................................521
Setting up a REORG corrective action ...........................................................................524
Working with restarts and batch JCL generation ........................................................ 532
   Using BMCTRIG-generated job restart ....................................................................532
   Restart and Startover processing .............................................................................533
   Overview of the restart program .............................................................................533
   Restarting jobs that BMCTRIG generates ...............................................................535
   Using batch processing ......................................................................................... 538

Chapter 9   Changing the secondary quantity  541
Prerequisites for changing the secondary quantity ..................................................... 541
Understanding the ALTERSEC utility .........................................................................541
Altering the secondary quantity (SQTY) .....................................................................544

Chapter 10   Producing reports  547
Before you begin working with reports ........................................................................ 547
Standard reports .......................................................................................................... 547
   Standard reports descriptions ..................................................................................548
   Organization of report options ................................................................................550
   Generating the Average length of RID chain report ...............................................551
   Event recording report .............................................................................................553
   Index analysis report ...............................................................................................556
   Index clustering analysis report ..............................................................................559
   Index combination statistics report ...........................................................................562
   Index keys analysis report .......................................................................................565
   Index leaf distribution report ..................................................................................567
   Indexspace space detail report ...............................................................................569
   Monthly detail by tablespace report .......................................................................571
   Page update analysis report ....................................................................................574
   Statistics exception report .......................................................................................576
   Table column detail report .......................................................................................581
   Table space extent summary report .........................................................................582
   Table space offset RIDs report ................................................................................585
   Tablespace space detail report ...............................................................................587
   Volume analysis report ............................................................................................590
   Volume free space detail report .............................................................................592
Model JCL for batch generation .....................................................................................594
QMF reports .................................................................................................................594
Customizable reports ....................................................................................................597
   Understanding customizable reports ...........................................................................598
Input parameters ................................................................. 600
Internal KB switch ............................................................... 603
Creating customizable reports ........................................... 603
Customizable report samples .............................................. 610
Action report ........................................................................ 610
Average Monthly Growth report ........................................ 612
Corrective action cross reference report ......................... 613
Data Set List by database name report ............................... 614
Database space detail report ................................................ 615
Database space trend report ................................................ 617
Database storage utilization report ...................................... 618
Exception report for BMCTRIG action ............................ 620
Index clustering analysis report .......................................... 621
Modified table space pages report ...................................... 623
Space estimation report ...................................................... 625
Space estimation trend report .............................................. 626
Space utilization by table report .......................................... 628
Table space and index storage limits report ...................... 630
Tables With More Than Average Growth report .............. 631
Table space and index extents report ................................. 632
Volume space trend report .................................................. 634

Chapter 11  Exporting and deploying product definitions 637
Overview of exporting object definitions ............................ 637
How the Export utility works ............................................... 637
Recommendations for setting up connections .................... 641
Connection list types .......................................................... 641
Required authorizations for using Export ......................... 642
SAF authority to access the CCL ......................................... 644
Task summary for exporting object definitions .................. 646
Preparing your environment for exporting ....................... 647
Accessing the Export utility .................................................. 648
Setting up connections ........................................................ 649
Setting up enterprise connections ...................................... 649
Setting up destination connections ...................................... 651
Specifying definitions to export ......................................... 654
Specifying action definitions .............................................. 654
Specifying service definitions ............................................ 655
Specifying object set definitions ........................................ 656
Specifying object-action priority definitions ..................... 656
Specifying exceptions definitions ..................................... 657
About this book

This book contains detailed information about the DASD MANAGER PLUS 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
- IBM z/OS operating system
- Job control language (JCL)
- Interactive System Productivity Facility (ISPF)

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

**Note**

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

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

Related publications

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

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

• View individual product documents (books and notices) within the “A – Z Supported Product List” (https://webapps.bmc.com/support/faces/az/supportlisting.jsp).

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

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

Conventions

This document uses the following special conventions:

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

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

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

Syntax statements

This topic explains conventions for showing syntax statements.
A sample statement follows:

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

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

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

**Syntax diagrams**

The following figure shows the standard format for syntax diagrams:
The following example illustrates the syntax for a hypothetical DELETE statement. Because the FROM keyword, alias variable, and WHERE clause are optional, they appear below the main command line. In contrast, the tableName variable appears on the command line because the table name is required. If the statement includes a WHERE clause, the clause must contain a search condition or a CURRENT OF clause. (The searchCondition variable appears on the main line for the WHERE clause, indicating that this choice is required.)

The following guidelines provide additional information about syntax diagrams:

- Read diagrams from left to right and from top to bottom.
- A recursive (left-pointing) arrow above a stack indicates that you may choose more than one item in the stack.
- An underlined item is a default option.
- If a diagram shows punctuation marks, parentheses, or similar symbols, you must enter them as part of the syntax.
- In general, IBM commands, keywords, clauses, and data types are displayed in uppercase letters. However, if an item can be shortened, the minimum required
portion might be shown in uppercase letters, with the remainder in lowercase (for example, CANcel).

- The following conventions apply to variables in syntax diagrams:
  - Variables are typically displayed in lowercase letters and are always italicized.
  - If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words (for example, databaseName).

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.

Version 11.2.00, May 2015

This release includes the following enhancements:

BMCSTATS enhancements

DASD MANAGER PLUS now supports the following BMCSTATS enhancements:

- Additional zIIP OFFLOAD
  BMCSTATS offloads additional eligible workloads to utilize the IBM System z Integrated Information Processor (zIIP), reducing the Total Cost of Ownership (TCO). You can enable zIIP offloading globally through the zIIP=E default options specification or you can define options through the zIIP E syntax.

- REPEATABLE keyword
  BMCSTATS page sampling has been enhanced to include a specified seed value through the usage of the REPEATABLE keyword. Usage of the default or a specified seed value enables the page sampling algorithm to ensure that repeated executions of BMCSTATS return the same results.

- Non Partition Index (NPI) key monitoring
  NPI key monitoring provides storage relief during key column statistics collection for multi-data sets (MUDS) through definable parameters. BMCSTATS now provides three new keywords, NPICACHEACTION, NPICACHETHRESH, and NPICACHEDSNUM, that can be used to tune high-storage consuming NPI objects. Values for these keywords can be globally established in the default
options module (DOPTs) and can be individually overridden in the BMCSTATS syntax.

- **RESET ACCESSPATH option**
  BMCSTATS supports a new RESET ACCESSPATH option that resets the access path statistics for all tables and indexes for the specified table space on the BMCSTATS statement.

- **RESETHISTORY option**
  BMCSTATS supports a new RESETHISTORY option. When you use RESET ACCESSPATH, RESETHISTORY ACCESSPATH, BMCSTATS inserts a row for each object in SYSIBM.SYSTABLES_HIST and SYSIBM.SYSINDEXES_HIST to indicate that a reset has been performed.

- **Ability to suppress null indexes**
  BMCSTATS supports suppressing null indexes. All table statistics derived from an index defined with the EXCLUDE NULL KEYS clause are adjusted by the number of excluded NULL values.

**BMCTTRIG enhancements**

DASD MANAGER PLUS now supports the following BMCTTRIG enhancements:

- The new ICPYPEND exception that detects table spaces and indexes placed in informational COPY-pending (ICOPY) status

- The new ACHKPEND exception that detects table spaces placed in Auxiliary CHECK-pending (ACHKP) status

- The new CHKPEND exception that detects table spaces placed in CHECK-pending (CHKP) status

- The new RBDPEND exception that detects indexes placed in REBUILD-pending (RBDP,RBDP*,PSRDB) status

**User interface enhancements**

The DASD MANAGER PLUS user interface now features an improved look and feel.

Enhancements include an improved process flow for defining and generating service actions, collecting and managing statistics, and detecting and correcting object exceptions. For detailed information on these options, see the utilities documentation. The following changes and new menus provide better organization of the DASD MANAGER PLUS features:

- **BMCCCOPY (COPY PLUS) option changes**
— INIT
— GENSYSPAGES
— RESYNC
— CHANGELIMIT
— SNAP

■ BMCMOD option changes
— MAXRECDAYS
— MAXIMUM DAYS
— MAXIMUM LOGS
— MINIMUM COPIES
— MINIMUM FULLCOPIES
— OFFSITE
— ON DSNOTFOUND
— ON ERROR NOTSUPPORTED (can now accept keyword END)
— ON ERROR BADSTATUS (can now accept keyword END)
— ON NOTRECOVERABLE
— SITETYPE
— SYSLLGRNX

■ BMCREORG option changes
— ARCHDDN
— ARCHFORMAT
— DDLIN
— FORCE (can now accept keyword REPORTONLY)
— FSFALLBACK
— FSTHRESHOLD
— IMPLICIT_TZ
— LOG NO
— MAPPINGDATABASE
— MAPTEXIT
— RBALRSN_CONVERSION
— REBLANCE ON
— SHORTMEMORY
— SORTDATA

■ REORG option changes
— LOGRANGES
— DRAIN_ALLPARTS
— SWITCHTIME
— NEWMAXRO
—SORTCLUSTER
—SORTNPSI
—RBALRSN_CONVERSION
—PARALLEL
—RECLUSTER
—HISTOGRAM NUMCOLS
—HISTOGRAM NUMQUANTILES
—SELECT COLGROUP COLUMNS
—COLUMN SELECTION
—COLGROUP FREQVAL COUNT
—COLGROUP HISTOGRAM NUMQUANTILES
—HISTOGRAM NUMCOLS
—HISTOGRAM NUMQUANTILES

- Runstats option changes
  —RESET ACCESSPATH
  —RESETHISTORY ACCESSPATH
  —TABLESAMPLE SYSTEM
  —REPEATABLE
  —USE PROFILE
  —INCLUDE NPI
  —SET PROFILE
  —FROM EXISTING STATS
  —UPDATE PROFILE
  —DELETE PROFILE

**JCL generation and execution changes**

This release also includes support for Extended Address Volume (EAV) data sets for the JCL generation and execution component of DASD MANAGER PLUS.

**Additional enhancements**

The following additional enhancements are included:

- Capture of statistical information

  DASD MANAGER PLUS now uses the ASU_HISTORY product table to capture statistical information about BMCSTATS and BMCTRIG executions. The LOGMETRICS=Y option enables you to collect this statistical information, which can be analyzed to see how these components are being used in your environment.
Archive Enabled tables

DASD MANAGER PLUS object sets now support archive tables. You can use object set tablespace specifications to include or exclude related archive table spaces. If you specify Include ACH Y in an object set definition, BMCSTATS and BMCTRIG include archive-related table spaces.

End of support for IBM DB2 Version 9

Starting with this release, DASD MANAGER PLUS does not support IBM DB2 Version 9.

DASD MANAGER PLUS supports the following versions of DB2:

- IBM DB2 Version 11
- IBM DB2 Version 10 new-function mode (NFM)

Version 11.1.00, June 2013

This release fixes known problems and includes the following enhancements:

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

- **31-bit 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 slows 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 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 0 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</td>
<td>REORGDATASPACE</td>
<td>REORDSPC</td>
<td>Allows you to specify whether reorganize a table space on the percentage of data that occupies the data set</td>
</tr>
<tr>
<td>Panel field</td>
<td>Keyword</td>
<td>Exception name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</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>REORGDELETES</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>REORGINserts</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>
<tr>
<td>Reorg on leaf page usage</td>
<td>REORGLEAF</td>
<td>REORLEAF</td>
<td>Allows you to specify whether to reorganize an index based on the percentage of leaf pages that are empty due to pseudo-deletes</td>
</tr>
<tr>
<td>No RTS Statistics</td>
<td>NORTSTATS</td>
<td>NORTSTAT</td>
<td>Allows you to specify whether to collect statistics when statistical data is not available in the DB2 RTS tables</td>
</tr>
<tr>
<td>No RTS Copy</td>
<td>NORTSCOPY</td>
<td>NORTSCPY</td>
<td>Allows you to specify whether to copy data when copy information is not available in the DB2 RTS tables</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 REALTIMESTATS option
  Current syntax using the REALTIMESTATS option automatically uses 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 for DB2
- RECOVERY 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
- TOTALPAGESPCT
- 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
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 runs 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.DBCLIB 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.

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

**End of support for DB2 Version 8**

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

**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 with 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 that are added to a unique index, 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 TIMESTAMP 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 whether the tablespace 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 “Estimating space requirement based on user-specified values” on page 416.

- 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_KEYTGTDIST (synonym BMCASU_SKTGDIST) contains one or more rows for the first key target of an extended index key.

  - BMCATSnn.RS_KEYTGTDISTSTATS (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 (AREOPENDING keyword and AREOPENP 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.

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 the chapter that discusses 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

The DASD MANAGER PLUS for DB2 User Guide 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 the DASD MANAGER PLUS for DB2 User Guide.
Summary of changes

- 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 is no longer supported in future releases of this product.
Introduction to DASD MANAGER PLUS for DB2

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.

For more information, view the Quick Course DASD MANAGER PLUS - Getting Started.

Overview of DASD MANAGER PLUS for DB2

By using DASD MANAGER PLUS, you can gather and store statistics for physical objects in a DB2 production environment.

In addition, you can 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
- Exports object definitions
Solution integration

DASD MANAGER PLUS for DB2 is a component of the Database Performance for DB2 solution and the new BMC Performance for DB2 Databases solution. Customers who acquire the new BMC Performance for DB2 Databases solution benefit from the features of the following individual products and technologies.

Product components

The Database Performance for DB2 solution includes the following product components:
- DASD MANAGER PLUS for DB2
- REORG PLUS for DB2
- SNAPSHOT UPGRADE FEATURE for DB2 (SUF)

In addition to the Database Performance for DB2 utilities, the new BMC Performance for DB2 Databases solution also includes BMC Workbench for DB2.

Technology components

The Database Performance for DB2 solution includes the following technologies:
- BMCSORT (AUP)
- JCL Generation and Execution (AEX)
- 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 DB2 Solution Common Code (SCC)
- DB2 Utilities Common Code (D2U)
- BMC Common Statistics

Features

This solution provides you with the following features:

- Help determining which required maintenance tasks for your DB2 objects can be automated
- An improved and easy-to-use graphical user interface helps you to navigate and perform tasks more efficiently
- A common interface that enables users of all experience levels to execute and automate database maintenance tasks
Improved DB2 performance, utilizing thresholds and statistics to generate reorganizations of the data only when needed. This feature also allows you to perform reorganizations without disrupting application access to the data.

The ability to enable you 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.

Help to determine when an exception condition warrants a corrective action.

The ability to 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.

Enhancements to 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

This table below describes the key benefits of DASD MANAGER PLUS.

Table 1: Product benefits

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring and managing of database</td>
<td>With BMCSTATS, you can perform the following tasks:</td>
</tr>
<tr>
<td>performance</td>
<td>■ Update DB2 catalog 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</td>
<td>DASD MANAGER PLUS provides object sets, which are enhanced filters. Object sets enable you to perform</td>
</tr>
<tr>
<td>performance</td>
<td>one action on a group of objects, for instance, by using patterns for including and excluding objects</td>
</tr>
<tr>
<td></td>
<td>from the object set specification and specifying SQL text.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Using object sets” on page 279.</td>
</tr>
</tbody>
</table>
Benefit | Description
--- | ---
**Improved problem detection and maintenance activities** | DASD MANAGER PLUS and the BMCTRIG utility provide exceptions to monitor for threshold violations. You can also define your own exceptions and thresholds. 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 “Analyzing objects by using BMCTRIG” on page 425. For a list of the exceptions that DASD MANAGER PLUS provides, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

**Increased productivity** | 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.

**Deploying object definitions** | 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 “Exporting and deploying product definitions” on page 637.

Also, you can use DASD MANAGER PLUS to solve some common problems in a DB2 production environment as shown in Table 2 on page 38:

**Table 2: Common problems that DASD MANAGER PLUS can solve**

<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 <strong>Update DB2 Catalog = Yes</strong>.</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 <strong>all table spaces</strong> (%.%) , <strong>Index = Yes</strong>, and <strong>Extents = 50</strong>. 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 <strong>80 percent</strong>. 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 provides the following features:

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 catalog 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 RUNSTATS, 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 IBM Resource Access Control Facility (RACF) authority to read the data set.

**Set and analyze 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
- 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 SQL Performance for DB2 product
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

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 Service 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 available in DASD MANAGER PLUS

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

**Customizable 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.

**Analyzing trends**

DASD MANAGER PLUS can help you perform trend analysis when you tune 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.

**Deployable object definitions**

If you 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**

The following table shows the versions of the IBM DB2 Universal Database (UDB) for OS/390 and z/OS that DASD MANAGER PLUS supports.

**Table 3: Supported versions of DB2 UDB for OS/390 and z/OS**

<table>
<thead>
<tr>
<th>DASD MANAGER PLUS version</th>
<th>DB2 9</th>
<th>DB2 10</th>
<th>DB2 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.00</td>
<td>Supported</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td>11.1.00</td>
<td>Supported</td>
<td>Supported</td>
<td>Not supported</td>
</tr>
<tr>
<td>11.2.00</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 in this publication, for the generally available (GA) release of Versions 10 and 11 of IBM DB2.
Version 11.2.00 of DASD MANAGER PLUS 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, DISCARDDN, 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

DASD MANAGER PLUS has the following system requirements:

- IBM System z10 processor or later 64-bit z/Architecture processor
- z/OS Version 1.10 or later

Software requirements

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

- Version 2.4.01 of BMCSORT
- Version 11.2.00 of the DB2 Utilities Common Code (D2U)
- Version 11.1.00 of the DB2 Solution Common Code (SCC)
- Version 11.2.00 of the BMC Common Statistics (ATS)
- Version 6.1.00 or later of either the EXTENDED BUFFER MANAGER (XBM) or the SNAPSHOT UPGRADE FEATURE (SUF) in order to offload eligible processing to a zIIP processor

If you use the XBMID option to specify an XBM subsystem, the specified subsystem must be at this maintenance level. If you do not specify a particular XBM subsystem and ZIIP ENABLED is in effect, DASD MANAGER PLUS searches for an XBM subsystem at this level.
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 for DB2 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 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
- RECOVERY MANAGER for DB2

DASD MANAGER PLUS 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:
Utility products for DB2

BMC Software offers the following utility products for DB2:

- LOADPLUS for DB2
- UNLOAD PLUS for DB2
- REORG PLUS for DB2

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 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 listed in the following table.

For more information, see the Installation System documentation.

Table 4: 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>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. This component is installed during DASD MANAGER PLUS installation but is maintained separately from DASD MANAGER PLUS.</td>
</tr>
</tbody>
</table>
Component | Description
---|---
DB2 Utilities Common Code (D2U) | D2U is a set of technologies that provides common processes for the BMC Utility products for DB2 and this product. The D2U code is automatically installed with the products that use it. This component is installed during DASD MANAGER PLUS installation but is maintained separately.
BMC Common Statistics component (ATS) | The BMC Common Statistics component is a technology that collects statistics and updates repository tables for DASD MANAGER PLUS. This component is installed during DASD MANAGER PLUS installation but is maintained separately.
User Interface Middleware (UIM) Server | 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.

**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 5 on page 49, Table 6 on page 50, Table 7 on page 50, and Table 8 on page 50.

**Table 5: Unsupported COPY PLUS options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>
### Table 6: Unsupported LOADPLUS options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 7: Unsupported REORG PLUS parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORKDDN</td>
<td>Overrides the default ddname or ddname prefix of the work data sets</td>
</tr>
</tbody>
</table>

### Table 8: Unsupported UNLOAD PLUS options

<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>
</tbody>
</table>
| CMRATIO     | 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:  
  - When HURBA is specified  
  - As a default because there are no DB2 statistics  
  - When the statistics are out of date |
| DELETEFILES | 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 |
| INFILE option VSAMDD | Specifies a data set other than the DB2 table space data set as the source of the input data |
Getting started with the ISPF interface

This chapter familiarizes you with the ISPF interface, navigation, and commands.

Overview of the ISPF interface

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. (To see the panel, see Getting started with the ISPF interface on page 51.)

The DASD MANAGER PLUS 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 the following:

- Options for defining and generating service actions
- Methods for collecting and managing statistics
- Options for detecting and correcting object exceptions
- Methods for generating reports
- Options for defining and managing services
- Options for defining and managing object sets
- Methods for managing user options which include setting user and graphic options and viewing the current environment
- Options for exporting object definitions
A list that describes newly supported features for the current version of the product, and other information about the product

--- Note ---

The figure in “Panel navigation” on page 55 displays a map of the primary panels that open when you select an 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 “Setting up DASD MANAGER PLUS” on page 63, 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**.

The following figure displays the DASD MANAGER PLUS main options panel.

**Figure 1: DASD MANAGER PLUS 11.02.00 Main Menu**

DEAE ------------------ DASD MANAGER PLUS 11.02.00 Main Menu -----------

Command ===> 

Select an option. Then press Enter.

1. Service Actions    - Define and Generate
2. Statistics         - Collect and Manage
3. Object Exceptions  - Detect and Correct
4. Reports            - Generate
5. Services           - Define and Manage
6. Object Sets        - Define and Manage
7. User Options       - Manage
8. Object Definitions - Export

?  About This Release

Current SQLID . . . ERVICEA

Table 9 on page 52 describes the common fields that appear on the Main Menu and on other DASD MANAGER PLUS panels.

**Table 9: Common information and input fields**

<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>
</tbody>
</table>
The title line of the Main Menu includes the product name, version, and maintenance level of DASD MANAGER PLUS that is currently installed. **Note:** If you call DASD MANAGER PLUS Customer Support for assistance, you need the version and maintenance level from the Main Menu panel.

An error message overwrites any panel number. Other types of messages, such as warnings, also appear in this space.

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.

You can use this field if you have authorization. The default value is your TSO logon ID.

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 z/OS subsystem in the upper left corner

## ISPF command descriptions

Most ISPF commands in DASD MANAGER PLUS work in the same way as they do in other ISPF applications.

Table 10 on page 54 describes the most commonly used ISPF commands.

### Table 10: Common ISPF commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCZOOM</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 to display. On most panels you can use the F4 key to view the full names. To expand a truncated value, place the cursor over the field and press F4. To return to the original display, press F3.</td>
</tr>
<tr>
<td>CANCEL</td>
<td>Returns to the previous panel without saving any change that you made on the current panel</td>
</tr>
<tr>
<td>DOWN (or F8)</td>
<td>Scrolls the panel down</td>
</tr>
<tr>
<td></td>
<td>More: + on a panel indicates that more information is available below the current line. Scrolling is available on a Model 2 3270 mainframe terminal, which uses a 24-line by 80-column display.</td>
</tr>
<tr>
<td>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 commands</td>
</tr>
<tr>
<td></td>
<td>For a sequence of related panels, pressing Enter validates the information on the current panel and displays the next panel in the sequence.</td>
</tr>
<tr>
<td>HELP (or F1)</td>
<td>Provides panel-level Help</td>
</tr>
<tr>
<td>LEFT (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 is available to the left.</td>
</tr>
</tbody>
</table>
Panel navigation

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 the table below.

Table 11: Navigation from the main menu options

<table>
<thead>
<tr>
<th>Option</th>
<th>Result</th>
</tr>
</thead>
</table>
| Service Actions | Make a selection and enter an action name or search criteria. Type an action in the action name field and press Enter. The product displays an editable list of actions matching your entry appear. You can also perform the following:  
  - Create new actions  
  - Edit the Services processed by the Action  
  - Generate action jobs  
  - Edit action properties  
  - Copy and delete actions  
  - Create reports for the Action |
<table>
<thead>
<tr>
<th>Option</th>
<th>Result</th>
</tr>
</thead>
</table>
| **Statistics**         | Collect, copy, and list statistics. (Space Estimation is available when listing statistics). You can also perform the following:  
■ Manage existing service actions for BMCSTATS, BMCCPRS, and BMCUPRS  
■ Create BMCSTATS, BMCCPRS, and BMCUPRS service actions |
| **Object Exceptions**  | Detect and correct object exceptions (changes) in databases by using BMCTRIG. You can also perform the following:  
■ Define and manage exception detection conditions  
■ Define and manage Object-Action priorities  
■ Manage existing and create new BMCTRIG service actions  
■ Restart WORKLIST format jobs as well as view logged exceptions that require Corrective Action generation |
| **Reports**            | Type a report option and press **Enter**. You can generate and view reports on the data accumulated in the DASD MANAGER PLUS database and the DB2 catalog. You can customize many of these reports. |
| **Services**           | List, edit and delete existing Services and service syntax options; and create new user-defined Services. You can also list Action impact. |
| **Object sets**        | List, copy, delete, and edit existing object set conditions; and create new object sets. You can also list action impacts and results. |
| **User Options**       | Select any of the following options for customization:  
■ General Options  
■ JCL generation options  
■ Graphic Display options  
■ Current environment information |
| **Object definitions** | Define object definitions for exporting. |
| **About This Release** | Review newly supported features for the current version of the product and other information about the product. |

Figure 2 on page 57 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.

**Figure 2: Organization of DASD MANAGER PLUS panels**

![Diagram of panel organization](image-url)
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. Table 12 on page 58 lists the products and commands.

**Table 12: 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 *BMC Products and Solutions for DB2 Customization Guide*.

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. You can initiate space estimation by entering **SSE** on the **COMMAND line** of DASD MANAGER PLUS, CATALOG MANAGER, or CHANGE MANAGER. For more information, see “Analyzing statistical trends” on page 379.
Help for panels

For information about how to use a panel while you are using it, press the F1 function key.

Wildcard characters

You can use wildcard characters to search for and to list DB2 and DASD MANAGER PLUS objects.

Table 13 on page 59 describes the wildcard characters that DASD MANAGER PLUS supports.

Table 13: 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>_ (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>* (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 74.
For viewing (and where applicable, editing) the contents of a long object name field, many of the panels also include a 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

If you want to display the information about the current DASD MANAGER PLUS environment, from the main menu select User Options => Current environment information.

Figure 3 on page 60 displays the DASD MANAGER PLUS plans and aliases panel.

#### Figure 3: DASD MANAGER PLUS plans and aliases

<table>
<thead>
<tr>
<th>Alias</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>ASU112.ACT_SEQ_SVC</td>
</tr>
<tr>
<td>BMCASU_ACTION</td>
<td>ASU112.ACTION</td>
</tr>
<tr>
<td>BMCASU_CORR_ACTS</td>
<td>ASU112.CORR_ACTS</td>
</tr>
<tr>
<td>BMCASU_EXCEPTION_DFN</td>
<td>ASU112.EXCEPTION_DFN</td>
</tr>
<tr>
<td>BMCASU_HISTORY</td>
<td>ASU112.ASU_HISTORY</td>
</tr>
<tr>
<td>BMCASU_KEYTARGETS</td>
<td>ATS112.RS_KEYTARGETS</td>
</tr>
<tr>
<td>BMCASU_KEYTARGETSTATS</td>
<td>ATS112.RS_KEYTARGETSTATS</td>
</tr>
<tr>
<td>BMCASU_LOBSTATS</td>
<td>ATS112.RS_LOBSTATS</td>
</tr>
<tr>
<td>BMCASU_OBJ_ACT</td>
<td>ASU112.OBJ_ACT</td>
</tr>
<tr>
<td>BMCASU_OBJ_ACT_JC</td>
<td>ASU112.OBJ_ACT_JC</td>
</tr>
<tr>
<td>BMCASU_SCOLDIST</td>
<td>ATS112.RS_SCOLDIST</td>
</tr>
<tr>
<td>BMCASU_SCOLDISTSTA</td>
<td>ATS112.RS_SCOLDISTSTA</td>
</tr>
<tr>
<td>BMCASU_SCOLLSTATS</td>
<td>ATS112.RS_SCOLLSTATS</td>
</tr>
<tr>
<td>BMCASU_SCOLLUMNS</td>
<td>ATS112.RS_SCOLLUMNS</td>
</tr>
<tr>
<td>BMCASU_SERVICE</td>
<td>ASU112.SERVICE</td>
</tr>
<tr>
<td>BMCASU_SEVENTS</td>
<td>ASU112.SEVENTS</td>
</tr>
<tr>
<td>BMCASU_SEXCEPTION2</td>
<td>ASU112.EXCEPTIONS2</td>
</tr>
<tr>
<td>BMCASU_SINDEXES</td>
<td>ATS112.RS_SINDEXES</td>
</tr>
<tr>
<td>BMCASU_INDEXEXIT</td>
<td>ATS112.RS_INDEXEXIT</td>
</tr>
<tr>
<td>BMCASU_INDEXPART</td>
<td>ATS112.RS_INDEXPART</td>
</tr>
<tr>
<td>BMCASU_SIXPART_DIS</td>
<td>ATS112.RS_SIXPART_DIS</td>
</tr>
<tr>
<td>BMCASU_SKGIDIST</td>
<td>ATS112.RS_SKGIDIST</td>
</tr>
<tr>
<td>BMCASU_SKGIDISTSTA</td>
<td>ATS112.RS_SKGIDISTSTA</td>
</tr>
<tr>
<td>BMCASU_SSTRGROUP</td>
<td>ATS112.RS_SSTRGROUP</td>
</tr>
<tr>
<td>BMCASU_STABLEPART</td>
<td>ATS112.RS_STABLEPART</td>
</tr>
<tr>
<td>BMCASU_STABLES</td>
<td>ATS112.RS_STABLES</td>
</tr>
<tr>
<td>BMCASU_STABLESPACE</td>
<td>ATS112.RS_STABLESPACE</td>
</tr>
<tr>
<td>BMCASU_STS_PART_DIS</td>
<td>ATS112.RS_STS_PART_DIS</td>
</tr>
<tr>
<td>BMCASU_SVC_DD</td>
<td>ASU112.SVC_DD</td>
</tr>
</tbody>
</table>
This panel shows the date of maintenance, options module, plan names, aliases, table names, and other useful information about the product.

If you want 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.

If you want 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.
Viewing current environment information
Setting up DASD MANAGER PLUS

This chapter shows you how to set up DASD MANAGER PLUS in your environment and includes instructions for advanced users.

Before you set up the Interactive System Productivity Facility (ISPF) interface

Before you set up DASD MANAGER PLUS for the Interactive System Productivity Facility (ISPF) interface, ensure that you have completed the installation process.

Note
When you logon to TSO, specify a minimum size of 8192.

For more information, see the Installation System documentation.

Control of your environment by using options

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 do the following:

- 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 206.

Note
For more information about user options for graphic display, see “Graphing statistics” on page 397. For information about the option for displaying the maintenance and current environment, see “Long object name fields” on page 59. You can also set the Graphic Display Options (from the User Options panel) to determine how statistics graphs will look. “Analyzing statistical trends” on page 379 describes this task fully.

Use of the installation options

Several values that are defined in the installation options module control the default operating environment. Components of DASD MANAGER PLUS use the global values that are stored in the installation options module to determine how to process information. When you install DASD MANAGER PLUS, the OS/390 and z/OS Installation System generates the installation options module. The module contains an assembly-language program with an options macro.

To customize the installation of DASD MANAGER PLUS for all users of the installation, edit the default values in the installation options module. The default name of the DASD MANAGER PLUS installation options module is ASUDOPTS. 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 205 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 are now stored in the product options file (POF). For more information, see “Use of the product options” on page 66.

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 “User-defined programs or services” on page 228. 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

1 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, see “Overriding options” on page 124.

**Note**

For more information about refreshing user options, see the *BMC Products and Solutions for DB2 Customization Guide*.

---

**Use of 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.UBMCCNTL data set. The POF does not require assembly and linkage and does not need to reside in an APF-authorized data set.

When you install the products, only one POF is created. This initial POF is initialized and populated with the default ISPF variables and values from the installation panels. This POF is shared among several products, if those products are installed at the same time. 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 125.

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.

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.
Example

- If the value of the BMC_COPY_LOAD keyword is BMC1010.ASU.D10.LOAD, and

- Then you installed version 11.2.00 of DASD MANAGER PLUS

Update the value of the keyword to a version 11.2.00 LINK library (BMC1120.ASU.D11.BMCLINK).

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 the following:

- “Using action POFs to reset JCL options” on page 125
- “Setting options and authorizations” on page 73
- In the Installation System Reference Manual, Configuring BMC products for DB2 => Use of the JCL Generation File Information dialog

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 124.
**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**

The following figure illustrates how DASD MANAGER PLUS uses the installation options module, product options (POF), user options, and override options.

**Figure 4: Functions of options**
Data set allocation

When you specify these data sets, use the guidelines in Table 14 on page 69.

Note
You must predefine partitioned data sets (PDSs). If a sequential file or member of a PDS does not exist, the product dynamically creates or allocates it.

Table 14: 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></td>
<td>PDS member</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></td>
<td>PDS member</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></td>
<td>PDS member</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 “Access to actions, services, and service syntax” on page 278.

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 311.
Execution and security setup

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 15 on page 70 lists the plans that the functions in DASD MANAGER PLUS use. In the plan names, vr represents the version and release.

### Table 15: DASD MANAGER PLUS plans

<table>
<thead>
<tr>
<th>Plan name</th>
<th>Function name</th>
<th>Plan description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASUvrDJ</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>ASUvrDR</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>ASUvrDS</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>ASUvrDZ</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>
## Execution component plans

You can restrict access to functions of the Execution component by using PLAN authorizations.

Table 16 on page 71 lists the plans that the Execution component uses.

### Table 16: Execution plans

<table>
<thead>
<tr>
<th>Plan name</th>
<th>Function name</th>
<th>Plan description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEXv7HA (ALTER and CHANGE MANAGER) AEXv7DA (DASD MANAGER PLUS)</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>
</tbody>
</table>
| AEXv7HM (ALTER and CHANGE MANAGER) AEXv7DM (DASD MANAGER PLUS) | Execution Monitor | Enables users 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:  
- AEXAUNLD unloads data from tables.  
- AEXSQLIO performs all worklist -SQL commands, including deletions before a -LOAD or -BMCL command that migrates data only.  
- 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.SYSSSTOGROUP
- 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 are set during the installation process. However, to manually set your options and authorizations, use the procedures described in the topics listed below:

- “Setting general user options” on page 74
- “Setting the JCL options for job cards” on page 75
- “Setting the JCL options for STEPLIBs” on page 78
- “Setting the JCL options for static data sets” on page 79
- “Setting the JCL options for tapes” on page 84
- “Setting the JCL options for temporary work data sets” on page 86
- “Setting the JCL options for permanent data sets” on page 88
- “Setting the JCL generation data group (GDG) options” on page 93
- “Setting the JCL debugging, display, and Execution options” on page 95
- “Setting the installation options module names for BMC utilities” on page 97
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.
Figure 5 on page 75 shows options that are specific to the DB2 subsystem and to generating actions.

Figure 5: General Options panel

```
----------------- DASD MANAGER PLUS 11.02.00 General Options -----------------
Command ===>

System Information
  DB2 SSID      : DEFF
  DB2 Catalog   : DEFFCAT
  Options Module: DSA2QDOP

Utility generation and processing options
  JCL DSN . . . . 'ASU.DEDK.V711S2.JCLLIB(&WKID)'
  Worklist DSN . . 'ASU.DEDK.V711S2.WORKLIST(&WKID)'
  Build Worklist. Y (Y/N - Build the Worklist)
  Edit Worklist . Y (Y/N - Edit the Worklist after build)
  Build JCL . . . Y (Y/N)
  Edit JCL . . . Y (Y/N - Edit the JCL file after build)
  Record Events . Y (Y/N)
  Submit Job . . Y (Y/N - Submit JCL for execution)
  Save Last Used. Y (Y/N - Applies to options above)

Panel Display Option
  Entry Field Delimiter.  1. Underscore  2. Reverse Video
                          3. Blink       4. None
  Truncation location.. M  (B/M/E - Beginning, middle, or end)
  Truncation characters. <>
  Preview Object Limit . 10

ISPF Color Attributes
  Type any color preference for the color attribute you wish to substitute.
  ISPF Color Attributes     Color Attribute You Wish to Substitute
  Blue . . . . . . . . . . .
  Green . . . . . . . . . . .
  Pink . . . . . . . . . . .
  Red . . . . . . . . . . .
  Turquoise . . . . . . . .
  White . . . . . . . . . .
  Yellow . . . . . . . . .
```

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

Use the Options panels of the JCL Generation component to define or modify the values in your ISPF profile and a user POF. Use the Jobcard Options panel to specify information about the job cards that the JCL uses.
To set the JCL options for job cards

1. Use the following menu selections to display the JCL Generation Jobcard Options Update panel (Figure 6 on page 76):

<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 Update - Main Menu</td>
<td>Jobcard Options</td>
</tr>
</tbody>
</table>

**Figure 6: JCL Generation Jobcard Options Update panel**

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

Is a TSO submit exit used to generate jobcards? . . N  (Y/N)
Enter Jobcards below:
//MVSCAL2I JOB ('&ZACCTNUM', '&PGMR',
//  CLASS=A, MSGLEVEL=(1,1), NOTIFY=&SYSUID
/*@ */
/*@ */
عائل
Jcllib . . . . . .
Sysexec. . . . . . ADM.INST1110.XXXREXX
Region size . . . OM  (See JCL Reference for valid options)
Memlimit. . . . . NOLIMIT (See JCL Reference for valid options)
Time parameter . . . . .  (See JCL Reference for valid options)
System MLIB. . . . SYS1.0000.ISPMENU
Runtime HLQ. . . . ADM.INSTXXXX
User HLQ . . . . . .
ULLQ . . . . . . . (Leave blank if using runtime enablement)
```

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

3. 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 partitioned data sets (PDSs):
   - A PDS that contains customized JCL to be included in the job
   - A PDS that specifies the cataloged procedures (PROCs) that are used for non-worklist JCL

5. In the Sysexec field, type the name of the PDS in which a REXX EXEC is a member.

**Tip**

To indicate the data set name for a different SSID, append the &SSID or &MSSID symbolic variable to the name.
In the **Region size** field, type the amount of memory to allocate for each step so that your job can run.

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

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

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

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

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

This value supports the following symbolic variables:

- &DB2V2 and &DB2V3, which resolve to the version of DB2
- &SSID, which resolves to the DB2 subsystem ID
  
  When you include the &SSID symbolic variable, the product can use a single POF with multiple subsystems.

In the **User HLQ** field, type the high-level qualifier used for the user-defined data sets for the installation environment.

**Note**

The **User HLQ** field is used only if the **LLQ** field is blank.

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.

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

**Note**

If the **User HLQ** field contains a value, the **ULLQ** value is ignored.
Press **END** to save your changes and return to the DASD MANAGER PLUS Main Menu.

### Setting the JCL options for STEPLIBs

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

#### To set the JCL options

1. Use the following menu selections to display the JCL Generation STEPLIB Options Update panel (Figure 7 on page 78):

<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 Update - Main Menu</td>
<td>Steplib Options</td>
</tr>
</tbody>
</table>

![Figure 7: JCL Generation STEPLIB Options Update panel](image)

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 In the field for the relevant product, type the data set names of the LINK libraries for the BMC products.

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

3 Press **END** to save your changes and return to the DASD MANAGER PLUS Main Menu.

### Setting the JCL options for static data sets

To define or modify the values in your ISPF profile and a user POF, 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:

<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>DASD MANAGER PLUS User Options</td>
<td>JCL generation options</td>
</tr>
</tbody>
</table>
2 In the Data set sizing option field, type N, B, C, or O to specify the sizing method, as shown in Table 17 on page 80.

**Note**

Whether or not data set sizing is performed, DB2 catalog access is required to resolve any symbolic variables.

### 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 4 on page 81.</td>
</tr>
</tbody>
</table>
| Sizing by using statistics from the DASD MANAGER PLUS tables | B    | - The DASD MANAGER PLUS product must be installed and interacting with ALTER or CHANGE MANAGER to use this option.  
- If a column is defined as a LONG VARCHAR, this option averages the row size.  
- If you select this option for Batch Execution JCL Generation, the DATASETSIZING B keyword is inserted into the AJXIN input stream.  
- Current BMCSTATS statistics should be available for objects in the worklist before you select this option. |
<table>
<thead>
<tr>
<th>For this method</th>
<th>Type</th>
<th>Considerations</th>
</tr>
</thead>
</table>
| Sizing by using statistics from the DB2 catalog | C    | ■ If a column is defined as a LONG VARCHAR, this option does not average the row size.  
■ If you select this option for Batch Execution JCL Generation, the DATASETSIZING C keyword is inserted into the AJXIN input stream.  
■ Current IBM RUNSTATS catalog statistics should be available for objects in the worklist before you select this option. |
| Estimating sizes based on physical, random sampling of VSAM data sets | O    | ■ If a column is defined as a LONG VARCHAR, this option averages the row size.  
■ If you select this option for Batch Execution JCL Generation, the DATASETSIZING O keyword is inserted into the AJXIN input stream.  
■ You can use this option if the statistical information in the DB2 catalog or in the DASD MANAGER PLUS tables is not current. However, JCL generation might take additional time to complete. |

You can specify whether to include comments in the Execution JCL that show statistics for determining data set sizes. For more information, see “Setting the JCL debugging, display, and Execution options” on page 95.

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

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

5 Specify the values that should be used for the following options when the maximum number of cylinders are exceeded:
   a In the **Max primary quantity** field, type the number of cylinders for the maximum primary quantity.
   b In the **Max secondary quantity** field, type the number of cylinders for the maximum secondary quantity.
   c In the **Max unit count** field, type the maximum number of volumes.

If you want to use multiple data sets on DASD, specify a value greater than 1 for **Max unit count**. On the JCL Generation Individual Data Set Options Update panels (see “Setting the JCL options for permanent data sets” on page 88), specify the name of a DASD unit.
In the Include data set cleanup step field, type Y or N to specify whether to generate a step in the JCL to delete the permanent work data sets. 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 (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 on page 82 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)</td>
<td>SYSDnnnn</td>
<td>LOADPLUS, IBM LOAD</td>
</tr>
<tr>
<td>Error</td>
<td>SYSERnnn</td>
<td>CHECK PLUS, LOADPLUS, IBM CHECK DATA, IBM LOAD</td>
</tr>
<tr>
<td>Map</td>
<td>SYSMAP</td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Punch</td>
<td>SYSPUNCH</td>
<td>REORG PLUS, IBM REORG</td>
</tr>
<tr>
<td>Unload (SYSREC)</td>
<td>SYSRnnnn or Rnnnnyyyy</td>
<td>REORG PLUS, 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 on page 83.</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 LOADPLUS and UNLOAD PLUS utilities and IBM LOAD utility are specified in the cleanup job step but are commented out. You must edit the Execution JCL and remove the comment delimiters to delete the other unload data sets automatically.
Table 19 on page 83 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>CHECK PLUS</td>
</tr>
<tr>
<td>SORTPnnn</td>
<td>LOADPLUS</td>
</tr>
<tr>
<td>SORTOnnn</td>
<td>REORG PLUS</td>
</tr>
<tr>
<td></td>
<td>IBM CHECK DATA</td>
</tr>
<tr>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td></td>
<td>IBM REORG</td>
</tr>
<tr>
<td>SYSUTnnn</td>
<td>CHECK PLUS</td>
</tr>
<tr>
<td>SUTnnn</td>
<td>LOADPLUS</td>
</tr>
<tr>
<td>WRKnnn</td>
<td>REORG PLUS</td>
</tr>
<tr>
<td></td>
<td>RECOVER PLUS</td>
</tr>
<tr>
<td></td>
<td>IBM CHECK DATA</td>
</tr>
<tr>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td></td>
<td>IBM REORG</td>
</tr>
<tr>
<td></td>
<td>IBM RECOVER INDEX</td>
</tr>
<tr>
<td></td>
<td>IBM REBUILD INDEX</td>
</tr>
</tbody>
</table>

7. In the **Return code for cleanup step** field, specify the value that should be compared against the condition code that is returned from any previous job step. You can specify any two-digit value for the return code; however, BMC recommends that you specify 4.

8. In the **Temporary unit** field, type the name of the unit that is used to allocate temporary files when JCL is generated. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

9. In the **Include SYSPRIN2 DD** field, type Y or N to specify whether to generate the //SYSPRIN2 DD SYSOUT=* DD in the JCL. The SYSPRIN2 output data set contains SYSPRINT output messages for versions 10.2 and later of the following BMC utilities:

   - CHECK PLUS
   - LOADPLUS
   - REORG PLUS
UNLOAD PLUS

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

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 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, 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 9 on page 85):

<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>
</tbody>
</table>
Figure 9: JCL Generation Tape Options Update panel

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

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

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

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

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

6 In the Tape TRTCH field, type the parity, data conversion, translation, and compression value for 7-track tape drives as shown in Table 20 on page 85.

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>(Leave the field blank)</td>
</tr>
<tr>
<td>Odd parity, conversion, and no translation</td>
<td>C</td>
</tr>
<tr>
<td>Even parity, no conversion, and no translation</td>
<td>E</td>
</tr>
<tr>
<td>Odd parity, no conversion, and translation</td>
<td>T</td>
</tr>
<tr>
<td>Even parity, no conversion, and translation</td>
<td>ET</td>
</tr>
<tr>
<td>Data compression</td>
<td>COMP</td>
</tr>
<tr>
<td>No data compression</td>
<td>NOCOMP</td>
</tr>
</tbody>
</table>

7 For each type of copy or product data set, type Y or N to specify whether the data sets should be stacked on a tape with data sets of the same type.
Setting the JCL options for temporary work data sets

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

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

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

### To set the JCL options for temporary work data sets

1. Use the following menu selections to display the JCL Generation Options For Sort Files Update panel (Figure 10 on page 86):

<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 UPDATE - 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>

**Figure 10: JCL Generation Options For Sort Files Update panel**

```
---------------------- JCL GENERATION OPTIONS FOR SORT FILES UPDATE ----------------------
COMMAND ===> 

Type data and press Enter.

Number of SORTWORK Data Sets... 5 (1 - 32)
SORTWORK unit name... SYSDA (SYSDA, 3380, etc)
Number of DATAWORK Data Sets... 5 (1 - 32)
DATAWORK unit name... SYSDA (SYSDA, 3380, etc)
Number of LOGSORT Data Sets... 1 (1 - 32)
LOGSORT unit name... SYSDA (SYSDA, 3380, etc)
Default Primary Quantity... 10 (Cylinders)
Default Secondary Quantity... 2 (Cylinders)
SMS Data Class... (Blank or Data Class Name)
SMS Storage Class... (Blank or Storage Class Name)
SMS Management Class... (Blank or Management Class Name)
$ORTPARM data set name (below)
```

2. Specify the options for SORTWORK data sets:

a. In the **Number of SORTWORK Data Sets** field, type the number of SORTWORK data sets.
b In the **SORTWORK unit name** field, type the name of the unit for SORTWORK data sets.

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

3 Specify the options for DATAWORK data sets:

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

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

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

4 Specify the options for LOGSORT data sets:

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

b In the **LOGSORT unit name** field, type the name of the unit for LOGSORT data sets.

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

5 If you typed **N** in the **Data set sizing option** field in **Step 2 on page 80**, 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.
In the $SORTPARM data set name field, type the name of the data set that provides parameters for SyncSort.

Press END to save your changes and 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, use the Options panels of the JCL Generation Update - Main Menu component. Use the individual data set options panels to specify information about the following permanent work data sets and image copy data sets:
- Sortout (WORKDDN or LOADDN)
- Sysut (WORKDDN)
- Copy (COPYDDN, RECOVERYDDN, RECOVERDDN, ICDDN, RECOVERYICDDN, OUTCOPYDDN, FCOPYDD, or EXPORTDDN)
- Sysrec (UNLDDN, INDDN, or UNLOADDN)
- Archive (ARCHDDN)
- Cntl file (CNTLDDN)
- *(CHANGE MANAGER only)* Baseline recovery
- Discard (DISCARDNN)
- Error (ERRDDN)
- Map (MAPDDN)
- Report
- Punch (PUNCHDDN)
- Filter (FILTERDDN)

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

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

The LOB SYSREC data sets is used only by the UNLOAD PLUS utility to unload and load data contained in LOB columns.

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

1. Use the following menu selections to display the panels for permanent work data sets:
Not all of the options are available on all of the data set options panels.
If you choose to dynamically allocate your copy or unload data sets, any changes that you make to the copy (SYSCOPY), unload (SYSREC), and baseline recovery (BLRP) (for CHANGE MANAGER only) data set options in the JCL Generation override panels do not take effect.

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 on page 89).

### 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 only) flashcopy</td>
<td>FCPY_SUPPRESS_SUFF</td>
</tr>
<tr>
<td>Local primary copy</td>
<td>PCPY1_SUPPRESS_SUFF</td>
</tr>
</tbody>
</table>
To specify a GDG for the local and recovery image copy data sets, add the `&GDG` symbolic variable to the end of the data set prefix (Figure 12 on page 90).

**Figure 12: Using the `&GDG` symbolic variable**

```
_TYPE DATA SET OPTIONS FOR LOCAL PRIMARY COPY UPDATE ===>
COMMAND ===> Type data and press Enter. Press PF3 or END to return to the main panel.

Enter Data Set Prefix below:
. . &PREFIX..&OBNOD(&GDG)

Unit Name . . . . . . . . . . . SYSDA    (SYSDA, TAPE, etc)
Primary Space . . . . . . . . . 10       (Cylinders)
Secondary Space . . . . . . . . . 2        (Cylinders)
Tape EXPDT. . . . . . . . . . .          (Blank or YYDD or YYYY/DDD)
Tape RETPD. . . . . . . . . . .          (Blank or 1 - 9999 days)
SMS Data Class. . . . . . . . .          (Blank or Data Class)
SMS Storage Class . . . . . . . .        (Blank or Storage Class)
SMS Management Class  . . . . . .       (Blank or Management Class)
Threshold Value . . . . . . . . 0        (Cylinders, 0 means no Threshold)
Alternate Unit Name . . . . . .         (SYSDA, TAPE, etc)
Alternate SMS Data Class . . .          (Blank or Data Class Name)
Alternate SMS Storage Class . .         (Blank or Storage Class Name)
Alternate SMS Management Class           (Blank or Management Class Name)
```

When you use the `&GDG` variable, JCL Generation resolves the data set name using the symbolic variable, and the name includes the GDG number (Figure 13 on page 90).

**Figure 13: Data set names resolved with the `&GDG` symbolic variable**

```c++
//SYC10001 DD DSN=RDACRJ.DEMOCJ.S9(+1),
//       DCB=(SYS1.MODEL),
//       DISP=(NEW,CATLG,CATLG),
//       SPACE=(CYL,(10,2),RLSE),
//       UNIT=SYSDA
//SYC10002 DD DSN=RDACRJ.DEMOCJ.S3(+1),
//       DCB=(SYS1.MODEL),
//       DISP=(NEW,CATLG,CATLG),
//       SPACE=(CYL,(10,2),RLSE),
//       UNIT=SYSDA
//SYC10003 DD DSN=RDACRJ.DEMOCJ.S2(+1),
//       DCB=(SYS1.MODEL),
//       DISP=(NEW,CATLG,CATLG),
//       SPACE=(CYL,(10,2),RLSE),
//       UNIT=SYSDA
```

<table>
<thead>
<tr>
<th>Data set</th>
<th>POF keyword</th>
</tr>
</thead>
<tbody>
<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>
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 Update - Main Menu allocates the data sets before it opens them.

- If you have very large data sets and want to avoid extents or multiple data sets on DASD, specify one of the following:
  - Tape
  - Virtual tape
  - Cartridge unit

- For the Error data set (SYSERnnn) and Map data set (SYSMAP), specify **SYSDA**.

  If you specify **TAPE** and need to restart the IBM LOAD utility, you must do the following:

  1. Uncatalog the existing data set

  2. Change the DD statements to a disposition of (NEW,CATLG,CATLG)
Tip
As an alternative to specifying tape for the unit, you can specify a DASD unit that spans multiple data sets, by doing the following:

1. Return to the JCL Generation Static Data Set Options panel (see the figure in “Setting the JCL options for static data sets” on page 79)

2. Specify a value for Max cylinders and a value greater than 1 for Max unit count

Best practice
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 in the Database Administration and BMC Object Administration for DB2 solutions.) If the ROWID SYSREC is set to SYSDA, you can improve the performance of the worklist commands for unloading the ROWID data set when the worklist is run in parallel.

4. If you typed N in the Data set sizing option field in Step 2 on page 80, specify the default primary and secondary quantities:
   a. In the Primary Space field, type the value for the primary quantity in cylinders.
   b. In the Secondary Space field, type the value for the secondary quantity in cylinders.

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

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

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

7. Specify the SMS definitions for the data set classes:
   a. In the SMS Data Class field, type the name of the data class.
   b. In the SMS Storage Class field, type the name of the storage class.
   c. In the SMS Management Class field, type the name of the management class.

8. In the Threshold Value field, type, in cylinders, the primary quantity for the data set.
If this value is exceeded, JCL Generation Update - Main Menu uses the alternate unit and the alternate SMS parameters. Zero indicates that a threshold is not specified for the unit. If you specify zero, JCL Generation Update - Main Menu does not use an alternate unit and the alternate SMS parameters.

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

9 In the **Alternate Unit Name** field, type the alternate name of the unit to be used if the threshold value specified in **Step 8 on page 92** 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 84.

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

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

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

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

11 Press **END** to save your changes and return to the DASD MANAGER PLUS Main Menu.

### Setting the JCL generation data group (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 options for generation data groups**

1 Use the following menu selections to display the JCL Generation Generation Data Group Options Update panel (**Figure 14 on page 94**):

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

### Figure 14: JCL Generation Generation Data Group Options Update panel

```
COMMAND ===> JCL GENERATION GENERATION DATA GROUP OPTIONS UPDATE

Type data and press Enter.

Define GDG base at JCL generation?.. N (Y/N)
Specify NSCR on GDG definition?.. N (Y/N)
Number of primary copy GDG entries.. 10 (1-255)
Number of recovery copy GDG entries .10 (1-255)
Type GDG Model data set below:
       .. SYS1.MODEL

NOTE: GDGs only apply to copy data sets.
```

2. In the **Define GDG base at JCL generation?** field, type **Y** or **N** to specify whether JCL Generation creates the base of the GDG.

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

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

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

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

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

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

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

7. Press **END** to save your changes and 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, use the Options panels of the JCL Generation component.

Use the Debugging, Display and Execution Options panel to specify information about how comments are handled in the JCL.

To set the JCL options for debugging, display, and Execution

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

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 Update - Main Menu</td>
<td>Debugging, Display and Execution Options</td>
</tr>
</tbody>
</table>

Figure 15: 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:
SYNDELETE . . . 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. \textbf{Table 22 on page 96} lists the comment codes that \textit{Execution} generates in the JCL.

\textbf{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: ( (\text{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>

\textbf{Figure 16 on page 96} shows example comments in the \textit{Execution} JCL.

\textbf{Figure 16: Execution JCL with comments}

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

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

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

\textit{JCL Generation} uses these variables to resolve the names of the data sets in the generated JCL.

In the \textbf{Suppress comments in JCL} field, type \textbf{Y} or \textbf{N} to specify whether to suppress all comments in the generated JCL.
Note
If you chose to include either the statistics comments or the variable comments in Step 2 on page 95, 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.

10 Press END to save your changes and return to the DASD MANAGER PLUS Main Menu.

Setting the installation options module names for BMC utilities

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

Use the BMC Utility Option Module Names panel to specify the name of the installation options module for the BMC utilities.

To set the JCL options for utility installation options module names

1 Use the following menu selections to display the JCL Generation BMC Utility Option Module Names Update panel (Figure 17 on page 98):
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 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 on page 99):

<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 Update - Main Menu</td>
<td>Utility Options</td>
</tr>
<tr>
<td>Utility Options</td>
<td>BMC Utility Option Module Names</td>
</tr>
</tbody>
</table>
From this menu | Select this item and press Enter
---|---
User Options | JCL generation options
JCL GENERATION UPDATE - MAIN MENU | Utility Options
UTILITY OPTIONS UPDATE | Non worklist JCL PROC Options

**Figure 18: PROC and STEP Names panel**

```plaintext
--- PROC AND STEP NAMES ---

COMMAND ===> 

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

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

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

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

*Tip*  You can specify the PDS that contains the PROCs in the **Jcllib** field on the JCL Generation Jobcard Options panel (see “Setting the JCL options for job cards” on page 75). To add JCLLIBs to the concatenation, modify the AJXJCLU SLIB. Alternatively, you can include the JCLLIB or PROCLIB in your jobcard JCL.

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

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

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

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

4 In the PROCNAME field, specify the name of a PROC for the corresponding utility.
Alternatively, you can define the name of a PROC in the following POF keywords in your POF:

- PROC_BMCCHECK_NAME =
- PROC_BMCCOPY_NAME =
- PROC_BMCCPRS_NAME =
- PROC_BMCLOAD_NAME =
- PROC_BMCRECOVER_NAME =
- PROC_BMCREORG_NAME =
- PROC_BMCSTATS_NAME =
- PROC_BMCSTOP_NAME =
- PROC_BMCTRIG_NAME =
- PROC_BMCUNLOAD_NAME =
- PROC_BMCUPRS_NAME =
- PROC_DSNUTILIB_NAME =
- PROC_DSN1COPY_NAME =
- PROC_IDCAMS_NAME =
- PROC_IEFBR14_NAME =
- PROC_TSO_NAME =
- PROC_USER_DEFINED =

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_BMCHECK_STEP =
PROC_BMCCOPY_STEP =
PROC_BMCCPRS_STEP =
PROC_BMCLOAD_STEP =
PROC_BMCRECOVER_STEP =
PROC_BMCREORG_STEP =
PROC_BMCSTATS_STEP =
PROC_BMCSTOP_STEP =
PROC_BMCTRIG_STEP =
PROC_BMCUNLOAD_STEP =
PROC_BMCUPRS_STEP =
PROC_DSNUTILB_STEP =
PROC_DSN1COPY_STEP =
PROC_IDCAMSS_STEP =
PROC_IEFBR14_STEP =
PROC_TSO_STEP =
PROC_USER_DEF_STEP =

6 Press END to save your changes and return to the DASD MANAGER PLUS Main Menu.

Setting the BMCTRIG Control-M options

To define or modify the values in your ISPF profile and a user POF, 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:

<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 Update - Main Menu</td>
<td>Utility Options</td>
</tr>
<tr>
<td>From this menu</td>
<td>Select this item and press Enter</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Utility Options</td>
<td>BMCTRIG Control-M Options</td>
</tr>
</tbody>
</table>

**Figure 19: BMCTRIG Control-M Options panel**

<table>
<thead>
<tr>
<th>COMMAND ====&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJXOCTLM ---------------------</td>
</tr>
<tr>
<td>BMCTRIG Control-M OPTIONS</td>
</tr>
</tbody>
</table>

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
  - 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 on page 103):

<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 Update - Main Menu</td>
<td>User Defined Variable Values</td>
</tr>
</tbody>
</table>

Figure 20: User Defined Variables Update panel

```
---------------------- USER DEFINED VARIABLES UPDATE ----------------------
COMMAND ===>
Type data and press Enter.
Character Variables:
User variable 1 . . . . . . (Symbolic &UVR1)
User variable 2 . . . . . . (Symbolic &UVR2)
User variable 3 . . . . . . (Symbolic &UVR3)
User variable 4 . . . . . . (Symbolic &UVR4)
User variable 5 . . . . . . (Symbolic &UVR5)
```

NOTE: Symbolic variables cannot be input to these values

2. Specify the values for the variables.

The maximum length of a variable name is eight characters.

3. Press END to save your changes, and to return to the DASD MANAGER PLUS Main Menu.

Creating a user POF

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

Use the Product Options File (POF) Functions panel to create a user POF or update the values in your ISPF profile. The panel displays the data set name of the initial POF. The panel also displays the value of the POFDATE parameter in the initial POF that was last used to update the ISPF profile.

To create a user POF

1. Use the following menu selections to display the JCL Generation Product Options File (POF) Functions panel (Figure 21 on page 104):
From this menu | Select this item and press Enter
--- | ---
DASD MANAGER PLUS Main Menu | User Options
User Options | JCL generation options
JCL Generation Update - Main Menu | Product Option File (POF) Functions

**Figure 21: JCL Generation Product Options File (POF) Functions panel**

---

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

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

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

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

--- Edit/Validate will update the User POFDATE --

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

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

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

4 Press **Enter** to create the user POF.

**Updating a user POF**

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

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

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

2 Use the following menu selections to display the JCL Generation Product Options File (POF) Functions panel:
<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 Update - 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.

5 Press **Enter**.

**To update the user POF directly**

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

2 In the **BROWSE, EDIT, VALIDATE User POF** field, type E.

---

**Note**

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

3 Edit and save the file.

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

---

**Note**

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

- To use the variables in the initial POF, enter `TSO POFRESET`.

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

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

5 Press **Enter**.
Using multiple POFs

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

Some sample scenarios follow:

Scenario 1

The Payroll department needs backup copies of their data sets on tape, but other departments do not need backup copies.

You can create one user POF for the Payroll department and one for the other departments. Whenever you work with payroll objects, you can specify your payroll POF name to reset the profile variables from that POF.

Scenario 2

Your company is a service provider for several customers.

By using a separate POF for each customer, you can accommodate each customer’s naming standards.

Scenario 3

You have different requirements for test and production data.

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

1 Follow the steps in “Creating a user POF” on page 103 to create an additional POF.

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

Note

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

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

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

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

**To refresh the initial POF**

1. Edit the initial POF outside of the product.
2. Change the value of the POFDATE keyword to the current date.
3. Append the refresh attribute (R) to the values that you want to update.
4. Save the POF.

After you save the changes, users receive the updated ISPF variables the next time they invoke one of the products. However, 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, users must 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 be sure to include the updated values.

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

Generating POF reports

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

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

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

**To generate the reports**

1. From the JCL Generation Product Options File (POF) Functions panel, in the **MSGCLASS for POF Diagnostic Messages** field, type the MSGCLASS for the SYSOUT field that is used to display messages.

Consider using a SYSOUT class that is designated to go to the held queue so that you can view the output. Two SYSOUT files are allocated: AJXPOFER and AJXPOFVL.

---

**Note**

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

---

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

**Reusing a POF in a subsequent installation**

Assume that you customized the values in your POF, and now you are installing a new release of a product. To avoid having to customize the values again, you can specify that the Installation System use your existing POF to populate the values in the new initial POF.

The new POF contains your current values plus any new keywords (and their values) for the new release.

**To reuse a POF**

1. Run the Installation System.
2 From the Install System JCL Generation File Information panel, in the **Use Existing POF to Populate the New Product Options File** field, type **Y**.

**Figure 22: Reusing an existing POF**

<table>
<thead>
<tr>
<th>AJXP041</th>
<th>Install System DB2 Administrative Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>====&gt;</td>
</tr>
<tr>
<td>POF</td>
<td>ZAJXBPOF</td>
</tr>
<tr>
<td>Use POF</td>
<td>Y (Y/N)</td>
</tr>
<tr>
<td>Press PF1 HELP for more information regarding the POF, its use, and how to avoid having to reenter the information for SSID installs.</td>
<td></td>
</tr>
<tr>
<td>Verify the ACS Common SQL API Alias Qualifier and the Collection ID. BMC Software Inc. recommends accepting the defaults. To make changes, type over the values shown.</td>
<td></td>
</tr>
<tr>
<td>Alias Creator</td>
<td>7CS111S1</td>
</tr>
<tr>
<td>ACS</td>
<td>ZCS111_D_MAIN</td>
</tr>
<tr>
<td>Execution for DB2</td>
<td>ZES111_D_MAIN</td>
</tr>
<tr>
<td>Collection ID</td>
<td>ZES111_D_MAIN</td>
</tr>
<tr>
<td>Press Enter to continue or F12 to go back.</td>
<td></td>
</tr>
</tbody>
</table>

3 Enter the names of the data set and member for the existing POF (**Figure 22 on page 109**).

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

**Figure 23: Specifying the name of the existing POF**

<table>
<thead>
<tr>
<th>AJXP044</th>
<th>Install System JCL Generation File Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>====&gt;</td>
</tr>
<tr>
<td>Enter the data set and member name of an existing Product Options File (POF) that is to be used to seed the new POF. An existence check will be performed, including verifying the specified member has a format consistent with the POF format.</td>
<td></td>
</tr>
</tbody>
</table>

**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 on page 110), the name of the primary copy data set resolves to the following name without modifications to the SLIB:

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

For data sets that are not dynamically allocated, JCL Generation appends the ddname to the prefix to create the name of the data set.

Figure 24: Specifying the prefix for a copy data set

For data sets that are not dynamically allocated, JCL Generation appends the ddname to the prefix to create the name of the data set.

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

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

Enter Data Set Prefix below:

.. ASSID..IC.T.ICPY.&DB..&TS

Unit Name . . . . . . . . . . . SYSDA (SYSDA, TAPE, etc)
Primary Space . . . . . . . . . 10 (Cylinders)
Secondary Space . . . . . . . . . 2 (Cylinders)
Tape EXPDT. . . . . . . . . . . . (Blank or YYDDDD or YYYY/ddd)
Tape RETPD. . . . . . . . . . . . (Blank or 0 - 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)

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 on page 110), the DBA can uphold your site’s naming standards.

Figure 25: Changing the SLIB variable for the copy data set in AJX#DSNS

```cm
)CM:-------------------------------------------------------------
)CM- &AJXC1PRF $= &Z
)SEL &AJXC1PRF ^= &Z
)SET SYSC1PR = &AJXC1PRF <- Resolved values from ISPF profile
)ENDSEL
)SEL &AJXC1PRF = &Z
)SEL &AJXSYCOP ^= &Z
)SET SYSC1PR = &AJXSYCOP
)ENDSEL
)SEL &AJXSYCOP = &Z
)SET SYSC1PR = &AJXHLQ..&AJXDB..&AJXTS
)ENDSEL
)ENDSEL
)SEL &AJXC2PRF ^= &Z
)SET SYSC2PR = &AJXC2PRF
)ENDSEL
)SEL &AJXC2PRF = &Z
)SEL &AJXSYCOP ^= &Z
)SET SYSC2PR = &AJXSYCOP
)ENDSEL
)SEL &AJXSYCOP = &Z
)SET SYSC2PR = &AJXHLQ..&AJXDB..&AJXTS
)ENDSEL
```

110  DASD MANAGER PLUS for DB2 User Guide
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 “Using the Skeleton Library compiler” on page 663.

Adding steps to the JCL

In CATALOG MANAGER or DASD MANAGER PLUS, you can include customized steps in the generated JCL (between product-generated steps or at the end of a job).

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

To add steps to the JCL

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

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 Update - Main Menu</td>
<td>Jobcard Options</td>
</tr>
</tbody>
</table>
2 From the JCL Generation Jobcard Options Update panel, in the Jcllib field, type the name of the partitioned data set and press END.

**Figure 26: JCL Generation Jobcard Options Update panel**

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

Type data and press Enter.

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

Enter Jobcards below:

```
//JOBCDBDC JOB ('&ZACCTNUM'), '&PGMR',
//  CLASS=A, MSGCLASS=X, MSGLEVEL=(1,1),
//  NOTIFY=&USERID
```
```
/*
*/
```

Jcllib . . . . . . SKH.INCLUDE.JCLLIB

Sysexec . . . . . .

Region size . . 0M (See JCL Reference for valid options)

Time parameter . . (See JCL Reference for valid options)

System MLIB. . . . ISP.SISPMENU

Runtime HLQ. . . . AEX.QA0101

LLQ . . . . . . . . BMC

ULLQ . . . . . . . (Leave blank if using runtime enablement)
```

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

**Figure 27: Including the JCL library**

```
/*-------------------------------------------------------
/* JCLLIB SPECIFIED
/*-------------------------------------------------------

JCLLIB ORDER=SKH.INCLUDE.JCLLIB
```

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

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: JCL Generation Debugging, Display and Execution Options Update panel**

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

Type data and press Enter.

Include data set sizing comments in JCL . . . . . . N (Y/N)

Include variable substitution comments in JCL . . . . . . N (Y/N)

Suppress comments in JCL . . . . . . . N (Y/N)

NOTE: Do not set suppress comments to Y if you have specified either sizing or variable substitutions.

Specify an Alternate Program for IKJEFT01 . . . . .

Post Step JCL INCLUDE member name . . . . . . STEPEND

Post Job JCL INCLUDE member name . . . . . . JOBEND

Include in AEXIN parameters:

SYNCDLFKEY . . . N (Y/N)  BINDFAIL . . . N (Y/N)

HASHFAIL . . . N (Y/N)  HASHWARNRC . . . (NUMERIC)

REBINDFAIL . . . N (Y/N)  REBINDRC . . . (NUMERIC)

2MEGSOL . . . N (Y/N)  NOFAILNOIMAGECOPY N (Y/N)

STOPWAIT . . . 3 (NUMERIC)

STOPWAIT SECS . . 10 (NUMERIC)
```
If you want to run the JCL between steps, type the member name in the **Post Step JCL INCLUDE member name** field.

The JCL Generation component generates the following statement in the JCL after each step:

**Figure 29: Including JCL between steps**

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

//  INCLUDE MEMBER=STEPEND
```

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

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

**Figure 30: Including JCL at the end of the jobs**

```plaintext
/*------------------------
* END OF JOB INCLUDE MEMBER
*------------------------

//  INCLUDE MEMBER=JOBEND
```

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

**Figure 31: Expanded INCLUDE members**

```plaintext
3 //  INCLUDE MEMBER=STEPEND
   XX** CUSTOMIZED PROGRAM TO DO SOMETHING BETWEEN STEPS
4 XXSTEPEND EXEC PGM=STPCOND,PARM=&SYSUID,COND=EVEN
   /*------------------------
5 //  INCLUDE MEMBER=JOBEND
   XX** STEP AT END OF THE JOB TO DO SOMETHING WHEN JOB COMPLETES
6 XXJOBEND EXEC PGM=STPCOND,PARM=&SYSUID,COND=EVEN
   /*------------------------
```

**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 more information on symbolic variables, see the supported symbolic variables table in “User-defined programs or services” on page 228.

DASD MANAGER PLUS passes pParameters (from the product options file (POF) and Execution panels) 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 (such as SORTWORK, and the permanent data sets that Execution uses) that DASD MANAGER PLUS passes from the JCL Generation option panels,

- 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 74). 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.

### Modification of 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.BMCSLIB` data set lists and briefly describes each non-DD and DD statement skeletons that JCL Generation uses. The member also lists the variables that JCL Generation uses to construct the default names for permanent data sets.

Each skeleton name in the `$AJXDOC` member has a corresponding member in the `HLQ.BMCSLIB` 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 on page 116. 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 102.
Table 23: User-defined variables

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

The *DASD MANAGER PLUS for DB2 Reference Manual* 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*. .

---

**Related Information**

- “Using the Skeleton Library compiler” on page 663

---

**JCL Generation data sets sizing function**

You can use the JCL Generation data set sizing function to tailor the data set sizes when the JCL is built.

The function gathers information from one of the following sources (shown in general order of accuracy, from most accurate to least accurate):

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

Whether it uses the BMCSTATS historical database or object sampling, or even if it does not use sizing, JCL Generation obtains some information from the DB2 system catalog.
The formulas for estimating data set size are taken from the documentation for the IBM DB2 utilities and from the documentation for the BMC utilities. Table 24 on page 117 shows the statistics that JCL Generation uses for space estimation and the source of the statistics.

Table 24: Data set sizing values and sources

<table>
<thead>
<tr>
<th>Value</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BMCSTATS</td>
</tr>
<tr>
<td>Number of active pages</td>
<td>X</td>
</tr>
<tr>
<td>Number of modified pages</td>
<td>X</td>
</tr>
<tr>
<td>Page size</td>
<td>NA</td>
</tr>
<tr>
<td>Maximum row length</td>
<td>NA</td>
</tr>
<tr>
<td>Average row length</td>
<td>X</td>
</tr>
<tr>
<td>Number of rows</td>
<td>X</td>
</tr>
<tr>
<td>Number of non-clustering indexes</td>
<td>NA</td>
</tr>
<tr>
<td>Longest key</td>
<td>X</td>
</tr>
<tr>
<td>Number of foreign keys</td>
<td>NA</td>
</tr>
<tr>
<td>Number of indexes</td>
<td>X</td>
</tr>
<tr>
<td>Longest foreign key</td>
<td>NA</td>
</tr>
</tbody>
</table>

Tip
To specify the data sizing method, see “Setting the JCL options for static data sets” on page 79.

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 (SYSC nnnn) and recovery (RECV nnnn) image-copy data set prefixes as needed.

When you use an &GDG variable (see Figure 32 on page 118), 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.

Figure 32: Use of the &GDG symbolic variable

| AJXODSPL --- JCL GENERATION DATA SET OPTIONS FOR LOCAL PRIMARY COPY UPDATE ---- |
| COMMAND ===>

Type data and press Enter. Press PF3 or END to return to the main panel.
Enter Data Set Prefix below:

| . . &PREFIX..&OBNOD(&GDG) |
| Unit Name ............... SYSDA (SYSDA, TAPE, etc) |
| Primary Space ........... 10 (Cylinders) |
| Secondary Space ......... 2 (Cylinders) |
| Tape EXPDT .............. (Blank or YYDDD or YYYY/DDD) |
| Tape RETPD .............. (Blank or 1 - 9999 days) |
| SMS Data Class .......... (Blank or Data Class) |
| SMS Storage Class ..... (Blank or Storage Class) |
| SMS Management Class (Blank or Management Class) |
| Threshold Value ........ 0 (Cylinders, 0 means no Threshold) |
| Alternate Unit Name .... (SYSDA, TAPE, etc) |
| Alternate SMS Data Class (Blank or Data Class Name) |
| Alternate SMS Storage Class (Blank or Storage Class Name) |
| Alternate SMS Management Class (Blank or Management Class Name) |

When you add the &GDG variable to the end of the field, the product resolves the local image-copy data sets as follows:

```csh
""" 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,(1,1),RLSE), ESTIMATE-C/H |
// UNIT=SYSDA |
""" DATA SET SIZE REQUIRED FOR DD C0002 IS AN ESTIMATE """
/C0002 DD DSN=RDAJXN2.DEAE.JEND30.JENS0130.C0002, |
// 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=RDAJXN2.DEAE.JEND30.C791278.D0001, |
// DISP=(NEW,CATLG,CATLG), |
// SPACE=(CYL,(1,1),RLSE), ESTIMATE-C/H |
// UNIT=SYSDA |
""" DATA SET SIZE REQUIRED FOR DD D0002 IS AN ESTIMATE """
/D0002 DD DSN=RDAJXN2.DEAE.JEND30.JENS0130.D0002, |
// DISP=(NEW,CATLG,CATLG), |
// SPACE=(CYL,(1,1),RLSE), ESTIMATE-C/H |
// UNIT=SYSDA |
Shown here is 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:

*************** Top of Data **********************
- TIME 000001 'yyyy-02-28-14.07.41.976612'                               91895345
- SSID 000001 DEAE                                                       29519769
- WKID 000002 JENTEST.UTILITY                                            40692400
- SYNC 000003                                                            08617204
  BMCC 000004
  COPY
  TABLESPACE JEND30.C791278
  SHRLEVEL REFERENCE
  FULL YES
  COPYDDN (C0001,D0001)
  RECOVERYDDN (RO001,V0001)
  42978305
  BMCC 000005
  COPY
  TABLESPACE JEND30.JENS0130
  SHRLEVEL REFERENCE
  FULL YES
  COPYDDN (C0002,D0002)
  RECOVERYDDN (RO002,V0002)

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

Shown here is 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:

/* *--------------------------------------------------------------------
/* 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
  UNIT=SYSDA
/* DATA SET SIZE REQUIRED FOR DD D0002 IS AN ESTIMATE
/*D0002 DD DSN=RDAJXN.CPY.DEAE.JEND30.JENS0130(+2),
  DCB=(SYS1.MODEL),
  DISP=(NEW,CATLG,CATLG),
  SPACE=(CYL,(1,1),RLSE), ESTIMATE-C/H
  UNIT=SYSDA
/* *--------------------------------------------------------------------
/* Utility copy DD statements
/* *--------------------------------------------------------------------
Using Copy DD options to specify GDGs

For finer control than the User Options panels (which 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 ("Control of your environment by using options" on page 63).

You can specify GDGs on the COPYDDN options panels of the following utilities:

- BMCCOPY
- FULLCOPY
- INCRCOPY
- MERGCOPY
- BMCREORG
Figure 33 on page 121 displays the BMCCOPY Copy Options panel.

**Figure 33: BMCCOPY Copy Options panel**

<table>
<thead>
<tr>
<th>ASUDUM</th>
<th>BMCCOPY TABLESPACE QZUD41.%</th>
<th>Row 49 to 85 of 106</th>
</tr>
</thead>
</table>

Service Syntax: BMCCOPY.DEMO_BMCCOPY

Enter data, then press end. More: + -

-------------------------------- Copy DD Options --------------------------------

| COPYDDN dd1 ... J | (N/J/D N-None, J-JCL, D-Dynamic allocation) |
| COPYDDN dd2 ... J | (N/J/D N-None, J-JCL, D-Dynamic allocation) |
| RECOVERYDDN dd1 ... J | (N/J/D N-None, J-JCL, D-Dynamic allocation) |
| RECOVERYDDN dd2 ... J | (N/J/D N-None, J-JCL, D-Dynamic allocation) |

****************************************************************************

| FULLDDN dd1 ... N | (N/D N-None, D-Dynamic allocation) |
| FULLDDN dd2 ... N | (N/D N-None, D-Dynamic allocation) |
| FULLRECCDDN dd1 ... N | (N/D N-None, D-Dynamic allocation) |
| FULLRECCDDN dd2 ... N | (N/D N-None, D-Dynamic allocation) |

****************************************************************************

| BIGDDN dd1 ... N | (N/D N-None, D-Dynamic allocation) |
| BIGDDN dd2 ... N | (N/D N-None, D-Dynamic allocation) |
| BIGRECCDDN dd1 ... N | (N/D N-None, D-Dynamic allocation) |
| BIGRECCDDN dd2 ... N | (N/D N-None, D-Dynamic allocation) |

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 34 on page 121 shows the DD options for COPYDDN dd1. The Dsname Prefix specifies a GDG variable:

**Figure 34: Local primary copy options with GDG variable specified**

<table>
<thead>
<tr>
<th>ASUDUM</th>
<th>COPYDDN ddname1 JCL TABLESPACE QZUD41.%</th>
<th>Row 1 to 10 of 10</th>
</tr>
</thead>
</table>

Service Syntax: BMCCOPY.DEMO_BMCCOPY

Enter data, then press end. Blank fields default to the Workid JCL Options.

-------------------------------- JCL Allocation --------------------------------

| UNIT ... | SYSDA | (Device type / Unit) |
| Dsname Prefix ... | ASU.CPY.ASSID .. &OBNOD(&GDG) |
| Dsname Prefix ... | ASU.CPY.ASSID .. &OBNOD(&GDG) |

(Prefix for output data set name)

| RETPD | (1-9999 Retention period) |
| EXPDT | (YYYY/DDD or YYDDD Expiration date) |

****************************************************************************** Bottom of data *******************************
Press **END** to display the DD options for RECOVERYDDN dd1, the recovery primary data set (Figure 35 on page 122). The **Dsname Prefix** specifies a GDG primary recovery data set.

**Figure 35: Recovery primary copy options with GDG variable specified**

```plaintext
ASUDUM     RECOVERYDDN ddname1 JCL TABLESPACE QZUD41.%        Row 1 to 10 of 10
Command ===>                                                 Scroll ===> CSR
Service Syntax: BMCCOPY.DEMO_BMCCOPY
Enter data, then press end.                                        More:
Blank fields default to the Workid JCL Options.
-------------------------- JCL Allocation -------------------------------------
UNIT . . . . . . . SYSDA    (Device type / Unit)
Dsname Prefix . .  ASU.RCV.&SSID..&OBNOD(&GDG)
(Prefix for output data set name)
RETPD . . . . . .           (1-9999 Retention period)
EXPDT . . . . . .           (YYYY/DDD or YYDDD Expiration date)
******************************* Bottom of data ********************************
```

Press **END** again to display the DD options for RECOVERYDDN dd2, the recovery backup data set (Figure 36 on page 122). The **Dsname Prefix** specifies a GDG data set.

**Figure 36: Recovery backup copy options with GDG variable specified**

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

The product resolves the image-copy data set names in the JCL:

```
-TIME 000000 'yyvy-05-11.09.40.13.00001'
* GENERAL BMCCOPY UTIL
-SSID 000001 DEBA
-WKID 000002 JENBCPY
-SYNC 000003
-BMCC 000004
-COPY
-TABLESPACE QZUD41.QZUS0141
-SHRLEVEL REFERENCE
-FULL YES
-COPYDDN (C0001,D0001)
-RECOVERYDDN (R0001,V0001)
-JCLP 000004 BMCC DDNAME C0001 DSNPREF
ASU.CPY.&SSID..&OBNOD(&GDG)
-JCLP 000004 BMCC DDNAME D0001 DSNPREF
ASU.CPY2.&SSID..&OBNOD(&GDG)
-JCLP 000004 BMCC DDNAME C0001 DEVTYPE SYSDA
-JCLP 000004 BMCC DDNAME D0001 DEVTYPE SYSDA
-JCLP 000004 BMCC DDNAME R0001 DSNPREF
```

122  **DASD MANAGER PLUS for DB2 User Guide**
In the JCL, the product resolves the image-copy data sets as follows:

```plaintext
/*--------------------------------------------------------------------
/* UTILITY COPY DD STATEMENTS
/*--------------------------------------------------------------------
C0001 DD DSN=ASU.CPY.DEBA.QZUD41.QZUS0141(+1),
       DCB=(SYS1.MODEL),
       DISP=(NEW,CATLG,CATLG),
       SPACE=(CYL,(10,2),RLSE),
       UNIT=SYSDA
/*--------------------------------------------------------------------
/* UTILITY DUAL COPY DD STATEMENTS
/*--------------------------------------------------------------------
D0001 DD DSN=ASU.CPY2.DEBA.QZUD41.QZUS0141(+1),
       DCB=(SYS1.MODEL),
       DISP=(NEW,CATLG,CATLG),
       SPACE=(CYL,(10,2),RLSE),
       UNIT=SYSDA
/*--------------------------------------------------------------------
/* UTILITY RECOVERY SITE IMAGE COPY DD STATEMENTS
/*--------------------------------------------------------------------
R0001 DD DSN=ASU.RCV.DEBA.QZUD41.QZUS0141(+1),
       DCB=(SYS1.MODEL),
       DISP=(NEW,CATLG,CATLG),
       SPACE=(CYL,(10,2),RLSE),
       UNIT=SYSDA
/*--------------------------------------------------------------------
/* UTILITY DUAL RECOVERY SITE IMAGE COPY DD STATEMENTS
/*--------------------------------------------------------------------
V0001 DD DSN=ASU.RCV2.DEBA.QZUD41.QZUS0141(+1),
       DCB=(SYS1.MODEL),
       DISP=(NEW,CATLG,CATLG),
       SPACE=(CYL,(10,2),RLSE),
       UNIT=SYSDA
/*--------------------------------------------------------------------
/* END OF JOBSTEP
/*--------------------------------------------------------------------
/* END OF JOB
/*--------------------------------------------------------------------
```

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 reassemble 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 37 on page 124 describes installation options for most DASD MANAGER PLUS utility functions.

Figure 37: DASD MANAGER PLUS installation options

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.

You can create an action options file by selecting 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**

POFDS is a keyword in the ALTER, CATALOG MANAGER, CHANGE MANAGER, and DASD MANAGER PLUS installation options modules. The POFDS parameter in the installations options module specifies an 80-character sequential file called the product options file (POF). The POF contains keywords and values for the JCL Generation options.

The POF is located in the `HLQ.UBMCCNTL` data set. When you install the products, only one POF is created. This POF, referred to as the *initial POF*, is initialized and populated with the default ISPF variables and values from the installation panels. Products that are installed at the same time share the initial POF.

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

**Example of product options**

This section provides an example of a product options file.

**Figure 38: Product options file**

```
*-------------------------------------------------------------
*    POF WRITTEN FROM VERSION:  V11.02.00
* 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
```
Customizing DASD MANAGER PLUS (for experienced users)

POFDATE = 2014/05/16 12:14:10
ACM_AMS = Y
ACM_ANALYSIS_SYSOUT = A
ACM_BASDIAG = SYSOUT
ACM_BRPTDIAG = SYSOUT
ACM_BRPTDSN = '&PREFIX..BASELINE.REPORT'
ACM_CDLDdsn = '&PREFIX..&SSID..CDL(CDL)'
ACM_CDLPs = 15
ACM_CDLS = 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_BRM1 =
ACM_BRM2 =
ACM_BRM3 =
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_JDSNBR = '&PREFIX..BASELINE(BLRPTJCL)'
ACM_JDSNC = '&PREFIX..COMPARE(&WORKID)'
ACM_JDSNCPL = '&PREFIX..TASKID(&TASKID)'
ACM_JDSNCPLo = '&PREFIX..EXEC(&WORKID)'
ACM_JDSNE = '&PREFIX..EXEC(&WORKID)'
ACM_JDSnl = '&PREFIX..IMPORT(&WORKID)'
ACM_PARALLEL_MAXINIT = 3
ACM_PARALLEL_MININIT = 2
ACM_PARALLEL_WORKLIST = 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 = BMCRMDS.V810INST.NONSMPE.LOAD,(R)
ADDLOAD2 = BMCRMDS.V810INST.UBMCLINK,(R)
ARCH_DATACLASS =
ARCH_DATACLASS_ALT =
ARCH_EXPDT =
ARCH_MGMTCLASS =
ARCH_MGMTCLASS_ALT =
ARCH_PREFIX = '&PREFIX..&WKID'
ARCH_PRIOTY = 10
ARCH_RETPD =
ARCH_SECQTY = 2
ARCH_STACK = N
ARCH_STORCLASS =
ARCH_STORCLASS_ALT =
ARCH_THRESH = 0
ARCH_UNIT = SYSDA
ARCH_UNIT_ALT =
ASU_XP_LOGD_DATAC=
ASU_XP_LOGD_MGMTC=
ASU_XP_LOGD_PRIQTY=10
ASU_XP_LOGD_SECQTY=2
ASU_XP_LOGD_STORC=
ASU_XP_LOGD_UNIT=SYSDA
ASU_XP_LOGDSN=&PREFIX..XPORT.LOG
ASU_XP_UIMSRVHOST=
ASU_XP_UIMSRVPORT=1
ASU_XP_UIMSRVTIMEOUT=300
BINDFAIL = N
BLRP_DATACLASS =
BLRP_DATACLASS_ALT =
BLRP_EXPDT =
BLRP_MGMTCLASS =
BLRP_MGMTCLASS_ALT =
BLRP_PREFIX = &PREFIX..&OBNOD
BLRP_PRIQTY = 10
BLRP_RETPD =
BLRP_SECQTY = 2
BLRP_STACK = N
BLRP_STORCLASS =
BLRP_STORCLASS_ALT =
BLRP_THRESH = 0
BLRP_UNIT = SYSDA
BLRP_UNIT_ALT =
BMC_CHECK_LOAD =
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_XBMID =
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 =
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
DATANK_NBR = 4
DATANK_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..&OBNOD
DISC_PRIQTY = 10
DISC_RETDP =
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 = &lt;\gt;
DISP_STATS = N
DISP_VAR_DBUG = N
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_RETDP =
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..&OBNOD..P&PART
FCPY_PRIQTY = 10
FCPY_RETDP =
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_RETDP =
FILT_SECQTY = 2
FILT_STORCLASS =
FILT_UNIT = SYSDA
GDG_MODEL = SYS1.MODEL
HASHFAIL = N
HASHWARNRC =
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
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_BMCOPY_NAME =
PROC_BMCOPY_STEP =
PROC_BMMCRS_NAME =
PROC_BMMCRS_STEP =
PROC_BMCLOAD_NAME =
PROC_BMCLOAD_STEP =
PROC_BMCRECOVER_NAME =
PROC_BMCRECOVER_STEP =
PROC_BMCREORG_NAME =
PROC_BMCREORG_STEP =
PROC_BMCSTATS_NAME =
PROC_BMCSTATS_STEP =
PROC_BMCSTOP_NAME =
PROC_BMCSTOP_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_IDCAMPS_NAME = 
PROC_IDCAMPS_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 = 
PUNCH_PREFIX = &PREFIX..&WKID..&STEPN 
PUNCH_PRIQTY = 1 
PUNCH_RETPD = 
PUNCH_SECQTY = 1 
PUNCH_STORCLASS = 
PUNCH_UNIT = SYSDA 
RCPY1_DATACLASS = 
RCPY1_DATACLASS_ALT = 
RCPY1_EXPDT = 
RCPY1_MGMTCLASS = 
RCPY1_MGMTCLASS_ALT = 
RCPY1_PREFIX = &PREFIX..&OBNOD..P&PART 
RCPY1_PRIQTY = 10 
RCPY1_RETPD = 
RCPY1_SECQTY = 2 
RCPY1_STACK = N 
RCPY1_STORCLASS = 
RCPY1_SUPPRESS_SUFF = N 
RCPY1_THRESH = 0 
RCPY1_UNIT = SYSDA 
RCPY1_UNIT_ALT = 
RCPY2_DATACLASS = 
RCPY2_DATACLASS_ALT = 
RCPY2_EXPDT = 
RCPY2_MGMTCLASS = 
RCPY2_MGMTCLASS_ALT = 
RCPY2_PREFIX = &PREFIX..&OBNOD..P&PART 
RCPY2_PRIQTY = 10 
RCPY2_RETPD = 
RCPY2_SECQTY = 2 
RCPY2_STACK = N 
RCPY2_STORCLASS = 
RCPY2_SUPPRESS_SUFF = N 
RCPY2_THRESH = 0 
RCPY2_UNIT = SYSDA 
RCPY2_UNIT_ALT = 
REBINDFAIL = N 
REBINDRC = 
REGION = 0M 
REORG_MAPDB = 
REORG_MAPTAB = 
REPT_DATACLASS = 
REPT_DATACLASS_ALT = 
REPT_EXPDT = 
REPT_MGMTCLASS = 
REPT_MGMTCLASS_ALT = 
REPT_PREFIX = &PREFIX..&WKID 
REPT_PRIQTY = 10 
REPT_RETPD = 
REPT_SECQTY = 2 
REPT_STORCLASS = 
REPT_STORCLASS_ALT = 
REPT_THRESH = 0
REPT_UNIT = SYSDA
REPT_UNIT_ALT = 
RUNTIME_HLO = &PREFIX.RNTM.(R)
SCHED_TRIG_CNTM_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
SRTOUT_STORCLASS = 
SRTOUT_STORCLASS_ALT = 
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_THRESHOLD = 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
TIMEPARM = 
TRTC = 
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
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 product option keywords

This section provides descriptions of the keywords in the product options file.

See also Example of product options on page 125.

2MEGSQL=N

For CATALOG MANAGER, this keyword indicates whether to allocate a 2-MB buffer for large SQL statements.

This keyword is not included in the AJXPOFIN input stream.

ACTWRK_MGMTCLASS

For CATALOG MANAGER, this keyword specifies the SMS definition for the management class associated with the work data sets.

ACTWRK_STORCLASS

For CATALOG MANAGER, this keyword specifies the SMS definition for the storage class associated with the work data sets.

ACTWRK_DATACLASS

For CATALOG MANAGER, this keyword specifies the SMS definition for the data class associated with the work data sets.

ACTWRK_UNIT

For CATALOG MANAGER, this keyword specifies the unit for the work data sets. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the ACTWRK_UNIT parameter from the JCL, specify NONE.

ACTWRK_PRIQTY

For CATALOG MANAGER, this keyword specifies the primary allocation (in cylinders) for the work data sets if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

ACTWRK_PRIQTY

For CATALOG MANAGER, this keyword specifies the secondary allocation (in cylinders) for the work data sets if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

ACM_ANALYSIS_SYSOUT =A
For ALTER and CHANGE MANAGER, this keyword sets the default output class to a value other than X.

**ACM_AMS=Y**

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Generates AMS statements (IDCAMS DELETE and DEFINE) in a worklist</td>
</tr>
<tr>
<td>N</td>
<td>Generates a -STOP worklist command that enables you to complete the DELETE and DEFINE commands before the DB2 object CREATE commands that are located later in the worklist</td>
</tr>
</tbody>
</table>

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

This keyword is not included in the AJXPOFIN input stream.

**ACM_BASDIAG=SYSOUT**

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

This keyword is not included in the AJXPOFIN input stream.

**ACM_BRPTDIAG=SYSOUT**

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

This keyword is not included in the AJXPOFIN input stream.

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

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

This keyword is not included in the AJXPOFIN input stream.

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

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

This keyword is not included in the AJXPOFIN input stream.
ACM_CDLPS=15

For CHANGE MANAGER, this keyword defines, in tracks, the default value for the primary space allocation of the CDL data set.

This keyword is not included in the AJXPOFIN input stream.

ACM_CDLSS=5

For CHANGE MANAGER, this keyword defines, in tracks, the default value for the secondary space allocation of the CDL data set.

This keyword is not included in the AJXPOFIN input stream.

ACM_CDLU=SYSDA

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

This keyword is not included in the AJXPOFIN input stream.

ACM_CMPDIAG=SYSOUT

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

This keyword is not included in the AJXPOFIN input stream.

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

For the CM/PILOT component of CHANGE MANAGER, this keyword defines the default data set name for generated CDL statements.

This keyword is not included in the AJXPOFIN input stream.

ACM_CPLDIAG=SYSOUT

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

This keyword is not included in the AJXPOFIN input stream.

ACM_CPLWDSN=’&PREFIX..&SSID..&TASKID’

For CHANGE MANAGER, this keyword specifies the default worklist data set name for a new TASKID used in the CM/PILOT component. CHANGE MANAGER dynamically allocates the data set the first time that the data set is used. This data set can be either a sequential file or a partitioned data set (PDS).
This keyword is not included in the AJXPOFIN input stream.

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

**ACM_IMPDIAG=SYSOUT**

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

This keyword is not included in the AJXPOFIN input stream.

**ACM_JDSN=’&PREFIX..ANALYSIS(&WORKID)’**

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

This keyword is not included in the AJXPOFIN input stream.

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

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

This keyword is not included in the AJXPOFIN input stream.

**ACM_JDSNBR=’&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_JDSNCPL0='&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 or BMC Object Administration for DB2 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 or BMC Object Administration for DB2 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 or BMC Object Administration for DB2 solution, this keyword indicates whether a CHANGE MANAGER worklist should be executed in parallel.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Executes the worklist in parallel</td>
</tr>
<tr>
<td></td>
<td>However, if adequate XIM resources are not available, the Execution function fails. In addition, if the required parallelism worklist commands (such as -BEGG and -ENDG) are not included in the worklist, the worklist is not executed in parallel.</td>
</tr>
<tr>
<td>N</td>
<td>Executes the worklist sequentially, even if the required parallelism worklist commands are included in the worklist</td>
</tr>
</tbody>
</table>

ACM_PARALLEL_XIMGRP=XIMACM

For the Database Administration or BMC Object Administration for DB2 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 or BMC Object Administration for DB2 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 or BMC Object Administration for DB2 solution, this keyword indicates whether the XIM technology should be started automatically.

ACM_PARALLEL_XIMTRCE=N

For the Database Administration or BMC Object Administration for DB2 solution, this keyword indicates whether tracing is used during the execution of a worklist.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Y         | Writes tracing records to the AEXPTRAC output data set
AEXPTRAC is dynamically allocated and the output is written to SYSOUT. |
| N         | Does not use tracing, even if an //AEXPTRAC DD statement is specified in the JCL |

ACM_PIC=N

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

This keyword is not included in the AJXPOFIN input stream.

ACM_SDSN=SYSOUT

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

This keyword is not included in the AJXPOFIN input stream.

ACM_SDSNE=SYSOUT

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

This keyword is not included in the AJXPOFIN input stream.

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

For ALTER and CHANGE MANAGER, this keyword defines the default data set name for a worklist that Analysis generates.
This keyword is not included in the AJXPOFIN input stream.

**ACM_WLORDER**

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Sorts the worklist by each table’s cardinality in descending sequence</td>
</tr>
<tr>
<td>N</td>
<td>Sorts the worklist by table order in ascending sequence, according to the table owner and table name</td>
</tr>
<tr>
<td>A</td>
<td>Sorts the worklist by either table cardinality or by table order, depending on whether the worklist is processed in parallel If the Database Administration or BMC Object Administration <em>for DB2</em> solution processes the worklist in parallel, Analysis sorts the worklist by table cardinality. Otherwise, it sorts the worklist by table order.</td>
</tr>
<tr>
<td>Blank</td>
<td>Generates the objects in the worklist in an unsorted, random order</td>
</tr>
</tbody>
</table>

This keyword is not included in the AJXPOFIN input stream.

**ACM_WLORDERMSG=Y**

For ALTER and CHANGE MANAGER, this keyword specifies whether to record in the SYSPRINT data set and in the worklist the amount of time to sort a worklist.

This keyword is not included in the AJXPOFIN input stream.

**ACM_WLPS=15**

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

This keyword is not included in the AJXPOFIN input stream.

**ACM_WLSS=5**

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

This keyword is not included in the AJXPOFIN input stream.

**ACM_WLU=SYSDA**

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

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

This keyword defines the additional LINK library.

--- Tip ---

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

---

ADDLOAD2= HLQ.UBMCLINK

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

--- Tip ---

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

---

ARCH_DATACLASS

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

ARCH_DATACLASS_ALT

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

ARCH_EXPDT

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

ARCH_MGMTCLASS

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

ARCH_MGMTCLASS_ALT

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

ARCH_PREFIX=&PREFIX..&WKID

This keyword specifies the prefix for the name of the archive discard data set.
ARCH_PRIQTY=10

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

ARCH_RETPD

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

ARCH_SECQTY=2

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

ARCH_STACK=N

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

ARCH_STORCLASS

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

ARCH_STORCLASS_ALT

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

ARCH_THRESH=0

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

ARCH_UNIT=SYSDA

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

ARCH_UNIT_ALT

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

ASU_XP_LOGD_DATAC=

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

ASU_XP_LOGD_MGMTC=

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

ASU_XP_LOGD_PRIQTY=10

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

ASU_XP_LOGD_SECQTY=2

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

ASU_XP_LOGD_STORC=10

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

ASU_XP_LOGD_UNIT=SYSDA

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

ASU_XP_LOGD_LOGDSN=&PREFIX..XPORT.LOG

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

ASU_XP_UIMSRVHOST=

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

ASU_XP_UIMSRVPORT=1

For DASD MANAGER PLUS, this keyword specifies the port number of the primary UIM server that contains the host definitions repository for the Export utility.
ASU_XP_UIMSRVTIMEOUT=300

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

BINDFAIL=N

This keyword specifies whether worklist execution continues if a bind fails:

- If BINDFAIL=Y, worklist execution stops with a return code of 8. The stop is noted in the sync tables, and an Execution restart continues with the command that caused the failure.

- If BINDFAIL=N, worklist execution continues.

BLRP_DATACLASS

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

BLRP_DATACLASS_ALT

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

BLRP_EXPDT

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

BLRP_MGMTCLASS

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

BLRP_MGMTCLASS_ALT

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

BLRP_PREFIX=&PREFIX..&OBNOD

For CHANGE MANAGER, this keyword defines the high-level qualifier, or prefix, used for data sets containing data stored for a baseline recovery point.
BLRP\_PRIQTY=10

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

BLRP\_RETPD

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

BLRP\_SECQTY=2

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

BLRP\_STACK=N

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

BLRP\_STORCLASS

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

BLRP\_STORCLASS\_ALT

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

BLRP\_THRESH=0

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

BLRP\_UNIT=SYSDA

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

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

**BMC_CHECK_LOAD**

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

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

**BMC_CHECK_OPTS=ACK$OPTS**

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

**BMC_COPY_LOAD**

This keyword specifies the name of the LINK library for the 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 COPY PLUS utility installation options module. This keyword replaces the COPYDOPT AJXPOFIN keyword. If both BMC_COPY_OPTS and COPYDOPT are included in the POF, the components use the value that is specified for COPYDOPT.

**BMC_LOAD_LOAD**

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

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

**BMC_LOAD_OPTS=AMU$OPTS**

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

**BMC_RECOVER_LOAD**

This keyword specifies the name of the LINK library for the 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 RECOVER PLUS utility. This keyword replaces the RECOVERDOPT AJXPOFIN keyword. If both BMC_RECOVER_OPTS and RECOVERDOPT are included in the POF, the components use the value that is specified for RECOVERDOPT.

**BMC_REORG_LOAD**

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

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

**BMC_REORG_OPTS=ARU$OPTS**

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

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_LOAD

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

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

CHECK+_LOAD

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

CHECKDOPT=ACK$MMS

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

CHGMAN_LOAD

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

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

CLEANUP_RC=4

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

CNTL_DATACLASS

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

CNTL_EXPDT

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

CNTL_MGMTCLASS

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

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

This keyword specifies the prefix for the name of the control data set that the BMC UNLOAD PLUS utility uses.
CNTL_PRIQTY=1

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

CNTL_RETPD

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

CNTL_SECQTY=1

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

CNTL_STORCLASS

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

CNTL_UNIT=SYSDA

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

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

For DASD MANAGER PLUS, this keyword specifies the name of the output data set that contains the job schedule name and the job sequence number of the non-IEFBR14 jobs.

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

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

COPY+_LOAD

This keyword specifies the name of the LINK library for the 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 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 COPY PLUS EXPORT command creates to
migrate data.

**CPYEXP_EXPDT**

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

**CPYEXP_MGMTCLASS**

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

**DASD_LOAD**

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

---

**Tip**

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

---

**DASDDOPT**

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

**DATA_PACKER_LOAD**

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

---

**Tip**

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

---

**DATASETSIZING=N**

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Does not perform data set sizing</td>
</tr>
<tr>
<td>C</td>
<td>Uses IBM RUNSTATS to perform data set sizing by using statistics from the DB2 catalog</td>
</tr>
<tr>
<td>B</td>
<td>Uses BMCSTATS to perform data set sizing by using the statistics from the BMC DASD MANAGER PLUS product tables</td>
</tr>
<tr>
<td>O</td>
<td>Physically and randomly samples the VSAM objects to estimate data set sizes</td>
</tr>
</tbody>
</table>

**DATAWK_NBR=4**

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

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

DB2EXIT

This keyword specifies the name of the DB2 EXIT library.

Tip

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

DB2LOAD

This keyword specifies the name of the DB2 LOAD library.

Tip

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

DEF_GDG_BASE=N

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

DEF_GDG_LIMIT=10

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

DEF_GDG_NOSCR=N

This keyword specifies whether the base of a generation data group (GDG) is defined in the IDCAMS DEFINE command as EMPTY (NSCR). If the GDG is defined as EMPTY (NSCR), the operating system uncatalogs the generation data set when it reaches the maximum number of generation data sets to keep (LIMIT). Otherwise, if the GDG is defined as SCRATCH (SCR), the operating system scratches (deletes) the generation data set when the data set is uncataloged.

DEF_GDG2_LIMIT=10

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

This keyword specifies the SYSOUT class that the components use for reporting incorrect entries in the POF. The default value is blank, which indicates that a report is not generated when the product is invoked. The asterisk (*) is a valid value in batch mode.

DISC_DATACLASS

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

DISC_DATACLASS_ALT

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

DISC_EXPDT

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

DISC_MGMTCLASS

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

DISC_MGMTCLASS_ALT

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

DISC_PREFIX=&PREFIX..&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.Allowopup=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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Indicates to display the object name in a dialog</td>
</tr>
<tr>
<td></td>
<td>If the name is too long to be displayed in a dialog, the product displays the name in a panel.</td>
</tr>
<tr>
<td>N</td>
<td>Indicates to display the object name in a panel</td>
</tr>
</tbody>
</table>
DISP_AUTO_TAB=+

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Indicates not to display an autotab character</td>
</tr>
<tr>
<td>any character other than N</td>
<td>Displays as the autotab character</td>
</tr>
</tbody>
</table>

DISP_LOCATION=M

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

The following values are valid:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Replaces characters at the left end (beginning) of the name</td>
</tr>
<tr>
<td>M</td>
<td>Replaces characters in the middle of the name</td>
</tr>
<tr>
<td>E</td>
<td>Replaces characters at the right end (end) of the name</td>
</tr>
</tbody>
</table>

DISP_OMIT_CHAR= <>

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

DISP_STATS=N

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

DISP_VAR_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).
Parameter | Description
---|---
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 formats are yyddd or yyyy/ddd.

**ERR_MGMTCLASS**

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

**ERR_MGMTCLASS_ALT**

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

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

ERR_PRIQTY=10

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

ERR_RETPD

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

ERR_SECQTY=2

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

ERR_STORCLASS

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

ERR_STORCLASS_ALT

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

ERR_THRESH=0

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

ERR_UNIT=SYSDA

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

ERR_UNIT_ALT

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

EXEC_LOAD

This keyword specifies the name of the Execution LINK library.

Tip

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

FCPY_DATACLASS

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

FCPY_EXPDT

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

FCPY_MGMTCLASS

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

FCPY_PREFIX=&PREFIX..&OBNOD..P&PART

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

FCPY_PRIQTY=10

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

FCPY_RETPD

For CATALOG MANAGER, this keyword specifies the retention period for the flashcopy data set on tape. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume). A data set cannot have an expiration date and a retention period. Valid values are 1 to 9999.
FCPY_SECQTY=2

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

FCPY_STORCLASS

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

FCPY_SUPPRESS_SUFF=N

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

FCPY_UNIT=SYSDA

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

FILT_DATACLASS

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

FILT_EXPDT

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

FILT_MGMTCLASS

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

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

This keyword specifies the prefix for the name of the filter data set.
FILT_PRIQTY=10

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

FILT_RETPD

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

FILT_SECQTY=2

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

FILT_STORCLASS

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

FILT_UNIT=SYSDA

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

GDG_MODEL=SYS1.MODEL

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

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

HASHFAIL=N

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

HASHWARNRC

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

Note
Do not use 8 for this value.
INCLUDE_SYSPRIN2=N

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

//SYSPRIN2 DD SYSOUT=*  

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

- CHECK PLUS
- LOADPLUS
- REORG PLUS
- UNLOAD PLUS

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

**Note**

SYSPRIN2 data sets have the following restrictions:

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

IOALOAD1

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

**Tip**

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

IOALOAD2

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

**Tip**

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

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

JCLLIB

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

JES3=N

JCL Generation no longer uses this keyword.

JOB_INCLUDE_MEMBER

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

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

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

LISTDEF_DSN

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

LL_CLIB=CLIB
LL_CLIB2
LL_CLIB3
LL_CLIB4
LL_CLIB5

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

LL_LINK=LINK
LL_LINK2
LL_LINK3
LL_LINK4
LL_LINK5

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

LL_MLIB=MLIB
LL_MLIB2
LL_MLIB3
LL_MLIB4
LL_MLIB5

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

LL_PLIB=PLIB
LL_PLIB2
LL_PLIB3
LL_PLIB4
LL_PLIB5

These keywords specify the LLQ for the panel and Help library data sets for the batch ISPF environment.

LL_SLIB=SLIB
LL_SLIB2
LL_SLIB3
LL_SLIB4
LL_SLIB5

These keywords specify the LLQ for the ISPF skeleton data sets for the batch ISPF environment.

LL_TLIB=TLIB
LL_TLIB2
LL_TLIB3
LL_TLIB4
LL_TLIB5

These keywords specify the LLQ for the ISPF table data sets for the batch ISPF environment.

LL_XML=XML
LL_XML2
LL_XML3
LL_XML4
LL_XML5

These keywords specify the LLQ for the utility generation data sets for the batch ISPF environment.
LLQ

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

*Note*

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

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).
MAP_THRESH=0

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

MAP_UNIT=SYSDA

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

MAP_UNIT_ALT

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

MAX_CYL=99999

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

MAX_PRIQTY=2000

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

MAX_SECQTY=200

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

MAX_UNITCNT

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

MEMLIMIT

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

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

PCPY1_DATACLASS

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

PCPY1_DATACLASS_ALT

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

PCPY1_EXPDT

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

PCPY1_MGMTCLASS

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

PCPY1_MGMTCLASS_ALT

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

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

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

PCPY1_PRIQTY=10

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

PCPY1_RETPD

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

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

PCPY1_STACK=N

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

PCPY1_STORCLASS

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

PCPY1_STORCLASS_ALT

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

PCPY1_SUPPRESS_SUFF=N

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

PCPY1_THRESH=0

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

PCPY1_UNIT=SYSDA

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

PCPY1_UNIT_ALT

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

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

PCPY2_DATACLASS_ALT

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

PCPY2_EXPDT

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

PCPY2_MGMTCLASS

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

PCPY2_MGMTCLASS_ALT

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

PCPY2_PREFIX=&PREFIX..&OBNOD..P&PART

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

PCPY2_PRIQTY=10

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

PCPY2_RETPD

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

PCPY2_SECQTY=2

This keyword specifies the secondary allocation (in cylinders) for the local backup copy data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.
PCPY2_STACK=N

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

PCPY2_STORCLASS

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

PCPY2_STORCLASS_ALT

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

PCPY2.Suppress_Suff=N

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

PCPY2_THRESH=0

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

PCPY2_UNIT=SYSDA

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

PCPY2_UNIT_ALT

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

POFDATE

This keyword shows the last date on which the initial product options file (POF) was updated. This value is created or updated when the POF is created or when it is updated by the AJXPOVAL or AJXPODAT edit macros.
**PRE_JOBSTEP_INCLUDE**

This keyword specifies the name of a JCL member to be included before each step in the JCL.

**PROC_BMCCHECK_NAME**

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

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCCHECK_STEP**

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

This keyword is not included in the AJXPOFIN input stream.

**PROC_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 COPY PLUS 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 COPY PLUS utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCCPRS_NAME**

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

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

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

This keyword is not included in the AJXPOFIN input stream.

PROC_BMCLOAD_NAME

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

This keyword is not included in the AJXPOFIN input stream.

PROC_BMCLOAD_STEP

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

This keyword is not included in the AJXPOFIN input stream.

PROC_BMCRECOVER_NAME

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

This keyword is not included in the AJXPOFIN input stream.

PROC_BMCREORG_NAME

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

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

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

This keyword is not included in the AJXPOFIN input stream.

PROC_BMCSTATS_NAME

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

This keyword is not included in the AJXPOFIN input stream.

PROC_BMCSTATS_STEP

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

This keyword is not included in the AJXPOFIN input stream.

PROC_BMCSTOP_NAME

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility. BMCSTOP refers to the part of the utility that issues a DB2 STOP command on an object and verifies the completion of the command.

This keyword is not included in the AJXPOFIN input stream.

PROC_BMCSTOP_STEP

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility. BMCSTOP refers to the part of the utility that issues a DB2 STOP command on an object and verifies the completion of the command.

This keyword is not included in the AJXPOFIN input stream.

PROC_BMCTRIG_NAME

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

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCTRIG_STEP**

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

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCUNLOAD_NAME**

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

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCUNLOAD_STEP**

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

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCUPRS_NAME**

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

BMCUPRS refers to the part of the utility that takes the statistics from the tables in the DASD MANAGER PLUS database (with the BMCSTATS utility) and updates the DB2 catalog with the statistics.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCUPRS_STEP**

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

BMCUPRS refers to the part of the utility that takes the statistics from the tables in the DASD MANAGER PLUS database (with the BMCSTATS utility) and updates the DB2 catalog with the statistics.
This keyword is not included in the AJXPOFIN input stream.

**PROC_DSNUTILB_NAME**

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

This keyword is not included in the AJXPOFIN input stream.

**PROC_DSNUTILB_STEP**

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

This keyword is not included in the AJXPOFIN input stream.

**PROC_DSN1COPY_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM DSN1COPY utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_DSN1COPY_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM DSN1COPY utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_GEN_SET_VAR=N**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies whether to include an SLIB that generates SET statements in the JCL for variables that you can use in catalog procedures (PROCs).

This keyword is not included in the AJXPOFIN input stream.

**PROC_IDCAMS_NAME**

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

This keyword is not included in the AJXPOFIN input stream.
PROC_IDCAM_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_IERFBR14_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 IERFBR14 job.

This keyword is not included in the AJXPOFIN input stream.

PROC_IERFBR14_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 IERFBR14 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..&OBNO..P&PART

This keyword specifies the prefix for the name of the recovery primary copy data set.
RCPY1_PRIQTY=10

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

RCPY1_RETPD

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

RCPY1_SECQTY=2

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

RCPY1_STACK=N

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

RCPY1_STORCLASS

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

RCPY1_STORCLASS_ALT

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

RCPY1_SUPPRESS_SUFF=N

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

RCPY1_THRESH=0

This keyword specifies the maximum anticipated size for the recovery primary copy data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.
RCPY1_UNIT=SYSDA

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

RCPY1_UNIT_ALT

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

RCPY2_DATACLASS

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

RCPY2_DATACLASS_ALT

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

RCPY2_EXPDT

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

RCPY2_MGMTCLASS

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

RCPY2_MGMTCLASS_ALT

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

RCPY2_PREFIX=&PREFIX..&OBNOD..P&PART

This keyword specifies the prefix for the name of the recovery backup copy data set.

RCPY2_PRIQTY=10

This keyword specifies the primary allocation (in cylinders) for the recovery backup copy data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.
**RCPY2\_RETPD**

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

**RCPY2\_SECQTY=2**

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

**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 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.

RECOVERRDOPT

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

REGION=0M

This keyword defines the REGION parameter in the EXEC statement.

REORG_MAPDB

This keyword specifies the mapping database that ALTER and CHANGE MANAGER provide to the IBM REORG utility as an override to the value of the DB2 subsystem parameter REORG_MAPPING_DATABASE. CATALOG MANAGER and DASD MANAGER PLUS do not use this value. The name can be from 1 to 8 characters long, and cannot include symbolic variables.
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.
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_CNTL_M_JOBS=N

This keyword specifies whether JCL Generation should generate BMC Control-M job schedule entries for jobs that do not contain IEFBR14 steps (Y or N). If the value is Y, JCL Generation generates an input data set (CNTLMSCH) and an output data set (CNTLMOUT) in the JCL for the BMCTRIG utility.

SORTWK_NBR=4

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

SORTWK_PRIQTY=10

This keyword specifies the primary allocation (in cylinders) for the DATAWORK, LOGSORT, or SORTWORK data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.
SORTWK_SECQTY=2

This keyword specifies the secondary quantity (in cylinders) for the DATAWORK, LOGSORT, or SORTWORK data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

SORTWK_UNIT=SYSDA

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

SQLEXP_LOAD

This keyword specifies the name of the LINK library for the BMC SQL Explorer for DB2 product.

Tip

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

SRTOUT_DATACLASS

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

SRTOUT_DATACLASS_ALT=CART

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

SRTOUT_EXPDT

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

SRTOUT_MGMTCLASS

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

SRTOUT_MGMTCLASS_ALT

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

SRTOUT_PREFIX=&PREFIX..&WKID..&STEPN

This keyword specifies the prefix for the name of the SORTOUT data set.
SRTOUT_PRIQTY=10

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

SRTOUT_RETPD=

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

SRTOUT_SECQTY=2

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

SRTOUT_STORCLASS

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

SRTOUT_STORCLASS_ALT

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

SRTOUT_THRESH=0

This keyword specifies the maximum anticipated size for the SORTOUT data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero (0) indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.

SRTOUT_UNIT=SYSDA

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

SRTOUT_UNIT_ALT=CART

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

This keyword specifies the name of a JCL member to be included after each step in the JCL.

STOPWAIT=3

This keyword specifies the number of intervals to wait for a DB2 STOP command to stop a database or table space. Zero (0) indicates that if the object does not stop, the worklist stops without waiting. The maximum value allowed for this keyword is 10 (which is 280 seconds).

STOPWTSECS=10

This keyword specifies the number of seconds to wait (during the first interval of the STOPWAIT keyword) for a DB2 STOP command to stop a database or table space.

SUPPRESS_COMMENTS=N

This keyword specifies whether to suppress the comments in the generated JCL (Y or N).

SYNCDELETE=N

This keyword specifies whether Execution should remove all sync entries when an Execution job completes with no errors (Y or N).

SYSEXEC

This keyword specifies the name of the partitioned data set in which a REXX EXEC is a member.

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

SYSTEM_MLIB

This keyword specifies the name of the system ISPF message library. You specify the value for this keyword during installation.

SYSUT_DATACLASS

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

SYSUT_DATACLASS_ALT

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

This keyword specifies the expiration date of the SYSUT data set on tape. A data set cannot have an expiration date and a retention period. The valid values are 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Does not use a 7-track drive</td>
</tr>
<tr>
<td>C</td>
<td>Uses odd parity, conversion on, and translation off</td>
</tr>
<tr>
<td>E</td>
<td>Uses even parity, conversion off, and translation off</td>
</tr>
<tr>
<td>T</td>
<td>Uses odd parity, conversion off, and translation on</td>
</tr>
<tr>
<td>ET</td>
<td>Uses even parity, conversion off, and translation on</td>
</tr>
<tr>
<td>COMP</td>
<td>Uses data compression on</td>
</tr>
<tr>
<td>NOCOMP</td>
<td>Uses data compression off</td>
</tr>
</tbody>
</table>

TSOPROGRAM

This keyword specifies an alternate TSO monitor program for standard JCL. TSOPROGRAM is available for nonworklist JCL.

TSOSUBEXIT=N

This keyword specifies whether to use a TSO submit exit to generate job cards. If TSOSUBEXIT=Y, no job cards are put in the JCL (Y or N).

ULLQ

This keyword specifies the LLQ for user-defined data sets for the installation environment. During installation, if you chose to use the runtime enablement feature, the Installation System set this value to blank, and you should not
change the value. If you chose not to use the feature, the Installation System set the value to UDB.

---

**Note**

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

---

**UNLD_FREF_DATACLASS**

For ALTER and CHANGE MANAGER, this keyword specifies the SMS definition for the data class associated with the file reference (SYSREC) data set.

**UNLD_FREF_DIRBLOCK = 250**

For ALTER and CHANGE MANAGER, this keyword specifies the number of directory blocks for the file reference (SYSREC) data set.

**UNLD_FREF_MGMTCLASS**

For ALTER and CHANGE MANAGER, this keyword specifies the SMS definition for the storage class associated with the file reference (SYSREC) data set.

**UNLD_FREF_PREFIX = &PREFIX..&MSSID..&WORKID8**

For ALTER and CHANGE MANAGER, this keyword specifies the prefix for the name of the file reference (SYSREC) data set.

**UNLD_FREF_PRIQTY = 10**

For ALTER and CHANGE MANAGER, this keyword specifies the primary allocation (in cylinders) for the file reference (SYSREC) data set. Valid values are 1 through 99999.

**UNLD_FREF_SECQTY = 2**

For ALTER and CHANGE MANAGER, this keyword specifies the secondary allocation (in cylinders) for the file reference (SYSREC) data set. Valid values are 1 through 99999.

**UNLD_FREF_STORCLASS**

For ALTER and CHANGE MANAGER, this keyword specifies the SMS definition for the storage class associated with the file reference (SYSREC) data set.

**UNLD_FREF_SUPPR_SUFF=N**

For ALTER and CHANGE MANAGER, this keyword specifies whether to suppress adding the DD name to the end of the name of the file reference.
(SYSREC) data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

**UNLD_FREF_UNIT = SYSDA**

For ALTER and CHANGE MANAGER, this keyword specifies the unit for the file reference (SYSREC) data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**UNLD1_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the primary unload (SYSREC) data set.

**UNLD1_DATACLASS_ALT**

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

**UNLD1_EXPDT**

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

**UNLD1_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set.

**UNLD1_MGMTCLASS_ALT**

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

**UNLD1_PREFIX=&USERID..&MSSID..&WORKID8**

This keyword specifies the prefix for the name of the primary unload (SYSREC) data set.

**UNLD1_PRIQTY=10**

This keyword specifies the primary allocation (in cylinders) for the primary unload (SYSREC) data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.
UNLD1_RETPD

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

UNLD1_SECQTY=2

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

UNLD1_STACK=N

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

UNLD1_STORCLASS

This keyword specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set.

UNLD1_STORCLASS_ALT

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

UNLD1_SUPPRESS_SUFF=N

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

UNLD1_THRESH=0

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

UNLD1_UNIT=SYSDA

This keyword specifies the unit for the primary unload (SYSREC) data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
UNLD1_UNIT_ALT

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

UNLD2_DATACLASS

This keyword specifies the SMS definition for the data class associated with the backup unload (SYSREC) data set.

UNLD2_DATACLASS_ALT

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

UNLD2_EXPDT

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

UNLD2_MGMTCLASS

This keyword specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set.

UNLD2_MGMTCLASS_ALT

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

UNLD2_PREFIX=&USERID..&MSSID..&WORKID8

This keyword specifies the prefix for the name of the backup unload (SYSREC) data set.

UNLD2_PRIQTY=10

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

UNLD2_RETPD

This keyword specifies the retention period for the backup unload (SYSREC) data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.
UNLD2_SECQTY=2

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

UNLD2_STACK=N

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

UNLD2_STORCLASS

This keyword specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set.

UNLD2_STORCLASS_ALT

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

UNLD2_SUPPRESS_SUFF=N

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

UNLD2_THRESH=0

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

UNLD2_UNIT=SYSDA

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

UNLD2_UNIT_ALT

This keyword specifies the alternate unit name for the backup unload (SYSREC) data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
UNLD3_DATACLASS

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the SMS definition for the data class associated with the ROWID SYSREC data set.

UNLD3_EXPDT

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the expiration date of the ROWID SYSREC data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.

UNLD3_MGMTCLASS

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the SMS definition for the storage class associated with the ROWID SYSREC data set.

UNLD3_PREFIX=&USERID..&MSSID..&WORKID8

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the prefix for the name of the ROWID SYSREC data set.

UNLD3_RETPD

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the retention period for the ROWID SYSREC data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

UNLD3_STORCLASS

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the SMS definition for the storage class associated with the ROWID SYSREC data set.

UNLD3_SUPPRESS_SUFF=N

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies whether to suppress adding the DD name to the end of the name of the ROWID SYSREC data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

UNLD3_UNIT=SYSDA

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the unit for the ROWID SYSREC data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
UNLD4_DATACLASS

For version 1.2 or earlier of the LOB DATA MOVER program in the Database Administration or BMC Object Administration for DB2 solution, this keyword specifies the SMS definition for the data class associated with the large object (LOB) SYSREC data set.

UNLD4_MGMTCLASS

For version 1.2 or earlier of the LOB DATA MOVER program in the Database Administration or BMC Object Administration for DB2 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 version 1.2 or earlier of the LOB DATA MOVER program in the Database Administration or BMC Object Administration for DB2 solution, this keyword specifies the prefix for the name of the large object (LOB) SYSREC data set.

UNLD4_STORCLASS

For version 1.2 or earlier of the LOB DATA MOVER program in the Database Administration or BMC Object Administration for DB2 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 version 1.2 or earlier of the LOB DATA MOVER program in the Database Administration or BMC Object Administration for DB2 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 version 1.2 or earlier of the LOB DATA MOVER program in the Database Administration solution or BMC Object Administration for DB2, this keyword specifies the unit for the large object (LOB) SYSREC data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

UNLOAD+_LOAD

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

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

USER_HLQ

This keyword specifies the high-level qualifier (HLQ) used by user defined data sets. This HLQ is also used for the runtime enablement (RTE) data sets.

USER_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 on page 203.

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 or corrective action</td>
</tr>
<tr>
<td>3</td>
<td>Utility parameters in the default or corrective 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 properties JCL Options as shown in the following figure, not the BMCTRIG action.

```
ASUIAPO ---------------------- Edit Action Properties ----------------------
Command ===>
Type Action data. Then press End.
Action . . . . : RDAPKM.DEMO Updated by : RDAPKM
Status . . . . : NOT GENERATED       Last Updated: yyyy-04-05
Description ..
Data Sets
  JCL Options   : ADM.INST1120.UBMCCNTL(AJXB2P00)
  JCL           :
  Worklist      :
  Diagnostics   :
Worklist Generated
  On . . . . . :
  By . . . . . :
  RC . . . . . : 0
```

**Note**
The BMCTRIG utility retrieves no information from the TSO ISPF profile or from user options.
Utilities other than BMCTRIG use information from the sources in Table 26 on page 204.

Table 26: Information sources for utilities other than BMCTRIG

<table>
<thead>
<tr>
<th>Order of use</th>
<th>Source</th>
<th>Affects the following:</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 “JCL Generation data sets sizing function” on page 116.

**Variables**

DASD MANAGER PLUS variables begin with one or two ampersands (&), depending upon their location.

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</td>
</tr>
<tr>
<td></td>
<td>The assembly code requires the installation options module to contain double ampersands to correctly convert the options into symbolic variables.</td>
</tr>
</tbody>
</table>

*a* 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.&&JOBCHAR 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, the &USERID installation option has one ampersand and a period delimiter between consecutive options.

**Figure 39: 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 resolves to **RND001**.

**Installation options**

This topic describes the options listing that the installation process creates for DASD MANAGER PLUS.

The figure below shows the installation options contained within the module that the installation process creates in $xnnDOPT and also in HLQ.BMCCNTL (where HLQ is the high-level qualifier). They have the same member name as the installation options module. For descriptions of individual installation options, see “Descriptions of the installation options” on page 206.

**Figure 40: DASD MANAGER PLUS installation options**

```
* MODULE : ASUDOPTS
* FUNCTION : DASD DEFAULT OPTIONS
* COPYRIGHT : COPYRIGHT BMC SOFTWARE INC., yyyy
* LEVEL : RELEASE vv.rr June yyyy
* FUNCTIONS : DEFINE THE DEFAULT PROFILE VARIABLES
* ASUDOPTS $ALUDOPT PRODUCT='DASD MANAGER',
  DATE=&SYSDATC,
  ZIIP=E,
  JDSNE='''&PREFIX..EXEC(&&WKID8)'',
  SYSTRIGS=N,
  Db2ntry=100,
  Db2Wait=3,
  Wdsn='''&PREFIX..&SSID..&WKID8''',
  Location=,
  EPP=IEX112DM,
  EAP=IEX112DA,
  EIP=BMIINSB2,
  PoFds=(ADM.INST1120.UBMCCNTL(AJXB2PO0),R),
  OPNDB2ID=(Y,R),
  Osauthchk=N,
  Statauth=(Y,R).
```
DPNAM=DPDELOAD, *  
ISPTLIB='ADM.INST1120.BMCTLIB', *  
UPDCATIXS=N, *  
UPDNUCAT=Y, *  
CPLAN=ISU112DC, *  
JPLAN=ISU112DJ, *  
SPLAN=ISU112DS, *  
ZPLAN=ISU112DZ, *  
RPLAN=ISU112DR, *  
WU=SYSDA, *  
WPS=10, *  
WSS=2, *  
BMCSYNC=Y, *  
FREQVAL=Y, *  
ATBWORKAREA=N, *  
ATBLARGEPEGS=N, *  
QUIESCEINTERVAL=0, *  
OPTIMIZECOMMIT=Y, *  
PERFADVLOC=, *  
LOGMETRICS=, *  
NPICACHEACTION=A, *  
NPICACHETHRESH=100, *  
NPICACHEDSNUM=25  
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.

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 “Control of your environment by using options” on page 63.

**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 BMC Products and Solutions for DB2 Customization Guide 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=ASU\$orDC

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 "Overview of product options" on page 125.

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 BMCUTIL, 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 BMCUTIL, 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.

**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 “Overview of product options” on page 125.
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 “Overview of product options” on page 125.

ISPTLIB='HLQ.TLIB'

This option is the fully qualified data set name for the ISPF TLIB. BMCTRIG uses this data set name.

JC1='

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 the supported symbolic variables table in User-defined programs or services on page 228.

The JOBCARD1, JOBCARD2, JOBCARD3, JOBCARD4, and JOBCARD5 keywords replace these installation options. For information about the POF keywords, see “Overview of product options” on page 125.

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.

JCLCLEANUP POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

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.

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 “Overview of product options” on page 125.

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).

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 “Access to actions, services, and service syntax” on page 278 and “Controlling access to Object Sets” on page 311.

POFDS='&&HLQ..BMCCNTL(&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'.

210 DASD MANAGER PLUS for DB2 User Guide
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 “Overview of product options” on page 125.

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 “Overview of product options” on page 125.

RECVPREF='&&PREFIX..&&OBNOD..P&&PART'

This option specifies the default prefix (high-level qualifier) for the RECV nnn recovery 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 RCPY1_PREFIX POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.
RECVPS=10

This option specifies the default primary space allocation, in cylinders, for RECV nnn recovery data sets.

The RCPY1_PRIQTY POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

RECVSS=2

This option specifies the default secondary space allocation, in cylinders, for RECV nnn recovery data sets.

The RCPY1_SECQTY POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

RECVUNIT=SYSDA

This option is the default unit that is used for creating RECV nnn recovery data sets.

The RCPY1_UNIT POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

RPLAN=ASU

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 “Overview of product options” on page 125.

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 “Overview of product options” on page 125.
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 “Overview of product options” on page 125.

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 “Overview of product options” on page 125.

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 “Overview of product options” on page 125.

SPLAN=ASU

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 “Overview of product options” on page 125.

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 “Overview of product options” on page 125.

**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 “Overview of product options” on page 125.

**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“.

**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 “Overview of product options” on page 125.

**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 “Overview of product options” on page 125.

**SYSCPREF='&&PREFIX..&&OBNOD..P&&PART'**

This option specifies the default prefix (high-level qualifier) for the SYSCO nnn 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 “Overview of product options” on page 125.

**SYSCPS=10**

This option specifies the primary space allocation, in cylinders, for SYSCO nnn data sets.

The PCPY1_PRIQTY POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

**SYSCSS=2**

This option specifies the secondary space allocation, in cylinders, for SYSRCO nnn data sets.

The PCPY1_SECQTY POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

**SYSCUNIT=SYSDA**

This option specifies the default UNIT for creating SYSCO nnn data sets.

The PCPY1_UNIT POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

**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 “Overview of product options” on page 125.

**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 “Overview of product options” on page 125.
**SYSRPREF='&&PREFIX..&&OBNOD'**

This option specifies the default prefix (high-level qualifier) for the SYSRE nnn data sets.

The UNLD1_PREFIX POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

**SYSRPS=10**

This option specifies the primary space allocation, in cylinders, for SYSRE nnn data sets.

The UNLD1_PRIQTY POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

**SYSRSS=2**

This option specifies the secondary space allocation, in cylinders, for SYSRE nnn data sets.

The UNLD1_SECQTY POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

**SYSRUNIT=SYSDA**

This option specifies the default UNIT for creating SYSRE nnn data sets.

The UNLD1_UNIT POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

**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 “Overview of product options” on page 125.

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 “Overview of product options” on page 125.

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 non-updatable 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 DATACLAS= in the JCL for workfiles.

The WORK_DATACLASS POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

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 MGMTCLAS= in the JCL for workfiles.

The WORK_MGMTCLASS POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.

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 “Overview of product options” on page 125.

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 STORCLAS= in the JCL for nontape work files.

The WORK_STORCLASS POF keyword replaces this installation option. For information about the POF keyword, see “Overview of product options” on page 125.
**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 “Overview of product options” on page 125.

**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 “Overview of product options” on page 125.

**ZIIP=E**

This option determines whether BMCSTATS attempts to use zIIPs.

- If you specify \texttt{E} and a zIIP is available, BMCSTATS attempts to offload eligible processing to the zIIP.
- If you specify \texttt{D}, the product prevents BMCSTATS from offloading to the zIIP.

To override this setting, specify ZIIP=E or ZIIP=D in the BMCSTATS syntax.

**ZPLAN=ASU\texttt{rDZ}**

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
  — LOW2KEY

■ Partial statistics rollup for COLCARRDATA 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 ignores SQL -911 errors and continue processing. Objects that receive SQL -911 errors still need to be reprocessed.

- 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.

---
**Note**

Specifying OPTIMIZECOMMIT NO increases BMCSTATS processing overhead.

---

- If you are processing a very large number of objects and have experienced main storage depletion problems, you can specify QUIESCEINTERVAL.

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.

### 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 SMFPRM xx 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.
Maintaining and generating Service Actions

This chapter explains how to create, modify, and generate Service Actions for the DASD MANAGER PLUS product.

Overview of Service Actions

A Service Action identifies one or more services (utilities) that you want to generate. You can create, edit, and maintain Service Actions. To use Service Actions, select Service Actions from the DASD MANAGER PLUS Main Menu.

Service Action maintenance and generation involves the following tasks:

■ Creating an action
■ Specifying object information
■ 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 “Analyzing objects by using BMCTRIG” on page 425.
Service Actions

A Service 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 41 on page 224, the Service Action called WEEKLYREORG has three Services:

- BMCREORG
- BMCSTATS
- BMCCOPY

Each of these Services refers to a named Service syntax. The Service 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.

Figure 41: Example relationship of Service Actions to Services

Services

Services are programs (or utilities) that DASD MANAGER PLUS runs as part of a Service 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 with default syntax parameters. You can copy the supplied BMC default syntax to your own syntax. You can also modify the syntax parameters as needed. DASD MANAGER PLUS can easily run these Services, which include most BMC Software and IBM utilities. You cannot delete these Services.</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>User-defined Services</td>
<td>You can define the attributes and syntax to include the Service in an Service Action.</td>
</tr>
</tbody>
</table>

**BMC Software required services**

The following table lists the Services that BMC Software provides. You can run these services in DASD MANAGER PLUS actions.

**Table 28: DASD MANAGER PLUS utilities and commands**

<table>
<thead>
<tr>
<th>Service type</th>
<th>Service name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics utilities</td>
<td>BMCCPRS</td>
</tr>
<tr>
<td></td>
<td>BMCSTATS</td>
</tr>
<tr>
<td></td>
<td>BMCTTRIG</td>
</tr>
<tr>
<td></td>
<td>BMCUPRIS</td>
</tr>
<tr>
<td></td>
<td>MODISTAT</td>
</tr>
<tr>
<td></td>
<td>RUNSTATS</td>
</tr>
<tr>
<td></td>
<td>STOSPACE</td>
</tr>
<tr>
<td>Image Copy utilities</td>
<td>BMCCOPY</td>
</tr>
<tr>
<td></td>
<td>BMCCOPYI</td>
</tr>
<tr>
<td></td>
<td>BMCMOD</td>
</tr>
<tr>
<td></td>
<td>FULLCOPY</td>
</tr>
<tr>
<td></td>
<td>INCRCOPY</td>
</tr>
<tr>
<td></td>
<td>DSN1COPY</td>
</tr>
<tr>
<td></td>
<td>MERGCOPY</td>
</tr>
<tr>
<td></td>
<td>MODICOPY</td>
</tr>
<tr>
<td>Reorganize utilities</td>
<td>BMCREORG</td>
</tr>
<tr>
<td></td>
<td>REORG</td>
</tr>
<tr>
<td></td>
<td>RESIZE</td>
</tr>
<tr>
<td></td>
<td>ALTERSEC</td>
</tr>
<tr>
<td>Note:</td>
<td>You cannot generate RESIZE online; you can generate it only through BMCTTRIG.</td>
</tr>
<tr>
<td>UNLOAD and LOAD utilities</td>
<td>BMCUNLOAD</td>
</tr>
<tr>
<td></td>
<td>UNLOAD</td>
</tr>
<tr>
<td></td>
<td>BMCLOAD</td>
</tr>
<tr>
<td></td>
<td>LOAD</td>
</tr>
<tr>
<td>Service type</td>
<td>Service name</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>DB commands and utilities</td>
<td>QUIESCE</td>
</tr>
<tr>
<td></td>
<td>REPAIR</td>
</tr>
<tr>
<td></td>
<td>START DB</td>
</tr>
<tr>
<td></td>
<td>STOP DB</td>
</tr>
<tr>
<td>Check and Report utilities</td>
<td>CHECK DATA</td>
</tr>
<tr>
<td></td>
<td>CHECK LOB</td>
</tr>
<tr>
<td></td>
<td>REPORT with RECOVERY parameter</td>
</tr>
<tr>
<td></td>
<td>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)</td>
</tr>
<tr>
<td></td>
<td>SYNC (BMC Execution Monitor)</td>
</tr>
</tbody>
</table>

Table 29 on page 226 describes the utilities by their command names and provides the names that are generated for the utilities in the worklists.

### Table 29: Command descriptions

<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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You must have installed COPY PLUS to use this function. See the <em>BMC New Generation Technology Copy for DB2 for z/OS Reference Manual</em>.</td>
</tr>
<tr>
<td>BMCCOPYI</td>
<td>BMCC</td>
<td>Runs the BMC Software COPY PLUS product to produce copies of image copies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You must have installed COPY PLUS to use this function. See the <em>BMC New Generation Technology Copy for DB2 for z/OS Reference Manual</em>.</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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You must have installed LOADPLUS to use this function. See the <em>LOADPLUS for DB2 Reference Manual LOADPLUS for DB2 Reference Manual</em>.</td>
</tr>
<tr>
<td>Command</td>
<td>Worklist form</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BMCMOD</td>
<td>BMCC</td>
<td>Runs the BMC Software C+/MODIFY product to perform maintenance on SYSCOPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You must have installed C+/MODIFY to use this function. See the <em>BMC New Generation Technology Copy for DB2 for z/OS Reference Manual</em>.</td>
</tr>
<tr>
<td>BMCREORG</td>
<td>BMCR</td>
<td>Runs the BMC Software REORG PLUS product to reorganize DB2 physical objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You must have installed REORG PLUS to use this function. See the <em>REORG PLUS for DB2 Reference Manual</em>.</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 CHECK PLUS product</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You must have installed CHECK PLUS to use this function. See the <em>UNLOAD PLUS for DB2 Reference Manual</em>.</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>
<tr>
<td>LOAD</td>
<td>LOAD</td>
<td>Runs the DB2 LOAD utility</td>
</tr>
<tr>
<td>MERGCOPY</td>
<td>MERG</td>
<td>Runs the DB2 MERGECOPY utility</td>
</tr>
<tr>
<td>MODICOPY</td>
<td>MODI</td>
<td>Runs the DB2 MODIFY utility</td>
</tr>
<tr>
<td>MODISTAT</td>
<td>MODI</td>
<td>Runs the DB2 MODIFY STATISTICS utility</td>
</tr>
<tr>
<td>QUIESCE</td>
<td>QUI</td>
<td>Runs the DB2 QUIESCE utility</td>
</tr>
<tr>
<td>RECOVER</td>
<td>RECV</td>
<td>Runs the DB2 RECOVER utility</td>
</tr>
<tr>
<td>REORG</td>
<td>REOR</td>
<td>Runs the DB2 REORG utility</td>
</tr>
<tr>
<td>REP REC</td>
<td>REPO</td>
<td>Runs the DB2 REPORT utility with the RECOVERY keyword</td>
</tr>
<tr>
<td>REP SET</td>
<td>REPO</td>
<td>Runs the DB2 REPORT utility with the TABLESPACESET keyword</td>
</tr>
<tr>
<td>REPAIR</td>
<td>REPX</td>
<td>Runs the DB2 REPAIR utility</td>
</tr>
<tr>
<td>RESIZE</td>
<td>SQL</td>
<td>Generates the ALTER commands to resize STOGROUP defined objects</td>
</tr>
<tr>
<td>RUNSTATS</td>
<td>RNST</td>
<td>Runs the DB2 RUNSTATS utility</td>
</tr>
<tr>
<td>Command</td>
<td>Worklist form</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>START DB</td>
<td>CMD</td>
<td>Runs the DB2 START DATABASE command</td>
</tr>
<tr>
<td>STOP</td>
<td>STOP</td>
<td>Runs the Stop worklist command</td>
</tr>
<tr>
<td>STOP DB</td>
<td>CMD</td>
<td>Runs the DB2 STOP DATABASE command</td>
</tr>
<tr>
<td>STOPSPACE</td>
<td>STOS</td>
<td>Runs the DB2 STOSPACE utility</td>
</tr>
<tr>
<td>SYNC</td>
<td>SYNC</td>
<td>Runs the Sync Point command</td>
</tr>
</tbody>
</table>

**User-defined programs or services**

DASD MANAGER PLUS supports user-defined programs (also called *user-defined Services*) as well as your other Services (previously called *job steps*) as part of a Service Action.

When you specify the JCL elements for generating the Service as shown in Table 30 on page 228, 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 information</td>
<td>■ Program name</td>
</tr>
<tr>
<td></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 information</td>
<td>■ ddnames including symbolic variables (such as sequence number)</td>
</tr>
<tr>
<td>(including STEPLIBS)</td>
<td>■ DD specifications including symbolic variables</td>
</tr>
<tr>
<td></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 on page 228 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 variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALID</td>
<td>Same as WKID</td>
</tr>
<tr>
<td>Symbolic variable</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>CR</td>
<td>Creator</td>
</tr>
<tr>
<td>CREATOR</td>
<td></td>
</tr>
<tr>
<td>DA</td>
<td>Date, same as DD</td>
</tr>
<tr>
<td>DATE</td>
<td>Date in YYMMD format</td>
</tr>
<tr>
<td>DATEJ</td>
<td>Date in YYYYYDDDD format</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>DD</td>
<td>Day of month</td>
</tr>
<tr>
<td>DDD</td>
<td>Julian day</td>
</tr>
<tr>
<td>DDSEQ</td>
<td></td>
</tr>
<tr>
<td>DSQ</td>
<td>DD sequence number</td>
</tr>
<tr>
<td>DOPTS</td>
<td>Default options module, for full automation mode only</td>
</tr>
<tr>
<td>DSNUM</td>
<td>Same as PART</td>
</tr>
<tr>
<td>DT</td>
<td>Date in YYMMD format</td>
</tr>
<tr>
<td>HH</td>
<td>Hour</td>
</tr>
<tr>
<td>HMS</td>
<td>Same as TIME</td>
</tr>
<tr>
<td>HO</td>
<td>Hour</td>
</tr>
<tr>
<td>IX</td>
<td>Index</td>
</tr>
<tr>
<td>IXCR</td>
<td>Index creator</td>
</tr>
<tr>
<td>IXNAME</td>
<td>Index name</td>
</tr>
<tr>
<td>JD</td>
<td>Date in YYDDD format</td>
</tr>
<tr>
<td>JDATE</td>
<td>Date in YYYYYDDDD format</td>
</tr>
<tr>
<td>JDAY</td>
<td>Julian day</td>
</tr>
<tr>
<td>JOBTYP</td>
<td>Job type, for online generation only</td>
</tr>
<tr>
<td>JQID</td>
<td>Same as WKID</td>
</tr>
<tr>
<td>JUL4Y</td>
<td>Date in YYYYYDDDD format</td>
</tr>
<tr>
<td>JULIAN</td>
<td>Date in YYDDDDD format</td>
</tr>
<tr>
<td>MEMBER</td>
<td>For online generation only</td>
</tr>
<tr>
<td>MI</td>
<td>Minutes</td>
</tr>
<tr>
<td>MM</td>
<td>month</td>
</tr>
<tr>
<td>MO</td>
<td></td>
</tr>
<tr>
<td>OBJKW</td>
<td>Object keyword</td>
</tr>
<tr>
<td>Symbolic variable</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>OBJNAME</td>
<td>Object name</td>
</tr>
<tr>
<td>OBJT</td>
<td>Object type</td>
</tr>
<tr>
<td>OBJTYPE</td>
<td>Object type</td>
</tr>
<tr>
<td>OBNAM</td>
<td>Object name</td>
</tr>
<tr>
<td>PART</td>
<td>Partition ID</td>
</tr>
<tr>
<td>PART2</td>
<td>Partition ID, at least 2-digits</td>
</tr>
<tr>
<td>PART3</td>
<td>Partition ID, at least 3-digits</td>
</tr>
<tr>
<td>PART4</td>
<td>Partition ID, is 4-digits</td>
</tr>
<tr>
<td>PGMR</td>
<td>For online generation only</td>
</tr>
<tr>
<td>PREFIX</td>
<td>Same as USERID, for online generation only</td>
</tr>
<tr>
<td>RUNTYP</td>
<td>Run type, for online generation only</td>
</tr>
<tr>
<td>SC</td>
<td>Seconds</td>
</tr>
<tr>
<td>SS</td>
<td></td>
</tr>
<tr>
<td>SPNAME</td>
<td>Either table space name or index space name</td>
</tr>
<tr>
<td>SSID</td>
<td>Subsystem ID</td>
</tr>
<tr>
<td>TIME</td>
<td>Time in HHMMSS format</td>
</tr>
<tr>
<td>TS</td>
<td>Table space</td>
</tr>
<tr>
<td>TSNAME</td>
<td>Table space name</td>
</tr>
<tr>
<td>UID</td>
<td>Same as USER, for online generation only</td>
</tr>
<tr>
<td>USER</td>
<td>For online generation only</td>
</tr>
<tr>
<td>USERID</td>
<td>For online generation only</td>
</tr>
<tr>
<td>UTILID</td>
<td>Utility ID, for full automation mode only</td>
</tr>
<tr>
<td>UTID</td>
<td></td>
</tr>
<tr>
<td>VCAT</td>
<td>VCAT name</td>
</tr>
<tr>
<td>WKID</td>
<td>Action name</td>
</tr>
<tr>
<td>WKID8</td>
<td>First 8 characters of WORKID or action</td>
</tr>
<tr>
<td>WORKID</td>
<td>Same as WKID</td>
</tr>
<tr>
<td>YEAR</td>
<td>Year in YYYY format</td>
</tr>
<tr>
<td>YY</td>
<td>Year in YY format</td>
</tr>
<tr>
<td>YYYY</td>
<td>Year in YYYY format</td>
</tr>
<tr>
<td>ZPREFIX</td>
<td>Same as PREFIX, for online generation only</td>
</tr>
<tr>
<td>ZUSER</td>
<td>Same as USERID, for online generation only</td>
</tr>
</tbody>
</table>
For an example of how to create a user-defined Service, see “Example of creating a User Defined service” on page 260.

Service Action Generation function

The Service Action Generation function helps you build routine utility jobs that you can reuse.

The production environment requires the execution of maintenance utility job streams. The following figure displays the relationship between an action and the other two major components, the worklist and worklist JCL.

Figure 42: Service Action, worklist, and worklist JCL
The following figure illustrates the process of generating an initial Service Action.

**Figure 43: Generating a Service Action for the first time**
The following figure illustrates the process of regenerating an Service Action.

**Figure 44: Regenerating a Service Action**

---

**Utility worklist**

A utility worklist consists of the 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.
The following figure illustrates a simple generated worklist.

**Figure 45: Simple generated worklist**

```
-TIME 000000 'yyyy-05-29-22.40.19.362484' 41312037
*                      TESTCASE BSP00028               26961273
-SSID 000001 DEAE                                      63931300
-WKID 000002 BSP00028.UTILITY                          06240917
-SYNC 000003                                           74205673
-BMCU 000004 ASUSMAIN
BMCSSTATS TABLESPACE QZUD4%.% EVENTS N KEYCARD Y NUMCOLS 8 COUNT 6 INDEX N 06678180
```

The following table describes the worklist commands.

**Table 32: 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>
<tr>
<td>Note:</td>
<td>DASD MANAGER PLUS bypasses generating a SYNC command in the worklist if the previous command in the worklist was also a SYNC command.</td>
</tr>
<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
For a comprehensive list of Worklist commands that DASD MANAGER PLUS uses, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

**Wildcard expansion**

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. An example of a TABLESPACE object name that uses a wildcard pattern is QZUD4%.%

**Worklist example**

The following figure displays a generated worklist for the COPY utility. The product reserves columns 73 to 80 for hash verification numbers.

**Figure 46: Example of a generated worklist for the COPY utility**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>yyyy-05-31-12.40.57.203157'</td>
</tr>
<tr>
<td>*</td>
<td>TESTCASE UBCAC005</td>
</tr>
<tr>
<td>-SSID</td>
<td>000001 DEAE</td>
</tr>
<tr>
<td>-WKID</td>
<td>000002 UBCAC005.UTILITY</td>
</tr>
<tr>
<td>-SYNC</td>
<td>000003</td>
</tr>
<tr>
<td>-BMCC</td>
<td>000004 COPY TABLESPACE QZUD11.QZUS0111 GROUP NO SHRLEVEL REFERENCE FULL YES INDEXES NO COPYDDN (C0001)</td>
</tr>
<tr>
<td>-JCLP</td>
<td>000004 BMCC DDNAME C0001 DSNPREF &amp;PREFIX..&amp;SSID..&amp;DBNAME..&amp;TSNAME</td>
</tr>
<tr>
<td></td>
<td>48170458</td>
</tr>
<tr>
<td></td>
<td>12823997</td>
</tr>
</tbody>
</table>

**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. The table below describes the column format of the worklist.

---

**WARNING**

Modifying the hash number might terminate worklist execution if you enable hash checking (HASHFAIL= Y).
## Table 33: Column format of worklists

<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 |
| 14–72  | Command text  
This text is free-format text that includes information for executing the command. For continuation lines, the text appears in columns 2 through 72.  
There is no implicit break or space between column 72 of one line and column 2 of the following continuation line. However, you cannot split keywords over multiple lines. |
| 73–80  | The hash verification number, a numerical hash value of the contents of this command  
Several components of DASD MANAGER PLUS use this number to detect changed and inserted commands. Do not insert or modify this code. The hash number appears only on the last line of a command.  
To override failures and warnings, use the HASHFAIL and HASHWARN parameters. |

### 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 (JCLGEN) function with the necessary information to generate JCL for execution.

The worklist and JCLGEN relieve you of all JCL generation tasks. For the standard, non-worklist JCL generation that BMCTRIG provides, see “Analyzing objects by using BMCTRIG” on page 425.

The following figure provides a high-level overview of the JCL generation process.

Figure 47: JCL Generation

If you specify Edit JCL when you generate the action, the product displays the JCL (Figure 48 on page 237).

Figure 48: Generated JCL for an action

```plaintext
//RDAJXN2U JOB (5213), 'UTILITY-JENTEST',
//CLASS=A,MSGCLASS=X
/*
/*JOBPARM SYSAFF=DB2A
/*
/****************************************************
//* CREATED BY : RDAJXN2
//* TIMESTAMP : 01/30/ yyyy.15.24.31
//* ENVIRONMENT: ISPF 5.2MVS TSO
//* RELEASE : Vvv.rr.mm 01/30/yyyy
//* DB2 VERSION: vv
/****************************************************
/****************************-********************************
/* DASD MANAGER WORKLIST EXECUTION
```
//STEP1 EXEC PGM=AEXEMAIN,REGION=OM,
// PARM='DS815ECA'
//STEP1B DD DSN=TIS.IVPDEC.DECA2.RNTM.BMCLINK,
// DISP=SHR
// DD DSN='SYS3.DECA.DSNEXIT',
// DISP=SHR
// DD DSN='CSGI.DB2V81M.DSNLOAD',
// DISP=SHR
// ABNLIGNR DD DUMMY
// DSSPRINT DD SYSOUT=* 
// SYSDUMP DD SYSOUT=* 
// SYSTERM 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$OPTS
REORGOPT ARU$OPTS
CHECKOPT ACK$OPTS
RECOVEROPT AFR$OPTS
//SYSIN DD SPACE=(CYL,(15,15)),UNIT=SYSDA,DISP=(NEW,PASS),
// DSORG=PS,LRECL=80,BKSIZE=3200,RECFM=FB
//SYSPRINT DD SPACE=(CYL,(15,15)),UNIT=SYSDA,DISP=(NEW,PASS)
/*JGENSRPT DD SYSOUT=*
/* 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
/* DATA SET SIZE REQUIRED FOR DD C0002 IS AN ESTIMATE
C0002 DD DSN=RDAJXN.CPY.DEAE.JEND30.JENS0130.C0002,
// DISP=(NEW,CATLG,CATLG),
// SPACE=(CYL,(1,1),RLSE), ESTIMATE-C/H
// UNIT=SYSDA
/* END OF JOBSTEP

238  DASD MANAGER PLUS for DB2 User Guide
Working with Service Actions

This section describes common tasks that you perform with Service Actions. The following tasks let you set up, maintain, and generate service actions:

- “Listing Service Actions” on page 239
- “Creating a Service Action” on page 241
- “Editing Service Action properties” on page 244
- “Copying a Service Action” on page 246
- “Deleting Service Actions” on page 248
- “Generating Service Actions” on page 248

On most DASD MANAGER PLUS artifact menus (i.e. Service Actions, Services, Object Sets, and so on) you can enter a name or a wildcard pattern in the field for the artifact name. Use these guidelines:

<table>
<thead>
<tr>
<th>If you want to</th>
<th>Type this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display a list of artifacts with similar names</td>
<td>Wildcard pattern Example: BMC%</td>
</tr>
<tr>
<td>Display a list of all artifacts</td>
<td>%</td>
</tr>
<tr>
<td>Create an artifact</td>
<td>New unique artifact name</td>
</tr>
<tr>
<td>Copy, delete, edit, or rename an artifact</td>
<td>Name of the artifact without wildcard characters)</td>
</tr>
</tbody>
</table>

You can select the desired artifact from the dialogs or from the Main Menu. If you do not know the name of the artifact, type a wildcard to display a list.

Listing Service Actions

Use this procedure to view a list of Service Actions.
To list Service Actions

1. On the DASD MANAGER PLUS Main Menu, select **Service Actions** and press **Enter**.

The Action Menu panel appears.

```
DEAE ------------------------------- Action Menu -------------------------
Command ===> 

An Action is an order set of Services used for generating utilities. Type a specific Action or type a wildcard pattern for a selection list.

Action . . . %.DEM%
```

Select an option. Then press Enter.
1. List Actions
2. Create a new Action
3. Edit Services for an Action
4. Generate Job for an Action
5. Edit Action Properties
6. Copy an Action
7. Delete an Action
8. Report Action Services
9. Report Trigger Action Object Exceptions
10. Report Trigger Action Default Corrective Action

For guidelines on entering a name or wildcard pattern, see “Working with Service Actions” on page 239.

2. On the Action Menu panel, do the following:

   a. In the **Action** field, type the name of a Service Action or a wildcard pattern to generate a list of Service Actions.

   b. Select **List Actions** and press **Enter**.

DASD MANAGER PLUS populates the Action List panel with the Service Actions that you specified.

```
DEAE ----------------- Action List ----------------           Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE
Action: %.DEM%

An Action is an order set of Services used for generating utilities. Select one or more options, then press Enter

C =Create  D =Delete  E =Edit  L =Like  P =Properties  G =Generate
RS =Rpt Services   RX =Rpt Exceptions   RD =Rpt Dflt Corrective Action

Act Action     Owner    Gen Date   Gen By   POF Ovr   Status
************************************************************* Top of data  *************************************************************
DEMO01         MVSPXS    yyyy-08-21 RDAPKM    Y        Gen
DEMO02         MVSPXS    yyyy-08-21 RDAPKM    Y        Gen
************************************************************* Bottom of data  *************************************************************
```

The following table describes the panel's fields.
Table 34: Action List panel fields

<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>Service Action name</td>
</tr>
<tr>
<td>Owner</td>
<td>Authorization ID of the Action owner</td>
</tr>
<tr>
<td>Gen Date</td>
<td>Last time that the Service Action was generated&lt;sup&gt;a&lt;/sup&gt; Date of the last action generation. A blank value means the action has not been generated.</td>
</tr>
<tr>
<td>Gen By</td>
<td>Last logon ID to generate the action&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
</tbody>
</table>
| POF Ovr | Indicates whether the Service Action is using a product options file (POF) other than the initial POF specified in the default options (DOPTs) module.  
  **Note:** The name of the overriding POF appears in the properties dialog or the generate dialog. The POF supplies the parameters for JCL generation. |
| Status  | Current status of the Service Action status from the DO_WORKIDS table.<sup>a</sup>. Possible values are as follows:  
  ■ blank—The product has not generated the Service Action job.  
  ■ Gen—The product generated the Service Action job but did not run it.  
  ■ Err—The Service Action job started but had an error.  
  ■ Exec—The service action job completed successfully. |
| Description | The description provided when the user created the action                  |

<sup>a</sup> This field acquires a value after the job is generated. A blank value means the action has not been generated.

Creating a Service Action

Use this procedure to create a Service Action and its associated properties.

Action properties contain the following elements:

- Description
- Status
- Maintenance ID
- Date
- Worklist
- JCL
- POF data set names (after the Action has been generated)
To create a Service Action

This procedure does the following:
- Creates a Service Action
- Associates to the Service Action the objects on which the Service Action executes
- Associates to the Service Action the Services that the Service Action executes

1. On the DASD MANAGER PLUS Main Menu, select Service Actions and press Enter.
   
   For guidelines on entering a name or wildcard pattern, see “Working with Service Actions” on page 239.
   
   The Action Menu panel is displayed. (For an example of the panel, see “Listing Service Actions” on page 239.)

2. In the Action field, type the name of a Service Action in the format owner.name, and press Enter.

   Note
   The Service Action name cannot contain an asterisk (*), percent sign (%), or embedded blank. The name part of the Service Action name must be unique. The default owner name is your TSO logon ID.

   DASD MANAGER PLUS validates that the Service Action does not already exist.

3. From the Action Menu, select Create a new Action and press Enter.

   As an alternate method, you can open the Service Action List panel from the Service Action panel by selecting List and specifying a wildcard pattern for the service action name, then, type C in an Act field and press Enter.

   The Create Action panel is displayed.

   DEAE --------------------------- Create Action -----------------------------
   Command ===> 
   Type new Action data. Then press Enter.
   Action . . . . PAYROLL.DEMO01
   Description . . . Service Action for payroll db reorg

4. (optional) Type the following information and press Enter:
   - In the Action field, modify the action owner and name.
   - In the Action Description field, enter a description to help identify the action.
The Action Object Specification panel is displayed. The Action field displays the name of the new Service Action. A message at the top of the panel displays the result of the create operation.

```
DEAE ----------------------- Action Object Specification --- Create successful
Command ===> PAYROLL.DEMO01

Please specify object information for the Service to use. Object information may consist of an Object Set, a fully qualified object name, or a wildcard pattern. Then press Enter.

Object Set
Name . . PAYROLL.DEMO01 (%= Object Set Select List)
Object Name or Pattern
Name . .
Type . . (IS,IX,SG,TS,TT,VL)
Part . . (1-4096 and not valid for wildcarded name)
```

5. Type the following information, and press Enter.

- (optional) In the Object Set Name field, type an Object Set Name. As an alternative method, type a Object Set name using wildcard characters, and press Enter to display an Object Set Selection List.

  If you are not using Object Sets, leave this field blank.

- In the Object Name or Pattern field, type the following information:
  — Object Name (you can use wildcard characters)
  — Object Type
  — Partition number (if you are executing services on specific partitions)

  To specify a partition number, you must have entered a fully qualified Object Name with no wildcard characters.

The Service Selection List panel is displayed. Use the Service Selection List to build and order your services (utilities).

```
DEAE --------------- Service Selection List ------------- Row 1 to 16 of 136
Command ===> Scroll ===> CSR
Action: PAYROLL.DEMO01

Select one or more Services for the Action to execute. (C)opy Service from the Available Services by indicating position in Selected Services using (A)fter/(B)efore. Then press Enter.

Selected Actions : L =Like  A =After  B =Before  M =Move  D =Delete
Available Actions: S =Select  C =Copy

More: +
Act Service Vendor Description
************************ Top of data ************************
SELECTED SERVICES
  1 <NONE> Add Services here
  2 BMCREORC BMC BMCREORG PLUS
AVAILABLE SERVICES
BMCCOPY BMC COPY PLUS
BMCCPYI BMC COPY PLUS COPY IMAGECOPY
BMCCPRES BMC DASD MGR COPY CATALOG STATS
BMCLoad BMC LOAD PLUS
BMCMOD BMC COPY PLUS MODIFY
```

Chapter 4  Maintaining and generating Service Actions  243
From the list of AVAILABLE SERVICES, select and add services to the SELECTED SERVICES list, specifying the order of execution, and press Enter.

The SELECTED SERVICES are used for the list of services on the Action Services panel. After you add services to the SELECTED SERVICES list, you can reorder or remove them.

The Edit Action Services panel is displayed.

Follow the steps in “Editing Services” on page 265.

Note
For information about Action Services, see “Working with Services” on page 252.

---

**Editing Service Action properties**

*(optional)* Use this procedure to modify an existing Service Action properties. You do not have to enter Service Action properties. Service Action properties do not affect how DASD MANAGER PLUS performs the Service Action. The properties contain the JCL Gen POF, which provides default values for the JCL options.

1. On the DASD MANAGER PLUS Main Menu, select **Service Actions** and press Enter.

   The Action Menu panel is displayed.

2. On the Action Menu panel, in the **Action** field, type the name of the Service Action to edit, select **Edit Action Properties**, and press Enter.
As an alternate method, open the Action List panel from the Action Menu panel by selecting List and specifying a wildcard pattern for the action name. (For an example of the Action List panel, see “Listing Service Actions” on page 239.) Then type P in the Act field next to an action and press Enter.

The Edit Action Properties panel is displayed.

- The Action field displays the fully qualified name of the selected Service Action, as well as maintenance information (such as who last updated the Action and when it was last updated). When the Worklist is generated, the product automatically updates the additional dataset and Worklist generated information, and status.

- The Status field displays 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>NOT GENERATED</td>
<td>The product generated the utility job but did not run it.</td>
</tr>
<tr>
<td>EXECUTED</td>
<td>The utility job completed successfully.</td>
</tr>
<tr>
<td>ERROR</td>
<td>The utility job started but had an error.</td>
</tr>
</tbody>
</table>

3 (optional) Enter the following information on the Edit Action Properties panel:

a. In the Description field, enter a description of up to 100 characters.

b. In the Datasets JCL Options field, enter a data set name for the product options file (POF).

To override the default installation POF dataset, provide a different POF data set. Providing a different POF data set sets the value in the POF Ovr (override) column on the Action list panel to Y.
If you do not specify a POF dataset, the product uses the dataset that the installation options module (DOPTS) specifies in the POFDS option.

*Note*

The remaining *Data Sets* fields are read-only. The product updates the *Worklist Generated* fields after you have generated the worklist.

After you have generated the worklist, the product sets the following read-only Data Set information:

- JCL—the dataset where the product saved the generated JCL
- Worklist—the dataset where the product saved the generated worklist
- Diagnostics—the dataset or SYSOUT where the product placed diagnostic information

The product sets the following read-only Worklist Generated information after you have generated the worklist:

- On—Date when you last generated the worklist
- By—AUTHID that generated the worklist.
- RC—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* refers to the data sets where the product placed the generated worklist and JCL. You specify the data sets on the Action Job Generation panel. For more information, see “Creating a Service Action” on page 241.

4 Press END to save the Service Action properties.

**Copying a Service Action**

Use this procedure to create a new Service Action that is based on an existing Service Action.

**To copy a Service Action**

1 On the DASD MANAGER PLUS Main Menu, select *Service Actions* and press Enter.
The Action Menu panel is displayed.

2 On the Action Menu panel, do the following:

a In the **Action** field, type the name of the Service Action (without wildcards) that you want to copy.

b Select **Copy an Action**.

c Press **Enter**.

---

*Note*

You can also use this alternate method.

1 On the Action Menu panel, specify a wildcard for the action name and select **List**.

2 On the List Action panel, type **L** in the **Act** field next to the action to want to copy, and press **Enter**.

The Copy Action panel is displayed.

**Figure 49: Copy Action panel**

```
DEAE ------------------------------- Copy Action ------------------------------
Command ===> Type new Action data. Then press Enter.
Action ...... MVSXL1.MVSXL1
Data Sets
  JCL ...... 'ASU.DEDK.V102QA.JCLLIB(&WKID)'
  Worklist .. 'ASU.DEDK.V102QA.WORKLIST(&WKID)'
Description ..
```

3 Enter a unique Service Action name in the **Action** field.

The Service Action name cannot contain a percent sign (%), asterisk (*), or embedded blank characters. If you plan to use &WKID or &WKID8 during JCL generation, the Service Action must be a valid PDS member name or data set node.

Optionally, you can enter a description. (For more information, see “Access to actions, services, and service syntax” on page 278.)

4 Press **Enter**.

The Action Menu panel is displayed. If you select List Actions, the Action List panel now shows the Service Action that you copied and the new Service Action that you created.
Deleting Service Actions

Use this procedure to delete a Service Action when you no longer need it.

To delete a Service Action

1. On the DASD MANAGER PLUS Main Menu, select Service Actions and press Enter.

2. On the Action Menu panel, do the following:
   a. In the Action field, specify the name of the Service Action that you want to delete, without using wildcard characters.
   b. Select Delete an Action.
   c. Press Enter.

   **Note**
   Alternately, you can access the List Actions panel from the Action menu to delete actions. On the Actions menu, enter a wild card character in the Action name field and press Enter. On the List Action panel, type D next to the action that you want to delete and press Enter.

The Delete Action panel is displayed.

```
DEAE ------------------------------ Delete Action -----------------------------
Command ==>
Action . . . . . : MVSNXL1.MVSNXL1
Updated by . . . . : MVSNXL1
Status . . . . . : NOT GENERATED
Last Updated . . : 2015-02-16
Description . . . :

Select delete option. Then press Enter.
Delete Options . . 2 1. Delete Action
                   2. Cancel Delete and Exit
```

3. Type 1 to confirm, and press Enter to delete the action.

Generating Service Actions

Use this procedure to generate or submit a Service Action.

To generate a Service Action, you specify data sets for the job and set 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.
To generate a Service Action

1. From the Action Menu panel, select **Generate Job for an Action**.

   The Action List panel is displayed.

2. In the **Act** field, type **G** next to a Service Action and press **Enter**.

   The *first* Action Job Generation panel is displayed.

   ![Action Job Generation Panel](image)

   **Action** field displays the Service Action that you are generating. The **Status** field shows the current status of the action. The DO_WORKIDS table provides the status. The field can contain one of the following values:

   - Not Generated—the action has not been generated
   - Generated Not Executed—the action has been generated but not submitted
   - Executed Successfully—the action has been generated, submitted, and ran successfully
   - Executed with Errors—the action has been generated, submitted, and completed unsuccessfully

3. The **JCL Type** field indicates what type of JCL to generate, depending on the Service 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. (You can select this option if the Action Status is **Executed with Errors**.) DASD MANAGER PLUS creates JCL from the old job. If you select this JCL type, you can use RESTART PARM to insert parameters into job stream.

To regenerate the JCL, on the next panel you must select **Build JCL**.

■ Type 3 to restart the execution job from the beginning. DASD MANAGER PLUS creates JCL from existing JCL.

4 In the **JCL Build Options** fields, perform the following steps:

a 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 into the EVENTS table.

This option is not valid if the JCL Type is **Build Restart JCL**.

b In the **Restart Parm** field, specify information to pass to a restart job. The **Restart Parm** field is available only if the JCL Type is **Build Restart JCL**.

c Press **Enter**.

The **second Action Job Generation panel** is displayed.

The **Action** field displays the Service Action that you are generating. The **Status** field shows the current status of the action. The DO_WORKIDS table provides the status. The field can contain one of the following:

■ Not Generated—the action has not been generated

■ Generated Not Executed—the action has been generated but not submitted
- Executed Successfully—the action has been generated, submitted, and ran successfully

- Executed with Errors—the action has been generated, submitted, and completed unsuccessfully

The **JCL POF** field displays the name of the Service 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 DASD MANAGER PLUS User Options panel. For more information, see “Setting up DASD MANAGER PLUS” on page 63.

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, &WKID uses only the first 8 characters of the action name.

5 Set the options as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>What to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JCL</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Worklist</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Diagnostics</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Override POF Values</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Build Worklist</strong></td>
<td>Type S to generate (or regenerate) the worklist.</td>
</tr>
<tr>
<td><strong>Edit Worklist</strong></td>
<td>Type S to display the worklist for viewing or editing after it is generated.</td>
</tr>
<tr>
<td><strong>Build JCL</strong></td>
<td>Type S to generate (or regenerate) the JCL.</td>
</tr>
<tr>
<td><strong>Edit JCL</strong></td>
<td>Type S to display the JCL for editing after it is generated.</td>
</tr>
<tr>
<td><strong>Submit JCL</strong></td>
<td>Type S to submit the JCL to execute the commands in the worklist. You can submit the job now or later.</td>
</tr>
</tbody>
</table>

6 Review your entries, and press **Enter** to process the job.
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 JCL** option, the product generates the worklist and JCL, and displays the Action Job Generation panel. The message **<ENTER> TO SUBMIT** appears at the top-right portion of the panel.

Press **Enter** to submit the job for processing, or press **END** to exit the Action Job Generation panel without submitting the job.

### Working with Services

This section describes common tasks that you perform with Services.

There are two types of Services: vendor supplied and user supplied. Vendor supplied are referred to as BMC Software required services. These Services cannot be created or deleted.

DASD MANAGER PLUS provides default syntaxes for each required service. Default supplied syntaxes cannot be edited, but they can be copied to a new name and then edited.

The following tasks let you set up, maintain, and generate services:

- “Listing Services” on page 252
- “Editing Service Syntax” on page 254
- “Editing Service properties” on page 255
- “Example of creating a User Defined service” on page 260
- “Deleting a User Defined Service” on page 264
- “Editing Services” on page 265
- “Previewing generated syntax” on page 272

### Listing Services

Use this procedure to view a list of Services.
To list Service Actions

1. On the DASD MANAGER PLUS Main Menu, select Services and press Enter.

The Services Menu panel appears.

```
DEAE ----------------------------- Services Menu -----------------------------
Command ===>                      
A Service is a program or utility that runs as part of an Action. Type a
specific Service or type a wildcard pattern for a selection list.
Service . . .
```

Select an option. Then press Enter.
1. List
2. Create a new User Defined Service
3. Edit a Service Syntax Options
4. Edit a Service Properties
5. Delete a User Defined Service
6. List Action Impact

2. On the Services Menu panel, do the following:

   a. In the Service field, type the name of a Service or a wildcard pattern to
generate a list of Services.

      For example, **BMC%**

   b. Select List Actions and press Enter.

The Service List panel is displayed.

```
DEAE ----------------- Service List -----------------       Row 1 to 10 of 10
Command ===>                                                  Scroll ===> CSR
Service: BMC%
A Service is a program or utility that runs as part of an Action. Select
one or more options. Then press Enter.
C =Create   D =Delete   E =Edit   P =Properties   A =Action Impact
More:       >
Act Service Vendor  Program   APF  Att  Description
******************************** Top of data *************************************************************
BMCCOPY   BMC     ACPMAIN    Y    8   BMC COPY PLUS
BMCCOPYI  BMC     ACPMAIN    Y    8   BMC COPY PLUS COPY IMAGECOPY
BMCCPRES  BMC     ASUSCPRES Y    8   BMC DASD MGR COPY CATALOG STATS
BMCLOAD   BMC     AMUUMAIN   Y    7   BMC LOAD PLUS
BMCMOD    BMC     ACPMAIN    Y    8   BMC COPY PLUS MODIFY
BMCLRDGR  BMC     ARUUMAIN   Y    7   BMC REDGR PLUS
BMCMSTATS BMC     ASUSMAIN   Y    8   BMC DASD MGR BMCMSTATS
BMCTRIG   BMC     ASUSTRIG  Y    8   DASD MGR DETECT EXCEPTIONS AND TRIGGE
BMCLUD    BMC     ADUUMAIN   Y    7   BMC UNLOAD PLUS
BMCPUPRS  BMC     ASUSUPRS  Y    8   BMC DASD MGR UPDATE CATALOG STATS
******************************** Bottom of data **************************************************************
```

The following table describes the panel's fields.
Table 35: Service List panel fields

<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>Service Action name</td>
</tr>
<tr>
<td>Vendor</td>
<td>One of the following vendor names:</td>
</tr>
<tr>
<td></td>
<td>■ BMC—BMC Software Service</td>
</tr>
<tr>
<td></td>
<td>■ IBM—IBM Software Service</td>
</tr>
<tr>
<td></td>
<td>■ USR—User Supplied Service</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the service</td>
</tr>
</tbody>
</table>

Editing Service Syntax

When you edit a Service, you list all of the Services syntaxes. The syntaxes contain the Service options that the product invokes when you generate Services. Perform these steps to edit a service.

A Service can have many syntaxes but only one default syntax. The default syntax is automatically assigned to the Service when the Service is added to the Service Action. You may assign another syntax containing options different than the default, thereby overriding the assigned default.

To edit a Service

1. On the DASD MANAGER PLUS Main Menu, select Services and press Enter.

   The Services Menu panel is displayed.

2. On the Services Menu panel, In the Service field, type a Service name, select Edit a Service Syntax Options, and press Enter.

   **Note**

   As an alternate method, you can access the Service Syntax List panel from the Service Menu panel to edit Service syntax. On the Service Menu panel, enter a wildcard character in the Service field and press Enter. On the List Service panel, type E next to a service and press Enter.
The Service Syntax List panel is displayed.

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

### Service Syntax List

- **Service**: BMCCOPY
- **Default Syntax**: BMC_DEFAULT

Select an option or change Display filter. Then press Enter. To change the default syntax, select another syntax with the 'F' option.

- **B**: Browse
- **C**: Create
- **D**: Delete
- **E**: Edit
- **L**: Like
- **P**: Properties
- **A**: Action Impact
- **V**: View Syntax
- **F**: Set default syntax

**Display filter**

Owner.Name . . . %.%

More:   -   >

<table>
<thead>
<tr>
<th>Act Dflt Name</th>
<th>Act</th>
<th>Owner</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y  BMC_DEFAULT</td>
<td></td>
<td>BMC</td>
<td>BMC supplied defau</td>
</tr>
<tr>
<td>1445049C</td>
<td></td>
<td>MVSCL2</td>
<td></td>
</tr>
<tr>
<td>PKM01</td>
<td></td>
<td>BDAPKM</td>
<td></td>
</tr>
<tr>
<td>TEST85-BMCCOPY</td>
<td></td>
<td>MVSEMUM3</td>
<td></td>
</tr>
</tbody>
</table>

******************************* Bottom of data ********************************

3. From the Service Syntax List panel, in the Act field, type the appropriate letter to select an option:
   - **B**—Browse syntax options
   - **C**—Create a syntax
   - **D**—Delete a syntax
   - **E**—Edit the syntax options
   - **L**—Like the syntax (use the syntax as a template to create a new syntax like the original)
   - **P**—Edit the syntax properties
   - **A**—View a list of Actions that use the syntax (simulate syntax generation)
   - **F**—Set the default syntax for a Service.

   When a default syntax is set, new Service Actions automatically use the default syntax. The default syntax is automatically set for a Service when the Service is selected.

   **Note**

   You can select a different default syntax from the Action Services panel.

4. *(optional)* In the **Owner.Name** field, edit the default value (%.%.) to filter the list of displayed syntaxes.

---

**Editing Service properties**

Use this procedure to edit the Service properties, including EXEC, DD and template syntax program details.
You can edit properties for user created Services only. You can browse BMC required Service properties, but you cannot edit them.

**To edit Service properties**

1. On the DASD MANAGER PLUS Main Menu, select **Services** and press **Enter**.

2. From the Services Menu panel, in the **Service** field, type the name of a Service, select **Edit a Service Properties**, and press **Enter**.

As an alternative method, you can access the List Services panel from the Services Menu panel to edit service properties.

1. On the Services Menu panel, enter a wild card character in the **Service** field and press **Enter**.

2. On the Service List panel, type **P** next to a service and press **Enter**.

The Edit Service Properties panel is displayed.

**Figure 50: Edit Service Properties panel**

```
DEAE ------------------------- Edit Service Properties ------------------------
Command ===> 
Type Service data. Then press End.
Service . . : IKJEFT01 Owner . . . . . : RDAPKM
Type . . . . : PROGRAM Vendor . . . . : USR
Program Name . IKJEFT01 (for EXEC PGM=) Updated by . . . : RDAPKM
APF . . . . : Y (Y/N APF authorized) Date last updated: 2016-03-04-11.09.55
Attach key . . 8 (1-8 Attach key)
Program Parm .
Description .
Select edit option. Then press Enter.
Program Details . . 1. DD statements
2. Syntax Template
```

3. On the Edit Service Properties panel, enter information into the following fields, and then press **Enter**:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>Documentation field, describes the type of service</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>PROGRAM, REPORT, COPY, LOAD, and so on</td>
</tr>
<tr>
<td><strong>Program Name</strong></td>
<td>Program name as it appears in the JCL in the PGM= keyword</td>
</tr>
<tr>
<td><strong>APF</strong></td>
<td>Indicates whether the program must run APF authorized</td>
</tr>
<tr>
<td><strong>Attach Key</strong></td>
<td>Indicates the key to use during the attach</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Program Parm</td>
<td>Parameter information as it appears in the JHCL in the PARM= keyword</td>
</tr>
<tr>
<td></td>
<td>The parameter list can contain symbolic variables.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the service</td>
</tr>
<tr>
<td>Program Details</td>
<td>Available edit options</td>
</tr>
<tr>
<td></td>
<td>For this task, select 1 (DD statements).</td>
</tr>
</tbody>
</table>

The DD Statement List panel is displayed.

```
DEAE ----------------------- DD Statement List --------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> CSR
Service: XL01COPY

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.

C =Create   D =Delete   E =Edit   L =Like   A =After  B =Before  M =Move

Act Name      Type      Dataset Name
******************************** Top of data **********************************
XLO1DD01      INFILE    RDAPKM.TEST.INFILE1
XLO1DD01      INFILE    RDAPKM.TEST.INFILE2
*************************************** Bottom of data ****************************
```

4 Enter DD statements as needed.

**Note**

For BMC Software required services, this list contains only additional DD statements that are not in the JCL product options file (POF) or syntax options. Do not add DD statements for BM 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 is allocated to a job entry subsystem (JES) input stream for standard JCL. The syntax DD is allocated to a temporary data set if you run it in worklist format under AEXEMAIN.

5 *(optional)* To edit DD statements, perform the following steps:

a Type **E** in the **Act** column next to the line item to be edited.
The Edit DD Statement panel is displayed.

DEAE ---------------------------- Edit DD Statement ---------------------------

Command ===> Type DD Statement data. Then press End.

Service . . : XL01COPY
DD name . . : XL01DD01
Type . . . : INFILE (INFILE, OUTFILE, REPORT, PGMLIB)
Data set name RDAPKM.TEST.INFILE1
Description .

DD Statement text (free-form)
//   DISP=SHR

Restart DD Statement text (free-form)
//   DISP=OLD

b On the Edit DD Statement panel, type information into the following fields:

<table>
<thead>
<tr>
<th>In this field</th>
<th>Enter this information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ddname</strong></td>
<td><em>ddname</em>, which can contain symbolic variables</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>For DDNAMES defined for the USERDEF Service, you must edit the JCL to add the procstep to the DDNAME as follows:</td>
</tr>
<tr>
<td></td>
<td>//processtep.ddnameDD</td>
</tr>
<tr>
<td><strong>Type</strong></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.</td>
</tr>
</tbody>
</table>
|                        | 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. |
|                        | - Use PGMLIB for program library name.                      |
|                        | For the automation component or AEXEMAIN, the library is TASKLIB. |
|                        | For Standard JCL, the library is STEPLIB.                    |
| **DSN**                | Data set name, which can contain symbolic variables         |
|                        | If the data set is a PDS, you can supply a member name. The member name can contain symbolic variables. |
| **Restart DD Statement text** | JCL control card names (which can contain symbolic variables) to include in the DD statements |
|                        | The control cards generate as comments for generation from DASD MANAGER PLUS. With the automation component, the control cards generate on a restart. |
|                        | Do not use DSN= in the text because DASD MANAGER PLUS generates it automatically. |

c Press **END** to save your changes and return to the Edit Service Properties panel (Figure 50 on page 256).
6 Edit the Syntax Template as follows:

a In the Program Details field, type 2 (Syntax Template). If your user-defined service has syntaxes already defined, the Syntax List panel is displayed. Skip to List item. on page 260.

If your user-defined service does not have syntaxes defined, the Create Service Syntax panel is displayed.

b Enter the following information, and then press Enter to enter syntax text:

<table>
<thead>
<tr>
<th>In this field</th>
<th>Enter this information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>Name of the selected service syntax</td>
</tr>
<tr>
<td>Owner</td>
<td>Authorization ID of the user who created the service syntax</td>
</tr>
<tr>
<td>DDName</td>
<td>JCL data definition to be allocated during execution of the service</td>
</tr>
<tr>
<td></td>
<td>The DDName must conform to standard DD naming conventions.</td>
</tr>
<tr>
<td>Description</td>
<td>Short explanation to help identify the service syntax</td>
</tr>
</tbody>
</table>

An edit session for the syntax text is displayed.

```
EDIT RDAPKM.T0291600.$04MAR16.$123259.$2854408 Syntax created
Command ===>

****** ************ Top of Data ************
****** DSN SYSTEM(&SSID)
****** DISPLAY DATABASE(&DBNAME) SPACE(&SPNAME) LIMIT(*
```

c Enter syntax in column 2 and then press END to save your changes.

---

**Note**

JCL Execution does not execute SQL commands in worklist format. BMCTRIG must generate SQL commands in standard JCL format.

The syntax for user-defined services can contain symbolic variables. For more information see the table of supported symbolic variables in “User-defined programs or services” on page 228.

---

If your user defined service has at least syntax already defined, the Syntax List panel is displayed. You can also access this panel by editing an entry on the Action Services List panel.
**Example of creating a User Defined service**

Following is an example of creating a user-defined service to be used with a worklist format.

In this example, you 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 a User-defined service**

1. On the DASD MANAGER PLUS Main Menu, select **Services** and press **Enter**.

   The Services Menu panel is displayed.

   ```
   DEAE  ------------------------------ Services Menu -----------------------------
   Command ===>
   A Service is a program or utility that runs as part of an Action. Type a specific Service or type a wildcard pattern for a selection list.
   Service . . . %
   Select an option. Then press Enter.
   1. List
   ```
2. Create a new User Defined Service
3. Edit a Service Syntax Options
4. Edit a Service Properties
5. Delete a User Defined Service
6. List Action Impact

2. From the Services Menu, select **Create a new User Defined Service** to display the Create User Defined Service panel.

The Create User Defined Service panel is displayed.

```
DEAE ----------------------- Create User Defined Service ----------------------
Command ===>
Type new Service data. Then press Enter.
Name . . . . 
Type . . . .  (i.e. PROGRAM, COPY, STAT, REPORT, etc...)
Program Name .  (Program for EXEC PGM=)
APF . . . .  N  (Y/N APF authorized)
Attach key . .  B  (1-8 Attach key)
Program Parm .
Description . 
Select edit option. Then press Enter.
Program Details . .  1. DD statements
2. Syntax Template
```

3. In the Create User Defined Service panel, enter the following information:

   a. In the **Name** field, type the Service name **IKJEFT01**.

   b. In the **Type** field, **PROGRAM**.

   c. In the **Program Name** field type the program name as it appears in the JCL in the **PGM=** keyword (in this example, **IKJEFT01**).

   d. In the **APF** field, type **Y** to authorize APF. (In this example, IKJEFT01 requires authorization.)

   e. In the **Select edit option** field, type **1** for DD statements, and press **Enter**.

*Figure 51: Edit Services Properties panel with information entered*
The Create DD Statement panel is displayed.

```
DEAE --------------------------- Create DD Statement ----------- Service added
Command ===> Create DD Statement data. Then press Enter.
Service . . : IKJEFT01
DD name . . :
Type . . . .  (INFILE, OUTFILE, REPORT, PGMLIB)
Data set name
Description .
DD Statement text (free-form)
Restart DD Statement text (free-form)
```

4 In the Create DD Statement panel, enter the following information:

a In the DD name field, type SYSTSPRT.

b In the Type field, type REPORT.

Figure 52: Create DD Statement with information entered

```
DEAE --------------------------- Create DD Statement ----------- Service added
Command ===> Create DD Statement data. Then press Enter.
Service . . : IKJEFT01
DD name . . : SYSTSPRT
Type . . . . REPORT  (INFILE, OUTFILE, REPORT, PGMLIB)
Data set name
Description .
DD Statement text (free-form)
Restart DD Statement text (free-form)
```

The DD name and Type fields tell the JCL Generation function where to send the output.

c Press END to save your changes and return to the DD Statement list panel, which now looks like this:

```
DEAE --------------------------- DD Statement List ---------------- Row 1 to 1 of 1
Command ===> DD Statement List
Service: IKJEFT01
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.
C =Create  D =Delete  E =Edit  L =Like  A =After  B =Before  M =Move
Act Name  Type  Dataset Name
*********** Top of data ***********
SYSTSPRT  REPORT
*********** Bottom of data ***********
```

Working with Services

262  DASD MANAGER PLUS for DB2 User Guide
For BMC Software required services, this list contains only DD statements that are not in the JCL Product Options File (POF) 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 is allocated to a job entry subsystem (JES) input stream for standard JCL. The syntax DD is allocated to a temporary data set if you run it in worklist format under AEXMAIN.

5 Press END to exit the DD Statement List panel and return to the Edit Services Properties panel.

6 In the Create User Defined Service panel, select 2 (Syntax Template) in the Select Edit Option field.

The Create Service Syntax panel is displayed.

7 In the Create Service Syntax panel, enter the following information:

a In the Syntax field, type DISPLAYDATABASE.

b In the DDName field, type SYSTSIN.

The panel should appear as shown in Figure 53 on page 263.

![Figure 53: Create Service Syntax](image)

The DEAE command is displayed.

- Press Enter to continue updating options and to include the syntax.

An editing panel is displayed.

8 Starting in column 2, enter the following syntax, and then press END to save the changes:

```
DSN SYSTEM(&SSID)
-DISPLAY DATABASE(&DBNAME) SPACE(&SPNAME) LIMIT(*)
```

---

**Note**

JCL Execution does not execute SQL commands in worklist format. BMCTRIG must generate SQL commands in standard JCL format.
The Service Syntax List panel appears.

```
DEAE --------- Service Syntax List ------- ---------         Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Service . . . : IKJEFT01
Default Syntax: DISPLAYDATABASE

Select an option or change Display filter. Then press Enter. To change
the default syntax, select another syntax with the 'F' option.

B =Browse      C =Create         D =Delete       E =Edit   L =Like
P =Properties  A =Action Impact  V =View Syntax  F =Set default syntax

Display filter
Owner.Name . . . %.%

More:   -   >

Act Dflt Name                                      Owner     Description
Y  DISPLAYDATABASE                           MVSNXL1

******************************* Bottom of data ********************************
```

9 Press END to return to Edit Services Properties.

10 Press END to complete the User Defined Exception and return to Services Main Menu.

11 Create an Action and associate the IKJEFT01 user defined service.

For more information about associating, see “Editing Services” on page 265.

**Deleting a User Defined Service**

Use this procedure to delete a User Defined Service when you no longer need it.

1 On the DASD MANAGER PLUS Main Menu, select **Services** and press **Enter**.

The Services Menu panel is displayed.

2 On the Services Menu panel, In the **Service** field, type a Service name, select **Delete a User Defined Service**, and press **Enter**.

**Note**

As an alternate method, you can access the Service List panel from the Service Menu panel to edit Service syntax by selecting **List**, entering a wildcard character in the **Service** field, typing **D** in the **Act** field next to a Service, and pressing **Enter**.

The Delete Service panel is displayed.

```
ASUISD0 -------------------------- Delete Service -----------------------------
Command ===>

Service . . IKJEFT01           Updated by . . .   RDAPKM
Owner . . . RDAPKM             Date last updated yyyy-03-04-12.02.52
Vendor . . . USR

More:   -   >

********************* Bottom of data *********************
```

**Working with Services**
3 In the **Delete Options** field, type 1 to confirm the delete and press **Enter**.

### Editing Services

Use this procedure to set up Services and syntax that DASD MANAGER PLUS uses to build the worklist and generate JCL for a new or existing Service Action.

**Before you begin**

Before you begin to edit the Services, you must have created a Service Action as instructed in "Creating a Service Action" on page 241.

**To edit Services**

1 Access the Edit Action Services panel as follows:

   a On the DASD MANAGER PLUS Main Menu, select **Service Actions** and press **Enter**.

   b On the Action Menu, in the **Action** field, type an Action name, select **Edit Services for an Action** and press **Enter**.

      The Action List panel displays the requested Service Actions.

As an alternative method, you can access the Edit Action Services panel from the Services Menu panel to edit Action Services. On the Action menu, enter a wildcard character in the **Action** field and press **Enter**. On the List Actions panel, type **E** next to an Action and press **Enter**.
On the Edit Action Services panel you do the following:

■ Associate Services (utilities) to be executed against DB2 Objects
  You can associate DB2 Objects by object set, object type and pattern, or object type and object name.

■ Manage the syntax options to use for a listed service
  Default syntaxes for a service are indicated with a (D) preceding the syntax name in the Syntax Name column.

The following are updatable fields and may be typed over:

■ Act
■ Service
■ Object Name/Pattern
■ Type
■ Part
■ Object Set name

On the Edit Action Services panel, you can perform the following actions:

<table>
<thead>
<tr>
<th>To do this</th>
<th>In the Act field, type the indicated keys and then press Enter</th>
</tr>
</thead>
</table>
| Add a blank line (when you know the Service name) | Type I beside the Service after which you want to insert another Service.  
  Tip: Tab to the service field of the new line and enter the Service name. If you do not know the Service name, type % in the Service field and press Enter to view the list of Services from which to select a Service. After you add a Service, you can edit the step. |
To do this | In the Act field, type the indicated keys and then press Enter
---|---
Duplicate a Service | Type L beside the Service that you want to duplicate.
Delete a Service from the action | Type D beside the Service that you want to delete.
Change the order of the Services | Type M besides the Service that you want to move. Type the one of following characters beside the target location of the Service.
- A for After
- B for Before
Edit the Services syntax options of the syntax listed in the Syntax Name field | Type O beside the Service whose syntax options you want to edit.
Apply the default syntax to a Service | Type U beside the Service to which you want to apply the default syntax.
*Note:* A D preceding the Syntax Name in the Syntax Name field indicates a default syntax.
Manage Object Sets for a Service | Type OS beside the Service that displays the Object Set Name that you want to manage.
Create a new Object Set for a Service | Type OS beside the desired Service and leave the Object Set Name field blank.
Simulate syntax based on the syntax options | Type V beside the Service whose syntax you want to simulate.
Edit the properties of the Service | Type P beside the Service whose syntax properties you want to edit.

2 Create or edit the action services, as follows:

a In the Service field, use a wild card to enter a Service name (for example, BMC % or %) to view a list of services, and press Enter.

The product displays a selection list of services and their descriptions. The Service List panel displays 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 260.

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 Step 2.d on page 268 and Step 2.e on page 268 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 “Analyzing objects by using BMCTRIG” on page 425.

d (optional) In the Object Name/Pattern 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 you edit utility options, you can use the same 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. The product stores object selections 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
Not all services support partition numbers.

g  (optional) In the Object Set Name field type an Object Set name. You may use a wildcard in the name to display a list of Object Sets.

<table>
<thead>
<tr>
<th>To do this</th>
<th>Do the following</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display an Object Set Name selection list</td>
<td>Type a value with wildcards in the Object Set Name field and press Enter.</td>
</tr>
<tr>
<td></td>
<td>Example: For example, PAYROLL.%</td>
</tr>
<tr>
<td>Edit an Object Set Specifications</td>
<td>Type OS in the Act field, an object set name in the Object Set Name field,</td>
</tr>
<tr>
<td></td>
<td>and press Enter.</td>
</tr>
<tr>
<td>Create a new Object Set</td>
<td>Type OS in the Act field and leave the Object Set Name field blank.</td>
</tr>
</tbody>
</table>

3  (optional) On the Edit Action Services panel Figure 54 on page 266, create, select, or edit the Service Action syntax by typing S in the Act field next to a Service Action to change the syntax.

Note
When you first add a Service, the product automatically selects the default syntax for that Service. In the Syntax Name field, the letter (D) precedes the default syntax.
The Service Syntax List panel displays the current syntax.

**Figure 55: Service Syntax List panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service . . . : REORG</td>
<td></td>
</tr>
<tr>
<td>Syntax . . . : BMC_DEFAULT</td>
<td></td>
</tr>
</tbody>
</table>

Select an option or change Display filter. Then press Enter. To change the default syntax, select another syntax with the 'S' option.

| B =Browse | C =Create | D =Delete | E =Edit | L =Like |
| P =Properties | A =Action Impact | V =View Syntax | S =Select Syntax |

Display filter
Owner.Name . . . %

<table>
<thead>
<tr>
<th>Act Dflt Name</th>
<th>Owner</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y BMC_DEFAULT</td>
<td>BMC</td>
<td>BMC supplied defau</td>
</tr>
<tr>
<td>JCLGEN-100001-REORG-001</td>
<td>MVSRX01</td>
<td>BMC SUPPLIED DEFAU</td>
</tr>
<tr>
<td>MVSRX01-A-REORG-001</td>
<td>MVSRX01</td>
<td>BMC SUPPLIED DEFAU</td>
</tr>
<tr>
<td>MVSRX01-SRDTDATANO-REORG-001</td>
<td>MVSRX01</td>
<td>BMC SUPPLIED DEFAU</td>
</tr>
<tr>
<td>MVSTAD1-ARCHDD</td>
<td>MVSTAD2</td>
<td>XXXX</td>
</tr>
<tr>
<td>MVSTAD1-NEWA</td>
<td>MVSTAD1</td>
<td>XXXX</td>
</tr>
</tbody>
</table>

The Service Syntax List panel lists all defined syntaxes for a Service. The current syntax contains the syntax options to be used by the service. You may change current syntax by selecting another. Changing the current syntax is discussed later.

At the top of the dialog, the **Syntax** field (commonly referred to as the current syntax) appears below the **Service** field.

BMC supplies default syntaxes for BMC and IBM services. These syntaxes contain descriptions of the default syntaxes. You cannot modify BMC-supplied default syntaxes. If you attempt to edit a BMC supplied default, the product launches the Edit Syntax Options panel in read-only mode.

On the Service Syntax List panel, you can perform the following actions:

<table>
<thead>
<tr>
<th>To do this</th>
<th>In the Act field, type the indicated keys and then press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse syntax options</td>
<td>Type B beside the syntax that you want to browse.</td>
</tr>
<tr>
<td></td>
<td>To avoid accidentally changing any options, browse instead of edit when you review.</td>
</tr>
<tr>
<td>Create a new syntax</td>
<td>Type C to create a new syntax using BMC cefault settings.</td>
</tr>
<tr>
<td>Delete a syntax from the Service</td>
<td>Type D beside the syntax that you want to delete.</td>
</tr>
<tr>
<td>Change syntax options</td>
<td>Type E to edit syntax options.</td>
</tr>
<tr>
<td>Duplicate a syntax</td>
<td>Type L beside the syntax that you want to duplicate.</td>
</tr>
<tr>
<td>Edit a Syntax Properties</td>
<td>Type P to edit syntax properties.</td>
</tr>
</tbody>
</table>
To do this | In the Act field, type the indicated keys and then press Enter
--- | ---
View the Service Actions that are using the syntax | Type A beside the syntax that is the subject of your query.
Simulate syntax based on the syntax options | Type V beside the syntax that you want to simulate.
Change the current syntax | Type S beside the syntax that you want to make current.
The content of the Syntax field should display the new syntax.

When you create a syntax, two panels are displayed.

- The first is to collect the syntax properties (for example, syntax name and description).
- The second is to collect the syntax options.

After the syntax is created and you return to the Service Syntax list panel, type S besides the new syntax to make it current.

Each service listed on the Action Service panel has a specified syntax. Each service has a unique set of syntax processing options. For example, you may have two REORGs each with a different syntax containing different processing options.

4 Press END to return to the Action Services panel.

5 (optional) On the Edit Action Services panel, type P in the Act field to view or modify the Action Service properties.

DEAE ------------------------- Action Service Options -------------------------
Command ===> Type Action Service data. Then press End.
Action . . . : MVSNXL1.MVSNXL1 Updated by :
Service . . . : REORG Last Updated:
Syntax . . . : PAYROLL_DB01
Description . .

Execution Type . 1 1. Sequential
2. Grouped
3. Parallel (Requires Full Automation)

Automation Options:
- Dependent on previous service. N (Y/N - Wait for all previous to complete)
- Maximum return code allowed . 4 (0 - 4095)

NOTE: To enter Dependent 'Y', Execution Type must be '3 - Parallel'

You can indicate whether to group objects together for the service. Grouped services let you 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 436.
For automation, when you process multiple objects, you can also use the **Dependent on previous service** indicator to specify whether the service depends on the completion of previous services. **Maximum Return Code Allowed** indicates the highest return code for continuing to process the next service. This feature is available only with the automation component. DASD MANAGER PLUS can generate multiple jobs to run in parallel, but the automation component generates complex job networks. For more information, see “Analyzing objects by using BMCTTRIG” on page 425.

6 *(optional)* On the Edit Action Services panel, type V in the Act field to view a simulation syntax based on the syntax options.

Refer to "previewing generated syntax" for more information ways DASD MANAGER simulates syntax generation.

7 *(optional)* On the Edit Action Services panel, type U in the Act field to reset the service to use the default syntax.

The Syntax Name field lists the current syntax for the service. If the Syntax name is not proceeded by a (D) indicator, than the current syntax is not the default syntax. Use the U command to change the current syntax back to the default syntax for the service.

**Previewing generated syntax**

DASD MANAGER PLUS provides two ways to simulate syntax generation. Simulation lets you preview the generated syntax before you generate the JCL.

You can preview syntax from the Edit Action Services panel or from the Service Syntax List panel.

**To view syntax from the Edit Action Services panel**

1 On the DASD MANAGER main menu, select Service Action and press Enter.

2 On the Action main menu, type an action owner and name in the format owner:name, select List, and press Enter.

3 On the Action List panel, type E in the Act field next to an Action and press Enter.
The Edit Action Services panel displays the Services that you specified for the Action.

```
EAE -------------- Edit Action Services --------------          Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR
Action: RDAPKM.DEM003
Listed is the ordered set of Services contained in the Action. Select one or more options. Then press Enter.
D =Delete  I =Insert  L =Like  A =After  B =Before  M =Move  P =Properties
S =Syntaxes  O =Syntax Options  V =View Syntax  U =Use default syntax
OS =Object Set

Act Service   Object Name/Pattern  Type  Part  Object Set Name     Syntax Name
*** **************************** Top of data **********************************
BMCSTATS                                   RDAPKM.OBJSET01     (D)ALANS-BMC
*** *************************** Bottom of data ********************************
```

4 In the Act field type V next to a Service and press Enter to view the Service syntax.

The generated Service syntax is displayed in an ISPF panel.

```
VIEW       RDAPKM.T0291600.$04MAR16.$130938.$0239085       Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ****************************
000001
000002  -BMCU 107795 ASUSMAIN
000003  BMCSTATS OBJECTSET RDAPKM.OBJSET01
000004  TABLE (ALL)
000005  UPDATEDB2 ALL
000006  FREQVAL YES
000007  NUMCOLS 15
000008  COUNT 15
000009  SORTDEVT SYSDA
000010  IXNUMQUANTILES 100
000011  NUMOCOLS 64
****** **************************** Bottom of Data ****************************
```

To view syntax from the Syntax List panel

1 On the DASD MANAGER main menu, select Service and press Enter.
2 On the Service Menu panel, in the Service field type a Service name, select List, and press Enter.
3 On the Service List panel, in the Act field type E next to a Service and press Enter.
The Service Syntax List panel displays the syntaxes for the Service.

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>DEAE------------------------</th>
<th>Service Syntax List</th>
<th>Row 3 to 18 of 28</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service . . : BMCSTATS</td>
<td>Default Syntax: BMC_DEFAULT</td>
<td>Select an option or change Display filter. Then press Enter. To change the default syntax, select another syntax with the 'F' option.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B =Browse</td>
<td>C =Create</td>
<td>D =Delete</td>
<td>E =Edit</td>
<td>L =Like</td>
<td>P =Properties</td>
</tr>
<tr>
<td>Display filter</td>
<td>Owner.Name . . . %.%</td>
<td>More: - + &gt;</td>
<td>Act Dflt Name</td>
<td>Owner</td>
<td>Description</td>
</tr>
<tr>
<td>Y</td>
<td>BMC_DEFAULT</td>
<td>BMC</td>
<td></td>
<td>BMC_STATS_CAT_COL_UPDATE</td>
<td>BMC_STATS_CAT_UPDATE</td>
</tr>
<tr>
<td>BMC</td>
<td>BMC</td>
<td>Used By Automation</td>
<td>BMC</td>
<td>BMC</td>
<td>Used By Automation</td>
</tr>
</tbody>
</table>

4 In the Act field type V next to a syntax and press Enter to view a Service syntax.

The Preview Service Syntax panel is displayed.

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>DEAE------------------------</th>
<th>Preview Service Syntax</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Syntax: BMCSTATS.BMC_DEFAULT</td>
<td>Specify the Object type to apply syntax. Then press Enter.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Type . . (TS, IX, TT, IS, SG, or VL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 In the Object Type field enter the code for an object type. Press Enter to apply the syntax to the object type that you entered.

To view a syntax from the Syntax List, you must enter the object type code manually.

An ISPF panel displays the generated Service syntax.

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>VIEW</th>
<th>RDAPKM.T0291600.$04MAR16.$131811.$1981590</th>
<th>Columns 00001 00072</th>
</tr>
</thead>
<tbody>
<tr>
<td>********</td>
<td>Top of Data</td>
<td>000001</td>
<td>000002</td>
<td>-BMCS 000001</td>
</tr>
</tbody>
</table>

Restarting a Service Action

Use this procedure to restart a Service Action that has stopped or had an error, or to start over a Service Action that is defined to contain multiple Action 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 “Analyzing objects by using BMCTRIG” on page 425.

To restart an action

1. Analyze the Worklist Execution Log in the job output to determine at which Action Service 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 277.
   - Check the sequence number to verify where the worklist will restart.

2. Correct the problem, as follows:
   - You might need to edit (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 Action Services completed successfully before the worklist failed, restart the worklist with the Action Service that failed. You don't need 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 Service Action to restart.

4. Type G in the Act field beside the Service Action.

   The first Action Job Generation panel appears. The Action field displays the action being generated.

   **Action Status** shows the current status of the action, obtained from the DO_WORKIDS table:
   - The field is blank if the product has not generated the action.
   - N indicates that you have generated the action, but have not run it.
   - R indicates that the action completed successfully.
   - S indicates that the action started but had an error.

5. In the JCL Type field, 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 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 Job Generation panel.

The second Action Job 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 Set the values as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>What to enter</th>
</tr>
</thead>
</table>
| JCL DSN     | 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 is. |
| Diagnostics | 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. |
### Understanding return codes for batch programs

The following table describes the return codes for ASUSMAIN, ASUSCPRS, and ASUSUPRS.

**Table 36: 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>
</tbody>
</table>
| 8    | One of the following events has occurred:  
  - Error gathering table space or index statistics  
  - Error writing statistics to DASD MANAGER PLUS or DB2 tables  
  - Initialization error (bad plan, options module, or syntax parse error) |
| 12   | A SYSIN DD statement open error occurred |

Table 37 on page 277 describes the return codes for ASUSTRIG.

**Table 37: 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>
</tbody>
</table>
## 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 DASD MANAGER PLUS Main Menu. If you edit the field, DASD MANAGER PLUS issues an SQL SET CURRENT SQLID command to verify your authorization to use that ID.
Using object sets

This chapter explains how to work with object sets in DASD MANAGER PLUS.

| For more information, view the Quick Course "DASD MANAGER PLUS—Understanding Object Sets". |

Before you begin using object sets

Before using object sets, you should have an understanding of DASD MANAGER PLUS Actions.

For more information, see “Maintaining and generating Service Actions” on page 223.

*Note*

During object resolution, object sets use a DB2 temporary table.

For information about setting up this temporary database, see the Installation System documentation.

Overview of object sets

Object Sets provides a facility for producing lists of objects on which services operate. For example, you can specify an object set if you want to apply an action only to certain objects or object types.

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. When working with object sets, first define and then associate them to one or more services when defining actions.
DASD MANAGER PLUS provides many types of object name patterns to select the DB2 objects to process and 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 38 on page 280 are wildcards and use an SQL LIKE operator for expansion.

**Table 38: Wildcard characters**

<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>
<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>

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.

**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 311.
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.

DASD MANAGER PLUS supports the types of object patterns as shown in Table 39 on page 281.

Table 39: Object patterns

<table>
<thead>
<tr>
<th>Pattern 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 object set name</td>
<td>CREATOR.OBJSET</td>
</tr>
<tr>
<td>PG</td>
<td>Package name pattern</td>
<td>COLLECTIONID.PACKAGE</td>
</tr>
</tbody>
</table>
| PL           | Plan name pattern | PLAN (one-part name)  
CREATOR.PLAN (two-part name) |
| SG           | Storage group name pattern | STOGROUP (one-part name)  
CREATOR.STOGROUP (two-part name) |
| SQ           | Dynamic SQL statement | Blank |
| TB           | Table name pattern | CREATOR.TBNAME |
| TS           | Table space name pattern | DBNAME.TSNAME (two-part name)  
DBNAME.TSNAME.CREATOR (three-part name) |

**Note**

If you specify OS in the **Obj Type** field of the Set Specification panel, DASD MANAGER PLUS 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, Include IX, Include RI, Include LOB, Include XML, Include HST, and Include ACH** 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 294.
Table 40 on page 282 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 40: Resulting object sets for pattern and resolution types

<table>
<thead>
<tr>
<th>Pattern type</th>
<th>Resolution type</th>
<th>Result set</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS</td>
<td>TS</td>
<td>No match</td>
</tr>
<tr>
<td></td>
<td>IX</td>
<td>Indexes that DBNAME and SPNAME identify</td>
</tr>
<tr>
<td></td>
<td>SG</td>
<td>Storage groups that contain indexes that DBNAME and SPNAME identify</td>
</tr>
<tr>
<td></td>
<td>Blank</td>
<td>Indexes that DBNAME and SPNAME identify</td>
</tr>
<tr>
<td>IX</td>
<td>TS</td>
<td>No match</td>
</tr>
<tr>
<td></td>
<td>IX</td>
<td>Indexes that CREATOR and IXNAME identify</td>
</tr>
<tr>
<td></td>
<td>SG</td>
<td>Storage groups that contain indexes that CREATOR and IXNAME identify</td>
</tr>
<tr>
<td></td>
<td>Blank</td>
<td>Indexes that CREATOR and IXNAME identify</td>
</tr>
<tr>
<td>OS</td>
<td>TS</td>
<td>Tables spaces that resolve from processing the specifications within the DASD MANAGER PLUS object set</td>
</tr>
<tr>
<td></td>
<td>IX</td>
<td>Indexes that resolve from processing the specifications within the DASD MANAGER PLUS object set</td>
</tr>
<tr>
<td></td>
<td>SG</td>
<td>Storage groups that resolve from processing the specifications within the DASD MANAGER PLUS object set</td>
</tr>
<tr>
<td></td>
<td>Blank</td>
<td>Table spaces and indexes that resolve from processing the specifications within the DASD MANAGER PLUS object set and indexes for resulting table spaces if <strong>Include IX</strong> is <strong>Yes</strong></td>
</tr>
<tr>
<td>Pattern type</td>
<td>Resolution type</td>
<td>Result set</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
<td>------------</td>
</tr>
<tr>
<td>PG</td>
<td>TS</td>
<td>Table spaces that are referenced within the packages that COLLID and PGNAME identify</td>
</tr>
<tr>
<td>IX</td>
<td></td>
<td>Indexes that are referenced within the packages that COLLID and PGNAME identify</td>
</tr>
<tr>
<td>SG</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 Yes</td>
<td></td>
</tr>
<tr>
<td>Blank</td>
<td></td>
<td>Table spaces and indexes that are referenced within the packages (that COLLID and PGNAME identify). If Include IX is Yes, the command also includes indexes that are referenced by the resulting table spaces.</td>
</tr>
<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>■ Storage groups that contain table spaces (tables) or indexes referenced within the plans (that PLNAME and CREATOR identify)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Indexes on the table spaces if Include IX is Yes</td>
<td></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></td>
<td>Storage groups that CREATOR and STOGROUP identify</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>Pattern type</td>
<td>Resolution type</td>
<td>Result set</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| SQ           | TS              | ■ Tables spaces (type TS is the first variable) that the SQL execution returns  
               ■ Table spaces using the storage group (type SG is the first variable) that the SQL execution returns |
| IX           |                 | ■ 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 |
| SG           |                 | ■ 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 |
| Blank        |                 | ■ 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 |
| TB           | TS              | Table spaces that contain tables that CREATOR and TBNAME identify |
| IX           |                 | Indexes that are defined on tables that CREATOR and TBNAME identify |
| SG           |                 | Storage groups that contain table spaces in which the tables that CREATOR and TBNAME identify are defined  
               All indexes on the Table spaces if Include IX is Yes |
| Blank        |                 | Table spaces that contain the tables that CREATOR and TBNAME identify, and the indexes on these tables  
               All indexes on the resulting table spaces if Include IX is Yes. |
| TS           | TS              | Table spaces that the DBNAME, TSNAME, and CREATOR identify |
| IX           |                 | Indexes defined on table spaces that DBNAME, TSNAME, and CREATOR identify |
| SG           |                 | Storage groups that contain table spaces that DBNAME, TSNAME, and CREATOR identify and their indexes if Include IX is Yes |
| Blank        |                 | Table spaces with DBNAME, TSNAME, and CREATOR and indexes on those tables if Include IX is Yes |

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.
Specify Object Set panel

This topic explains the Specify Object Set panel.

Figure 56: Specify Object Set panel

The panel displays the following elements of an object set:

- **Object Set:** displays the object set name (MVSNLX1.TESTIDD).

- **Act** (also called the *action column*) is used for the line commands that are listed in the panel.

  **Example**

  To Like a line, type the L line command in the **Act** column.

  **Note**

  Some date might be truncated, which is indicated by the <> characters. Press **F4** to zoom or expand truncated data.

- **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 do not return any table spaces or indexes for objects that are not defined in the catalog (for example, DSND8B01). To ensure good performance, when specifying which objects to exclude from an object set, do not add excludes for objects that are not defined in the catalog.

- **Obj Type** indicates the type of object pattern for the specification.
Note
If you specify OS in the Obj Type field of the Set Specification panel, DASD MANAGER PLUS 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, Include IX, Include RI, Include LOB, Include XML, Include HST, and Include ACH fields to further qualify an object set.

■ Name or Name Pattern is the explicit name or pattern (for more information, see “Patterns for object sets” on page 281). If the name or name pattern is longer than the field can show on the panel, you 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.

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 are excluded.

For more information, see “Example of defining object sets” on page 288. If the Begin Part and End Part fields are not zero, and you do not set By Part to Y, DASD MANAGER PLUS 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 related LOB Objects.
If a name pattern matches a base LOB object, the value in the Include column determines how the product processes related auxiliary objects.
An auxiliary object might be included because:
— It is related to the base object, or
— Its name matches the specified pattern
Valid values are:
<table>
<thead>
<tr>
<th>Value</th>
<th>Specifies to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Include base objects that match the naming pattern, but exclude auxiliary objects that match the naming pattern unless they match the naming pattern</td>
</tr>
<tr>
<td>Y</td>
<td>Include base objects that match the naming pattern and include their related auxiliary objects</td>
</tr>
<tr>
<td>B</td>
<td>Include base objects and exclude all auxiliary objects</td>
</tr>
<tr>
<td>O</td>
<td>Exclude base object and include auxiliary objects that match the naming pattern</td>
</tr>
</tbody>
</table>

- **Include XML** specifies how to include related XML objects.
  
  If a name pattern matches a base XML object, the value in the **Include** column determines how the product processes related auxiliary objects.

  An auxiliary object might be included because:
  
  - It is related to the base object, or
  - Its name matches the specified pattern

  Valid values are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Specifies to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Include base objects that match the naming pattern, but exclude their related auxiliary objects unless they match the naming pattern</td>
</tr>
<tr>
<td>Y</td>
<td>Include base objects that match the naming pattern and include their related auxiliary objects</td>
</tr>
<tr>
<td>B</td>
<td>Include base objects and exclude all auxiliary objects</td>
</tr>
<tr>
<td>O</td>
<td>Exclude base object and include auxiliary objects that match the naming pattern</td>
</tr>
</tbody>
</table>

- **Include HST** specifies how to include related temporal history table Objects.

  If a name pattern matches a base object that has a temporal history object, the **Include** column determines how the product processes related temporal history objects.

  An temporal history object might be included because:

  - It is related to the base object, or
  - Its name matches the specified pattern

  Valid values are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Specifies to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Include base objects that match the naming pattern, but excludes related temporal history objects unless they match the naming pattern</td>
</tr>
<tr>
<td>Y</td>
<td>Include base objects that match the naming pattern and include their related temporal history objects</td>
</tr>
<tr>
<td>B</td>
<td>Include base objects and exclude all temporal history objects</td>
</tr>
<tr>
<td>O</td>
<td>Exclude base object and include temporal history objects that match the naming pattern</td>
</tr>
</tbody>
</table>
• **Include ACH** specifies how to include related archive table objects.
  
  If a name pattern matches a base object that has an archive-enabled object, the **Include** column determines how the product processes related archive-enabled objects.
  
  An archive-enabled object might be included because:
  
  — It is related to the base object, or
  
  — Its name matches the specified pattern
  
  Valid values are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Specifies to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Include base objects that match the naming pattern, but excludes related archive-enabled objects unless they match the naming pattern</td>
</tr>
<tr>
<td>Y</td>
<td>Include base objects that match the naming pattern and include their related archive-enabled objects</td>
</tr>
<tr>
<td>B</td>
<td>Include base objects and exclude all archive-enabled objects</td>
</tr>
<tr>
<td>O</td>
<td>Exclude base object and include archive-enabled objects that match the naming pattern</td>
</tr>
</tbody>
</table>

**Example of defining object sets**

The following figures shows an example of the Specify Object set panel.

**Figure 57: Example Set Specification panel**

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**

As of version 11.2 of DASD MANAGER PLUS, DASD MANAGER PLUS object sets support archive tables. You can use object set tablespace specifications to include or exclude related archive table spaces. If you specify Include ACH Y in an object set definition, BMCSTATS and BMCTRIG include archive-related table spaces.

Actual object results depend on the object set resolution type. For more information, see “Viewing object set results” on page 304.

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 “Analyzing objects by using BMCTRIG” on page 425.

---

## Working with object sets

The following table shows some common tasks for setting up and maintaining object sets.

**Table 41: Object set tasks**

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an object set.</td>
<td>“Creating an Object Set” on page 291 or “Associating a new object set with an action service” on page 306</td>
</tr>
<tr>
<td>Create dynamic SQL object sets</td>
<td>“Creating dynamic SQL Object Sets” on page 294</td>
</tr>
<tr>
<td>Edit the specifications of an object set:</td>
<td>“Editing Object Sets specifications” on page 298</td>
</tr>
<tr>
<td>■ Include additional objects in or remove objects from an object set</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>
</tbody>
</table>
Defining and Maintaining object sets

Working with object sets, can be divided into two process. The first process is to manage object sets, this includes actions such as creating, deleting, copy, renaming. The second process is to associate the object set to a service, this identifies what objects are to be used by a service.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit the properties of an object set</td>
<td>“Editing object set properties” on page 304</td>
</tr>
<tr>
<td>Rename an object set</td>
<td>“Renaming object sets” on page 301</td>
</tr>
<tr>
<td>Copy an object set</td>
<td>“Copying object sets” on page 302</td>
</tr>
<tr>
<td>Delete an object set</td>
<td>“Deleting object sets” on page 303</td>
</tr>
<tr>
<td>View objects for a particular specification and use the X command to</td>
<td>“Viewing object set results” on page 304 or “Editing Object Sets specifications” on page</td>
</tr>
<tr>
<td>exclude objects from a wildcard specification</td>
<td>298</td>
</tr>
<tr>
<td>Identify actions associated with an object set</td>
<td>“Identifying actions associated with an object set” on page 300</td>
</tr>
<tr>
<td>Create an object set for a service</td>
<td>“Associating a new object set with an action service” on page 306</td>
</tr>
<tr>
<td>Associate an existing object set with an action service</td>
<td>“Associating an existing object set with an action service” on page 307</td>
</tr>
<tr>
<td>Associate an object set with a new action service</td>
<td>“Associating an existing object set with a new action” on page 309</td>
</tr>
</tbody>
</table>
By selecting **Object Sets** from the DASD MANAGER PLUS Main Menu, you can define and maintain object sets. The Object Set Main menu provides means to manage the object set.

**Figure 58: Object Set menu processing**

Managing object sets through the Object Set menu 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 an Object Set**

Use this procedure to create an Object Set outside a Service Action by using the **Object Sets** option on the Main Menu.

For more information, view the Quick Course "DASD MANAGER PLUS—Creating Object Sets."
To create an Object Set

1. On the DASD MANAGER PLUS Main Menu, select **Object Sets** and press **Enter**.

   The Object Sets menu panel is displayed.

   **DEAE ----------------------------- Object Set Menu -----------------------------**
   **Command ===>**
   An Object Set is an ordered group of DB2 Objects. Type a specific Object Set or type a wildcard pattern for a selection list.
   **Object Set . . . BMCDE30.FIOBJSET01**
   Select an option. Then press Enter.
   1. List
   2. Create a new Object Set
   3. Edit an Object Set Specification
   4. Edit an Object Set Properties
   5. Rename an Object Set
   6. Copy an Object Set
   7. Delete an Object Set
   8. View Object Set Result Set
   9. List Action Impact

2. In the **Object Set** field, enter an object set name in the form of **owner.name** and press **Enter**.

   DASD MANAGER PLUS validates that it does not already exist.

   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 311. Together, the owner name and Object Set name must form a unique name. The default owner name is your TSO logon ID.

3. Select Create a new Object Set and press **Enter**.

   The Create Object Set panel appears:

   **DEAE ---------------------------- Create Object Set ---------------------------**
   **Command ===>**
   Type new Object Set data. Then press Enter.
   **Object Set . . . BMCDE30.FIOBJSET01**
   **Description . . .**

   **Note**

   As an alternative method, perform the following steps to open the Object Set List from the main object set panel:

   1. Select **List** and specify a wildcard pattern for the object set name.
   2. Type C in the **Act** field.
4 On the Create Object Set panel, optionally enter an Object Set description and press Enter.

**Note**
The Object set name that you specified on the Object Set Main menu is carried forward. You can change the Object Set name if necessary.

The Specify Object Set panel is displayed.

5 Enter the following information:

<table>
<thead>
<tr>
<th>In this field</th>
<th>Enter this information</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCL/EXCL</td>
<td>Plus (+) sign to include, or minus (-) sign to exclude.</td>
</tr>
<tr>
<td>Obj Type</td>
<td>TS, IX, TB, SG, PL, PG, IS, OS, or SQ</td>
</tr>
<tr>
<td>Name or Name Pattern</td>
<td>Type a name or name pattern</td>
</tr>
<tr>
<td>By Part</td>
<td>Type Y if you want to specify partition numbers, or N if you don't want to specify partition numbers.</td>
</tr>
<tr>
<td>Begin Part</td>
<td>Enter partition numbers. If you entered N in the By Part field, these values are set to zero.</td>
</tr>
<tr>
<td>End Part</td>
<td></td>
</tr>
<tr>
<td>INCLUDE OPTIONS</td>
<td></td>
</tr>
<tr>
<td>IX</td>
<td>Enter Y to include these objects, or N to exclude them.</td>
</tr>
<tr>
<td>RI</td>
<td></td>
</tr>
<tr>
<td>LOB</td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td></td>
</tr>
<tr>
<td>HST</td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td></td>
</tr>
<tr>
<td>ACH</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>(optional) Enter a description.</td>
</tr>
</tbody>
</table>
Note

If you enter **OS** in the **Obj Type** field, 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, Include IX, Include RI, Include XML, Include HST,** and **Include ACH** 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 281.

6 *(optional)* Use the following commands to edit the Object Set:
- **I** inserts a line for you to add a new specification.
- **L** inserts a line that repeats the previous line (for faster editing).
- **D** deletes a specification line.
- **E** displays the Object Set Specification Edit panel. This panel displays the line in detail format.
- **C** copies a line to another location. You must use the **A(fter) or B(efore)** in the **Act** column to designate a destination for the copied line.
- **M** moves a line to another location. You must use the **A(fter) or B(efore)** in the **Act** column to designate a destination for the moved line.
- **A** designates the line after the current line as the destination for the Move or Copy action
- **B** designates the line before the current line as the destination for the Move or Copy action
- **S** allows you to add SQL for a **SQ**-type specification.
- **X** expands wild cards that you have entered

Note

Use the PF4 key to zoom into the contents of a long object name field. Place the cursor over the truncated field and press **F4** to see the entire field.

7 Press **END** to create the Object Set.

Creating dynamic SQL Object Sets

Use this procedure to create an object set that contains dynamic SQL. You can use dynamic SQL object sets to create a SQL statement to fetch the objects that you want the services to operate upon. When you enter the SQL statement, DASD MANAGER PLUS executes an SQL PREPARE for validation purposes before saving the SQL statement to the repository.

DASD MANAGER PLUS reads the temporary data set, verifying it by performing SQL PREPARE. Then, 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 Set menu panel appears.

   DEAE ----------------------------- Object Set Menu -----------------------------
   Command ===>  
   An Object Set is an ordered group of DB2 Objects. Type a specific Object Set or type a wildcard pattern for a selection list.
   Object Set . . .  BMCDEC30.FIOBJSET01
   Select an option. Then press Enter.
   1. List
   2. Create a new Object Set
   3. Edit an Object Set Specification
   4. Edit an Object Set Properties
   5. Rename an Object Set
   6. Copy an Object Set
   7. Delete an Object Set
   8. View Object Set Result Set
   9. List Action Impact

2. In the **Object Set** field, enter an object set name in the form of `owner.name` and press Enter.

   DASD MANAGER PLUS validates that it does not already exist.

   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 311. Together, the owner name and Object Set name must form a unique name. The default owner name is your TSO logon ID.

3. Select Create a new Object Set and press Enter.

   The Create Object Set panel appears:

   DEAE ---------------------------- Create Object Set -----------------------------
   Command ===>  
   Type new Object Set data. Then press Enter.
   Object Set . . .  BMCDEC30.FIOBJSET01
   Description . . .
As an alternative method, perform the following steps to open the Object Set List from the main object set panel:

1. Select **List** and specify a wildcard pattern for the object set name.

2. Type **C** in the **Act** field.

4. On the Create Object Set panel, optionally enter an Object Set description and press **Enter**.

*Note*

The Object set name that you specified on the Object Set Main menu is carried forward. You can change the Object Set name if necessary.

The Specify Object Set panel is displayed.

<table>
<thead>
<tr>
<th>In this field</th>
<th>Enter this information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Act</strong></td>
<td><strong>S</strong></td>
</tr>
<tr>
<td><strong>INCL/EXCL</strong></td>
<td>Plus (+) sign to include, or minus (-) sign to exclude.</td>
</tr>
<tr>
<td><strong>Obj Type</strong></td>
<td><strong>SQ</strong></td>
</tr>
<tr>
<td><strong>Name or Name Pattern</strong></td>
<td>Type a name or name pattern</td>
</tr>
<tr>
<td><strong>By Part</strong></td>
<td>Type <strong>Y</strong> if you want to specify partition numbers, or <strong>N</strong> if you don't want to specify partition numbers.</td>
</tr>
<tr>
<td><strong>Begin Part</strong></td>
<td>Enter partition numbers. If you entered <strong>N</strong> in the <strong>By Part</strong> field, these values are set to zero.</td>
</tr>
<tr>
<td><strong>End Part</strong></td>
<td></td>
</tr>
<tr>
<td><strong>INCLUDE OPTIONS</strong></td>
<td></td>
</tr>
</tbody>
</table>
DASD MANAGER PLUS opens ISPF Edit on a temporary data set for you to enter SQL statements. An ISPF Edit on a temporary dataset is used for creating and changing SQL statements.

6 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>In this field</th>
<th>Enter this information</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX</td>
<td>Enter <strong>Y</strong> to include these objects, or <strong>N</strong> to exclude them.</td>
</tr>
<tr>
<td>RI</td>
<td></td>
</tr>
<tr>
<td>LOB</td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td></td>
</tr>
<tr>
<td>HST</td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td></td>
</tr>
<tr>
<td>ACH</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td><em>(optional)</em> Enter a description.</td>
</tr>
</tbody>
</table>
### Example

**Table 42: 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 SYSIBM.SYSTABLESPACE WHERE DBNAME LIKE 'QZU%'&lt;br&gt;SELECT 'TS', DBNAME, TSNAME, PARTITION FROM SYSIBM.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 SYSIBM.SYSINDEXES WHERE TBCREATOR='QZU'&lt;br&gt;SELECT 'IX', IXCREATOR, IXNAME, PARTITION FROM SYSIBM.SYSINDEXPART WHERE IXCREATOR='QZU' AND PARTITION &gt; 100 AND PARTITION &lt;= 200</td>
</tr>
<tr>
<td>SG</td>
<td>SELECT 'SG','ANY', STORNAME FROM SYSIBM.SYSTABLEPART WHERE DBNAME='QZUD11' AND TSNAME='QZUS0111'&lt;br&gt;SELECT 'SG','ANY', STORNAME, PARTITION FROM SYSIBM.SYSTABLEPART WHERE DBNAME='QZUD11' AND TSNAME='QZUS0111' AND PARTITION BETWEEN 1 AND 4</td>
</tr>
</tbody>
</table>

7 Press END to the ISPF edit and return to the Object Set Specification panel.

8 Press END again to create the Object Set.

**Editing Object Sets specifications**

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 On the DASD MANAGER Plus Main Menu, select **Object Sets** and press **Enter**.

2 On the Object Set Menu panel, type the name of an Object Set, select **Edit Object an Set Specification** and press Enter.
As an alternative method, perform the following steps to open the Object Set List from the main object set panel:

1. Select List and specify a wildcard pattern for the object set name.

2. Type E in the Act field.

The Edit Specification panel is displayed.

DEAE --------------- Specify Object Set ---------------  Row 1 to 1 of 1
Command ===>                                                   Scroll ===> CSR

Object Set: BMCDE30.FIOBJSET01

I =Insert L =Like D =Delete E =Edit C =Copy M =Move A =After B =Before
S =SQL text (for type SQ)  X =Expand Wildcard

Include/Exclude options: Plus sign = Include, Minus sign = Exclude.
Object types: TS, IX, TB, SG, PL, PG, IS, OS, SQ

---

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 X (Expand wildcard), and press Enter.

The Object Set Specification Expansion panel is displayed.

DEAE --- Object Set Specification Expansion ---  Row 1 to 3 of 3
Command ===>                                                   Scroll ===> CSR

Object Set . : BMCDE30.FIOBJSET01
Object Type . : TS
Object Pattern: QZUD11.%
Option . . : EXCLUDE

Type action next to name. Then press Enter.
X =generate an explicit exclude/include for an object

Act Object Name
QZUD11.QZUS0111.ASUQA
QZUD11.QZUS0211.ASUQA
QZUD11.QZUS0311.ASUQA

***************************************************************************************
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, 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.

5 Press **END** to save your changes.

### Identifying actions associated with an object set

You might need to know with what actions an object set is associated. You can use the List Action Impact to view Actions using an 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 On the DASD MANAGER PLUS Main Menu, select **Object Sets** and press **Enter**.

2 On the Object Set menu, enter the Object set name to view associated actions, select **List Action Impacts**, and press **Enter**.
Note

As an alternative method, perform the following steps to open the Object Set List from the main object set panel:

1. Select **List** and specify a wildcard pattern for the object set name.
2. Type **A** in an **Act** field.

The Object Set Action Impact List panel is displayed.

<table>
<thead>
<tr>
<th>Action Name</th>
<th>Sequence Number</th>
<th>Service Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDDDEMO</td>
<td>1</td>
<td>BMCSTATS</td>
</tr>
</tbody>
</table>

The fields on the Object Set Action Impact List are as follows:

**Table 43: Fields on the Object Set Action Impact List**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Set</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>

3. Press **END** to return.

**Renaming object sets**

When you rename an object set, all actions associated with the object set are updated to use the rename value.

**To rename an object set**

1. On the DASD MANAGER PLUS Main Menu, select **Object Sets** and press **Enter**.
2. On the Object Set menu, enter the Object set name to rename, select **Rename Object Set**, and press **Enter**.
Note

Alternately, you can open the Object Set List from the main object set panel by selecting List and specifying a wildcard pattern for the 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 appears:

```
DEAE ---------------------------- Rename Object Set ----------------------------
Command ===>
Type new Object Set data. Then press Enter.
Old Object Set . . . : MVSNXL1.TESTIDD
New Object Set . . . : MVSNXL1.TESTIDD
Description . . . :
```

3 In the New Object Set field, type the new name.

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 311.

4 Press Enter to save your changes.

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 Object Set menu, enter the Object set name to copy, select Copy an Object Set, and press Enter.

Note

Alternately, you can open the Object Set List from the main object set panel by selecting List and specifying a wildcard pattern for the object set name. Then, enter L in the Act field next to the object set that you want to copy.

The Copy Object Set panel opens.

```
DEAE ----------------------------- Copy Object Set ----------------------------
Command ===>
Type new Object Set data. Then press Enter.
Object Set . . . : 01803370.0S37001
```
3 In the **Object Set** field, enter the name for the copy of the object set.

Optionally, you can enter an description of the Object Set.

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 311.

4 Press **Enter** to save your changes.

### Deleting object sets

Before deleting an object set, verify that it is currently not associated with any actions. Object sets being used by actions must not be deleted. Before deleting an object set, you must remove all associations.

To identify actions associated with an object set refer to “Identifying actions associated with an object set” on page 300.

#### To delete an object set

1 On the DASD MANAGER PLUS Main Menu, select **Object Sets** and press **Enter**.

2 On the Object Set menu, enter the Object set name to delete, select **Delete an Object Set**, and press **Enter**.

    **Note**

    As an alternative method, perform the following steps to open the Object Set List from the main object set panel:

    1 Select **List** and specify a wildcard pattern for the object set name.

    2 Type **D** in the **Act** field next to an object.

    The Delete Object Set panel is displayed.
3 In the **Delete Options** field, type 1 to confirm the delete and press **Enter**.

### Editing object set properties

Use the following procedure to edit object set properties.

1 In the DASD MANAGER PLUS main menu select **Object Sets** and press **Enter**.

2 In the Object Sets Main Menu, enter the object set name.

3 Select **Edit an Object Set Properties**, and press **Enter**.

**Note**

As an alternative method, perform the following steps to open the Object Set List from the main object set panel:

1 Select **List** and specify a wildcard pattern for the object set name.

2 Type **P** in the **Act** field next to an Object Set.

The Edit Object Set Properties panel appears.

```
DEAE ----------------------- Edit Object Set Properties -----------------------
Command ===>
Type a description. Then press Enter.
Object Set . . . : BMCDE30.FIOBJSET01
Number of objects: -1           (From Last Expansion)
Date Created . . : yyyy-09-15.32.09.503757
Created by . . . : RDAPKM
Date last updated: yyyy-09-15.32.09.503757
Updated by . . . : RDAPKM
Description . . .
```

4 *(optional)* Type an Object Set description.

5 Press **END** to save your changes.

### Viewing object set results

Use this procedure to view object set results.
To view object set results

1. From the DASD MANAGER PLUS Main Menu select **Object sets**, and press **Enter**.

2. In the Object Set Menu panel, enter the object set name, then select **View object set result set** and press **Enter**.

   *Note*

   As an alternative method, perform the following steps to open the Object Set List from the main object set panel:

   1. Select **List** and specify a wildcard pattern for the object set name.

   2. Type **V** in the **Act** field next to an object.

   The View Result Set panel is displayed.

   DEAE ----------------------------- View Result Set -----------------------------
   Command ===>                                                   Scroll ==> PAGE
   Object Set: BMCDEMO.FIOBJSET01
   Select resolve option. Then press Enter.
   Resolve Options . .   1. Tablespaces (TS)
                       2. Indexes (IX)
                       3. Stogroups (SG)
                       4. Tablespaces and Indexes (TS, IX)
   Additional wildcard pattern.
   X =Generate explicit exclude entry  Z =Zoom
   Act Type
   ************************************************************************
   Bottom of data ************************************************************************

3. Select what object types you want to resolve, and press Enter.

   *Example*

   Enter **2** to view indexes that match the object name wildcard pattern specified for indexes.

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 specification.

   DASD MANAGER PLUS displays a list of the objects in the object set.

5. Press **END** to return.
Associating object sets with action services

Once you have define your object set, you will need to associate them with a service. By associating an object set for a service you are designating what objects the services is to process. Associating an object set with a service is accomplished while working with actions. More specifically, you specify the object set for an action service.

There are three methods to associating an object set to a service:

- “Associating a new object set with an action service” on page 306
- “Associating an existing object set with an action service” on page 307
- “Associating an existing object set with a new action” on page 309

Associating a new object set with an action service

This method is used to create a new object set on demand. Use the following procedure to create and associate an on demand object sets

1. On the DASD MANAGER PLUS main menu select **Service Actions** and press **Enter**.

2. On the Service Actions menu, enter an Action name, select **Edit Services for an Action**, and press **Enter**.

   **Note**

   As an alternative method, perform the following steps to open the Object Set List from the main object set panel:

   1. Select **List** and specify a wildcard pattern for the object set name.

   2. Type **E** in the **Act** field next to an action and press **Enter**.

   The Edit Action Services panel is displayed.

<table>
<thead>
<tr>
<th>Command</th>
<th>Action: PAYROLL.DEMO01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listed is the ordered set of Services contained in the Action. Select one or more options. Then press Enter.</td>
<td></td>
</tr>
<tr>
<td>D =Delete I =Insert L =Like A =After B =Before M =Move P =Properties S =Syntaxes O =Syntax Options V =View Syntax U =Use default syntax OS =Object Set</td>
<td></td>
</tr>
<tr>
<td>Act Service</td>
<td>Object Name/Pattern</td>
</tr>
<tr>
<td>More: &gt;</td>
<td></td>
</tr>
</tbody>
</table>
3 In the Act field, enter OS, blank out or leave the Object Set Name field blank, and press Enter.

4 Follow the instructions provided in “Creating an Object Set” on page 291 to create an object set.

When you have finished creating the object set, the newly created object set is automatically associated with the action service.

Example

DEAE --------------- Edit Action Services ---------------     Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR
Action: PAYROLL.DEMO01
Listed is the ordered set of Services contained in the Action. Select one or more options. Then press Enter.

D =Delete  I =Insert  L =Like  A =After  B =Before  M =Move  P =Properties
S =Syntaxes  O =Syntax Options  V =View Syntax  U =Use default syntax
OS =Object Set

More:       >

Act Service   Object Name/Pattern  Type  Part  Object Set Name     Syntax Name

5 (optional) If you need to add or remove objects from the object set definition, type OS in the Act field and press Enter.

The Specify Object Set panel is displayed.

Associating an existing object set with an action service

This method is used to specify an existing object set by either specifying the name or by selecting the object set from a list. Use the following procedure to associate an existing object set.

1 On the DASD MANAGER PLUS main menu select Service Actions and press Enter.

2 On the Service Actions menu, enter an Action name, select Edit Services for an Action, and press Enter.

Alternatively you can open the Edit Services for an Action panel by selecting List and specifying a wildcard pattern for the action name. Then, enter E in the Act field next to the Action that you want to edit and press Enter.
The Edit Action Services panel appears.

```
Action: PAYROLL.DEMO01

Listed is the ordered set of Services contained in the Action. Select one or more options. Then press Enter.

D =Delete  I =Insert  L =Like  A =After  B =Before  M =Move  P =Properties
S =Syntaxes  O =Syntax Options  V =View Syntax  U =Use default syntax
OS =Object Set

Act Service   Object Name/Pattern  Type  Part  Object Set Name     Syntax Name
*** **************************** Top of data **********************************
BMCREORG                                                       (D)BMC_DEFAU

*** *************************** Bottom of data ********************************
```

3 Specify the name of the object set in the Object Set Name field.

If you do not know the name:

1 Enter OS in the Act field.

2 Enter a wildcard pattern in the Object Set Name field

3 Press Enter.

The Object Set Selection List panel appears.

```
Object Set: PAYROLL.%

Select an Object Set. Then press Enter.

Act Owner   Name                    Description
*******************************************************************************
PAYROLL DEMO01                  OBJECT SET FOR PAYROLL
PAYROLL DEMO02                  OBJECT SET FOR PAYROLL 2

******************************* Bottom of data ********************************
```

4 Enter S in the Act field next to the required object set and press Enter.

The selected object set is automatically associated with the action service.

5 (optional) If you need to add or remove objects from the object set definition, enter OS in the Act field and press Enter.

This displays the Specify Object Set panel.
Associating an existing object set with a new action

Use this method to specify an existing object set while defining a new action. Use the following procedure to associate an existing object set with a new action.

1. On the DASD MANAGER PLUS main menu select Service Actions and press Enter.

2. On the Service Actions menu, enter an new Action name, select Create a new Action.

3. In the Create Action panel, enter a unique Action name, optionally enter a description, and press Enter.

   The Action Object Specification panel appears.

   ```
   DEAE ----------------------- Action Object Specification ----------------------
   Command ===>
   Action  . : PAYROLL.DEM003
   Please specify object information for the Service to use. Object information may consist of an Object Set, a fully qualified object name, or a wildcard pattern. Then press Enter.
   Object Set
   Name . . PAYROLL.DEM002               (%= Object Set Select List)
   Object Name or Pattern
   Name . .
   Type  . .        (IS,IX,SG,TS,TT,VL)
   Part  . .        (1-4096 and not valid for wildcarded name)
   ```

4. Perform one of the following actions to select an existing object set:

   - In the Object Set Name field, enter the name of an existing object set, and press Enter.
   - Look up an Object set name:
     1. Enter a wildcard pattern in the Object Set Name field.
     2. Press Enter.

   The Object Set Selection List is displayed.

   ```
   DEAE --------------------- Object Set Selection List ---- Row 1 to 10 of 10
   Command ===>                                               Scroll ===> CSR
   Object Set: P%.%
   Select an Object Set. Then press Enter.
   ```

<table>
<thead>
<tr>
<th>Act Owner</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAYROLL</td>
<td>DEM001</td>
<td>OBJECT SET FOR PAYROLL</td>
</tr>
<tr>
<td>PAYROLL</td>
<td>DEM002</td>
<td>OBJECT SET FOR PAYROLL 2</td>
</tr>
<tr>
<td>POFGEN</td>
<td>DS0001</td>
<td></td>
</tr>
<tr>
<td>PUBLIC</td>
<td>ACPMRDB_ALL</td>
<td>DSNUM ALL</td>
</tr>
</tbody>
</table>

Chapter 5 Using object sets 309
The Action Object Specification panel is updated with your selection.

5 Press **Enter**.

The Service Selection List is displayed.

6 Select the Service by entering **S** in the **Act** field and press **Enter**.

The selected service is automatically associated with the action service.

7 Click **END** to return to the Edit Action Services panel.

8 Continue with the remaining panels to complete the Action definition.
Controlling access to Object Sets

By default, all users have can maintain Object Sets. To restrict maintenance to only the owner of Object Sets, use the OSAUTHCHK option in the installation options module. 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 either of the following conditions exists:
  - 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 explains how to collect, view, and manage statistics.

Overview of collecting and managing statistics

The DASD MANAGER PLUS product provides a set of statistics tools. These tools 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 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 “Analyzing objects by using BMCTRIG” on page 425) 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

You should run BMCSTATS to collect statistics before you try to use the statistics display, update, and management tasks that this chapter describes. For more information, see “Updating the DB2 catalog by using BMCSTATS” on page 363.

**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 336.

Other statistics tasks in this chapter use Service Actions. If you are not familiar with Service Actions, see “Maintaining and generating Service Actions” on page 223.

**Quick reference of statistics-related tasks**

The following table shows typical ways to use the statistics tasks.

**Table 44: Programs for collecting and updating statistics**

<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>

Table 45 on page 314 lists the detailed tasks.

**Table 45: 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 336</td>
</tr>
<tr>
<td>Task</td>
<td>Reference</td>
</tr>
<tr>
<td>-------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>List DB2 object data sets.</td>
<td>“Listing data sets used by a database” on page 338</td>
</tr>
<tr>
<td>Specify BMCSTATS job parameters.</td>
<td>“Updating the DB2 catalog by using BMCSTATS” on page 363</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></td>
</tr>
<tr>
<td>Compare DB2 statistics and BMCSTATS statistics.</td>
<td>“DB2 and BMCSTATS statistics” on page 341</td>
</tr>
<tr>
<td>Update DB2 statistics with your own values.</td>
<td>“Updating the DB2 catalog interactively” on page 348</td>
</tr>
<tr>
<td>Update DB2 statistics with values from the DASD MANAGER PLUS database.</td>
<td>“Updating the DB2 catalog by using BMCUPRS” on page 370</td>
</tr>
</tbody>
</table>

**Tips for collecting statistics**

Use the following suggestions to improve the performance of BMCSTATS:

---

**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
- Tablespaces with hash organization

---

- 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 ASURSDEL 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 319.
- 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. For more information, see the *IBM DB2 for z/OS SQL Reference*.

BMCSTATS utility

The BMCSTATS utility 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 REORGNLEVELS (the number of index levels 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.

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 59 on page 318 shows how the BMCSTATS utility gathers statistics.

**Figure 59: Gathering statistics using the BMCSTATS utility**

---

**DASD MANAGER PLUS historical database**

The following table 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 DASD MANAGER PLUS for DB2 Reference Manual*.

**Table 46: DASD MANAGER PLUS historical database**

<table>
<thead>
<tr>
<th>Table name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCATS nn.RS_COLDIST</td>
<td>Distribution statistics on columns</td>
</tr>
<tr>
<td>BMCATS nn.RS_COLDISTSTAT</td>
<td>Distribution statistics on columns at the partition level</td>
</tr>
<tr>
<td>BMCATS nn.RS_COLSTATS</td>
<td>Statistics on each column in a partition</td>
</tr>
<tr>
<td>BMCATS nn.RS_COLUMNS</td>
<td>Table columns</td>
</tr>
<tr>
<td>Table name</td>
<td>Contents</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>BMCATS nn.RS_INDEXES</td>
<td>Indexes</td>
</tr>
<tr>
<td>BMCATS nn.RS_INDEXPART</td>
<td>Index partitions</td>
</tr>
<tr>
<td>BMCATS nn.RS_INDEXPART_DIST</td>
<td>Index partition page group</td>
</tr>
<tr>
<td>BMCATS nn.RS_KEYTARGETS</td>
<td>Key targets</td>
</tr>
<tr>
<td>BMCATS nn.RS_KEYTARGETSTATS</td>
<td>Key targets partitions</td>
</tr>
<tr>
<td>BMCATS nn.RS_KEYTGTDIST</td>
<td>Distribution statistics on key target columns</td>
</tr>
<tr>
<td>BMCATS nn.RS_KEYTGTDISTSTATS</td>
<td>Distribution statistics on key target columns at the partition level</td>
</tr>
<tr>
<td>BMCATS nn.RS_LOBSTATS</td>
<td>LOB statistics</td>
</tr>
<tr>
<td>BMCATS nn.RS_STOGROUP</td>
<td>Storage groups</td>
</tr>
<tr>
<td>BMCATS nn.RS_TABLEPART</td>
<td>Table space partitions</td>
</tr>
<tr>
<td>BMCATS nn.RS_TABLES</td>
<td>Tables</td>
</tr>
<tr>
<td>BMCATS nn.RS_TABLESPACE</td>
<td>Table space statistics</td>
</tr>
<tr>
<td>BMCATS nn.RS_TSPART_DIST</td>
<td>Table space partition page group</td>
</tr>
<tr>
<td>BMCATS nn.RS_VOLUMES</td>
<td>Volumes</td>
</tr>
</tbody>
</table>

**BMCSTATS options**

When you collect statistics, you choose parameters on the first BMCSTATS panel. As the following sample shows, this panel lists the parameters by category.
Figure 60: BMCSTATS parameters

<table>
<thead>
<tr>
<th>Command ====&gt;</th>
<th>BMCSTATS</th>
<th>Scroll ===&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Service Syntax:</strong> BMCSTATS.I431937S</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Type Service Syntax options. Then press End.**

---

**What to Collect**

- **TABLE** . . . .  N   (Y/N/S Y=ALL Tables, N=No Tables, S=Select Tables)
- **INDEX** . . . .   Y   (Y/N Collect column statistics on all indexes)
- **SPACEONLY** . . . N   (Y/N Collect space information only)

---

**Distribution Stats:**

- **NUMCOLS** . . . 1   (1-64 Max index key columns to concatenate)

**Histogram Stats:**

- **IXNUMQUANTILES**   (1-100 Number of quantiles to collect)
- **NUMQCOLS** . . .   (1-64 Number of columns for quantiles)

---

**FREQVAL** . . . . N   (Y/N Collect frequent value statistics)

---

**Table Space Options**

- **COUNT** . . . .   (1-300 Max number of frequent values to collect)
- **FREQTYPE** . . . M   (M/L/B M=Most L=Least B=Both)

---

**BMC Stats Reporting and Update Options**

- **SAVESTATS** . . Y   (Y/N Save statistics in STATS DB)
- **REPORT** . . . .   Y   (Y/N Print statistics report)

---

**DB2 Catalog Update Options**

- **UPDATEDB2** . . N   (N/A/P/S N=None, A=All, P=Accesspath, S=Space)
- **DELETEHISTAGE** 32767 (0-32767 Days for deleting history table entries)
- **RESETACCESSPATH** N   (N/Y Reset accesspath statistics for all tables)
- **RESETHISTORY** . N   (N/Y Inserts history rows for which access path statistics are reset)
- **OMITCARD0** . . N   (Y/N - Bypass catalog update if cardinality is 0)

---

**Stats Processing Options**

- **BADOBJECTRC** . . 4   (0-8 Return code when object is bypassed)
- **RECALL** . . . .   N   (Y/N Recall archived data datasets)
- **FORCEROLLUP** . . N   (Y/N Produce aggregate stats when missing parts)
- **MSGLEVEL** . . . 0   (0/1 0-Normal msgs, 1-Additional msgs)
- **911ACTION** . . . I   (I/S Ignore or Stop at -911 SQL errors)

---

**Stats Tuning Options**

- **TASKS** . . . .   1-16   (Multitasking level)
- **TSSAMPLEPCT** . . N   (Y/N/1-50 Random sampling tablespace statistics)
- **IXSAMPLEPCT** . . N   (Y/N/1-50 Random sampling for index statistics)
- **ATMBORKAREA** . . N   (Y/N Freq/card work areas above the 2G bar)
- **OPTIMIZECOMMIT** . . N   (Y/N Reduce DELETE, INSERT, UPDATE commits)
- **QUIESCENAME** . . (0/10000-1000000 Partitions processed before storage reorganization)

---

**Sort Options**

- **SORTNUM** . . . .  (0-255 Number of SORTNks for Histogram Stats)
- **SORTDEVT** . . . . (Device type for sort data sets for Histogram Stats)

---

**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:
  - **Y**—Collects statistics on all table columns (see “BMCSTATS options for tables” on page 326).
— N—Does not collect statistics on table columns.
— S—Collects statistics on specific tables and columns.

You can also select column groups for collection.

**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.

- **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.

- **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.

- **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 PLUS to collect this type of statistic. Otherwise, DASD MANAGER PLUS 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 60 on page 320). 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 Catalog History tables.

■ **DELETEHISTAGE** specifies how long to keep the DB2 Catalog History table entries before deleting them.
- **RESET ACCESSPATH** resets access path statistics in the DB2 Catalog to -1 for all tables in the specified table space and related indexes. Real-time statistics and space statistics in the DB2 Catalog are not reset.

- **RESETHISTORY** inserts rows into the following tables, for the objects specified in the RESET ACCESSPATH command:
  - SYSIBM.SYSTABLES_HIST for tables
  - SYSIBM.SYSINDEXES_HIST for indexes

- **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 47 on page 324 defines the sample table space parameter options.

Table 47: BMCSTATS 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.</td>
</tr>
<tr>
<td>1–50</td>
<td>Sample the specified percentage of pages for statistics.</td>
</tr>
</tbody>
</table>

  ^a The numPages/numTables must be greater than the minimum pages listed, or BMCSTATS will process all pages.

For more information about sampling, see “Sampling statistics” on page 332.

- **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.

  **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

  Table 48 on page 325 defines sample index parameter options.
Table 48: BMCSTATS 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 332.

- **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** (100001000000) 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 (Sx78 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 SORTDEVT field. You can specify an integer value of 0 through 99.

- **SORTDEVT** 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, the BMCSTATS Select Tables panel is displayed:

**Figure 61: BMCSTATS Select Tables panel**

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** is displayed in “Figure 62” on page 327.

- 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 62: BMCSTATS Options for Table `tableName` panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Options for Table TABLE.QZU.QZUT01_D10S01</th>
<th>Scroll</th>
<th>Row 1 to 33 of 33</th>
</tr>
</thead>
</table>

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)
  COLGROUPPREQYES          (Y/N Collect frequent value statistics)
  COLGROUPPREQTYPE         (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)
  COLGROUPPREQYES          (Y/N Collect frequent value statistics)
  COLGROUPPREQTYPE         (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)
  COLGROUPPREQYES          (Y/N Collect frequent value statistics)
  COLGROUPPREQTYPE         (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)
  COLGROUPPREQYES          (Y/N Collect frequent value statistics)
  COLGROUPPREQTYPE         (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)
  COLGROUPPREQYES          (Y/N Collect frequent value statistics)
  COLGROUPPREQTYPE         (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)
```

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 n** 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 74 on page 365. 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 74 on page 365) 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 74 on page 365). 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. 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 Figure 63 on page 329 appears:

**Figure 63: BMCSTATS Select Columns panel**

You can select up to a maximum of 80 column names on which to gather statistics, as follows:

- To select a column, type S in the Sel field for the column.
- To unselect a column, blank out any character in the Sel field.

**BMCSTATS performance and tuning**

The following topics discuss offloading workloads to a zIIP processor and performing non-partition Indexes processing.

**zIIP offloading**

DASD MANAGER PLUS can use BMCSTATS to offload eligible workloads to an IBM System z Integrated Information Processor (zIIP) on IBM z9 and z10 systems. Offloading workloads provides the following benefits:

- Frees general computing capacity
- Reduces the IBM monthly licensing charges
- Reduces your mainframe's total cost of ownership (TCO)

You can offload the following types of workload:

- All DB2 page file I/O processing
- Table space data analysis
- Index space data analysis

To use zIIP processing, you must have installed version 5.6 or later of the EXTENDED BUFFER MANAGER (XBM) product (with PTF BPE0313) or
SNAPSHOT UPGRADE FEATURE for DB2 (SUF) technology. This zIIP offload does not apply to BMC Utilities executing BMCSTATS inline.

You can set the zIIP option for BMCSTATS as the default by setting the zIIP option to zIIP=E in the default options module (DOPTs). When the zIIP option is the default option, all BMCSTATS jobs that use the DOPTs module offload all zIIP-eligible processing. If no zIIP is available, BMCSTATS executes the workload on the general processor.

To disable automatic zIIP offloading, set the zIIP option to zIIP=D.

Use the BMCSTATS ZIIP syntax options to override the DOPTs setting for individual BMCSTATS jobs. In the BMCSTATS syntax, ZIIP E enables BMCSTATS to zIIP offloading; and ZIIP D disables zIIP offloading.

Since XBM handles zIIP usage for BMCSTATS, BMCSTATS and XBM use the same dispatching priority. To lower the BMCSTATS dispatching priority, create another XBM with a lower priority and use the lower priority XBM in the BMCSTATS job. Use the BMCSTATS syntax option XMBID to cause BMCSTATS to execute using the XBMID with the desired priority dispatching.

Non Partition Indexes (NPIs) processing

BMCSTATS uses a key caching algorithm when it processes indexes. Large multi-datasets (MUDS) Non Partition Indexes (NPIs) can consume large amounts of memory during the caching process. BMCSTATS provides storage relief during key column statistics collection for NPIs. You can use the following three BMCSTATS keywords to tune high storage-consuming objects:
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Used to</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPICACHEACTION N:</td>
<td>Activate or deactivate the key caching processing</td>
<td>&lt;NONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ NONE— No storage limit is enforced for multi-dataset non-partitioned index key caching. The values that NPICACHESTHRESH and NPICACHEDSNUM specify are ignored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ LIMIT—For multi-dataset non-partitioned indexes, cache storage below the bar is limited to the value that NPICACHESTHRESH sets. When the storage limit is exceeded, key caching stops.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ ATBCACHE— For multi-dataset non-partitioned indexes, cache storage below the bar is limited to the value that NPICACHESTHRESH sets. When the storage limit is exceeded:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— The key cache is moved above the 2GB bar.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Caching continues.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Cache storage held below the bar is released.</td>
</tr>
<tr>
<td>NPICACHESTHRESH</td>
<td>Specify the amount of storage in megabytes that can be used below the</td>
<td>&lt;1–500&gt; 100</td>
</tr>
<tr>
<td></td>
<td>2GB bar before the action that NPICACHEACTION N specifies takes place.</td>
<td></td>
</tr>
<tr>
<td>NPICACHEDSNUM M</td>
<td>Specify the number of datasets required in a multi-dataset non-partitioned</td>
<td>&lt;2–4096&gt; 25</td>
</tr>
<tr>
<td></td>
<td>index before the values that NPICACHEACTION N and NPICACHESTHRESH specify are observed.</td>
<td></td>
</tr>
</tbody>
</table>

Note

If NPICACHEACTION=NONE, NPICACHEDSNUM and NPICACHESTHRESH are ignored.
To set these keywords globally for all BMCSTATS NPI processing, edit the Default Options module (DOPTs). To override the default settings for processing selected jobs, use the BMCSTATS syntax options.

**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:

```
Example
TSSAMPLEPCT 25
IXSAMPLEPCT 50
```

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 `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 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).

Using an index that has fewer than 1000 pages

### 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 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, you can use SAMPLE T or SAMPLE I 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.
Sampling by using a seed value

BMCSTATS uses random pages to sample at the page level unless you use the REPEATABLE clause.

If you specify REPEATABLE with an integer, BMCSTATS uses the integer as the seed value for the sample. Consequently, the product samples the same pages, and you can perform the following operations:

- Rerun BMCSTATS with the same REPEATABLE value
- Produce statistics based on the same pages

If you do not specify REPEATABLE, BMCSTATS uses a random seed value for each run. Consequently, the product might produce different table space sampling statistics every time you run BMCSTATS.

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
1 On the DASD MANAGER PLUS Main Menu, select **Statistics** and press **Enter**.

The Collect and Manage Statistics panel appears.

```
DEAE ---------------------- Collect and Manage Statistics ---------------------
Command ===> 
```

DASD provides the ability to collect, copy and list statistics. Space Estimation is available when listing statistics.

Select an option. Then press Enter.
1. List BMC and DB2 Statistics
2. Manage existing BMCSTATS, BMCCPRS and BMCUPRS Service Actions
3. Create BMCSTATS Service Action to collect statistics
4. Create BMCCPRS Service Action to copy DB2 Catalog to BMC stats database
5. Create BMCUPRS Service Action to copy BMC Stats database to DB2 Catalog

2 On the Collect and Manage Statistics panel, select **List BMC and DB2 Statistics** and press **Enter**.

The Display DB2 Object Selection panel is displayed.

```
DECA -------------------------- DB2 Object Selection --------------------------
COMMAND ===> 
```

Specify the object type(s) to be included in a list, then press Enter.

Specify either an object name or a pattern.

```
Database . . . . QZUD44%                   (DB Name)
Tablespace . . .                             (DB.TS Name)
Table. . . . . .                              (Own.TB Name)
Index. . . . . .                              (Own.IX Name)
Stogroup . . . .                             (SG Name)
Volume . . . . .                             (VL Name)
```

You can specify the objects to include in a list on the DB2 Object Selection panel. You initiate most of the DASD MANAGER PLUS statistics display functions from the Display DB2 Object List. You can run many of the statistics functions interactively or in batch.

### Displaying DB2 object lists

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

---

Chapter 6  Collecting and managing statistics  335
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 that a database uses, 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** and press **Enter**.
   
   The Collect and Manage Statistics panel is displayed.

2. On the Collect and Manage Statistics panel, select **List BMC and DB2 Statistics** and press **Enter**.
   
   The Display DB2 Object List panel is displayed. (For an example of the panel, see Accessing the statistics display features on page 334.)

3. To specify the 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 (Figure 64 on page 337) 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 64: Display DB2 Object List panel**

<table>
<thead>
<tr>
<th>Act Source</th>
<th>Status</th>
<th>Object-Type</th>
<th>Object Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>.</td>
<td>.</td>
<td>QZUD44</td>
</tr>
<tr>
<td>TS</td>
<td>.</td>
<td>.</td>
<td>QZUD44</td>
</tr>
<tr>
<td>TB</td>
<td>.</td>
<td>.</td>
<td>.QZUT01_D44S01</td>
</tr>
<tr>
<td>IXC</td>
<td>QZU</td>
<td>.</td>
<td>.QZUX01_D44S01T01</td>
</tr>
<tr>
<td>IX</td>
<td>QZU</td>
<td>.</td>
<td>.QZUX02_D44S01T01</td>
</tr>
<tr>
<td>TS</td>
<td>.</td>
<td>.</td>
<td>.QZUS044</td>
</tr>
<tr>
<td>TB</td>
<td>.</td>
<td>.</td>
<td>.QZUT01_D44S02</td>
</tr>
<tr>
<td>IXC</td>
<td>QZU</td>
<td>.</td>
<td>.QZUX01_D44S02T01</td>
</tr>
<tr>
<td>IX</td>
<td>QZU</td>
<td>.</td>
<td>.QZUX02_D44S02T01</td>
</tr>
</tbody>
</table>

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 49 on page 337 shows.

**Table 49: 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>

The following table shows possible values in the **Status** field:

**Table 50: 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 4 on page 337.)</td>
</tr>
</tbody>
</table>

4 In the **Act** field, specify a statistics operation for an object.

**Table 51 on page 338** shows valid commands.
Table 51: 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

1. On the DASD MANAGER PLUS Main Menu, select **Statistics**

   The Collect and Manage Statistics panel is displayed.

2. On the Collect and Manage Statistics panel, select **List BMC and DB2 Statistics** and press **Enter**.

   The Display DB2 Object List panel is displayed. (For an example of the panel, see Accessing the statistics display features on page 334.)

3. On the DB2 Object Selection panel, specify the databases containing the data sets that you want to list.

   You can type the name of a single database or a wildcard pattern.

   The Display DB2 Object List panel appears. This panel shows all objects from the DB2 subsystem that match the information that you specified on the DB2 Object Selection panel.

4. Create a list of data sets for a database by typing **I** in the **Act** field next to a database name and pressing **Enter**.
If you select more than one database at a time, each subsequent database appears when you press **END**.

The DB2 Data Set List panel displays all data sets that the specified database uses:

```
Database Name: QZUD44
Data Set Name                                 Trks Ext AU  Pqty Sqty Volume
---------------------------------------------------------------------------
DEBACAT.DSNDBD.QZUD44.QZUS0144.I0001.A001       130  11 T    30   10 OAQ0003
DEBACAT.DSNDBD.QZUD44.QZUS0144.I0001.A002        30   1 C     2    1 OAQ0003
DEBACAT.DSNDBD.QZUD44.QZUS0144.I0001.A003        30   1 T    30   10 OAQ0003
DEBACAT.DSNDBD.QZUD44.QZUS0144.I0001.A004        60   3 C     2    1 OAQ0003
DEBACAT.DSNDBD.QZUD44.QZUS0244.I0001.A001       100  10              OAQ020
DEBACAT.DSNDBD.QZUD44.QZUS0244.I0001.A001        30   1 T    30   10 OAQ0003
DEBACAT.DSNDBD.QZUD44.QZUS0244.I0001.A002        30   1 C     2    1 OAQ0003
DEBACAT.DSNDBD.QZUD44.QZUS0244.I0001.A003        30   1 T    30   10 OAQ0003
```

5 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.
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. “Analyzing statistical trends” on page 379 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 “Analyzing statistical trends” on page 379).

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 336.

The Display DB2 Object List panel 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

The Tablespace Statistics Browse panel displays the DB2 catalog and BMC Software statistics for the specified table space.

After viewing the table space statistics in the Tablespace Statistics Browse panel (Figure 66 on page 342), 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 66: Tablespace Statistics Browse panel

| ASUCTSPB ------------- Tablespace Statistics Browse -------------------------- |
|-----------------|------------------|------------------|
| COMMAND ====>   | Tablespace Name: QZUDAC.QZUS01AC | Type : Partitions: 4 |
|                 |                  | Clone : N Tables. : 1 |
|                 |                  | MaxParts: 0 Segsize : 0 |
|                 |                  | Member Cluster: N Implicit : N |
| |----------------------------------------------------|
| |----------------------------------------------------|
| | |              DB2 Catalog |      BMCSTATS      |
| | | yyyy-11-15-10.47 | yyyy-12-23-12.54 |
| |----------------------------------------------------|
| Select additional panels to display and press Enter. |
| Display Table statistics? | Display Partition Statistics? |

To display updatable table statistics from the Display Table Statistics? field

1 In the Display Table Statistics? field, type S and press Enter.

To display table statistics

1 Use one of the following methods:

- In the Display Table Statistics? field, type S and press Enter.
- On the COMMAND line, type T and press Enter.
From the Object Selection List panel, type B in the Act field next to the table name.

**To display table partition statistics**

1. You can use either of the following methods:
   - In the Display Partition Statistics? field, type S and press Enter.
   - On the COMMAND line, type P and press Enter.

**DB2 table statistics**

When you select Display Table Statistics on the Tablespace Statistics Browse panel, if the table space contains more than one table, the Table Selection panel is displayed:

```
COMMAND ===>
Type S to select an entry and press Enter.

Act Owner    Table Name                           Card   Npages Pages   Type
-------------------------------------------------------------------------------
QZU      QZUT01_DA1S02                        2036        0  100     T
QZU      QZUT02_DA1S02                           0        0    0     T

To select a table from the list in the Table Selection panel

1. 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:

```
<table>
<thead>
<tr>
<th>DB2 Catalog</th>
<th>BMCSTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card . . .</td>
<td>65000</td>
</tr>
<tr>
<td>Npages . . .</td>
<td>21669</td>
</tr>
<tr>
<td>PctPages</td>
<td>93</td>
</tr>
<tr>
<td>PctRowComp</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2010-11-15-10.47</td>
</tr>
</tbody>
</table>
```

Select additional panels to display and press Enter.

Display Column statistics?  Display Index statistics?
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

You can select **Display Partition Statistics** on the Tablespace Statistics Browse panel.

If more than one partition exists in the table space, the Table Statistics Browse panel (Figure 67 on page 344) opens, allowing you to choose the partition you need to scan for statistics.

**Figure 67: Tablespace Partition Selection panel**

```plaintext
<table>
<thead>
<tr>
<th>Act</th>
<th>Part</th>
<th>Nactive</th>
<th>Card</th>
<th>Npages</th>
<th>Pages</th>
<th>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></td>
</tr>
<tr>
<td>4</td>
<td>360</td>
<td>1667</td>
<td>105</td>
<td>29</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
```

**To select a table from the list**

1. Type `S` in the **Act** field beside the table space partition name and press **Enter**.
The Figure 68 on page 345 displays the DB2 catalog statistics for the partition.

**Figure 68: Tablespace Partition Table Statistics Browse panel**

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.

**DB2 column statistics**

The Column Selection panel lists the columns in the specified table. The 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.

---

Chapter 6  Collecting and managing statistics  345
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:
   
   ```plaintext
   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:</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>91</td>
</tr>
<tr>
<td>High2key . .</td>
<td>WEST ORA</td>
</tr>
<tr>
<td></td>
<td>ECEE4DDC</td>
</tr>
<tr>
<td></td>
<td>65230691</td>
</tr>
<tr>
<td>Low2key . .</td>
<td>ALLENTOW</td>
</tr>
<tr>
<td></td>
<td>CDDCDEDE</td>
</tr>
<tr>
<td></td>
<td>13355366</td>
</tr>
</tbody>
</table>
   ``

   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.

   **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

This topic 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 69: Index Statistics Browse panel**

ASUCIXPB ---------------- Index Statistics Browse ----------------------------

Index Name: QZU.QZUX01_D09S01T01                       DB Name  : QCHD09
Table Name: QZU.QZUT01_D09S01
1stKeyCol : COLUMN_1
Compress  : N                                  IX Extension Type:
Hash . . .: N                                  Sparse . . . . : N
More:     +

----------------------------------------------------
|              DB2 Catalog     |     BMCSTATS      |
| FirstKeyCard:         -1     |             0     |
| FullKeyCard :         -1     |             0     |
| Nleaf . . . :      4295M     |             0     |
| Nlevels . . :         -1     |             0     |
| ClusterRatio:          0     |             0     |
| DataRepeat  :         -1     |            -1     |
| High2key  . :                |                   |
|                 44444444     |      44444444     |
|                 00000000     |      00000000     |
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.SYSSCOLUMNS 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**

1. In the **Display Partition Statistics?** field, type **S** and press **Enter**.

Typing **P** on the **COMMAND** line produces the same result.
DB2 index partition statistics

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.

<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

1. Type S in the Act field beside the index partition name and press Enter.

The Index Partition Statistics Browse panel (Figure 70 on page 348) appears, displaying the DB2 catalog statistics for the partition.

Figure 70: Index Partition Statistics Browse panel

The product obtains the DB2 catalog statistics from the SYSIBM.SYSINDEXSTATS 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

You can update the DB2 catalog with the most recent BMCSTATS results from the DASD MANAGER PLUS database or other values that you choose.
1 Create a DB2 object list as described in “Creating a DB2 object list” on page 336.

**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 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

This topic shows a sample Tablespace Statistics Update panel, which you can use to compare DB2 catalog and BMC Software statistics.

The Tablespace Statistics Update panel (Figure 71 on page 350) displays 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.

**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.
To update table space statistics

1 On the Tablespace Statistics Update panel, type new values for the appropriate fields and press Enter.

**Figure 71: Tablespace Statistics Update panel**

![Figure 71: 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**

This topic 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.
**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.

---

**To update table statistics**

1. On the Table Statistics Update panel, type new values for the appropriate fields and press Enter. The Table Statistics Update panel appears.

   
   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>Last Update 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**

This topic 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.
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.

---

**To update table space partition table statistics**

1. Type new values for the appropriate fields and press **Enter**. The Table Space Partition Table Statistics Update panel appears:

   DEBA --------- Tablespace Partition Table Statistics Update -----  
   COMMAND ===>  
   Type data and press Enter.  
   Table Name . . : ZZU.ZZUT01_D11S01  
   Tablespace Name: ZZUD11.ZZUS0111  
   Partition . . : 4  
   
<table>
<thead>
<tr>
<th>New Value</th>
<th>DB2 Catalog</th>
<th>BMCSTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card . . . : 19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>Npages . . : 2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>PctPages . . : 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Nactive . . : 360</td>
<td>360</td>
<td>360</td>
</tr>
<tr>
<td>PctRowComp . : 0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

   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**

Use this procedure to update statistics 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.

---

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.
To update column statistics

1. In the Column Statistics Update panel, type a value in the **New Value** column beside the appropriate fields and press **Enter**.

   **DECA ---------------- Column Statistics Update ----------------**

   **COMMAND ==>>**

   **Type data and press Enter.**

   **Column Name:** COLUMN_1  
   **Table Name:** QZU.QZUT01_DAIS03  
   **Partition:** 4

<table>
<thead>
<tr>
<th>New Value</th>
<th>DB2 Catalog</th>
<th>BMCSTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card . . . 1667</td>
<td>1667</td>
<td>1667</td>
</tr>
<tr>
<td>High2key . 0.6....</td>
<td>0.6....</td>
<td>0.6</td>
</tr>
<tr>
<td>81CA2222</td>
<td>81CA2222</td>
<td>81CA0000</td>
</tr>
<tr>
<td>0BEE0000</td>
<td>0BEE0000</td>
<td>0BEE0000</td>
</tr>
<tr>
<td>Low2key . ø.ó....</td>
<td>ø.ó....</td>
<td>ø.ó</td>
</tr>
<tr>
<td>81BD2222</td>
<td>81BD2222</td>
<td>81BD0000</td>
</tr>
<tr>
<td>0BEF0000</td>
<td>0BEF0000</td>
<td>0BEF0000</td>
</tr>
<tr>
<td>Highkey . ø.ó®....</td>
<td>ø.ó®....</td>
<td>ø.ó®</td>
</tr>
<tr>
<td>81CA2222</td>
<td>81CA2222</td>
<td>81CA0000</td>
</tr>
<tr>
<td>0BEF0000</td>
<td>0BEF0000</td>
<td>0BEF0000</td>
</tr>
<tr>
<td>Lowkey . ø.∞û....</td>
<td>ø.∞û....</td>
<td>ø.∞û</td>
</tr>
<tr>
<td>81BD2222</td>
<td>81BD2222</td>
<td>81BD0000</td>
</tr>
<tr>
<td>0BF80000</td>
<td>0BF80000</td>
<td>0BF80000</td>
</tr>
<tr>
<td>yyyy-11-17-02.18</td>
<td>yyyy-11-09-10.25</td>
<td></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.

   - 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**

Use this procedure to update the index statistics. The Index Statistics Update panel 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.

*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.

**To update index statistics**

1. Type a value in the **New Value** column beside the appropriate fields.

   **Figure 72: Index Statistics Update panel**

   ![Index Statistics Update panel]

   - **ASUCIXPA**
   - **COMMAND ====>** Type data and press Enter.
   - **Table Name:** QZU.QZUT01_D09S01
   - **DB Name:** QCHD09
   - **1stKeyCol:** COLUMN_1
   - **Compress:** N
   - **Hash ...:** N
   - **IX Extension Type:** Hash
   - **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.

   **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

Use this procedure to update index partition statistics. The Index Partition Statistics Update pane displays the DB2 catalog and BMC Software statistics for the index partition that you specified.

Access this panel from the Index Statistics Update panel. 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.

**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.

#### To update index partition statistics

1. Type a value in the **New Value** column beside the appropriate fields.

   **Figure 73: Index Partition Statistics Update panel**

   ASUCIPPA ----------------- Index Partition Statistics Update-----------------
   COMMAND ===>Type data and press Enter.
   Index Name: QZU.QZUX01_DA1S05T01
   Partition : 1

<table>
<thead>
<tr>
<th></th>
<th>New Value</th>
<th>DB2 Catalog</th>
<th>BMCSTATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstKeyCard</td>
<td>10719</td>
<td>10719</td>
<td>10719</td>
</tr>
<tr>
<td>FullKeyCard</td>
<td>10719</td>
<td>10719</td>
<td>10719</td>
</tr>
<tr>
<td>Nleaf</td>
<td>62</td>
<td>62</td>
<td>62</td>
</tr>
<tr>
<td>Nlevels</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>ClusterRatio</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DataRepeat</td>
<td>190</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>Keycount</td>
<td>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>
<td>2009-10-29-02.26</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

If you want to run BMCSTATS, BMCUPRS, and BMCCPRS, you must have one of the types of authority shown in the following table.

If using DB2 security exits for authorization, the programs invoke the exit to verify access.

Table 52: Authorization requirements

<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.SYSDMAUTH = G or Y</td>
</tr>
<tr>
<td>SQLADM</td>
<td>SYSUSERAUTH.SYSSQLADMAUTH = 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 *BMC Products and Solutions for DB2 Customization Guide*.

### Updating DB2 columns by DASD MANAGER PLUS

The following tables 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.
Table 53: Table columns that are updated in the DB2 catalog

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIBM.SYSCOLDIST</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>LOWVALUE</td>
</tr>
<tr>
<td></td>
<td>NAME</td>
</tr>
<tr>
<td></td>
<td>NUMCOLUMNS</td>
</tr>
<tr>
<td></td>
<td>QUANTILENO</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td></td>
<td>TBOWNER</td>
</tr>
<tr>
<td></td>
<td>TBNAME</td>
</tr>
<tr>
<td></td>
<td>TYPE</td>
</tr>
<tr>
<td>SYSIBM.SYSCOLSTATS</td>
<td>COLCARD</td>
</tr>
<tr>
<td></td>
<td>HIGHKEY</td>
</tr>
<tr>
<td></td>
<td>HIGH2KEY</td>
</tr>
<tr>
<td></td>
<td>LOWKEY</td>
</tr>
<tr>
<td></td>
<td>LOW2KEY</td>
</tr>
<tr>
<td></td>
<td>STATS_FORMAT</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td>SYSIBM.SYSCOLUMNS</td>
<td>COLCARD</td>
</tr>
<tr>
<td></td>
<td>COLCARDF</td>
</tr>
<tr>
<td></td>
<td>HIGH2KEY</td>
</tr>
<tr>
<td></td>
<td>LOW2KEY</td>
</tr>
<tr>
<td></td>
<td>STATS_FORMAT</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td>Table</td>
<td>Columns</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>SYSIBM.SYSINDEXES</td>
<td>AVGKEYLEN, CLUSTERED, CLUSTERRATIO, CLUSTERRATIOF, DATAREPEATFACTORF, FIRSTKEYCARD, FIRSTKEYCARDF, FULLKEYCARD, FULLKEYCARDF, NLEAF, NLEVELS, SPACE, SPACEF, STATSTIME</td>
</tr>
<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>Table</td>
<td>Columns</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>SYSIBM.SYSKEYTGTDIST</td>
<td>CARDF</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>IXSCHEMA</td>
</tr>
<tr>
<td></td>
<td>IXNAME</td>
</tr>
<tr>
<td></td>
<td>KEYGROUPKEYNO</td>
</tr>
<tr>
<td></td>
<td>KEYSEQ</td>
</tr>
<tr>
<td></td>
<td>KEYVALUE</td>
</tr>
<tr>
<td></td>
<td>LOWVALUE</td>
</tr>
<tr>
<td></td>
<td>NUMKEYS</td>
</tr>
<tr>
<td></td>
<td>QUANTILENO</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td></td>
<td>TYPE</td>
</tr>
<tr>
<td>SYSIBM.SYSTABLES</td>
<td>AVGROWLEN</td>
</tr>
<tr>
<td></td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>CARDF</td>
</tr>
<tr>
<td></td>
<td>NPAGES</td>
</tr>
<tr>
<td></td>
<td>NPAGESF</td>
</tr>
<tr>
<td></td>
<td>PCTPAGES</td>
</tr>
<tr>
<td></td>
<td>PCTROWCOMP</td>
</tr>
<tr>
<td></td>
<td>SPACEF</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<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>PERCACTIVE</td>
</tr>
<tr>
<td></td>
<td>PERCDROP</td>
</tr>
<tr>
<td></td>
<td>PQTY</td>
</tr>
<tr>
<td></td>
<td>SECQTYI</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>
</tbody>
</table>
### Table 54: Partitioned Table Space/Index Space Columns Updated in the DB2 Catalog

<table>
<thead>
<tr>
<th>Partitioned Table/Index Space</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIBM.SYSTABLESPACE</td>
<td>AVGROWLEN</td>
</tr>
<tr>
<td>SYSIBM.SYSCOLDISTSTATS</td>
<td>CARDF</td>
</tr>
<tr>
<td>SYSIBM.SYSINDEXSTATS</td>
<td>CLUSTERRATIO</td>
</tr>
<tr>
<td>Partitioned Table/Index Space</td>
<td>Columns</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>----------------------------------------------</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>
<tr>
<td>SYSIBM.SYSKEYTGTSTATS</td>
<td>CARDF</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>IXSCHEMA</td>
</tr>
<tr>
<td></td>
<td>IXNAME</td>
</tr>
<tr>
<td></td>
<td>KEYGROUPKEYNO</td>
</tr>
<tr>
<td></td>
<td>KEYSEQ</td>
</tr>
<tr>
<td></td>
<td>KEYVALUE</td>
</tr>
<tr>
<td></td>
<td>LOWVALUE</td>
</tr>
<tr>
<td></td>
<td>NUMKEYS</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>TYPE</td>
</tr>
<tr>
<td>SYSIBM.SYSLOBSTATS</td>
<td>AVGSIZE</td>
</tr>
<tr>
<td></td>
<td>FREESPACE</td>
</tr>
<tr>
<td></td>
<td>ORGRATIO</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td>SYSIBM.SYSTABSTATS</td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>CARDF</td>
</tr>
<tr>
<td></td>
<td>NACTIVE</td>
</tr>
<tr>
<td></td>
<td>NPAGES</td>
</tr>
<tr>
<td></td>
<td>PCTPAGES</td>
</tr>
<tr>
<td></td>
<td>PCTROWCOMP</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
</tbody>
</table>
Updating DB2 real-time statistics

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 55 on page 363 lists the columns in table SYSIBM.SYSTABLESPACESTATS that BMCSTATS and BMCUPRS reset for each partition and the reset value.

Table 55: Statistics reset in table SYSIBM.SYSTABLESPACESTATS

<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>
<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 56 on page 363 lists the columns in table SYSIBM.SYSINDEXSPACESTATS that BMCSTATS and BMCUPRS reset for each partition and the reset value.

Table 56: 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>STATSLASTTIME</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 370.)

---

**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 a Service Action” on page 241.

   A user-defined Service Action is a group of services (utilities) within a DASD MANAGER PLUS job. This Service Action creates a worklist to run the BMCSTATS utility. For more information about actions, see “Maintaining and generating Service Actions” on page 223.

   After you specify the action and press Enter, the Action List appears.

2. Add the BMCSTATS service to the Service Action, as follows:

   **Note**

   Use the Edit Action Services panel to select and edit services and service syntax for the action. (For more information, see the link below to the relevant topic.)

   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 `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)
As an alternative method, specify the object name and object type by using an object set. For more information, see the link below to the relevant topic.

e To insert a service, type I in the Act field, type % in the Service field, and press Enter. The Service List panel displays a list of the services that you can select.

f Select the BMCSTATS utility as a service in your new Service Action by typing S in the Act field beside the service name and pressing Enter.

g 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 typing I.

Figure 74 on page 365 appears. 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 74: BMCSTATS panel

<table>
<thead>
<tr>
<th>Command</th>
<th>BMCSTATS</th>
<th>Row 1 to 26 of 55</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEAE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Syntax: BMCSTATS.I431937S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type Service Syntax options. Then press End. More: +</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABLE . . . . * N (Y/N/S Y=ALL Tables, N=No Tables, S=Select Tables)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEX . . . . Y (Y/N Collect column statistics on all indexes)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPACEONLY . . . N (Y/N Collect space information only)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distribution Stats:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMCOLS . . . 1 (1-64 Max index key columns to concatenate)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Histogram Stats:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IXNUMQUANTILES (1-100 Number of quantiles to collect)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMCOLS . . . (1-64 Number of columns for quantiles)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Space Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FREQVAL . . . (Y/N Collect frequent value statistics)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table and Index Space Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COUNT . . . . 10 (1-300 Max number of frequent values to collect)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5 Set the BMCSTATS utility parameters on the main panel (or accept the defaults).

6 When you return to the service syntax list, select the syntax by typing S in the Act field.

7 Press END to return to the Edit Action Services panel.

The Edit Action Services panel reappears and shows the BMCSTATS utility as a service (Figure 75 on page 366).

**Figure 75: Edit Action Services panel for BMCSTATS**

<table>
<thead>
<tr>
<th>Command</th>
<th>Edit Action Services</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action: PAYROLL.DEMO03</td>
<td></td>
<td>CSR</td>
</tr>
</tbody>
</table>

Listed is the ordered set of Services contained in the Action. Select one or more options. Then press Enter.

D =Delete  I =Insert  L =Like  A =After  B =Before  M =Move  P =Properties  S =Syntaxes  O =Syntax Options  V =View Syntax  U =Use default syntax  OS =Object Set

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 Service field of the blank line and press Enter.

c Select a service by typing S to the left of the service and pressing Enter.

The Edit Action Services panel is displayed. (For more information about editing action services, For more information, see the link below to the relevant topic.

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 Job Generation panel appears. The Action Job 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 Service Action. The DO_WORKKIDS 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 Service Actions” on page 248.

11 Depending on the options that you specified on the second Action Job Generation panel, choose one of the following actions:

- To override the Product Options File (POF) settings, select Override POF Values.
- To build the worklist, select Build Worklist.
To review the built worklist, select **Edit Worklist**.
For detailed information about the worklist commands, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

To build the JCL, select **Build JCL**.

To review the built JCL, select **Edit JCL**.
When you review the built JCL, you can submit it for execution by typing **SUB** in the Command line.

To run the job automatically after building the worklist and JCL, select **Submit JCL**.
As an alternate method (if you selected **Edit JCL**) type **SUB** on the ISPF Edit panel.
Asterisks (*) in the **Build Worklist** and **Build JCL** fields indicate that the product has generated them.

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 76 on page 368 displays the JCL for a sample BMCSTATS utility job.

**Figure 76: JCL for BMCSTATS utility job**

```
//RDAJXN4U JOB (5213),"UTILITY-JENBMCS",  // CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),  // NOTIFY=RDAJXN4  //***************************************************  //* CREATED BY : RDAJXN4  //* TIMESTAMP : .17.27.54  //* ENVIRONMENT: ISPF 7.1MVS TSO  //* RELEASE :  V  //* DB2 VERSION:  //***************************************************  //** DASD MANAGER WORKLIST EXECUTION  //** -----STEP1 EXEC PGM=AEXEMAIN,REGION=0M,  //** PARM="DSB15ECA"  //** STEPLIB DD DISP=SHR,DSN=DB2.MSTRPLAN.LOAD  //** DD DISP=SHR,DSN=SYS3.DEBA.DSNEXIT  //** DD DISP=SHR,DSN=SYS2.DB2V8IM.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  //** ***************************************************```
Figure 77 on page 369 provides an example of a BMCSTATS worklist.

**Figure 77: BMCSTATS worklist**

```
- TIME 000000 '-06-24-17.41.50.00005'
*   GENERIC BMCSTATS
- SSID 000001 DEBA
- WKID 000002 JENBMCS
- SYNC 000003

- BMCU 000004 ASUSMAIN
BMCSTATS TABLESPACE JEND30.%
TABLE (ALL)
EVENTS N
TASKS 2
KEYCARD Y
NUMCOLS 2
COUNT 5
```

********************* Top of Data *********************
- TIME 000000 '-06-24-17.41.50.00005'
*   GENERIC BMCSTATS
- SSID 000001 DEBA
- WKID 000002 JENBMCS
- SYNC 000003

- BMCU 000004 ASUSMAIN
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 a Service Action” on page 241.

   A user-defined Service Action is a group of services (utilities) within a DASD MANAGER PLUS job. This Service Action creates a worklist to run the BMCUPRS utility. For more information about actions, see “Maintaining and generating Service Actions” on page 223.

   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 Service 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)
— TT (table space set)
Alternatively, specify the object name and object type by using an object set.

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 BMCUPRS parameters panel for a table space (Figure 78 on page 371).
Optionally, you can add new syntax by typing I.

Figure 78: BMCUPRS parameters panel for a table space

ASUDUM                BMCUPRS TABLESPACE QZUD40.%             Row 1 to 20 of 20
Command ===>                                                  Scroll ===> CSR

Service Syntax: BMCUPRS.DEMO

Enter data, then press end.                                         More:
-------------------------- What to Update -------------------------------------
Tablespace Options:
TABLE . . . . . * N      (Y/N/S Y=ALL Tables, N=No Tables, S=Select Tables)
INDEX . . . . . Y        (Y/N Update column stats on all indexes)

-------------------------- DB2 Catalog Update Options ------------------------
UPDATEDB2 . . . . A        (A/P/S A=All, P=Accesspath, S=Space)
HISTORY . . . . N          (N/A/P/S N=None, A=All, P=Accesspath, S=Space)
DELETEHISTAGE 32767       (0-32767 Days for deleting history table entries)
OMITCARDO . . . . N        (Y/N - Bypass catalog update if cardinality is 0)
RESETRTS . . . . N         (Y/N - Reset realtime stats in the catalog)

-------------------------- 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)
**********************************************************************

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.
- Type N if you don’t want 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 does 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 Job Generation panel by typing G beside an action on the Action List panel and pressing Enter.

The Action Job Generation panel appears.

7 Complete the fields on the Action Job Generation panel, as follows:

a In the Build Worklist, Build JCL, and Submit Job fields type S.

b (optional) In the Edit Worklist and Edit JCL fields, type S to review the built worklist and built JCL.

c Press Enter.

8 Depending on the options that you specify on the Action Job Generation panel, complete the following tasks:

■ To override the Product Options File (POF) settings, select Override POF Values.

■ To build the worklist, select Build Worklist.

■ To review the built worklist, select Edit Worklist.

For detailed information about the worklist commands, see the DASD MANAGER PLUS for DB2 Reference Manual.

■ To build the JCL, select Build JCL.

■ To review the built JCL, select Edit JCL.

When you review the built JCL, you can submit it for execution by typing SUB in the Command line.

■ To run the job automatically after building the worklist and JCL, select Submit JCL.

As an alternate method (if you selected Edit JCL) type SUB on the ISPF Edit panel.

Asterisks (*) in the Build Worklist and Build JCL fields indicate that the product has generated them.

9 Press Enter to confirm that you want to submit the BMCUPRS 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
- Tablespaces with hash organizations

To update DASD MANAGER PLUS statistics by using BMCCPRS

1. Create an action as described in “Creating a Service Action” on page 241.

   A user-defined Service 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 “Maintaining and generating Service Actions” on page 223.

   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 Nam** 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).

   c. Press **Enter** to display the Service Syntax List panel.
d Type **E** in the **Act field** next to the syntax and press **Enter** to display the BMCCPRS panel (Figure 79 on page 375). Optionally, you can add new syntax by typing **I**.

**Figure 79: BMCCPRS Parameters panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>BMCCPRS TABLESPACE QZUD40.%</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Service Syntax: BMCCPRS.MVSTAD1-BMCCPRS

Type Service Syntax options. Then press End.

-------------------------- What to Migrate --------------------------
TABLE .* .* .* N       (Y/N/S Y=All Tables, N=No Tables, S=Select Tables)
INDEX .* .* .* N        (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.
- 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 Job Generation panel by typing **G** next to an action on the Action List panel and pressing **Enter**.

The Action Job Generation panel appears.

6 Complete the fields on the Action Job Generation panel and press **Enter**.
In the **Build Worklist**, **Build JCL**, and **Submit JCL** fields, type **S**.

**(optional)** In the **Edit Worklist** and **Edit JCL** fields, type **S** to review the built worklist and built JCL.

Depending on the options that you specified on the Action Job Generation panel, complete the following tasks:

- To override the Product Options File (POF) settings, select **Override POF Values**.
- To build the worklist, select **Build Worklist**.
- To review the built worklist, select **Edit Worklist**. For detailed information about the worklist commands, see the *DASD MANAGER PLUS for DB2 Reference Manual*.
- To build the JCL, select **Build JCL**.
- To review the built JCL, select **Edit JCL**. When you review the built JCL, you can submit it for execution by typing **SUB** in the **Command** line.
- To run the job automatically after building the worklist and JCL, select **Submit JCL**. As an alternate method (if you selected **Edit JCL**) type **SUB** on the ISPF Edit panel.
  - Asterisks (*) in the **Build Worklist** and **Build JCL** fields indicate that the product has 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 BMC Statistical Repository 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 described in “Deleting BMC Software statistics” on page 377.

- Manually alter and run the sample SQL that BMC provides in the *HLQ.CNTL (ASURSDEL)* data set.

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 336.

   The Display DB2 Object List appears.

2. To display the BMC Software Statistics for the object, type D in the *Act* field beside the object name and press Enter.

   See “Analyzing statistical trends” on page 379.

   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.
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.

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 explains how to analyze trends in statistics that have been gathered using BMCSTATS or RUNSTATS.

Before you begin analyzing statistical trends

Before you can analyze statistical trends, you must collect statistics from BMCSTATS, RUNSTATS, or some other source.

See “Collecting and managing statistics” on page 313, which 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. The query 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.

Analysis of BMC Software statistics

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 what actions to take and when to take them.

DASD MANAGER PLUS timestamps BMCSTATS statistics and stores them in a separate database, and 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 that the BMCSTATS utility was run. You can also browse BMC Software statistics beside DB2 catalog statistics to compare the values. (See “Browsing and updating statistics” on page...
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 hard-copy 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.

Note
For more information see “Producing reports” on page 547.

Analyzing BMC Software statistics

Use this procedure to access the DASD MANAGER PLUS statistics display feature.

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

   The Collect and Manage Statistics panel is displayed.


   For an example of the panel see the figure in “Accessing the statistics display features” on page 334.

   The Display DB2 Object List panel is displayed.

3. On the DB2 Object Selection panel, specify the objects with stats that you want to analyze and press Enter.

   You can type the name of a single database or a wildcard pattern.

   The Display DB2 Object List panel displays all objects from the DB2 subsystem that match your specifications on the DB2 Object Selection panel.

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 “Collecting and managing statistics” on page 313.

**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 336.

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**

This topic shows a sample Tablespace Statistics panel, which is a high-level view of the statistics.

Figure 80 on page 381 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.

**Figure 80: Tablespace Statistics panel**

<table>
<thead>
<tr>
<th>Tablespace Name: QZUDAC.QZUS01AC</th>
<th>Partitions: 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace type:</td>
<td>Member Cluster: N</td>
</tr>
<tr>
<td>Implicit . . .: N</td>
<td>MaxParts: 0</td>
</tr>
<tr>
<td>Segsize : 0</td>
<td>Segsize : 0</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Active Pages : 23220</td>
<td>Prev Run %Change 23220 0</td>
</tr>
<tr>
<td>Space (Trks) : 1935</td>
<td>1935 0</td>
</tr>
<tr>
<td>Reorgspace : 1830</td>
<td>1830</td>
</tr>
<tr>
<td>2010-12-23-12.54</td>
<td>2010-12-23-12.17</td>
</tr>
</tbody>
</table>

Select additional panels to display and press Enter.

- Display Partition information?
- Display Table information?
- Graph Pages?
- Display Entry List?
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).

To display table space statistics

1. Perform any of the following actions:
   - 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 Tablespace 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 Tablespace Statistics List panel appears.

Tips for analyzing table space statistics

The following tips can help you analyze table space 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

This topic shows a sample Tablespace Partition Statistics panel, which is a more detailed view of the table space statistics.
Figure 81 on page 383 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 81: Tablespace Partition Statistics panel

<table>
<thead>
<tr>
<th>Tablespace Name: CCBPART.TSPART</th>
<th>Pctfree: 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition: 1</td>
<td>Freepage: 0</td>
</tr>
<tr>
<td>Stogroup: SYSDEFLT</td>
<td>Volume: DEV231</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Last Run</th>
<th>Prev Run</th>
<th>%Change</th>
<th>1st Run</th>
<th>%Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Pages:</td>
<td>540</td>
<td>540</td>
<td>0</td>
<td>540</td>
<td>0</td>
</tr>
<tr>
<td>Pctactive:</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Pctdrop:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Full Pages:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Dirty Pages:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Space (Trks):</td>
<td>45</td>
<td>45</td>
<td>0</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>Extents:</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Reorgspace:</td>
<td>30</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

- Analysis of BMC Software statistics

**Chapter 7  Analyzing statistical trends**

To display tablespace partition statistics

1. Perform any of the following actions:
   - In the **More Part Stats** field, type **S** to display additional tablespace 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 Tablespace Partition Statistics for dates other than those shown (or type L at the COMMAND line) and press Enter. The Tablespace Partition Statistics panel appears.

In the PageGroup Stats field, type S to display Tablespace 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.

Table statistics for a partitioned table space

The Additional Tablespace Partition Statistics panel displays the last, previous, and first statistics from the DASD MANAGER PLUS database for the specified table in a table space partition.

Figure 82: Additional Tablespace Partition 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>Cardinality</td>
<td>200</td>
<td>200</td>
<td>0</td>
<td>200</td>
</tr>
<tr>
<td>Pages</td>
<td>200</td>
<td>200</td>
<td>0</td>
<td>200</td>
</tr>
<tr>
<td>Farindref</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Nearindref</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PctPages</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>PctRowComp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Avg Row Len</td>
<td>203</td>
<td>203</td>
<td>203</td>
<td>203</td>
</tr>
<tr>
<td>PageSave</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Tips for analyzing table space partition statistics

The following tips can be helpful for analyzing space partition 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.

- If you want 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 that 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.

- If you want 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 space partition pagegroup statistics**

The figure below lists all page group statistics in the DASD MANAGER PLUS database for the specified table space partition.

---

**Note**

A zero in the **Partition** field indicates that the table space is not partitioned.

---

**Figure 83: Tablespace Partition Pagegroup Statistics panel**

<table>
<thead>
<tr>
<th>Begin Page</th>
<th>Card</th>
<th>Faroff</th>
<th>Nearoff</th>
<th>Dirty</th>
<th>Full</th>
<th>Act</th>
<th>Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1667</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

----------------------------- BOTTOM OF DATA -----------------------------
To enter page group statistics in the historical database

1. Access the Tablespace Partition Pagegroup Statistics panel through the Table Space Partition Statistics panel.

2. Run BMCSTATS with a nonzero value for the GROUPSIZE keyword.

Tips for partition pagegroup statistics analysis

The following tips can be helpful for analyzing partition pagegroup 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

The following example of a Table Statistics 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.

Figure 84: Table 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>Cardinality</td>
<td>2036</td>
<td>2036</td>
<td>0</td>
<td>2036</td>
</tr>
<tr>
<td>Pages</td>
<td>129</td>
<td>129</td>
<td>0</td>
<td>129</td>
</tr>
<tr>
<td>Indrefs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PctPages</td>
<td>8</td>
<td>99</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>PctRowComp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Avg Row Len</td>
<td>176</td>
<td>176</td>
<td>0</td>
<td>241</td>
</tr>
</tbody>
</table>

Select additional panels to display and press Enter.

Graph Cardinality?  
Display Column Information?
Graph Pages?  
Display Index Information?
Graph Row-Avg.?  
Display Entry List?
Graph PctPages?
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).

To display table statistics

1 Perform one of the following actions:
   - 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 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 to display the Column Statistics panel (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 display table statistics 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

The Column Statistics 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 380.

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 85: Column Statistics panel

<table>
<thead>
<tr>
<th>Coltype: INTEGER</th>
<th>Length : 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardinality</td>
<td>2036</td>
</tr>
<tr>
<td>Avg Len</td>
<td>4</td>
</tr>
<tr>
<td>Min Len</td>
<td>4</td>
</tr>
<tr>
<td>Max Len</td>
<td>4</td>
</tr>
<tr>
<td>#Nulls</td>
<td>0</td>
</tr>
<tr>
<td>High2key</td>
<td>81C0000</td>
</tr>
<tr>
<td>Low2key</td>
<td>81A0000</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.

- 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 column statistics analysis**

The following tips can help you analyze column 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

BMCSTATS automatically collects the ten most frequent occurrences of the values in a 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 86: Column Value Statistics panel

<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 column value statistics analysis

The following tips can help you analyze column value 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

The Index Statistics 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 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.

**Figure 87: Index Statistics panel**

![Index Statistics panel](image)

**Note**

If BMCSTATS encounters negative data points, it will not display them in the cardinality statistics, cluster ratio, or pages 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).

**To display index statistics**

1. You can perform any of the following tasks:

   - 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**.

   - 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 index statistics analysis**

The following tips can help you analyze index 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**

The Index Partition Statistics panel displays the last, previous, and first statistics found in the DASD MANAGER PLUS database for the specified index partition.
A zero in the **Partition** field indicates the index is not partitioned.

### Figure 88: Index Partition Statistics panel

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASUZIPPA</td>
<td>Index Partition Statistics</td>
</tr>
<tr>
<td>COMMAND</td>
<td></td>
</tr>
<tr>
<td>Stogroup</td>
<td>QZUALL</td>
</tr>
<tr>
<td>Index Name</td>
<td>QZU.QZUX01_DA1S05T01</td>
</tr>
<tr>
<td>Partition</td>
<td>1</td>
</tr>
<tr>
<td>Pctfree</td>
<td>0</td>
</tr>
<tr>
<td>Freepage</td>
<td>0</td>
</tr>
<tr>
<td>Volume</td>
<td>OAQ011</td>
</tr>
<tr>
<td>FirstKeyCard</td>
<td>10719</td>
</tr>
<tr>
<td>FullKeyCard</td>
<td>10719</td>
</tr>
<tr>
<td>Nleaf</td>
<td>62</td>
</tr>
<tr>
<td>ClusterRatio</td>
<td>99</td>
</tr>
<tr>
<td>DataRepeat</td>
<td>190</td>
</tr>
<tr>
<td>Nlevels</td>
<td>2</td>
</tr>
<tr>
<td>Keycount</td>
<td>32396</td>
</tr>
<tr>
<td>Faroffpos</td>
<td>21593</td>
</tr>
<tr>
<td>Nearoffpos</td>
<td>8970</td>
</tr>
<tr>
<td>LeafDist</td>
<td>1</td>
</tr>
<tr>
<td>Space (Trks)</td>
<td>10</td>
</tr>
</tbody>
</table>

Select additional panels to display and press Enter:
- Graph Cardinality?
- Graph Leafdist?
- Graph PageGroup?
- Graph Pages?
- List PageGroup?
- Graph Extents?
- Display Entry List?

### To display index partition statistics

1. 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 Index partition statistics analysis

The following tips can help you analyze Index partition 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

The Index partition pagegroup statistics 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 Tablespace Partition Statistics panel.

Figure 89: Index Partition Pagegroup Statistics panel

<table>
<thead>
<tr>
<th>Page</th>
<th>Card</th>
<th>FullKey</th>
<th>Faroff</th>
<th>Leafdist</th>
<th>Nearoff</th>
<th>#Leaf</th>
<th>Kbytes</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2036</td>
<td>2036</td>
<td>674</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Analysis of BMC Software statistics
Note
A zero in the **Partition** field indicates that the table space is not partitioned.

---

**Storage group statistics**

The Storage Group Statistics 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 336.

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**.

**Figure 90: Storage Group Statistics panel**

<table>
<thead>
<tr>
<th>Tracks  . . :</th>
<th>Last Run</th>
<th>Prev Run</th>
<th>%Change</th>
<th>1st Run</th>
<th>%Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>29126</td>
<td>29126</td>
<td>0</td>
<td>15750</td>
<td>84</td>
</tr>
<tr>
<td>TS Space</td>
<td>13742</td>
<td>13742</td>
<td>0</td>
<td>6908</td>
<td>98</td>
</tr>
<tr>
<td>TS Count</td>
<td>177</td>
<td>177</td>
<td>0</td>
<td>136</td>
<td>30</td>
</tr>
<tr>
<td>IX Space</td>
<td>15384</td>
<td>15384</td>
<td>0</td>
<td>8842</td>
<td>73</td>
</tr>
<tr>
<td>IX Count</td>
<td>369</td>
<td>369</td>
<td>0</td>
<td>242</td>
<td>52</td>
</tr>
</tbody>
</table>


Select additional panels to display and press Enter.

**To display storage group statistics**

1. Perform any of the following actions:

   - 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**.

**Tips for storage group statistics 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**

The Volume Statistics panel displays the last, previous, and first statistics in the DASD MANAGER PLUS database for the specified volume.

**Figure 91: Volume Statistics panel**

<table>
<thead>
<tr>
<th>Volume: AUS001</th>
<th>Unit: 3390</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Data Sets : 1959</td>
<td>1617</td>
</tr>
<tr>
<td>DB2 Tracks : 41830</td>
<td>36254</td>
</tr>
<tr>
<td>% DB2 Data Sets : 83</td>
<td>72</td>
</tr>
<tr>
<td>Total Data Sets : 1969</td>
<td>1627</td>
</tr>
<tr>
<td>% Volume Used : 99</td>
<td>88</td>
</tr>
<tr>
<td>DSCBS : 4499</td>
<td>4499</td>
</tr>
<tr>
<td>% DSCBS Used : 47</td>
<td>40</td>
</tr>
<tr>
<td>Free DSCBS : 2342</td>
<td>2690</td>
</tr>
<tr>
<td>Free Cyls : 0</td>
<td>358</td>
</tr>
<tr>
<td>Free Trks : 104</td>
<td>5680</td>
</tr>
<tr>
<td>Free Extents : 38</td>
<td>56</td>
</tr>
</tbody>
</table>

To display volume statistics

1. Perform any of the following actions:
   - 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 volume statistics analysis**

The Volume Statistics panel 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.

**Best practice**

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 400. You must have a graphics-capable terminal to display statistics graphs.

Table 57 on page 397 describes the graphs that you can display or print.

<table>
<thead>
<tr>
<th>Table 57: Statistics graphs summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Graph</strong></td>
</tr>
</tbody>
</table>
| Index Partition Cardinality         | Graphs the number of rows of the index partition in relation to the number of NEAROFFPOS and FAROFFPOS references over time (months)  
If NEAROFFPOS and FAROFFPOS increase in relation to the cardinality, the table becomes more disorganized, and you should consider a reorganization. |
| Index Partition Extents             | Graphs the number of extents in the partition over time (months)  
The maximum number that VSAM allows is 7257 per data set. |
<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 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) Optimum LEAFDIST is 100, indicating that all leaf pages are adjacent to each other. A value of 200 indicates 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) A large difference between allocated pages and active pages indicates that 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 causes 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 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 causes 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 An increasingly high percentage of indirect references indicates that 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>
<tr>
<td>Table Space Partition Cardinality</td>
<td>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.</td>
</tr>
<tr>
<td>Graph</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 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  
- 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 causes 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 causes 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 336.

   The DB2 Objection Selection List appears. From this panel, 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.

**Note**
You can also create a graph by typing its abbreviation at the COMMAND line of the statistics panel and pressing Enter.

The graph appears.

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.
You can specify graphic display and printing options to obtain the most desirable display or print. **Figure 92 on page 401** shows the components of a statistics graph for which you can set options.

**Figure 92: Statistics graph components**

1. 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 X and Y axes
7. Legend describing the plotted data (markers and lines)

**To set graphic display options**

1. Access the Graphics Options panel as follows:
On the DASD MANAGER PLUS Main Menu, select **User Options** and press **Enter**.

On the User Options panel, select **Graphic Display** and press **Enter**.

The Graphics Options panel appears:

**Figure 93: Graphics Options panel**

<table>
<thead>
<tr>
<th>Header Attributes</th>
<th>Legend Key Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color . . . 7</td>
<td>Color . . . 0 (0 - 8)</td>
</tr>
<tr>
<td>Char Size 125</td>
<td>Char Size 75 (75 - 200)</td>
</tr>
<tr>
<td>Char H/W 125</td>
<td>Char H/W 100 (75 - 200)</td>
</tr>
<tr>
<td>Markers . . MARK (MARK, NOMA)</td>
<td>Lines . . . LINE (LINE, NOLI)</td>
</tr>
<tr>
<td></td>
<td>Justify . . C (C, R, T, B)</td>
</tr>
<tr>
<td></td>
<td>Order . . KNOR (KNOR, KREV)</td>
</tr>
<tr>
<td></td>
<td>Box . . . KBOX (KBOX, NKBO)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X Title Attributes</th>
<th>Y Title Attributes</th>
<th>Color Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color . . . 7</td>
<td>Color . . . 7</td>
<td>0 - Default</td>
</tr>
<tr>
<td>Char Size 100</td>
<td>Char Size 100</td>
<td>1 - Blue</td>
</tr>
<tr>
<td>Char H/W 100</td>
<td>Char H/W 100</td>
<td>2 - Red</td>
</tr>
<tr>
<td>Axis Color 7</td>
<td>Axis Color 7</td>
<td>3 - Pink</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 - Green</td>
</tr>
<tr>
<td>Label Attributes</td>
<td>Box Attributes</td>
<td>5 - Turquoise</td>
</tr>
<tr>
<td>Color . . . 7</td>
<td>Color . . . 7</td>
<td>6 - Yellow</td>
</tr>
<tr>
<td>Char Size 100</td>
<td>Char Size 100</td>
<td>7 - Neutral</td>
</tr>
<tr>
<td>Char H/W 100</td>
<td>Char H/W 100</td>
<td>8 - Background</td>
</tr>
</tbody>
</table>

**Note**

To identify the graph components, see Figure 92 on page 401.

---

2 Specify the Header Attributes as follows to determine how to display graph headers:

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** as follows to determine how to display the legends:
a Specify the Color Key, Char Size, and Char H/W in the same way as in Step Setting graphic display options on page 402.

b In the Orient field, specify H for horizontal orientation or V for vertical orientation of the legend.

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 in the same way as you did in Step Setting graphic display options on page 402.
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** to determine the appearance of the axis 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 Printer Graphics Options panel (Figure 94 on page 404) appears.

**Figure 94: Printer Graphics Options panel**

<table>
<thead>
<tr>
<th>Device Name</th>
<th>L317F2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copies</td>
<td>1</td>
</tr>
<tr>
<td>Page Size</td>
<td></td>
</tr>
<tr>
<td>Rows</td>
<td>60</td>
</tr>
<tr>
<td>Columns</td>
<td>120</td>
</tr>
<tr>
<td>Print Only</td>
<td>N</td>
</tr>
</tbody>
</table>

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
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.

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. The default values are the most current values in the DASD MANAGER PLUS statistics tables for the object.

**Note**
Space estimation is not available for XML objects or for objects that have partial statistics.

You can edit some of these fields in the Tablespace Estimation panel (nonpartitioned) (Figure 95 on page 407) and then 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 95: Tablespace Estimation panel (nonpartitioned)

<table>
<thead>
<tr>
<th>DEAE</th>
<th>Tablespace Estimation</th>
<th>Row 1 to 1 of 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ===&gt;</td>
<td>SCROLL ===&gt; CSR</td>
<td></td>
</tr>
</tbody>
</table>

Type data and press Enter.

Tablespace Name QZUDA1.QZUS01A1  Type . . . (G, I or blank)
Pagesize . . . . 4 (4,8,16,32) Maxrows 255 (0-255)
Segsize . . . . 4 (0-64) Dssize . 0 ------ Estimated ------
Pages . . 130 MaxParts 0 Pages . . 130
Space (Trks) 90 Space (Trks) 11
Extents . . 1 Space (KB) 528
Priqty . . 90 Extents . . 1
Secqty . . 3 Data Sets . . . 1
Pctfree . . 0 (0-99) Freepage 0 (0-3)
Alloc Unit . . T (T or C) Compress (Y or blank)
Device Type . . 3390

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.

- 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.

The following information appears for partitioned table spaces:

- The Tablespace Estimation (Partitioned) panel shows currently used space, growth information, and the type of table space on the left and center of the panel.

**Figure 96: Tablespace Estimation (Partitioned) panel**

<table>
<thead>
<tr>
<th>Part</th>
<th>#Rows</th>
<th>Priqty</th>
<th>Secqty</th>
<th>Devt</th>
<th>Alloc</th>
<th>Unit</th>
<th>Free</th>
<th>Page</th>
<th>Space</th>
<th>Ext</th>
<th>Free</th>
<th>Page</th>
<th>/Kbytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>56</td>
<td>T</td>
<td>5</td>
<td>0</td>
<td>56</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4053</td>
<td>7</td>
<td>3390</td>
</tr>
<tr>
<td>2</td>
<td>40839</td>
<td>56</td>
<td>T</td>
<td>5</td>
<td>0</td>
<td>140</td>
<td>26</td>
<td>133</td>
<td>12</td>
<td></td>
<td>48</td>
<td>T</td>
<td>3390</td>
</tr>
<tr>
<td>3</td>
<td>27180</td>
<td>56</td>
<td>T</td>
<td>5</td>
<td>0</td>
<td>98</td>
<td>26</td>
<td>99</td>
<td>6</td>
<td></td>
<td>6384</td>
<td>T</td>
<td>3390</td>
</tr>
<tr>
<td>4</td>
<td>52813</td>
<td>56</td>
<td>T</td>
<td>5</td>
<td>0</td>
<td>182</td>
<td>26</td>
<td>171</td>
<td>18</td>
<td></td>
<td>8208</td>
<td>T</td>
<td>3390</td>
</tr>
</tbody>
</table>

- **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
  - **K** for large cluster members
  - **I** for normal cluster member
  - **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 is 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.

The following information appears for partition-by-growth table spaces:
The Tablespace Estimation (Partitioned by Growth) panel (Figure 97 on page 410) shows currently used space, growth information, and the type of table space on the left and center of the panel:

**Figure 97: Tablespace Estimation (Partitioned by Growth) panel**

ASUZPGES           Tablespace Estimation (Partitioned by Growth)
COMMAND ===>                                                   SCROLL==> PAGE

Tablespace Name   ARMDBPBG.TS40UTS1   Type . . .  G (I, G or blank)
Table Name . . .  JTR.T40UTS1
#Rows  . .      9009900
Pagesize . . . .    4 (4,8,16,32) Maxrows  255 (0-255)
Segsize . . . .    4 (0-64) Dssize . . 0 ------- Estimated -------
Pages . . . .    936 MaxParts  4 Pages . . .  237244
Space (Trks)  78 Alloc Parts 1 Space (Trks)  19771
MaxRow Len . .  93 AvgRow Len 93 Extents . . .  1319
Extents . . .    3 Est. Parts . . .  1
Priqty . . . . .  3 Secqty . .  15
Pctfree . . . . .  5 (0-99) Freepage 0 (0-3)
Alloc Unit . . . T (T or C) Compress (Y or blank)
Device Type . .  3390

Parts        #Rows            Rows/Page       Tracks       Kbytes        Ext
------------------------------------------------------------------------------
< 1 9009900         38        19771       949008       1319

— **Type** indicates the type of table space:
  — G for partition-by-growth table spaces
  — I for partition-by-growth table spaces
  — 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.
—\textbf{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.

\textbf{Note}
Space estimation is not available for XML objects or for objects that have partial statistics.

\section*{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 336.

2. Select the Space Estimation function by typing \textbf{S} in the \textbf{Act} field beside the table space for which you want space estimates.

   The Tablespace Estimation panel appears.

3. Estimate space requirements for the entire table space or for individual tables or partitions (or both) as follows:
   \begin{enumerate}
   \item Type over the existing values and press \textbf{Enter} to update the \textbf{Estimated} fields.
   \item You can repeat this step as many times as necessary.
   \end{enumerate}

4. When you have finished estimating space requirements, press \textbf{END} to exit.

\section*{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.

\textbf{Note}
Space estimation is not available for XML objects, LOB objects, or objects that have partial statistics.

\section*{To estimate space requirements for a non partitioned 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 336.

   The Display DB2 Object List panel appears showing the indexes that match your specification.
2 Enter S in the **Act** field beside the index for which you are estimating space, and then press **Enter**.

The Index Estimation panel appears. (*Figure 98 on page 413* shows the panel for nonpartitioned 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, then press **Enter**.

Other fields are for information only. The Tab key moves between fields that you can modify.

---

*Figure 98: Index Space Estimation panel (nonpartitioned)*

<table>
<thead>
<tr>
<th>Type data and press Enter.</th>
<th>Index Name . . MVSTAD1.BASENLOG</th>
<th>DB Name . . MVSTAD1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name . . MVSTAD1.BASENLOG</td>
<td>1stKey Col . . ID</td>
<td></td>
</tr>
<tr>
<td>Unique . . . D (U/P/N/D/C/R/G)</td>
<td>Pagesize . . 4</td>
<td></td>
</tr>
<tr>
<td>Compress . . . N (Y or blank)</td>
<td>TS type . . (L/G/R or blank)</td>
<td></td>
</tr>
<tr>
<td>Pct Compressed 0 (0 - 99)</td>
<td>Piece size . 2097152</td>
<td></td>
</tr>
<tr>
<td>Keylength . . 2 (1-2000)</td>
<td>-- Estimated -------</td>
<td></td>
</tr>
<tr>
<td>Avgkeylen . . 0</td>
<td>Pages . . 3</td>
<td></td>
</tr>
<tr>
<td>Pages . . 5</td>
<td>Space (Trks) . 15</td>
<td></td>
</tr>
<tr>
<td>Space (Trks) . 15</td>
<td>Space (KB) . 720</td>
<td></td>
</tr>
<tr>
<td>Levels . . . 2</td>
<td>Levels . . . 2</td>
<td></td>
</tr>
<tr>
<td>Data Sets . . . 1</td>
<td>Data Sets . . . 1</td>
<td></td>
</tr>
</tbody>
</table>

---

The following information appears for a nonpartitioned index as *Figure 98 on page 413* 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 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.

5 (optional) If you want to see additional estimates based on different values, repeat Step 4 on page 414.

6 When finished, press END to exit.

To estimate space requirements for a partitioned 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 336.

The Display DB2 Object List panel appears showing the indexes that match your specification.

2 Enter S in the Act field beside the index for which you are estimating space, and then press Enter.

The Index Estimation panel appears. (Figure 99 on page 415 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.

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, Dvet, 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, then press Enter.

Other fields are for information only. The Tab key moves between fields that you can modify.

Figure 99: Index Space Estimation panel (partitioned)

The following information appears for a partitioned index as Figure 99 on page 415 shows:

- Current used 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.

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 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.

5 (optional) If you want to see additional estimates based on different values, repeat List item. on page 416

6 When finished, press END to exit.

**Estimating space requirement 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 that are shown in boldface in Figure 100 on page 417. 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 box entitled "Estimated" on the center-right, and table-level estimates at the bottom-right of the panel.

For a description of each field that accepts user-specified values, see the online Help (by pressing the F1 key).

Figure 100: Tablespace Estimation panel

<table>
<thead>
<tr>
<th>Type data and press Enter.</th>
<th>Tablespace Estimation</th>
<th>Row 1 to 1 of 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pagesize . . . . 4 (4,8,16,32)</td>
<td>Pages . .</td>
<td>Pagesize</td>
</tr>
<tr>
<td>Segsize . . . . 4 (0-64)</td>
<td>Space (KB)</td>
<td>Maxrows . . . . 255 (0-255)</td>
</tr>
<tr>
<td>Dsize . . . . 0 (0,1,2,4,8,16,32,64,128,256)</td>
<td>Data Sets . .</td>
<td>Pctfree . . . . 0 (0-99)</td>
</tr>
<tr>
<td>Freepage . . . . 0 (0-255)</td>
<td></td>
<td>Compress . . . . N (Y or blank)</td>
</tr>
<tr>
<td>Priqty . . . .</td>
<td>Alloc Unit . . . T (T or C)</td>
<td></td>
</tr>
<tr>
<td>Secqty . . . .</td>
<td>Device Type . . 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>Avg Row</th>
<th>#Rows</th>
<th>------ Estimated ------</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cmd</td>
<td>Length</td>
<td>Rows/Page</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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.

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. The default values are the most current values in the DASD MANAGER PLUS statistics tables for the object.
Space estimation is not available for XML objects or for objects that have partial statistics.

You can edit some of these fields in the Tablespace Estimation panel (nonpartitioned) (Figure 95 on page 407) and then 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 101: Tablespace Estimation panel (nonpartitioned)**

<table>
<thead>
<tr>
<th>DEAE</th>
<th>Tablespace Estimation</th>
<th>Row 1 to 1 of 1</th>
<th>SCROLL===&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type data and press Enter.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tablespace Name  QZUDA1.QZUS01A1</td>
<td>Type . . . (G, I or blank)</td>
<td>Pagesize . . . . 4 (4,8,16,32) Maxrows 255 (0-255)</td>
<td>Segsize . . . . 4 (0-64) Dssize . 0</td>
</tr>
<tr>
<td>Alloc Unit . . . T (T or C) Compress (Y or blank)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device Type . . . 3390</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

- 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.

The following information appears for partitioned table spaces:

- The Tablespace Estimation (Partitioned) panel shows currently used space, growth information, and the type of table space on the left and center of the panel.

**Figure 102: Tablespace Estimation (Partitioned) panel**

<table>
<thead>
<tr>
<th>Tablespace Name: QZUD83.QZUS01B3</th>
<th>Type . . . (L, R, P, or blank)</th>
<th>Row 1 to 4 of 4</th>
<th>SCROLL===&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pagesize . . . . 4 (4,8,16,32)</td>
<td>Maxrows. 255</td>
<td>Pages . .</td>
<td>Estimated</td>
</tr>
<tr>
<td>Space (Trks) 476</td>
<td>Segsize . 0</td>
<td>Space (Trks) 394</td>
<td>Space (KB) 18912</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table Name: QZU.QZUT01_DB3S01</th>
<th>MaxRow Len . 4053</th>
<th>AvgRow Len . 144</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Part</th>
<th>#Rows /AvgRow</th>
<th>Priqty /Secqty</th>
<th>Unit /Dev (Y)</th>
<th>Pct Free /Ext</th>
<th>Page Fre /Compress</th>
<th>Space /Ext</th>
<th>Rows/Tracks /Ext</th>
<th>---- Estimated ----</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4053</td>
<td>7</td>
<td>3390</td>
<td>T</td>
<td>5</td>
<td>0</td>
<td>56</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>40839</td>
<td>7</td>
<td>3390</td>
<td>T</td>
<td>5</td>
<td>0</td>
<td>140</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>27180</td>
<td>7</td>
<td>3390</td>
<td>T</td>
<td>5</td>
<td>0</td>
<td>98</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>52813</td>
<td>7</td>
<td>3390</td>
<td>T</td>
<td>5</td>
<td>0</td>
<td>182</td>
<td>0</td>
</tr>
</tbody>
</table>

---

- **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
  - **K** for large cluster members
  - **I** for normal cluster member
  - **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 is 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.

The following information appears for partition-by-growth table spaces:

- The Tablespace Estimation (Partitioned by Growth) panel (Figure 97 on page 410) shows currently used space, growth information, and the type of table space on the left and center of the panel:

**Figure 103: Tablespace Estimation (Partitioned by Growth) panel**

| Tablespace Name | Type | Table Name | #Rows | Pagesize | Segsize | Maxrows | Dssize | Pages | Space (Trks) | Alloc Parts | Space (KB) | Extents | Est. Parts | Priqty | Secqty | Pctfree | Alloc Unit | Device Type | -- | -- | -- | -- |
|-----------------|------|------------|-------|----------|---------|---------|--------|--------|-----------|------------|-----------|---------|---------|---------|-------|--------|---------|-----------|------------|---|---|---|---|
| ARMDBPBG.TS40UTS1 | G/I | JTR.T40UTS1 | 9009900 | 4(8,16,32) | 4(0-64) | 255(0-255) | 0 | 237244 | 19771 | 949008 | 1319 | 1 |
| ASUZPGES | Tablespace Estimation (Partitioned by Growth) | COMMAND ====> | SCROLL===> | PAGE |

**Type** indicates the type of table space:
- G for partition-by-growth table spaces
- I for partition-by-growth table spaces
- 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.

---

**Note**

Space estimation is not available for XML objects or for objects that have partial statistics.

---

**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 336.

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.

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 a table space
Analyzing objects by using BMCTRIG

This chapter shows you how to analyze objects using BMCTRIG.

Before you begin to use BMCTRIG to analyze objects

You should be familiar with creating and editing actions and should refer to the following sections:
- “Maintaining and generating Service Actions” on page 223 covers working with actions.
- “Using object sets” on page 279 covers working with Object sets.
- “Collecting and managing statistics” on page 313 covers BMCTRIG and BMCSTATS.

For more information, view the following Quick Courses:
- "DASD MANAGER PLUS—Using BMCTRIG"
- "DASD MANAGER PLUS—JCLGEN Situations"

When you run the BMCTRIG utility, the utility evaluates the statistics in the DASD MANAGER 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 PLUS statistics for evaluation, you do not need to run BMCSTATS on the objects before running BMCTRIG. If you are not using DB2 RTS, collect or update the BMCSTATS statistics in the database before running BMCTRIG to ensure that you are monitoring current BMCSTATS values.

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 104 on page 426 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 104: Initiating a reorganization based on exception values**

![Diagram showing the process of BMCTRIG initiating a reorganization based on exception values.]

For the contents of the DASD MANAGER PLUS statistical database, see “DASD MANAGER PLUS historical database” on page 318 and 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 466.

- You determine what to monitor, defining 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 429 and Job generation on page 432.

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 466.

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 445.

**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.

For information about creating and editing actions, see “Maintaining and generating Service Actions” on page 223. For information about linking corrective actions to exceptions, see “Managing corrective actions” on page 469.
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 “Maintaining and generating Service Actions” on page 223.) 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 an 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 58 on page 434** shows three example actions:
### Table 58: Example actions for eliminating duplicate actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>COPY named &quot;weekly copy&quot;</td>
</tr>
<tr>
<td>2</td>
<td>QUIESCE REORG Shrlevel N reorganization</td>
</tr>
<tr>
<td></td>
<td>COPY named &quot;weekly copy&quot;</td>
</tr>
<tr>
<td></td>
<td>STATS named &quot;update catalog&quot;</td>
</tr>
<tr>
<td>3</td>
<td>STATS named &quot;update catalog&quot;</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 436.

For procedures for using object-action priorities, see Defining system-level object-action priorities on page 473.

BMCTRIG generates work for the objects and actions that it identifies as follows:

- If any priorities exist or you have specified JCLWLB Y (Workload balancing), 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 439.
- 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.

As an alternative method, you can split the processing into two independent phases as follows:

1. 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 List dialog panel (by choosing **Object Exceptions => View logged Exceptions needing Corrective Action generation** 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.

2. 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 following services as grouped services when the Execution type is G (grouped):

- BMCCOPY
- BMCCOPYI
- BMCMOD
- BMCREORG
- BMCSTATS
- BMCUPRS
- BMCCPRS
- FULLCOPY
- INCRCOPY
- QUIESCE
- REORG
- RESIZE
- RUNSTATS

**Note**

For REORG, BMCREORG, COPY, and BMCCOPY grouped services, if any object within the group is specified as not logged, SHRLEVEL CHANGE is not generated. Instead, SHRLEVEL CHANGE is changed automatically to SHRLEVEL REFERENCE.

**Example**

In the following example, the first three table spaces in the worklist are ungrouped:

```sql
-BMCCOPY TABLESPACE DB.TS1
-BMCCOPY TABLESPACE DB.TS2
-BMCCOPY TABLESPACE DB.TS3
```

Grouping them as follows leads to greater efficiency:

```sql
-BMCCOPY TABLESPACE DB.TS1 TABLESPACE DB.TS2 TABLESPACE DB.TS3
```

Grouped services allow you to process multiple partitions as well.

**Note**

Note the following considerations for REORG and BMCREORG:

- Partitions in a table space can be grouped, but multiple table spaces are not grouped.

- If you specify a non-zero value for Max Reorg Parts, BMCTRIG will not include more than the specified number of partitions in a single invocation of REORG or BMCREORG when it is grouped. If you specify zero, no limit is set.
Example

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:

- BMCC COPY TABLESPACE DB.TS1 grouped service
- BMCL LOAD TABLESPACE DB.TS1 service that cannot be grouped
- BMCU BMCSTATS TABLESPACE DB.TS1 grouped service
- BMCC COPY TABLESPACE DB.TS2 grouped service
- BMCL LOAD TABLESPACE DB.TS2 service that cannot be grouped
- BMCU BMCSTATS TABLESPACE DB.TS2 grouped service

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 (BMCCOPY and BMCSTATS) can be grouped for the TABLESPACE DB.TS1 objects. BMCTRIG then determines that the first and last services (BMCCOPY and BMCSTATS) can be grouped for the TABLESPACE DB.TS2 objects. In this scenario, BMCTRIG generates the following worklist:

- BMCC COPY TABLESPACE DB.TS1 TABLESPACE DB.TS2
- BMCL LOAD TABLESPACE DB.TS1
- BMCL LOAD TABLESPACE DB.TS2
- BMCU BMCSTATS TABLESPACE DB.TS1 TABLESPACE DB.TS2

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:

1. QUIESCE grouped service
2. UNLOAD service that cannot be grouped
3. QUIESCE grouped service [ignored]
4. LOAD service that cannot be grouped
5. COPY grouped service
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.

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 “Using object sets” on page 279.

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.

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, see “Creating an action with BMCTRIG” on page 477 which includes following topics:

- Excluding objects
- Worklist format or standard JCL
- Workload balancing
- Resizing
- Rebinding
- JCL cleanup

Table 59 on page 441 shows the general BMCTRIG processing options.

### Table 59: 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. 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>

### Sample workflows

The following sections show examples of workflows for the majority of BMCTRIG processing options.

Before you evaluate any exceptions, you can perform the following steps to define system thresholds:

- Review or set up thresholds ([Thresholds in BMCTRIG syntax on page 457](#)).
- Set up shop-specific exceptions and thresholds for the exceptions (Setting exception thresholds on page 445).
- Set up corrective actions (Setting a corrective action in BMCTRIG syntax on page 470).
- Associate corrective actions with the exceptions (Corrective actions for exceptions on page 431).
- Set priorities for objects and corrective actions (Defining system-level object-action priorities on page 473).

**Evaluating exceptions and generating corrective actions**

This topic 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 as described in Figure 105 on page 442

**Figure 105: Evaluating exceptions and generating corrective actions**
**Evaluating BMCTRIG exceptions**

This topic describes setting 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 106: Evaluating exceptions**

- **Evaluation**
  - **START**
  - Decide which objects to monitor

- **Generation**
  - Set up BMCTRIG job to evaluate Exceptions
  - Analyze the exceptions report in your job output or create an exceptions report
  - Take the necessary corrective action or use the Resume option

**Generating jobs without evaluating exceptions**

This topic shows how to set up a job that generates maintenance utilities without evaluating exceptions.
For this functionality, as seen in Figure 107 on page 444, use the Generate Jobs processing option from the BMCTRIG main menu.

**Figure 107: Generating jobs**

![Flowchart diagram showing the steps of generating jobs.]

**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 108 on page 445 shows how to set up a job that evaluates exceptions for the Database Performance for DB2 automation component.

**Figure 108: Evaluating exceptions for automation**

This topic 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.

When encountering an object that initiates an exception, BMCTRIG records the exception in the DASD MANAGER PLUS exceptions table. To determine what
corrective action to take, if needed, 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 “Creating user-defined exceptions” on page 466. For information about threshold evaluation and BMCTRIG rules, see the DASD MANAGER PLUS for DB2 Reference Manual.

Shared BMCTRIG exceptions

The following table lists the exceptions that BMCTRIG provides for both table spaces and indexes.

Table 60: BMCTRIG exceptions for both table spaces and indexes

<table>
<thead>
<tr>
<th>Exception definition name</th>
<th>Condition</th>
<th>Use this exception to do the following</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREOPEND</td>
<td>Advisory pending</td>
<td>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>ARERPEND</td>
<td>Advisory REORG pending</td>
<td>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>AYSYRAT</td>
<td>I/O ratio</td>
<td>Identify objects for which the ratio of asynchronous to synchronous I/O are over a specified percentage</td>
</tr>
<tr>
<td>BSTATAGE</td>
<td>BMCSTATS Age number of days since you collected statistics</td>
<td>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>CARD</td>
<td>Card percentage of cardinality increase</td>
<td>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>CHKPEND</td>
<td>Check pending</td>
<td>Identify objects that have been LOADed with the ENFORCE NO option.</td>
</tr>
<tr>
<td>COPYMODS</td>
<td>Mods since copy percentage of rows that have changed since last copy</td>
<td>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>COPYPEND</td>
<td>Copy pending</td>
<td>Initiate an image copy on table spaces in copy pending status.</td>
</tr>
<tr>
<td>Exception definition name</td>
<td>Condition</td>
<td>Use this exception to do the following</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>CSTATAGE</td>
<td>Catalog Upd Age number of days since the statistics were updated in the DB2 catalog</td>
<td>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>DSEXTENT</td>
<td>Dataset Extents number of data set extents</td>
<td>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>EXTENTS</td>
<td>Extents number of extents</td>
<td>Identify table spaces or indexes that have reached a certain number of extents (evaluated at the partition level).</td>
</tr>
<tr>
<td>GIPCDEC</td>
<td>GETPAGES decrease</td>
<td>Identify objects that GETPAGES per synchronous I/O decreased by a specified percentage over a normal base value</td>
</tr>
<tr>
<td>ICPYPEND</td>
<td>Informational copy pending</td>
<td>Initiate an image copy on table spaces in informational copy pending status.</td>
</tr>
<tr>
<td>LOADCOPY</td>
<td>Load after copy</td>
<td>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>LOADSTAT</td>
<td>Load after stats</td>
<td>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>NORTSCPY</td>
<td>No RTS copy data</td>
<td>Identify objects for which no DB2 RTS image copy statistics exist (evaluated at the partition level).</td>
</tr>
<tr>
<td>NORTSTAT</td>
<td>No RTS statistics</td>
<td>Identify objects for which DB2 statistical data is not available in the DB2 RTS tables.</td>
</tr>
<tr>
<td>NOSTATS</td>
<td>Nostats Objects for which no BMCSTATS are detected</td>
<td>Identify objects for which no BMCSTATS statistics exist.</td>
</tr>
<tr>
<td>PERFDEG</td>
<td>Performance degradation</td>
<td>Identify objects for which exceptions AYSYRAT, GIPCDEC, and SYPCINC were all triggered.</td>
</tr>
<tr>
<td>REORCOPY</td>
<td>Reorg after copy</td>
<td>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>
<tr>
<td>REORDELS</td>
<td>Deletes since reorg</td>
<td>Initiate a reorganization of a table space or index in which a percentage of rows were deleted since the last REORG (evaluated at the partition level).</td>
</tr>
<tr>
<td>REORINS</td>
<td>Inserts after reorg</td>
<td>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).</td>
</tr>
<tr>
<td>Exception definition name</td>
<td>Condition</td>
<td>Use this exception to do the following</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>REORSTAT</td>
<td>Reorg after stats</td>
<td>Initiate a statistics collection of a table space or index in which there has been a REORG after the last time statistics were updated (evaluated at the partition level).</td>
</tr>
<tr>
<td>REORMDEL</td>
<td>Mass del reorg</td>
<td>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>REORMODS</td>
<td>Mods since reorg</td>
<td>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>REORSPACE</td>
<td>Reorgspace</td>
<td>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>Initiate 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>STATMDEL</td>
<td>Mass del stats</td>
<td>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>STATMODS</td>
<td>Percent modified</td>
<td>Identify objects for which a percentage of rows or keys were modified since stats were collected.</td>
</tr>
<tr>
<td>SYPCINC</td>
<td>GETPAGES increase</td>
<td>Identify objects that GETPAGES per synchronous I/O increased by a specified percentage over a normal base value.</td>
</tr>
</tbody>
</table>

1. If you specify Partlvl Y, the product evaluates this exception at the partition level. Otherwise, the product evaluates the exception at the object level.

You can evaluate this exception using DB2 RTS or BMC statistics. You must specify **DB2 RTS Y** or **O** in the Exception options panel to make BMCTRIG evaluate the exception using DB2 RTS data. For more information, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

You must specify **DB2 RTS 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.
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 `TRIGNOREORY`, BMCTRIG does not analyze these exceptions.

Table 61 on page 449 lists the reorganization-related exceptions.

### Table 61: Reorganization-related exceptions

<table>
<thead>
<tr>
<th>Exception definition name</th>
<th>Condition</th>
<th>Use this exception to do the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPNDINS a</td>
<td>Append inserts</td>
<td>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>AREOPEND</td>
<td>AREO* pending</td>
<td>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>ARERPEND</td>
<td>AREOR pending</td>
<td>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>AYSYRAT</td>
<td>I/O ratio</td>
<td>Identify objects for which the ratio of asynchronous to synchronous I/O are over a specified percentage</td>
</tr>
<tr>
<td>CARD</td>
<td>CARD Percentage of cardinality increase</td>
<td>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>DSEXTENT b</td>
<td>Dataset Extents Number of data set extents</td>
<td>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>EXTENTS</td>
<td>EXTENTS Number of extents</td>
<td>Identify table spaces or indexes that have reached a certain number of extents (evaluated at the partition level).</td>
</tr>
<tr>
<td>Exception definition name</td>
<td>Condition</td>
<td>Use this exception to do the following:</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>FARIND (^b)</td>
<td>FARIND</td>
<td>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>
<tr>
<td>FAROFF</td>
<td>FAROFF</td>
<td>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. (^c) <strong>Note:</strong> This exception is evaluated only for clustering indexes.</td>
</tr>
<tr>
<td>GIPCDEC</td>
<td>GETPAGES decrease</td>
<td>Identify objects that GETPAGES per synchronous I/O decreased by a specified percentage over a normal base value.</td>
</tr>
<tr>
<td>REORDELS (^a)</td>
<td>Del since reorg</td>
<td>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>LEAFDIST</td>
<td>LEAFDIST</td>
<td>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.</td>
</tr>
<tr>
<td>LEAFFOFF (^b)</td>
<td>LEAFFAROFF</td>
<td>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).</td>
</tr>
<tr>
<td>LEAFTOFF (^b)</td>
<td>LEAFTOTOFF</td>
<td>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).</td>
</tr>
<tr>
<td>LEVELINC (^b)</td>
<td>LEVELINC</td>
<td>Identify when the number of index levels has increased compared with the previous statistics entry. (^c)</td>
</tr>
<tr>
<td>LEVELMIN (^b)</td>
<td>LEVELMIN</td>
<td>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. (^c)</td>
</tr>
<tr>
<td>Exception definition name</td>
<td>Condition</td>
<td>Use this exception to do the following:</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>LEVELS b</td>
<td>LEVELS Number of index levels</td>
<td>Identify reaching a specific number of index levels. The default is 99. 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>LOBFRSPC</td>
<td>LOB Freespace</td>
<td>Initiate a reorganization based on the percentage of the LOB that is freespace.</td>
</tr>
<tr>
<td>LOBORGR</td>
<td>LOB ORGRATIO</td>
<td>Initiate a reorganization based on the percentage of organization of the table space.</td>
</tr>
<tr>
<td>MINSPACE</td>
<td>Percentage of space used over the space required</td>
<td>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 reorganized. If the REORGSPACE_KB statistic is less than or equal to zero (indicating that no statistics were collected), the MINSPACE exception is not available.</td>
</tr>
<tr>
<td>PACTHI</td>
<td>PctActivHi Maximum percentage of active pages</td>
<td>■ Initiate a reorganization on a table space containing a high percentage of active pages ■ Validate design assumptions by determining how much of the allocated space that the table space is using You can use this exception to identify table spaces that are running out of space.</td>
</tr>
<tr>
<td>PACTLO</td>
<td>PctActivLo Minimum percentage of active pages</td>
<td>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>PCTCLUS</td>
<td>PCTCLUST Cluster ratio of the object</td>
<td>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. <strong>Note:</strong> You must specify INDEX Y or INDEX followed by an index name for this exception to be initiated.</td>
</tr>
<tr>
<td>PCTDROP</td>
<td>Pct Dropped Rows Percentage of space that dropped rows used</td>
<td>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>PERFDEG</td>
<td>Performance degradation</td>
<td>Identify objects for which exceptions AYSYRAT, GIPCDEC, and SYPINC were all triggered.</td>
</tr>
<tr>
<td>PSEUDODL b</td>
<td>Pseudo Del Key Percentage of pseudo-deleted keys</td>
<td>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>Exception definition name</td>
<td>Condition</td>
<td>Use this exception to do the following:</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------</td>
<td>----------------------------------------</td>
</tr>
</tbody>
</table>
| RBDPEND                   | RBDP pending  
Rebuild pending          | Identify indexes in the restrictive status of rebuild pending status. This status usually indicates a broken index. |
| REORGIN$^a$              | Ins since reorg | 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). |
| REORMDEL$^a$              | Mass del reorg       | 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). |
| REORMODS$^a$              | Mods since reorg  
Percentage of rows modified since last Reorg | 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). |
| REORDSPC$^a$              | Pct over alloc     | Use this exception to determine the percentage by which space allocated to a table space or partition exceeds space occupied by row data. |
| REORGLOB$^a$              | Reorg DISORG LOB   | 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. |
| REORLEAF$^a$              | Reorg Leaf         | Initiate a reorganization of an index based on a percentage of total pages in comparison to number of active leaf pages. |
| REORPEND                  | Reorg Pend  
Reorganization pending | 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. |
| REORSPAC                  | REORSPAC  
Percentage of space used over the space required | 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 an alias 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. |
| SPACE                     | SPACE  
Percentage of increase in tracks that an object uses | 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. |
| SYPCINC                   | GETPAGES increase | Identify objects that GETPAGES per synchronous I/O increased by a specified percentage over a normal base value. |
| TOTALIND$^b$              | TOTALIND  
Percentage of rows in a table that are not in their optimal positions | 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. |
Statistics and miscellaneous 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 does not analyze these exceptions.

Table 62 on page 453 lists the statistics-related exceptions.

**Table 62: Statistics-related exceptions**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Exception definition name</th>
<th>How to use as an exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACHKP pending</td>
<td>ACHKPEND</td>
<td>Use this exception to identify base table spaces in the restrictive auxiliary check-pending status. You can use the REPAIR utility, followed by CHECK DATA, to reset the ACHKP status. 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>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>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>Catalog Upd Age</td>
<td>CSTATAGE^a</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>CHKP pending</td>
<td>CHKPEND</td>
<td>Use this exception to identify tables with the check pending status.</td>
</tr>
<tr>
<td>Load after stats</td>
<td>LOADSTAT^b</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^b</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^b</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>NORTSTAT^b</td>
<td>Use this exception to identify objects for which no DB2 RTS RUNSTATS statistics exist.</td>
</tr>
<tr>
<td>NONUNIFORM</td>
<td>NONUNIFORM</td>
<td>Use this exception to monitor the nonuniformity of an index based on the values that appear in the COLDIST table. NONUNIFORM shows the percentage of index rows that contain the most frequently occurring value for the first key column.</td>
</tr>
<tr>
<td>NOSTATS</td>
<td>NOSTATS</td>
<td>Use this exception to identify objects for which no BMCSTATS statistics exist. This exception is evaluated at the partition level if you specify <strong>Partlvl Y</strong>. Otherwise, the exception is evaluated at the object level.</td>
</tr>
<tr>
<td>RBDP pending</td>
<td>RBDPEND</td>
<td>Use this exception to identify indexes in the restrictive status of rebuild pending status. This status usually indicates a broken index. This exception is evaluated at the partition level if you specify <strong>Partlvl Y</strong>. Otherwise, the exception is evaluated at the object level.</td>
</tr>
</tbody>
</table>
### Condition

<table>
<thead>
<tr>
<th>Condition</th>
<th>Exception definition name</th>
<th>How to use as an exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebd after stats</td>
<td>REBDSTAT b</td>
<td>Use this exception to initiate a statistics collection of an index based on whether a REBUILD occurred after the last time statistics were collected.</td>
</tr>
<tr>
<td>Reorg after stat</td>
<td>REORSTAT b</td>
<td>Use this exception to initiate a statistics collection of a table space or index in which there has been a REORG after the last time statistics were updated (evaluated at the partition level).</td>
</tr>
<tr>
<td>ROWS/KEY average</td>
<td>ROWS/KEYS</td>
<td>Use this exception to determine the uniqueness of an index or to identify unexpected changes in the uniqueness of an index. BMCTRIG tests for a value greater than or equal to the value specified.</td>
</tr>
</tbody>
</table>

**Note**

- 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 DB2 RTS 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.

#### Copy-related exceptions

Use the exceptions in this category to identify conditions in which you need to copy an object, including the following conditions:

- The existing COPY is out of date because of a reorg, rebuild, or load utility
- The object data has changed significantly
- DB2 status conditions

**Note**

If you specify TRIGNOSTATS Y, BMCTRIG does not analyze these exceptions.

Table 63 on page 455 lists the copy-related exceptions.

### Table 63: Copy-related exceptions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Exception definition name</th>
<th>Use this exception to accomplish the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy Pend (Copy pending)</td>
<td>COPYPEND</td>
<td>Initiate an image copy on table spaces in copy pending status. If you specify Partlvl Y, BMCTRIG evaluates the exception at the partition level. Otherwise, BMCTRIG evaluates the exception at the object level</td>
</tr>
<tr>
<td>Condition</td>
<td>Exception definition name</td>
<td>Use this exception to accomplish the following:</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>Dirty Pages (IX Dirty Pages)</td>
<td>IXDIRTY</td>
<td>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</td>
<td>Initiate a full image copy based on the percentage of modified pages.</td>
</tr>
<tr>
<td>Dirty Inrement</td>
<td>DIRTYIC</td>
<td>(with the DIRTY exception) Initiate an incremental copy on objects based on the percentage of modified pages.</td>
</tr>
<tr>
<td>Full Copy Age</td>
<td>FICAGE</td>
<td>Initiate a full image copy of this object (evaluated at the partition level).</td>
</tr>
<tr>
<td>ICOPY pending</td>
<td>ICPYPEND</td>
<td>Initiate an image copy on table spaces or indexes in informational copy pending stats. If you specify Partlvl Y, BMCTRIG evaluates the exception at the partition level. Otherwise, BMCTRIG evaluates the exception at the object level.</td>
</tr>
<tr>
<td>Image Copy Age (IX Copy Age)</td>
<td>IXICAGE</td>
<td>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</td>
<td>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</td>
<td>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</td>
<td>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</td>
<td>Identify objects for which no DB2 RTS image copy statistics exist (evaluated at the partition level).</td>
</tr>
<tr>
<td>NumIncremnt</td>
<td>COPY#IIC</td>
<td>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>REBDCOPY</td>
<td>Initiate a copy of an index based on whether a REBUILD occurred after the last copy.</td>
</tr>
<tr>
<td>Condition</td>
<td>Exception definition name</td>
<td>Use this exception to accomplish the following:</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Reorg after copy</td>
<td>REORCOPY</td>
<td>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>

1. You can evaluate this exception using DB2 RTS or BMC statistics. Specify DB2 RTS Y or O in the exception options panel for BMCTRIG to evaluate the exception using DB2 RTS data. For more information, see the DASD MANAGER PLUS for DB2 Reference Manual.

2. Specify DB2 RTS Y or O in the exception options panel (to indicate that you want to use DB2 real time statistics for analysis) to have this exception evaluated.

Thresholds in BMCTRIG syntax

In the syntax for the BMCTRIG utility, you can specify thresholds for each exception that BMC Software supplies.

**Before you begin**

To set the thresholds by using the BMCTRIG syntax, you must first perform the following operations:

1. Create an action that contains BMCTRIG as instructed in “Creating a Service Action” on page 241.

2. Create syntax for BMCTRIG to use with the BMCTRIG action as instructed in “Copying a Service Action” on page 246.

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.

   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 BMCTRIG 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 a Service Action” on page 241.

b Edit the BMCTRIG service as instructed in “Copying a Service Action” on page 246.

The main BMCTRIG panel (Figure 109 on page 458) appears.

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 109: BMCTRIG main panel

<table>
<thead>
<tr>
<th>Command</th>
<th>BMCTRIG TABLESPACE QZUD40.%</th>
<th>Scroll ===&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEGG</td>
<td>SERVICE Syntax: BMCTRIG.TRIGDEMO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enter data, then press end. More: +</td>
<td></td>
</tr>
<tr>
<td>System Triggers</td>
<td>What to Analyze: (Y/N - use exceptions and actions in DASD tables)</td>
<td></td>
</tr>
<tr>
<td>DB2 RTS</td>
<td>(Y/N/O - Y-Use RTS. N-Ignore RTS, O-Use Only RTS)</td>
<td></td>
</tr>
<tr>
<td>Bypass Exceptions:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorg Exceptions</td>
<td>(Y/N - Bypass analysis of Reorg-Related Exceptions)</td>
<td></td>
</tr>
<tr>
<td>Stats Exceptions</td>
<td>(Y/N - Bypass analysis of Stats-Related Exceptions)</td>
<td></td>
</tr>
<tr>
<td>Copy Exceptions</td>
<td>(Y/N - Bypass analysis of Copy-Related Exceptions)</td>
<td></td>
</tr>
<tr>
<td>Indexes</td>
<td>(Y/N - Include all indexes in tablespace)</td>
<td></td>
</tr>
<tr>
<td>Save</td>
<td>Saving and Reporting Exception Options: (Y/N - Save Exceptions in Exceptions table)</td>
<td></td>
</tr>
<tr>
<td>DELETEAGE</td>
<td>(Delete Exceptions older than this many days)</td>
<td></td>
</tr>
<tr>
<td>REPORT</td>
<td>(Y/N - Print Report on objects with exceptions)</td>
<td></td>
</tr>
<tr>
<td>FKEYOMSG</td>
<td>(Y/N - Allow notification of empty indexes)</td>
<td></td>
</tr>
<tr>
<td>NOTIFY</td>
<td>(Notify this User id on Exceptions)</td>
<td></td>
</tr>
<tr>
<td>MSGLEVEL</td>
<td>(0/1 0-Normal msgs, 1-Additional msgs)</td>
<td></td>
</tr>
<tr>
<td>IX Escalate</td>
<td>Eliminate Duplicates and Index Escalation Options: (Y/N/A - Escalate IX's to TS's during generation)</td>
<td></td>
</tr>
<tr>
<td>IX Copy</td>
<td>(Y/N - Copy Indexes)</td>
<td></td>
</tr>
<tr>
<td>Remove dup actions</td>
<td>(Y/N - Eliminate duplicate actions)</td>
<td></td>
</tr>
<tr>
<td>Detect inline opts</td>
<td>(Y/N - Eliminate actions based on inline options)</td>
<td></td>
</tr>
</tbody>
</table>
2 Scroll down until Override System Trigger Exceptions appears, as shown in Figure 110 on page 459.

Figure 110: Override Thresholds panel in BMCTRIG Override System Trigger panel

<table>
<thead>
<tr>
<th>Service Syntax: BMCTRIG.TRIGDEMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt; BMCTRIG.TRIGDEMO</td>
</tr>
<tr>
<td>Scroll ===&gt; CSR</td>
</tr>
<tr>
<td>Enter data, then press end.</td>
</tr>
<tr>
<td>More: + -</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>Overrize System Trigger Exceptions -------------------------------</td>
</tr>
<tr>
<td>Reorg-Related Override Exceptions -------------------------------</td>
</tr>
<tr>
<td>DB2 RTS Specific Reorg Exceptions:</td>
</tr>
<tr>
<td>Shared Tablespace/Index Exceptions:</td>
</tr>
<tr>
<td>Mods since reorg (1-100 % rows modified since reorg)</td>
</tr>
<tr>
<td>Ins since reorg (1-100 % rows inserted since reorg)</td>
</tr>
<tr>
<td>Del since reorg (1-100 % rows deleted since reorg)</td>
</tr>
<tr>
<td>Mass del reorg N (Y/N - Mass delete since last reorg)</td>
</tr>
<tr>
<td>Tablespace Exceptions:</td>
</tr>
<tr>
<td>Unclust inserts (1-99 Percent unclustered inserts)</td>
</tr>
<tr>
<td>Reorg DISORG LOB (1-100 % disorganized LOBs inserted since reorg)</td>
</tr>
<tr>
<td>Pct over alloc (Space Allocated / Space Used &gt; % Specified)</td>
</tr>
<tr>
<td>Index Exceptions:</td>
</tr>
<tr>
<td>Append inserts (1-100 Percentage of keys appended)</td>
</tr>
<tr>
<td>Reorg Leaf . . (1-100 Percentage that NPAGES exceeds NLEAF)</td>
</tr>
<tr>
<td>DB2 RTS or BMCSTATS Reorg Exceptions:</td>
</tr>
<tr>
<td>Shared Tablespace/Index Exceptions:</td>
</tr>
<tr>
<td>Dataset Extents (1-7257 Number of extents per dataset)</td>
</tr>
<tr>
<td>AREO* pending N (Y/N - Advisory REORG pending)</td>
</tr>
<tr>
<td>AREOR pending N (Y/N - Advisory REORG pending and pending DDL)</td>
</tr>
<tr>
<td>Tablespace Exceptions:</td>
</tr>
<tr>
<td>FARIND . . . . (0-99 Percent far indirect reference)</td>
</tr>
<tr>
<td>TOTALIND . . . . (0-99 Percent total indirect reference)</td>
</tr>
<tr>
<td>Reorg Pend . . . N (Y/N Reorg pending flag is on)</td>
</tr>
<tr>
<td>Index Exceptions:</td>
</tr>
</tbody>
</table>

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 “Setting up DASD MANAGER PLUS” on page 63.

Accessing system thresholds

Use this procedure to access system thresholds.

1 On the DASD MANAGER PLUS main menu, select Object Exceptions.
The Detect and Correct Object Exceptions panel is displayed.

DEAE ------------------ Detect and Correct Object Exceptions ---------------- Command ===> 

BMCTRIG is a DASD MANAGER utility that identifies and logs changes (Exceptions) in a database and optionally creates maintenance jobs (Corrective Actions) in response to these changes.

Select an option. Then press Enter.
1. Define and manage Exception detection conditions
2. Define and manage Object-Action priorities
3. Create a new BMCTRIG Service Action
4. Manage existing BMCTRIG Service Actions
5. Restart a WORKLIST format job that was generated by BMCTRIG
6. View logged Exceptions needing Corrective Action generation

2 Select Define and manage Exception detection conditions.

The Exceptions Menu is displayed.

DEEA ----------------------------- Exceptions Menu ---------------------------- Command ===> 

An Exception is a condition you want DASD MANAGER to detect. Type a specific Exception or type a wildcard pattern for a selection list.

Exception . . %

Select an option. Then press Enter.
1. List
2. Create a new User Define Exception
3. Edit Exception Thresholds
4. Edit Exception Corrective Actions
5. Edit Exception Properties
6. Copy an Exception
7. Delete an Exception

3 Select List

The Exceptions List is displayed.

DEAE ---------------- Exceptions List --------------- Row 1 to 6 of 1061 Command ===> Scroll ===> PAGE 

Exception: %

An Exception is a condition you want DASD MANAGER to detect. Select one or more actions. Then press Enter.

Sort by. . 1 (1=Name, 2=Category)
C =Create   D =Delete   P =Properties   L =Like   A =Active Exceptions   R =Edit Corrective Actions   T =Edit Thresholds
More:     + >

************* Top of data *******************
... Y Y 255 Y CTGRYE1 User Defined Exception
..--.. Y Y 255 Y CTGRYE1 User Defined Exception
.-.. Y Y 255 Y CTGRYE1 User Defined Exception
.-- Y Y 255 Y CTGRYE1 User Defined Exception
.-.-- Y Y 255 Y CTGRYE1 User Defined Exception
-.- Y Y 255 Y CTGRYE1 User Defined Exception

4 In the Act field type T next to the exception whose thresholds you want to edit.
The Exception Threshold List panel is displayed.

An Exception Threshold is a specific value that BMCTRIG monitors. When the value is exceeded, BMCTRIG logs the Exception and can generate a Corrective Action. Select an option then press Enter.

Sort by . . 1 ( 1 =Name, 2 =Object)
C =Create  D =Delete  L =Like  P =Properties

More: >

Act Exception Value Apply To Object P
**************************************** Top of data ****************************************
BSTATAGE >= 30 Days IX %.%
BSTATAGE >= 30 Days TS %.%
**************************************** Bottom of data ****************************************

The Exception Thresholds List 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:

- Create a threshold (“Creating a new threshold” on page 462)
- Edit an existing threshold (“Editing an existing threshold” on page 463)
- Delete a threshold (“Deleting a threshold” on page 464)
- Copying a threshold (“Copying a threshold” on page 465)

Before proceeding, note that BMCTRIG applies rules in the order shown in Table 64 on page 461 when determining which system-level thresholds to apply to each object:

**Table 64: Applying system-level thresholds to objects**

<table>
<thead>
<tr>
<th>Order for applying rules</th>
<th>Qualifier</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Equal</td>
<td>Equal</td>
<td>Equal</td>
</tr>
<tr>
<td>2</td>
<td>Set owner</td>
<td>Set name</td>
<td>OS a</td>
</tr>
<tr>
<td>3</td>
<td>Equal</td>
<td>Wildcard match</td>
<td>Equal</td>
</tr>
<tr>
<td>4</td>
<td>Wildcard match</td>
<td>Equal</td>
<td>Equal</td>
</tr>
<tr>
<td>5</td>
<td>Wildcard match</td>
<td>Wildcard match</td>
<td>Equal</td>
</tr>
</tbody>
</table>

a 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 “Using object sets” on page 279.

---

**Creating a new threshold**

Use this procedure to add a new threshold.

1. Type C in the **Act** field of a threshold entry that has the exception name for which you want to set a threshold.

   ```
   DEBA ----------------------- Exception Threshold List --- Row 1 to 2 of 2
   Command ===> Scroll ===> PAGE
   Exception: %
   An Exception Threshold is a specific value that BMCTRIG monitors. When the value is exceeded, BMCTRIG logs the Exception and can generate a Corrective Action. Select an option. Then press Enter.
   Sort by. . 1 ( 1 =Name, 2 =Object)
   C =Create  D =Delete  L =Like  P =Properties
   More:       >
   Act Exception Value Apply To Object
   ******************************** Top of data ********************************
   BSTATAGE >= 30 Days IX %.%
   BSTATAGE >= 30 Days TS %.%```

2. Press **Enter**
The Create Exception Threshold panel appears. DASD MANAGER PLUS automatically enters the exception name and description.

```
DEBA ----------------------- Create Exception Threshold -----------------------
Command ===> Type new Exception Threshold data. Then press Enter.
Exception:
  Name . . . . . . : BSTATAGE
  Category . . . . : STATS
  Description . . : DAYS SINCE LAST UPDATE OF BMCSTATS TABLES
Threshold:
  Value . . . . . . 0     (Range=Low Value 0 High Value 32767)
  Apply To:
    Obj Type . . . . TS    (TS, IX, OS)
    Obj Pattern . . . . . . %.%
  Priority . . . . 0     (0-255 to indicate the priority)
  Enabled . . . . . Y     (Y/N to indicate whether Exception is enabled)
```

3 Complete the panel as follows:

- In the **Value** field, enter a threshold within the valid range defined on the panel.
- In the **Obj Type** field, specify the type of object (TS, IX, or OS). Use object type OS to specify an object set name.
- In the **Obj Pattern** field, specify the objects to which the threshold applies.
- (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. BMCTRIG uses only enabled thresholds.

4 Press **END** to save the threshold and return to the previous panel.

**Editing an existing threshold**

Use this procedure to edit an existing threshold.

1 On the Exception Thresholds panel (displayed in “Accessing system thresholds” on page 459), type **P** in the **Act** field next to the threshold that you want to edit.
The Edit Exception Threshold panel is displayed.

```
DEBA ------------------------ Edit Exception Threshold -------------------
Command ===> 
Type Exception Threshold data. Then press End.
Exception:
  Name . . . . . . : BSTATAGE
  Category . . . . : STATS
  Description . . : DAYS SINCE LAST UPDATE OF BMCSTATS TABLES
Threshold:
  Value . . . . . . : 0  (Range=Low Value 0 High Value 32767 )
  Apply To:       Obj Type . . . . : TS  (TS, IX, OS)
                  Obj Pattern . . : .%  
  Priority . . . . : 0  (0-255 to indicate the priority)
  Enabled . . . . : Y  (Y/N to indicate whether Exception is enabled)
```

2 Edit the threshold as follows:

- In the **Value** field, enter a threshold within the valid range defined on the panel.
- In the **Obj Type** field, specify the type of object (TS, IX, or OS). Use object type OS to specify an object set name.
- In the **Obj Pattern** field, specify the objects to which the threshold applies.
- *(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. BMCTRIG uses only enabled thresholds.

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 466.

3 Press **END** to save your changes and return to the previous panel.

### Deleting a threshold

Use this procedure to delete a threshold.

1 On the Exception Threshold List panel (for an example of the panel, see “Editing an existing threshold” on page 463), type **D** in the **Act** field next to the threshold that you want to delete.
The Delete Exception Threshold panel appears.

DEBA ----------------------- Delete Exception Threshold -----------------------
Command ===>

Exception:
Name . . . . . . : CARD
Category . . . . : REORG
Description . . : PERCENT OF CARDINALITY INCREASE SINCE LAST STATS

Threshold:
Value . . . . . . : 1 Updated By : MVSTAD1
Priority . . . . : 1 Last Updated: 2016-03-08
Enabled . . . . : Y

Apply To:
Obj Type . . . : TS
Obj Pattern . . : MVSTAD1.%

Select delete option. Then press Enter.
Delete Options . . 2 1. Delete Threshold 2. Cancel Delete and Exit

2 Type 1 in the Delete Options field and press Enter.

Copying a threshold

Use this procedure to copy a threshold.

1 On the Exception Thresholds List panel (displayed in “Accessing system thresholds” on page 459), type L in the Act field next to the threshold that you want to copy.

The Copy Exception Threshold panel is displayed.

DEAE ------------------------ Copy Exception Threshold ------------------------
Command ===>

Type new Exception Threshold data. Then press Enter.

Exception:
Name . . . . . . : BSTATAGE
Category . . . . : STATS
Description . . : DAYS SINCE LAST UPDATE OF BMCSTATS TABLES

Threshold:
Value . . . . . . : 30 (Range=Low Value 0 High Value 32767 )
Priority . . . . : 0 (0-255 to indicate the priority)
Enabled . . . . : N (Y/N to Indicate whether Exception is enabled)

Obj Type . . . : TS (TS, IX, OS)
Obj Pattern . . : %.%

2 Edit the threshold as follows:

- In the Value field, edit a threshold within the valid range defined on the panel.
- In the Obj Type field, specify the type of object (TS, IX, or OS). Use object type OS to specify an object set name.
- In the Obj Pattern field, specify the objects to which the threshold applies.
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. BMCTRIG uses only enabled thresholds.

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 “Setting up DASD MANAGER PLUS” on page 63. For more information and samples of the REXX program requirements for user-defined exceptions, see the DASD MANAGER PLUS for DB2 Reference Manual.

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 Object Exceptions. The Detect and Correct Object Exceptions panel is displayed.

DEBA ------------------ Detect and Correct Object Exceptions ------------------

BMCTRIG is a DASD MANAGER utility that identifies and logs changes (Exceptions) in a database and optionally creates maintenance jobs (Corrective Actions) in response to these changes.

Select an option. Then press Enter.
1. Define and manage Exception detection conditions
2. Define and manage Object-Action priorities
3. Create a new BMCTRIG Service Action
4. Manage existing BMCTRIG Service Actions
5. Restart a WORKLIST format job that was generated by BMCTRIG
6. View logged Exceptions needing Corrective Action generation
2 Select **Define and manage Exception detection conditions** and press **Enter**.

The Exceptions Menu panel is displayed.

```
DEBA ----------------------------- Exceptions Menu -----------------------------
Command ===>  
An Exception is a condition you want DASD MANAGER to detect. Type a specific Exception or type a wildcard pattern for a selection list.
Exception . . . %
Select an option. Then press Enter.
   1. List
   2. Create a new User Define Exception
   3. Edit Exception Thresholds
   4. Edit Exception Corrective Actions
   5. Edit Exception Properties
   6. Copy an Exception
   7. Delete an Exception
```

3 *(optional)* If you know the name of the desired exception name, enter the exception name, select **Create a new User Defined Exception**, and then skip to List item. on page 468.

4 Select **List** and press **Enter**.
The Exceptions List is displayed.

The Exceptions List is displayed.

Object: %

An Exception is a condition you want DASD MANAGER to detect. Select one or more actions. Then press Enter.

Sort by. . 1 (1=Name, 2=Category)

C =Create    D =Delete   P =Properties   L =Like   A =Active Exceptions
R =Edit Corrective Actions   T =Edit Thresholds

More:     + >

Apply To   Aging
Act  Name      Ts  Ix    Incr   User  Category  Identifies
******************************** Top of data ********************************

ACTPEND    Y   N      0      N   CHECK      DB2 Status Condition
APPENDINS  N   Y      0      N   REORG      Disorganized Data
AR Pend    N   Y      0      N   REORG      DB2 Status Condition
AR PEND    Y   Y      0      N   REORG      DB2 Status Condition
ASY SYRAT   Y   Y      0      N   REORG      Object Performance Degradation
BSTATAGE   Y   Y      0      N   STATS      Out of Date Stats/Copy
CARD       Y   Y      0      N   REORG      Change in Amount of Data
CHKPEND    Y   Y      0      N   CHECK      DB2 Status Condition
COPY#IIC   Y   N      0      N   COPY       Change in Object Data
COPY#AGE   Y   N      0      N   COPY       Out of Date Stats/Copy
COPYMODS   Y   Y      0      N   COPY       Change in Object Data
COPYPEND   Y   Y      0      N   COPY       DB2 Status Condition
CSTATAGE   Y   Y      0      N   STATS      Out of Date Stats/Copy
DIRTY      Y   N      0      N   COPY       Change in Object Data
DIRTYIC    N   Y      0      N   COPY       Change in Object Data
DSEXTENT   Y   Y      0      N   REORG      Data Set Fragmentation
EXTENTS    Y   Y      0      N   REORG      Data Set Fragmentation
FARIND     Y   N      0      N   REORG      Disorganized Data
FAROFF     N   Y      0      N   REORG      Disorganized Data
FICAGE     Y   N      0      N   COPY       Out of Date Stats/Copy
GIPCDEC    Y   Y      0      N   REORG      Object Performance Degradation
ICPYPEND   Y   Y      0      N   COPY       DB2 Status Condition
IXDIRTY    Y   Y      0      N   COPY       Change in Object Data
IXICAGE    N   Y      0      N   COPY       Out of Date Stats/Copy
LEAFDIST   N   Y      0      N   REORG      Disorganized Data
LEAFFOFF   N   Y      0      N   REORG      Disorganized Data

5 Add a new exception by typing C in the Act field next to an Exception.

The Create User Exception panel is displayed

DEBA ------------------------- Create User Exception -------------------------
Command ===>

Type new User Exception data. Then press Enter.

Name . . . . . .
Category . . . . (COPY, REORG, STATS, REPORT, OTHER)
Applies to TS . . Y (Y/N applies to table spaces)
Applies to IX . . Y (Y/N applies to indexes)
Aging Increment . 0 (0-255 to increment priority per day)
Description . . .
Long Description

6 Enter a unique name for the exception, keeping in mind the following considerations:

- BMCTRIG uses the name for reporting the exception.
You should specify whether the exception applies to table spaces, indexes, or both.

(optional) You can specify the aging priority (the same as you would for exceptions that BMC Software supplies).

(optional) You can specify the exception Category (COPY, REORG, STATS, REPORT, or OTHER).

7 Press END to save your changes.

The Create Exception Threshold panel is displayed. (For an example of the panel, see Creating a new threshold on page 462).

**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. See “Analyzing statistical trends” on page 379, which 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, 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.
3. 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 SYSTEM TRIGGERS 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
- Add a corrective action
- Edit an existing corrective action
- Delete a corrective action
To access system corrective actions

1. On the DASD MANAGER PLUS main menu, select Object Exceptions.

The Exceptions Menu panel is displayed.

```
DEBA ----------------------------- Exceptions Menu -----------------------------
Command ===>
```

An Exception is a condition you want DASD MANAGER to detect. Type a specific Exception or type a wildcard pattern for a selection list.

Exception . . . %

Select an option. Then press Enter.
1. List
2. Create a new User Define Exception
3. Edit Exception Thresholds
4. Edit Exception Corrective Actions
5. Edit Exception Properties
6. Copy an Exception
7. Delete an Exception

2. Enter a wildcarded exception name and select Edit Exception Corrective Actions.

The Corrective Actions List panel (Figure 111 on page 471) is displayed.

```
Figure 111: Corrective Actions List panel
```

```
DEBA --------------------- Corrective Actions List --- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> PAGE
Exception: %
```

A Corrective Action is the Action to initiate when the Exception Threshold Value has been reached. Associate an Action Name as the Corrective Action to initiate when the Exception is detected and specify for which objects that action is to be used.

Sort by. . 1 ( 1 =Exception, 2 =Action, 3 =Object)

C =Create  D =Delete  L =Like  P =Properties

--- More: > ---

```
Act Excepton Action Name           Apply To Object            Updated
******************************** Top of data ********************************
BSTATAGE  BMC_BMCSTATS_DEFAULT  TS  %.%                    2004-07-22-17.31
CARD      BMC_BMCSTATS_DEFAULT  TS  %.%                    2004-07-22-17.31
UDEFINED  BMCCOPY               TS  MVSTAD1.%              2016-03-08-07.59
******************************** Bottom of data ********************************
```

The panel displays all of the corrective actions that are defined for your subsystem. In the Sort by field, you can 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 Actions List panel (Figure 111 on page 471), type C in the Act field next to a corrective action entry, and press Enter.
The Create Corrective Action panel is displayed.

<table>
<thead>
<tr>
<th>Command</th>
<th>Create Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception: BSTATAGE</td>
<td></td>
</tr>
</tbody>
</table>

Create a Corrective Action to generate when the Exception is detected. Use ‘%’ to list or create Actions. Type new data. Then press Enter.

<table>
<thead>
<tr>
<th>Action Name</th>
<th>Apply to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj Type</td>
<td>Obj Pattern</td>
</tr>
</tbody>
</table>

2 If you know the name of the corrective action, type them in the appropriate fields.

**Tip**

To select from a list, type % in the **Action Name** field.

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 the table in “Shared BMCTRIG exceptions” on page 446 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.

3 Press **END** to save your changes and return to the previous panel.

**To edit an existing corrective action**

1 On the Corrective Actions List panel (Figure 111 on page 471), type **P** in the Act field next to the corrective action that you want to edit.
The Edit Corrective Action Properties panel is displayed.

```
DEBA ----------------- Edit Corrective Action Properties -----------------
Command ===>
Type Corrective Action data. Then press End.
Exception:
 Name . . . . : UDDEFINED
 Category . . : OTHER
 Description . : USER DEFINED
Corrective Action:
 Action Name . . BMCCOPY          (%=List or Create)
 Apply to:
 Obj Type . . TS     (TS, IX, OS)
 Obj Pattern . %.%
```

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 Actions List panel (Figure 111 on page 471), type P in the Act field next to the corrective action that you want to delete.

The Delete Corrective Action panel is displayed.

2 Type 2 in the Delete Options 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
- Add an object-action priority
- Edit an existing object-action priority
- Delete an object-action priority

To access system object-action priorities

1 On the DASD MANAGER PLUS Main Menu, select Object Exceptions.
The Detect and Correct Object Exceptions panel is displayed.

DEBA ------------------ Detect and Correct Object Exceptions ------------------
Command ===> 

BMCTRIG is a DASD MANAGER utility that identifies and logs changes (Exceptions) in a database and optionally creates maintenance jobs (Corrective Actions) in response to these changes.

Select an option. Then press Enter.
1. Define and manage Exception detection conditions
2. Define and manage Object-Action priorities
3. Create a new BMCTRIG Service Action
4. Manage existing BMCTRIG Service Actions
5. Restart a WORKLIST format job that was generated by BMCTRIG
6. View logged Exceptions needing Corrective Action generation

2 Select Define and manage Object-Action priorities.

The Object-Action Priorities List panel (Figure 112 on page 474) appears.

Figure 112: Object-Action Priorities List panel

DEBA ------------------ Object-Action Priorities List Row 1 to 3 of 3 
Command ===> Scroll ===> PAGE

Object-Action Priorities enables ordering the work generated for specific objects when a particular Corrective Action is initiated. Object-Action priorities override Exception Threshold priorities. Select one or more options. Then press Enter.

C =Create   D =Delete   P =Properties

More:       >

<table>
<thead>
<tr>
<th>Act Prty</th>
<th>Apply To Object</th>
<th>Part</th>
<th>Corrective Action</th>
<th>BMC Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 IX %.%</td>
<td>0 IX A2345678.A</td>
<td>0 %</td>
<td>BMCREORG</td>
<td>THIS IS A V</td>
</tr>
</tbody>
</table>

Note
Angle brackets (<>) are included within a long name that has been truncated. To view or modify the full name, type P.

Figure 112 on page 474 shows all defined priorities 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 65 on page 474 when applying priority.

Table 65: Applying object-action priority

<table>
<thead>
<tr>
<th>Priority</th>
<th>Qualifier</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>equal</td>
<td>equal</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 list panel ([Figure 112 on page 474](#)), type C in the **Act** field of an object-action priority, and then press **Enter**.

The Create Object-Action panel is displayed.

```
DEBA -------------------------- Create Object-Action --------------------------
Command ===>
```

Type new Object-Action data. Then press Enter.

```
Priority . . . . 0      (0-255)
Apply to:
Obj Type . . . . TS     (TS, IX)
Obj Pattern . .        (0-4096)
Corrective Action                       (%=List or Create)
Description . .
```

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 **Corrective Action** 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. In the **Description** field, type a description for the Object-Action.

7. Press **END** to save your changes and return to the previous panel.
To edit an existing object-action priority

1. On the Object-Action Properties List panel (Figure 112 on page 474), type E in the Act field next to the object-action priority that you want to edit.

   The Edit Object-Action Priorities panel is displayed.

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 Priorities list panel (Figure 112 on page 474), type D in the Act field next to the object-action priority that you want to delete.

   The Delete Object-Action panel appears.

2. Type 2 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 “Using object sets” on page 279.)

2. Identify which exceptions to monitor (by using system triggers or overrides).

   To set thresholds that apply to both table spaces and indexes, see “Thresholds in BMCTRIG syntax” on page 457.

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


2. Add the service BMCSTATS in the action (before BMCTRIG so that your BMC Software statistics are current) as instructed in “Copying a Service Action” on page 246.

   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 113 on page 478.

**Figure 113: Example of Edit Action Services panel with BMCTRIG**

<table>
<thead>
<tr>
<th>Act Service</th>
<th>Object Name/Pattern</th>
<th>Type</th>
<th>Part</th>
<th>Object Set Name</th>
<th>Syntax Name</th>
<th>More:</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMSTATS</td>
<td>TZU%.%</td>
<td>TS</td>
<td></td>
<td></td>
<td>(D)</td>
<td></td>
</tr>
<tr>
<td>BMCTRIG</td>
<td>TZU%.%</td>
<td>TS</td>
<td></td>
<td></td>
<td>(D)</td>
<td></td>
</tr>
</tbody>
</table>

6 To view the syntax options for the BMCTRIG step, type **S** 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 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 **C** in the Act field next to any entry.

b Enter information in the Create Service Syntax panel, and then press **Enter** to continue.
The BMCTRIG main panel appears.

**Figure 114: BMCTRIG main panel for example job EXCEPT1**

<table>
<thead>
<tr>
<th>DEGG</th>
<th>BMCTRIG TABLESPACE T2U%.%</th>
<th>Row 1 to 37 of 211</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td></td>
<td>Scroll ===&gt; CSR</td>
</tr>
<tr>
<td>Service Syntax: BMCTRIG.TRIGDEMO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enter data, then press end.</td>
<td></td>
<td>More: +</td>
</tr>
</tbody>
</table>

**Processing Options**
- Select . . . . . . . . 1
  1. Evaluate and Generate
  2. Evaluate Exceptions
  3. Generate Jobs
  4. Evaluate Exceptions for Automation
  5. Resume Generation

**What to Analyze**
- System Triggers Y (Y/N - use exceptions and actions in DASD tables)
- DB2 RTS . . . . . Y (Y/N/O - Y-Use RTS, N-Ignore RTS, O-Use Only RTS)
- Bypass Exceptions:
  - Reorg Exceptions N (Y/N - Bypass analysis of Reorg-Related Exceptions)
  - Stats Exceptions N (Y/N - Bypass analysis of Stats-Related Exceptions)
  - Copy Exceptions N (Y/N - Bypass analysis of Copy-Related Exceptions)
- Indexes . . . . . Y (Y/N - Include all indexes in tablespace)

**Saving and Reporting Exception Options**
- Save . . . . . . . Y (Y/N - Save Exceptions in Exceptions table)
- DELETEAGE . . . 32767 (Delete Exceptions older than this many days)
- REPORT . . . . . Y (Y/N - Print Report on objects with exceptions)
- FKEYOMSG . . . . Y (Y/N - Allow notification of empty indexes)
- NOTIFY . . . . . (Notify this User id on Exceptions)
- MSGLEVEL . . . . 0 (0/1 0-Normal msgs, 1-Additional msgs)

**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 processing options

For the action with BMCTRIG that you created, select one of the following processing options from the BMCTRIG menu:

- Evaluate and Generate
- Evaluate Exceptions
- Generate Jobs
- Evaluate Exceptions for Automation
- Resume Generation

For more information about these options, see “BMCTRIG processing options” on page 440.

Specifying What to Analyze options

These options specify what trigger analysis uses during the evaluation process.

These 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. These options also give you the ability to suppress exceptions from evaluation based on categories such as REORG, STATISTICS, and COPY.

Specifying Analyze options

1. Specify these options as shown in Table 66 on page 480.

Table 66: What to Analyze options

<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 SYSTEM TRIGGERS Y in the command syntax (or SYSTRIGS Y 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 SYSTEM TRIGGERS N in the command syntax, BMCTRIG evaluates only thresholds that you specify on the BMCTRIG panels.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</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>
<tr>
<td>Bypass Exceptions</td>
<td>This option provides the ability to suppress categories of exceptions from evaluation. You can suppress reorganization, statistics, or copy exceptions from evaluation.</td>
</tr>
<tr>
<td></td>
<td>These options are particularly useful when system triggers are defined. For example, if you have system triggers defined for statistics and reorganization exceptions, and for this trigger run you are only interested in evaluating reorganization exceptions, you can suppress the statistics evaluations by specifying Stats Exceptions Y.</td>
</tr>
<tr>
<td>Indexes</td>
<td>This option indicates whether index exceptions are also evaluated when processing table spaces or table space sets. Y is the default. The PCTCLUS, FAROFF, and TOTALOFF exceptions are detected at the Index level, but cause generation of table space utilities. To detect these three types of executions, you must specify Indexes Y or run against indexes instead of table spaces.</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 the BMCTRIG main panel. (See “Creating an action with BMCTRIG” on page 477.) Specify these options as shown in Table 67 on page 481.

<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>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 REPORT Y to generate the report.</td>
</tr>
<tr>
<td>FKEY0MSG</td>
<td>This option indicates whether to generate message BMC16637 when DASD MANAGER encounters indexes with no keys (0 FULLKEYCARD). Specify FKEY0MSG Y to report on indexes with no keys. Otherwise these indexes are not processed. BMCTRIG does not evaluate indexes with 0 FULLKEYCARD.</td>
</tr>
</tbody>
</table>
| NOTIFY    | This option is used to notify (with a TSO NOTIFY message) a user when the product finds exceptions. In the NOTIFY field, type a TSO logon ID. To ensure that you receive messages from other terminal users, you must turn the INTERCOM option on. To review your profile and turn the INTERCOM option on, follow these steps:  

1. From the TSO command prompt (ISPF option 6), type `PROFILE` and press `Enter` to display your current settings.  
2. To receive messages, type `PROFILE INTERCOM` at the TSO Command Prompt. |
| MSGLEVEL  | This option is used to specify whether you want normal or additional messages. MSGLEVEL 0 is the default.                                                                                           |

### 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 115 on page 483.

Figure 115: 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</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 non-copyable 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 433.</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>

Specify these options as shown in Table 68 on page 483.

Table 68: Eliminate Duplicates and Index Escalation options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX Escalate</td>
<td>(Y/N/A - Escalate IX's to TS's during generation)</td>
</tr>
<tr>
<td>IX Copy</td>
<td>(Y/N - Copy indexes)</td>
</tr>
<tr>
<td>Remove dup actions</td>
<td>(Y/N - Eliminate duplicate actions)</td>
</tr>
<tr>
<td>Detect inline opts</td>
<td>(Y/N - Eliminate actions based on inline options)</td>
</tr>
<tr>
<td>TS Tracks &lt;</td>
<td>(0-999999999 Exclude TS/TP's with &lt; this many tracks)</td>
</tr>
<tr>
<td>TS Tracks &gt;</td>
<td>(0-999999999 Exclude TS/TP's with &gt; this many tracks)</td>
</tr>
<tr>
<td>IX Tracks &lt;</td>
<td>(0-999999999 Exclude IX/IP's with &lt; this many tracks)</td>
</tr>
<tr>
<td>IX Tracks &gt;</td>
<td>(0-999999999 Exclude IX/IP's with &gt; this many tracks)</td>
</tr>
<tr>
<td>ExcludeEmpty</td>
<td>(Y/N - Exclude TS/TSP objects with 0 cardinality)</td>
</tr>
<tr>
<td>Archived</td>
<td>(Y/N - Exclude objects whose datasets are archived)</td>
</tr>
<tr>
<td>Bad status</td>
<td>(Y/N - Exclude objects not in RW status)</td>
</tr>
<tr>
<td>PARTLVL</td>
<td>(Y/N - Generate actions at a partition level)</td>
</tr>
<tr>
<td>Generate priority</td>
<td>(0-255 Min. exception priority to generate)</td>
</tr>
<tr>
<td>Group objects</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>

Specifying Exclude Objects from Generation options

If you are evaluating exceptions, you can specify options to exclude objects from corrective actions. 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 “Using object sets” on page 279. 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 the following table are applicable if you are generating corrective actions.

Table 69: Corrective Action Generation options

<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. 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. <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>
<tr>
<td>Group objects</td>
<td>Typing Y generates all objects in a single invocation when the service allows it. When you use this option with Evaluate Exceptions for Automation Y, the option also determines whether objects are registered individually or all in one registration.</td>
</tr>
</tbody>
</table>
Option Description
Max reorg parts To limit the number of partitions that will be included in a single invocation of a grouped REORG or BMCREORG service, type a number from 1 through 4096. To not limit the number of partitions, keep the default, 0.

**Specifying Rebind and Resize Object options**

This section familiarizes you with the Rebind and Resize options. The following panel displays these options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebind . . . . N</td>
<td>(Y/N/PG/PL - Rebind affected Packages or Plans, Y=Both)</td>
</tr>
<tr>
<td>Rebind Version . . L</td>
<td>(L/A - L=Latest version, A=All versions)</td>
</tr>
<tr>
<td>Resize jobs . . Y</td>
<td>(Y/N - Split REORG/RESIZE into separate jobs)</td>
</tr>
<tr>
<td>Resize Down . . Y</td>
<td>(Y/N - Allow objects to be resized downward)</td>
</tr>
<tr>
<td>Resize % pqty . . 0</td>
<td>(0-999 - Percentage increase over Reorgspace)</td>
</tr>
<tr>
<td>Resize % sqty . . 0</td>
<td>(0-999 - Percentage of primary quantity)</td>
</tr>
<tr>
<td>Round qty by . . A</td>
<td>(A/C - A=Allocation type, C=Cylinder)</td>
</tr>
<tr>
<td>Maximum qty . . 0</td>
<td>(0-67108864 - Max primary quantity in K)</td>
</tr>
<tr>
<td>Pqty % if max . . 0</td>
<td>(0-100 - Pct of Reorgspace if pqty reaches max)</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 70 on page 485 lists the utilities that update DB2 catalog statistics.

1 Use the values in the table below to specify rebinds:
   - In the corrective action
   - With the parameters shown in the following table

**Table 70: Triggered Rebind utilities**

<table>
<thead>
<tr>
<th>Utility</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCCOPY</td>
<td>RUNSTATS = Yes or UPDATE = Accesspath</td>
</tr>
<tr>
<td>BMCLOAD</td>
<td>UPDATEDB2STATS = Yes</td>
</tr>
<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 or UPDATEDB2STATS = Accesspath</td>
</tr>
</tbody>
</table>
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 the preceding table 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 the following table are applicable if you are generating corrective actions.

### Table 71: 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 <strong>Y</strong> in the Rebind field. To rebind packages only, type <strong>PG</strong>. To rebind plans only, type <strong>PL</strong>. Otherwise, type <strong>N</strong>.</td>
</tr>
<tr>
<td>Rebind Version</td>
<td>To include only the latest version for the value selected for Rebind, type <strong>L</strong> in the Rebind Version field. To include all versions, type <strong>A</strong>.</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.

1. Apply the options in the following table if you are generating corrective actions:
Table 72: Resize options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Resize jobs** | 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.  
**Note:** 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. |
| **Resize Down** | 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. |
| **Resize % pqty** | 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:  
\[
\text{NEW PRIQTY} = \text{REORGSPACE} + (2 \times \text{SQTY})
\]  |
| **Resize % sqty** | Use this option to change the secondary quantity allocation to a percent of the calculated primary quantity. If you do not specify a value, the secondary quantity is not changed. |
| **Round qty by** | Type A to round by the allocated type as noted in the BMCSTATS tables (the default) or C to calculate on a cylinder boundary. |
| **Maximum pqty** | Use this option to provide an upper limit for the calculated primary quantity when resizing.  
**Note:** Note that DB2 rules for the maximum data set size always apply, no matter what value you set. However, if you use this option, you can specify **Pqty % if max** as an alternate value to apply to Reorgspace, if the calculated primary quantity exceeds this limit. |
| **Pqty % if max** | 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** value. |

The **Resize** option in the corrective action for the IBM REORG and BMCREORG functions specifies the scope of the resizing. The following figure shows the **Resize** option on the first IBM REORG utility panel. If the corrective action...
contains a RESIZE service, it applies to all generated VCAT or storage group objects.

Figure 116: REORG with Resize option

<table>
<thead>
<tr>
<th>Value</th>
<th>Instruction to BMCTRIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>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 <em>table space partitions qualified for generation</em> 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 <em>index partitions qualified for generation</em> 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 <em>objects qualified for generation</em> (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 product expands or contracts the primary space quantity 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 the return value as the primary quantity. If BMCTRIG does not find a resize exit, it uses calculations and user options instead. The shipped default for the resize exit uses the following formula to calculate primary quantity:

\[
PRIQTY = REORGSPACE + (2 \times SQTY)
\]
Note
You should also consider the following:

■ DB2 rules determine the maximum primary quantity.

■ 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 does not resize the object.

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.

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

The following figure shows the Job and JCL Generation options.

**Figure 117: Job and JCL Generation options**

<table>
<thead>
<tr>
<th>Job Generation Options</th>
<th>(Default corrective action to generate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Action</td>
<td>(Y/N - Automatically Submit the Util DSN)</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>Standard JCL Options</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard JCL</td>
<td>(Y/N - Y=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>Output Dataset Options</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Util DSN</td>
<td></td>
</tr>
<tr>
<td>Interim WL DSN</td>
<td></td>
</tr>
</tbody>
</table>

Note: Interim WL DSN is required when Standard JCL=Y. Util DSN (JCL target dataset) is always required.

--- Jobcard(s) for BMCTRIG generated jobs(s) ---

| Jobcard1                | //JOBC JOB ('ZACCTNUM'),'PGMR', |
| Jobcard2                | // CLASS=A, MSGLEVEL=(1,1) |
| Jobcard3                | // |
| Jobcard4                | // |
| Jobcard5                | // |

If you are generating corrective actions, you can specify job processing options.

**To specify a data set name to hold the JCL**

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 491.

**To specify the maximum number of DB2 objects (Max Objects)**

Max Objects specifies the maximum number of DB2 objects to place in each generated job.

1. In the Max Objects option on the JCL Generation panel, type one the following values:
### User Input | Description
---|---
1 | Generates one object for each job  
**Note:** Do not use this option if you are using workload balancing
User specified number | When the number that you specify is reached, a new job is generated.
Blank | The default number of objects, 20, will be placed in the job

**To specify data set names (Util DSN)**

1 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. The option 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.
Enter the number of jobs that you require the job scheduler to generate.

You must match the jobnaming 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 actions:

- 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**
  
  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

This topic describes the Reorg-Related Override Exceptions options.

Figure 118 on page 493 shows the Reorg-Related Override Exceptions options and 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.

**Figure 118: Reorg-Related Override Exception**

<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                  (1-100 % rows modified since reorg)</td>
</tr>
<tr>
<td>Ins since reorg                   (1-100 % rows inserted since reorg)</td>
</tr>
<tr>
<td>Del since reorg                   (1-100 % rows deleted since reorg)</td>
</tr>
<tr>
<td>Mass del reorg N                  (Y/N - Mass delete since last reorg)</td>
</tr>
<tr>
<td><strong>Tablespace Exceptions:</strong></td>
</tr>
<tr>
<td>Unclost inserts                   (1-99 Percent unclustered inserts)</td>
</tr>
<tr>
<td>Reorg DISORG LOB                  (1-100 % disorganized LOBs inserted since reorg)</td>
</tr>
<tr>
<td>Pct over alloc                    (Space Allocated / Space Used &gt; % Specified)</td>
</tr>
<tr>
<td><strong>Index Exceptions:</strong></td>
</tr>
<tr>
<td>Append inserts</td>
</tr>
<tr>
<td>Reorg Leaf ...                    (1-100 Percentage that NPAGES exceeds NLEAF)</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                      (1-7257 Number of extents per dataset)</td>
</tr>
<tr>
<td>AREO* pending N                     (Y/N - Advisory REORG pending)</td>
</tr>
<tr>
<td>AREOR pending N                     (Y/N - Advisory REORG pending and pending DDL)</td>
</tr>
<tr>
<td><strong>Tablespace Exceptions:</strong></td>
</tr>
<tr>
<td>FARIND ...                          (0-99 Percent far indirect reference)</td>
</tr>
<tr>
<td>TOTALIND ...                        (0-99 Percent total indirect reference)</td>
</tr>
<tr>
<td>Reorg Pend ...                      (Y/N Reorg pending flag is on)</td>
</tr>
<tr>
<td><strong>Index Exceptions:</strong></td>
</tr>
<tr>
<td>LEAFTOTOFF ...                      (1-99 Percent leaf pages not in optimal position)</td>
</tr>
<tr>
<td>LEAFFAROFF ...                      (1-99 Percent leaf pages far from previous page)</td>
</tr>
<tr>
<td>LEVELMIN ...                        (Y/N - Levels &gt; minimum Levels)</td>
</tr>
<tr>
<td>Pseudo Del Key ...                  (Y/N - Increase in levels)</td>
</tr>
<tr>
<td>LEVELS ...                          (1-99 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 ...                           (0-9999% Space increase)</td>
</tr>
<tr>
<td>REORGSSPACE ...                    (0-9999% Over reorgspace)</td>
</tr>
<tr>
<td>EXTENTS ...                         (1-928896 Total number of extents)</td>
</tr>
<tr>
<td>CARD ...                            (0-9999% Card increase)</td>
</tr>
<tr>
<td><strong>Tablespace Exceptions:</strong></td>
</tr>
<tr>
<td>FAROFF ...                         (0-99 Percent far off position)</td>
</tr>
<tr>
<td>TOTALOFF ...                       (0-99 Percent total off position)</td>
</tr>
<tr>
<td>PCTCLUST ...                      (1-99 Percent clustered)</td>
</tr>
<tr>
<td>Pct Dropped Rows                   (1-99 Percent space occupied by dropped rows)</td>
</tr>
<tr>
<td>PctActvLvO ...                     (1-99 Percent less than percent active)</td>
</tr>
<tr>
<td>LOB ORGRATIO ...                  (1-100 Percent of organization in the LOB space)</td>
</tr>
<tr>
<td>LOB Freespace                      (1-100 Percent of free space available)</td>
</tr>
<tr>
<td><strong>Index Exceptions:</strong></td>
</tr>
<tr>
<td>LEAFDIST ...                       (100-99999999 Leaf distribution)</td>
</tr>
</tbody>
</table>
Specifying statistics-related and miscellaneous override exceptions

This topic describes the statistics-related and miscellaneous override exceptions options.

The panel below displays the Statistics-Related and Miscellaneous Override Exceptions options. For descriptions of the exceptions, see the link below. 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.

---

Figure 119: Statistics-Related and Miscellaneous 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</td>
</tr>
<tr>
<td>Load after stats N</td>
</tr>
<tr>
<td>Mods since stats</td>
</tr>
<tr>
<td>Mass del stats N</td>
</tr>
<tr>
<td>NO RTS stats dat N</td>
</tr>
</tbody>
</table>

---

DB2 RTS Index Exceptions:

- Rebd after stats N | (Y/N - Rebuild after last stats)

---

DB2 RTS or BMCSTATS Specific Statistics/Miscellaneous Exceptions:

<table>
<thead>
<tr>
<th>Shared Tablespace/Index Exceptions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog Upd Age</td>
</tr>
<tr>
<td>CHKP pending . . N</td>
</tr>
<tr>
<td>Tablespace Exceptions:</td>
</tr>
<tr>
<td>ACHKP pending N</td>
</tr>
<tr>
<td>Index Exceptions:</td>
</tr>
<tr>
<td>RBDP pending . . N</td>
</tr>
</tbody>
</table>

---

BMCSTATS Specific Statistics/Miscellaneous Exceptions:

<table>
<thead>
<tr>
<th>Shared Tablespace/Index Exceptions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOSTATS . . . N</td>
</tr>
<tr>
<td>BMCSTATS Age .</td>
</tr>
<tr>
<td>Index Exceptions:</td>
</tr>
<tr>
<td>NONUNIFORM . .</td>
</tr>
<tr>
<td>ROWS/KEY . . .</td>
</tr>
</tbody>
</table>

---

Related Information

- “Statistics and miscellaneous related exceptions” on page 453

---

Specifying Copy-Related Override Exceptions

This topic describes the Copy-Related Override Exceptions options.

Table 63 on page 455 provides descriptions of the exceptions. The DASD MANAGER PLUS for DB2 Reference Manual provides formulas for the exceptions.

Figure 120 on page 495 shows the Copy-Related Override Exceptions options panel.
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 120: Copy-Related Exceptions options**

<table>
<thead>
<tr>
<th>DB2 RTS Specific Copy Exceptions:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Tablespace/Index Exceptions:</td>
<td></td>
</tr>
<tr>
<td>Reorg after copy N</td>
<td>(Y/N - Reorg after last copy)</td>
</tr>
<tr>
<td>Load after copy N</td>
<td>(Y/N - Load after last copy)</td>
</tr>
<tr>
<td>Mods since copy</td>
<td>(1-100 % rows modified since copy)</td>
</tr>
<tr>
<td>No RTS copy data N</td>
<td>(Y/N No Copy data in RTS tables)</td>
</tr>
<tr>
<td>Index Exceptions:</td>
<td></td>
</tr>
<tr>
<td>Rebd after copy N</td>
<td>(Y/N - Rebuild after last copy)</td>
</tr>
<tr>
<td>Dirty Pages</td>
<td>(1-100 % of modified index pages)</td>
</tr>
</tbody>
</table>

| DB2 RTS or BMCSTATS Copy Exceptions: |  |
| Shared Tablespace/Index Exceptions: |  |
| Copy Pend ... N                    | (Y/N Copy pending flag is on) |
| ICOPY pending Y                    | (Y/N - Advisory Informational COPY pending) |
| Tablespace Exceptions:            |  |
| Dirty Pages                       | (0-99% Percentage of modified pages) |
| Image Copy Age                    | (0-32766 Days since last copy) |
| Full Copy Age                     | (0-32766 Days since last full copy) |
| NumIncremt ...                    | (1-99 Incremental copies since last full) |
| DtyIncremt ...                    | (0-99% Dirty pages to trigger incremental copy) |
| Index Exceptions:                 |  |
| Image Copy Age                    | (0-32766 Days since last copy of index) |

### Generating a BMCTRIG job

Use this procedure to specify and generate the JCL for a BMCTRIG job. After you create an action to run BMCTRIG (and BMCSTATS), you generate the JCL.

1. From the DASD MANAGER PLUS main menu, select **Service Actions**.

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 Job Generation panel appears. (See “Generating Service Actions” on page 248).

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 BMCTRIG worklist (Figure 121 on page 496) is displayed.

**Figure 121: BMCTRIG worklist**

```
| -TIME 000000 'yyyy-08-03-13.46.42.00004'
| -SSID 000001 DEAE
| -WKID 000002 EXCEPT1
| -SYNC 000003
| -BMCU 000004 ASUSMAIN
| BMCTSTATS TABLESPACE TZU%.% TABLE (ALL)
| -BMCU 000005 ASUSTRIG
| BMCTSTATS TABLESPACE TZU%.% SYSTEMTRIGGERS Y

SPACE 30 COPYAGE 7 FAROFF 20 LEVELINC Y IXCOPYAGE 10 CATSTATAGE 30
STDJCL Y JCLGEN Y NUMJOBS 5 BEGJOBSEQ 001 UTILJOB MYREORG UTILSEQ 0001
STDJCLDSN 'ASU.DEAE.V710QA.RGRTRIG.TRIGNWJ'
UTILITYDSN '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.

You can also create an exceptions report (see “Statistics exception report” on page 576), view exceptions online, or view active exceptions.

8 (optional) Add the job to your scheduler to run routinely.

**Note**

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.
Using worklist format or standard JCL format

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 format 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 the table below when you use standard JCL:

Table 73: Standard JCL tasks

<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 501.</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 491.</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 122: Generating standard JCL

Standard format JCL—more information

This topic provides the following information on standard format JCL:
- “Standard format JCL generation options” on page 498
- “Example procedures” on page 499
- Interim worklist on page 499
- “Job card exit and job name” on page 501

Standard format JCL generation options

The BMCTRIG JCL Generation panel (Figure 126 on page 509) shows the JCL generation options that are required for creating standard JCL. The required options are as follows:
- Standard JCL = Y
- Data set names:
Example procedures

The interim worklist and generated JCL that Figure 123 on page 500 and Figure 124 on page 503 show are the output of procedures in the following tasks:

- “To set up a REORG corrective action” on page 525
- “To generate the BMCTRIG service that uses the REORG corrective action” on page 531

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 “Setting up DASD MANAGER PLUS” on page 63.

The interim worklist contains worklist commands for standard JCL, including the commands shown in the table below.

Table 74: 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>

The figure below 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 123: Interim worklist

```
-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)
- 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
- REOR 000003
REORG TABLESPACE QZUD40.QZUS0140
UNLDDN S0002
WORKDDN SYSUT001
SORTDATA
SORTDEVT SYSDA
UNLOAD CONTINUE
- COPY 000004
COPY TABLESPACE QZUD41.QZUS0141
SHRLEVEL REFERENCE
COPYDDN (C0003)
RECOVERYDDN (R0003)
- JCLP 000004 COPY DDNAME C0003 DSNPREF
RDAJXN.DEAE.LP.&OBNOD
- JCLP 000004 COPY DDNAME C0003 DEVTYPE SYSDA
- JCLP 000004 COPY DDNAME R0003 DSNPREF
RDAJXN.DEAE.RP.&OBNOD
- JCLP 000004 COPY DDNAME R0003 DEVTYPE SYSDA
- REOR 000005
REORG TABLESPACE QZUD41.QZUS0141
UNLDDN S0004
```
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. The table below describes the variables and how to use them.
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.

### Table 75: 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>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></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></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.

**Generated standard JCL**

The generated standard JCL in Figure 124 on page 503 shows the structure of the interim worklist, described in Table 76 on page 503.
Table 76: Generated interim worklist structure

<table>
<thead>
<tr>
<th>Job</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First job</td>
<td>The job illustrated in the figure below (Generated standard JCL) is the first of the four jobs of the worklist. The job consists of four services, one for each utility and object combination:</td>
</tr>
<tr>
<td></td>
<td>■ COPY QZUD40.QZUS0140</td>
</tr>
<tr>
<td></td>
<td>■ REORG QZUD40.QZUS0140</td>
</tr>
<tr>
<td></td>
<td>■ COPY QZUD41.QZUS0141</td>
</tr>
<tr>
<td></td>
<td>■ REORG QZUD41.QZUS0141</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Second job</td>
<td>The second job consists of the following services:</td>
</tr>
<tr>
<td></td>
<td>■ COPY QZUD42.QZUS0142</td>
</tr>
<tr>
<td></td>
<td>■ REORG QZUD42.QZUS0142</td>
</tr>
<tr>
<td>Third job</td>
<td>The third job has one step, a IEFBR14</td>
</tr>
</tbody>
</table>

Figure 124: Generated standard JCL

```
//TBMC001 JOB (5213), 'DASDTRIG-STDJCL',
// CLASS=A, MSGCLASS=X
/*/ JOBPARM SYSAFF=DB2A
/*/ /*** **********************************************
/*/ ** CREATED BY: MVSTAD1
/*/ ** TIMESTAMP: 03/09/2016.15.25.07
/*/ ** ENVIRONMENT: ISPF 7.1MVS   TSO
/*/ ** RELEASE: V12.01.00 05/06/2016
/*/ ** DB2 VERSION: XEPM
/*/ **********************************************
//STEP1 EXEC PGM=DSNUTILB,
// ** PARM=(DEAE,'TBMC001.BMCTR',
// **     RESTART),
// **     RESTART),
// ** PARM=(DEAE,'TBMC001.BMCTR',
// ** REGION=0M, COND=(4,LT)
//STEP1B 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=* 
//SYSDUMP DD SYSOUT=* 
//UPTPRINT DD SYSOUT=* 
//SYSDOUT DD SYSOUT=* 
//SYSIN DD *
COPY TABLESPACE QZUD40.QZUS0140
SHRLEVEL REFERENCE
COPYDDN (C0001)
RECOVERYDDN (R0001)
*/ ** UTILITY COPY DD STATEMENTS
*--------------------------------------------------------------------
* 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*
```
Setting up a BMCTRG job

```
// SPACE=(CYL,(21,5),RLSE), ESTIMATE-C/H
// UNIT=SYSDA
//
// END OF JOBSTEP
//
//STEP2 EXEC PGM=DSNUTILB,
//  PARM=(DEAE,'TBMC001.BMCTR',
//         RESTART),
//  PARM=(DEAE,'TBMC001.BMCTR'),
//  REGION=0M,COND=(4,LT)
//STELIB DD DISP=SHR,DSN=DB2.MSTRPLAN.LOAD
//        DD DISP=SHR,DSN=SYS3.DEAE.DSNEXIT
//        DD DISP=SHR,DSN=SYS2.DB2V71M.DSNLOAD
//ABNLIGNR DD DUMMY
//DSSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//UTPRINT  DD SYSOUT=*  
//SYSIN DD *
//REORG TABLESPACE QZUD40.QZUS0140
//UNLDN S0002
//WORKDDN SYSUT001
//SORTDATA
//SORTDEVT SYSDA
//UNLOAD CONTINUE
//
// END OF JOBSTEP
```

```
// 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
//
// END OF JOBSTEP
//
//STEP2 EXEC PGM=DSNUTILB,
//      PARM=(DEAE,'TBMC001.BMCTR',
//            RESTART),
//      PARM=(DEAE,'TBMC001.BMCTR'),
//      REGION=0M,COND=(4,LT)
//STELIB DD DISP=SHR,DSN=DB2.MSTRPLAN.LOAD
//  DD DISP=SHR,DSN=SYSDA
//  DD DISP=SHR,DSN=SYSS.DBE52.V71M.DSNLOAD
//  DD DISP=SHR,DSN=AUS.ASU810.D71.LOAD
//ABNLIGNR DD DUMMY
//DSSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//UTPRINT  DD SYSOUT=*  
//SYSIN DD *
//REORG TABLESPACE QZUD40.QZUS0140
//UNLDN S0002
//WORKDDN SYSUT001
//SORTDATA
//SORTDEVT SYSDA
//UNLOAD CONTINUE
//
// END OF JOBSTEP
```

```
// DATA SET SIZE REQUIRED FOR DD S0002 IS AN ESTIMATE
//S0002 DD DSN=RDAJXN2.TBMC001.SYSREC.S0002,
//      DISP=SHR, ***RESTART*
//      DISP=(NEW,CATLG,CATLG), ***INITIAL*
//      SPACE=(CYL,(21,5),RLSE), ESTIMATE-C/H
//      UNIT=SYSDA
//
// END OF JOBSTEP
//
//DATA SET SIZE REQUIRED FOR DD SYSUT001 IS AN ESTIMATE
//SYSUT001 DD DSN=RDAJXN2.TBMC001.STEP2.SYSUT001,
//      DISP=SHR, ***RESTART*
//      DISP=(NEW,CATLG,CATLG), ***INITIAL*
//      SPACE=(CYL,(45,11)), ESTIMATE-C/H
//      UNIT=SYSDA
//
// END OF JOBSTEP
//
//DATA SET SIZE REQUIRED FOR DD SORTOUT IS AN ESTIMATE
//SORTOUT DD DSN=RDAJXN2.TBMC001.STEP2.SORTOUT,
//      DISP=SHR, ***RESTART*
//      DISP=(NEW,CATLG,CATLG), ***INITIAL*
//      SPACE=(CYL,(45,11)), 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.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'),
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=* SYSIN DD *
SYSPRINT DD SYSOUT=* SYSPRINT DD SYSOUT=* SYSDUMP DD SYSOUT=* UTPRINT DD SYSOUT=* SYSDUMP DD SYSOUT=* SYSIN DD *
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,
//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=*  
//SYSOUT DD SYSOUT=*  
//SYSIN DD *
REORG TABLESPACE QZUD41.QZUS0141
UNLDDN S0004
WORKDDN SYSUT001
SORTDATA
SORTDEV SYSDA
UNLOAD CONTINUE

/* UTILITY UNLOAD DD STATEMENTS */

/* DATA SET SIZE REQUIRED FOR DD S0004 IS AN ESTIMATE */
S0004 DD DSN=RDAJXN2.TBMC001.SYSREC.S0004,
  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 SYSUT001 IS AN ESTIMATE */
SYSUT001 DD DSN=RDAJXN2.TBMC001.STEP4.SYSUT001,
  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 */
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 77 on page 507.

Table 77: 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>
</tbody>
</table>
To generate worklist JCL instead of standard JCL, specify **Standard JCL N** on the JCL Generation panel of the BMCTРИG utility when you are setting the BMCTРИG 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

The figure below illustrates worklist JCL generation.

**Figure 125: Generating worklist JCL**
JCL generation options for worklist format JCL

The BMCTRIG JCL generation options are displayed under the multiple sections beginning with Job Generation Options on the BMCTRIG Syntax Options panel.

Figure 126: Worklist JCL Generation options panel

For worklist format, type N in the Standard JCL field.

For information about all of the other JCL generation options, see the DASD MANAGER PLUS for DB2 Reference Manual.

If you want 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. The UTDEF DD must reference the XML data set by default instead of the CLIST.
Worklist format JCL—more information

This topic provides the following information on worklist format JCL:
- “Generated standard JCL” on page 502
- “Worklist format JCL example procedures on page 510”
- “Generating worklist format JCL” on page 513

Worklist format JCL generation options

Figure 127: Worklist JCL Generation options panel

<table>
<thead>
<tr>
<th>Command</th>
<th>BMCTRIG TABLESPACE TZU%.%</th>
<th>Service Syntax: BMCTRIG.DEMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDEDK</td>
<td></td>
<td>Enter data, then press end.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>More: + -</td>
</tr>
<tr>
<td></td>
<td>Default Action</td>
<td>(Default corrective action to generate)</td>
</tr>
<tr>
<td></td>
<td>AUTOSUBMIT . . . . N</td>
<td>(Y/N - Automatically Submit the Util DSN)</td>
</tr>
<tr>
<td></td>
<td>Worklist only . . N</td>
<td>(Y/N - Generate only the worklist)</td>
</tr>
<tr>
<td></td>
<td>When No Objects S</td>
<td>(S/C - S-Stop, C-Continue)</td>
</tr>
<tr>
<td></td>
<td>Standard JCL . . N</td>
<td>(Y/N - Standard JCL, N=Worklist JCL)</td>
</tr>
<tr>
<td></td>
<td>Max Objects . . 20</td>
<td>(1-255 - Maximum number of objects per job)</td>
</tr>
<tr>
<td></td>
<td>Max Steps . . 255</td>
<td>(1-255 - Maximum number of steps per job)</td>
</tr>
<tr>
<td></td>
<td>Number of Jobs</td>
<td>(0-46655 - Number of jobs to generate)</td>
</tr>
<tr>
<td></td>
<td>Workload Balance N</td>
<td>(Y/N - Balance work across jobs)</td>
</tr>
<tr>
<td></td>
<td>Util DSN . . . . . .</td>
<td>Note: Interim WL DSN is required when Standard JCL=Y.</td>
</tr>
<tr>
<td></td>
<td>Interim WL DSN . . .</td>
<td>Util DSN (JCL target dataset) is always required.</td>
</tr>
<tr>
<td></td>
<td>------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td></td>
<td>Jobcard(s) for BMCTRIG generated jobs(s)</td>
<td>Job Sequence Type N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(N/A/blank - N=Numeric, A=Alpha blank=Numeric)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Beginning Seq . . .</td>
</tr>
<tr>
<td></td>
<td>Jobcard1 . . . . . JOB1 JOB (JZACCTNUM.&quot;&amp;PGMR&quot;),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jobcard2 . . . . . CLASS=A,MSGCLASS=X,</td>
<td></td>
</tr>
</tbody>
</table>
|         | Jobcard3 . . . . . /*
|         | Jobcard4 . . . . . /*
|         | Jobcard5 . . . . . /* |

Worklist format JCL example procedures

The worklist format job that (Figure 128 on page 510) shows is the output of procedures that are described in the following tasks:
- “To set up a REORG corrective action” on page 525
- “To generate the BMCTRIG service that uses the REORG corrective action” on page 531

Figure 128: Generated worklist JCL

```
//RDAJXN2U JOB (5213), 'UTILITY-JENTEST',
// CLASS=A, MSGCLASS=X
/*
*JOBPARM SYSAFF=DB2A
*/
```

---

510  DASD MANAGER PLUS for DB2 User Guide
//STEP1 EXEC PGM=AEXEMAIN,REGION=0M,
//             PARM='DS121EAE'
//STEPLIB DD DISP=SHR,DSN=AUS.ASU121.D11.LOAD
//        DD DISP=SHR,DSN=SYS3.DEAE.DSNEXIT
//        DD DISP=SHR,DSN=SYS2.DB2V11M.DSNLOAD
//* ----------------------------------------------------------------- *
//ABNLIGNR DD DUMMY
//DSSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=* 
//SYSTERM  DD SYSOUT=* 
//UTPRINT  DD SYSOUT=* 
//SYSOUT   DD SYSOUT=* 
//AEXIN DD * ASU
SSID DEAE
DASDTRIG LINES 54 STATS
UTILITYID TJENTEST.S001
COPYOPT  ACP$OPTS
UNLOADOPT  ADU$OPTS
LOADOPT  AMU$OPTS
REORGOPT ARU$OPTS
CHECKOPT ACK$OPTS
RECOVEROPT AFR$OPTS
//SYSIN DD SPACE=(CYL,(15,15)),UNIT=SYSDA,DISP=(NEW,PASS),
//        DSORG=PS,LRECL=3200,RECFM=FB
//SYSPRINT DD SPACE=(CYL,(15,15)),UNIT=SYSDA,DISP=(NEW,PASS)
*/JGENSRPT DD SYSOUT=*
//SYSTSIN DD SPACE=(TRK,(1,1)),UNIT=SYSDA,DISP=(NEW,PASS),
//        DSORG=PS,LRECL=3200,RECFM=FB
//SYSTSPRT DD DISP=(NEW,PASS),
//        DSN=%&TSPRT1,
//        SPACE=(CYL,(5,5)),UNIT=SYSDA,
//        DSORG=PS,LRECL=3155,RECFM=VBA
//AEXPRINT DD SYSOUT=* 
//ACPGDGLP DD *
//DEFINE GDG (NAME(&BASE) LIMIT(10) SCR)
//ACPGDGLBP DD *
//DEFINE GDG (NAME(&BASE) LIMIT(10) SCR)
//ACPGDGRP DD *
//DEFINE GDG (NAME(&BASE) LIMIT(10) SCR)
//ACPGDGRB DD *
//DEFINE GDG (NAME(&BASE) LIMIT(10) SCR)
//WORKLIST DD *
-TIME 000001 'yyyy-03-06-11.19.25.628396' 57489450
* FOR TRIGGER EXCEPTION 25145407
74734598
-COPY 000001
COPY TABLESPACE JEND30.C791278
SHRLEVEL REFERENCE
CONCURRENT
COPYDDN (C0001)
-JCLP 000001 COPY DDNAME C0001 DSNPREF &PREFIX..&SSID..&OBIND
-JCLP 000001 COPY DDNAME C0001 DEVTYPE SYSDA

-REOR 000002
REORG TABLESPACE JEND30.C791278
UNLDDN S0002
WORKDDN SYSUT001
SORTDATA
SORTDEVT SYSDA
UNLOAD CONTINUE

-COPY 000003
COPY TABLESPACE JEND30.JENS0130
SHRLEVEL REFERENCE
CONCURRENT
COPYDDN (C0003)

-REOR 000004
REORG TABLESPACE JEND30.JENS0130
UNLDDN S0004
WORKDDN SYSUT001
SORTDATA
SORTDEVT SYSDA
UNLOAD CONTINUE

//**  UTILITY UNLOAD DD STATEMENTS
//**  ---------------------------------
//**  M = ESTIMATE IS THE MAXIMUM DATASET SIZE FOUND FOR THIS WORKDD TYPE
//SO002 DD DSN=RDAJXN2.JENTEST.SYREC.SO002,  
//     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
//SO004 DD DSN=RDAJXN2.JENTEST.SYREC.SO004,  
//     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  
//     UNIT=SYSDA
//**  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
// ** ALTERN 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
// **----------------------------------------------------------
// ** END OF DATASET CLEANUP STEP
// **----------------------------------------------------------
// ** END OF JOB
// **----------------------------------------------------------

Generating worklist format JCL

The generated worklist JCL (see the figure in “Worklist format JCL” on page 507) 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

1. 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 129 on page 514 shows a sample list of exceptions.

**Figure 129: 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 Action Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC16673</td>
<td>BSTATAGE</td>
<td>TDU1</td>
<td>IX</td>
<td>0</td>
<td>N/A</td>
<td>0000030</td>
</tr>
<tr>
<td>BMC16673</td>
<td>COPYAGE</td>
<td>TDU1</td>
<td>TS</td>
<td>0</td>
<td>N/A</td>
<td>0000014</td>
</tr>
<tr>
<td>BMC16673</td>
<td>REORSPAC</td>
<td>TDU1</td>
<td>TS</td>
<td>0</td>
<td>N/A</td>
<td>0000030</td>
</tr>
<tr>
<td>BMC16673</td>
<td>DIRTY</td>
<td>TDU4</td>
<td>TS</td>
<td>99</td>
<td>N/A</td>
<td>0000020</td>
</tr>
<tr>
<td>BMC16673</td>
<td>DIRTY</td>
<td>TDU1</td>
<td>TS</td>
<td>99</td>
<td>N/A</td>
<td>0000020</td>
</tr>
<tr>
<td>BMC16673</td>
<td>DSEXTENT</td>
<td>TDU1</td>
<td>TS</td>
<td>0</td>
<td>N/A</td>
<td>0000030</td>
</tr>
<tr>
<td>BMC16673</td>
<td>SPACE</td>
<td>TDU1</td>
<td>TS</td>
<td>0</td>
<td>N/A</td>
<td>0000030</td>
</tr>
<tr>
<td>BMC16673</td>
<td>SPACE</td>
<td>TDU1</td>
<td>TS</td>
<td>0</td>
<td>N/A</td>
<td>0000030</td>
</tr>
<tr>
<td>BMC16673</td>
<td>SPACE</td>
<td>TDU1</td>
<td>TS</td>
<td>0</td>
<td>N/A</td>
<td>0000030</td>
</tr>
<tr>
<td>BMC16673</td>
<td>SPACE</td>
<td>TDU1</td>
<td>TS</td>
<td>0</td>
<td>N/A</td>
<td>0000030</td>
</tr>
</tbody>
</table>

Figure 129 on page 514 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 “Producing reports” on page 547.

To view exceptions by object online

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

2 On the Collect and Manage Statistics panel, select List BMC and DB2 Statistics.

3 In the relevant field, type the object name or pattern for objects that you want to view, and press Enter.

   The Display DB2 Object List panel appears.

4 Type E in the Act field next to an 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 **Object Exceptions** and press **Enter**.

2 On the Detect and Correct Object Exceptions panel, select **View logged Exceptions needing Corrective Action generation**.

The Active Exceptions List panel is displayed.

```
DEAE --------------------- Active Exceptions List ----
Command ===>                                  Scroll ===> PAGE

Active Exceptions are logged Exceptions without Corrective Action generation. Use BMCTRIG resume processing on the Objects to generate maintenance JCL. Change display by changing display filter, then press Enter.

Display filter:
Object Name . %.% (ALL, TS, IX, or TB)
Object Type . ALL (0 for all parts, or exact part number)
Part . . . . 0
Exception . %
D =Delete  P =Properties

More:       >
Current     Compare
Act Object                               Part  Exception     Value       Value
******************************** Top of data ********************************
IX QCH.QCHX02_DE01S09T02                0  IXICAGE            0           0
******************************* Bottom of data ********************************
```

3 Edit the selection criteria, and then press **Enter** to retrieve the rows.

By default, you see all active exceptions. You might want to limit the selection criteria.

4 Type **P** in the **Act** field next to an exception type to view or edit it.

The Object Exception Edit panel is displayed.

```
DEAE -------------------------- Object Exception Edit -------------------------
Command ===>

Type Object Exception data. Then press End.

Object Name . . . : TZUD10.TZUS0110
Object Type . . . : TS
Partition . . . . : 1
Exception Name  . : UNCLUST
Exception Type  . : REORG
Current Value . . : 40
Compare Value . . : 0
Trigger Value . . : 25
Timestamp . . . . : 2015-05-06-15.18.56.961076
Updated By . . . . : RDAMLC
Updated Timestamp : 2015-05-06-16.15.32.363933
JCL Dataset . . . :
Action Taken . . .:
Registration ID . : 0
Active . . . . . : Y        (Y/N)
Priority . . . . : 0        (0-255)
Corrective Action . : (%=List or Create)
```

5 *(optional)* In the **Active** field, to deactivate the exception, specify **N**.
Note
You cannot deactivate an exception that is currently registered to the automation component and is awaiting execution. An exception is registered if the Registration ID field contains a value other than zero. You must use the Automation component to delete an active registration. For more information about the automation component, see “Solution integration” on page 36.

6 (optional) Alter the priority or corrective action if you want to use BMCTRIG resume generation later to generate the corrective action.

Note
If you use BMCTRIG RESUME Y processing, BMCTRIG selects active exception rows for the identified objects and uses the priority and corrective action from the exceptions table to continue processing.

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 \( \text{%Pseudo-deleted Keys} \) 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:

  \[
  \frac{\text{RS_INDEXPART.PSEUDO_DEL_ENTRIES}}{\text{RS_INDEXPART.CARDF}} \geq \text{specifiedValue}
  \]

  - 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:

\[
\frac{\text{RS\_INDEXPART\_LEAFNEAR} + \text{RS\_INDEXPART\_LEAFFAR}}{\text{RS\_INDEXPART\_NLEAF}} \geq \text{specifiedValue}
\]

— 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:

\[
\frac{\text{RS\_INDEXPART\_LEAFFAR}}{\text{RS\_INDEXPART\_NLEAF}} \geq \text{specifiedValue}
\]

— 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 SYSEMTRIGGERS Y. For additional information about creating and adding thresholds, see Thresholds in BMCTRIG syntax on page 457.

---

**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 a Service Action” on page 241.
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 “Copying a Service Action” on page 246.

3 On the Action List panel, type E 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 “Job generation” on page 432:
   - **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** until you return to the DASD MANAGER PLUS main menu.

16 Set up the related corrective action as described in “Setting up a REORG corrective action” on page 524.

17 Press **END** until you return to the DASD MANAGER PLUS main menu.

18 On the DASD MANAGER PLUS main menu, select **Service Actions** and press **Enter**.
19 On the Action Menu panel, select the action that you created in “To generate an automatic reorganization with BMCTRIG” on page 521.

20 Generate the BMCTRIG job as described in To generate the BMCTRIG service that uses the REORG corrective action on page 531.

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 576.

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 “Setting a corrective action in BMCTRIG syntax” on page 470.) Be aware that, unlike a regular action, a corrective action specifies actions but often does not specify the DB2 objects to process.

**Note**

The corrective action (formerly known as a skeleton work ID) in this procedure runs the REORG utility with the inline copy feature. Typically, you use this type of corrective action for thresholds that indicate when to perform a reorganization. Use this procedure as a guideline, and alter the action to meet your needs.

If you do not use the inline copy feature, specify two services: one step for COPY and one for REORG.

Also, you can use the symbolic variables in Table 78 on page 524 in the mapping table for IBM REORG and BMCREORG when defining service syntax options:

**Table 78: Supported REORG mapping table variables**

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>TSNAME</td>
<td>Table space name</td>
</tr>
<tr>
<td>JOBNMBR</td>
<td>Job number</td>
</tr>
<tr>
<td>WORKID</td>
<td>Work ID (maximum of eight characters)</td>
</tr>
</tbody>
</table>
To set up a REORG corrective action

1. Create a BMCTRIG job as described in “Generating an automatic reorganization with BMCTRIG” on page 521.

2. Create an action as described in “Creating a Service Action” on page 241. When you create the action, select REORG from the Service Selection List panel.

   Note
   This procedure uses an action named SKREORG.

3. Type E (or S) in the Act field of the REORG service.

   The Service Syntax List appears.

4. Create the syntax or select existing syntax and press Enter.

   The REORG panel is displayed.

5. 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 79 on page 525 lists the choices for this field.

Table 79: 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>
</tbody>
</table>
**Note**

You must set `Shrlevel=N` for LOB table spaces.

`Shrlevel=C` (Change) are not generated on an object that is not logged. Instead, this option is changed automatically to `Shrlevel=R` (Reference).

Table 80 on page 526 describes the Shrlevel options.

### Table 80: Shrlevel options

<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>
<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. YES 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>
6 Specify Reorg options.

The Reorg Options panel is displayed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REUSE</td>
<td>Type Y 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 Y 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 is ignored for LOB, XML and XML base objects.</td>
</tr>
<tr>
<td>LOG</td>
<td>Type Y to log records during the RELOAD phase of reorganization. If you type N, the product does not log records and turns on the copy pending restriction.</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>Use the default value C to specify continue after unloading data.</td>
</tr>
<tr>
<td>KEEPDICTIONARY</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>ROWFORMAT</td>
<td>Specifies the output row format in the affected table space or partition</td>
</tr>
<tr>
<td></td>
<td>BRF specifies that the reorganized or replaced table space or partition being is converted to (or remains in) basic row format</td>
</tr>
<tr>
<td></td>
<td>RRFSpecifies that the reorganized or replaced table space or partition is converted to (or remains in) reorder row format</td>
</tr>
</tbody>
</table>

Table 81 on page 527 describes the Reorg options.

Table 81: Reorg options
Type N, T, I, or A. Table 82 on page 530 shows valid values for the Resize field. The Resize option is valid only for a corrective action that a BMCTRG service references. Selection of DB2 data sets for resizing is based on the latest BMCSTATS data. The primary space quantity expands or contracts based on the values for the REORGSPACE and secondary quantity statistics, as in the following formula:

\[
\text{NEW PRIQTY} = \text{REORGSPACE} + (2 \times \text{SQTY})
\]

For BMCTRG, you can specify other options to control the calculation. Also, the RESIZE exit is available.

Use the default N to avoid generating RESTART PHASE parameters in the worklist for this reorganization step. Specifying **Restart Y** might be useful if a previous reorganization step specified **Unload = Pause**.

For hash-organized table spaces, specifies that REORG automatically calculates and formats the size of the fixed hash space. Specifying **AUTOESTSPACE Y** might reduce the number of rows in the overflow area.

- **Y YES**— (default) Specifies that DB2 uses real time statistics (RTS) values to adjust the size of the hash space.
- **N NO**— Specifies that DB2 uses the HASH SPACE value specified for CREATE TABLE or ALTER TABLE.

These values are stored in the SYSIBM.SYSTABLESPACE and SYSIBM.SYSTABLEPART catalog tables.

Specify the Copy options.

The Copy DD Options panel is displayed.

```plaintext
Copy DD Options

FLASHCOPY . . . N (N/Y/C N-No, Y-Yes, C-Consistent Flash Copy)

COPYDDN dd1  . . N (N/J/D N-None, J-JCL, D-Dynamic allocation)
COPYDDN dd2  . . N (N/J/D N-None, J-JCL, D-Dynamic allocation)
RECOVERYDDN dd1 N (N/J/D N-None, J-JCL, D-Dynamic allocation)
RECOVERYDDN dd2 N (N/J/D N-None, J-JCL, D-Dynamic allocation)
FCCOPYDDN dd1 . . N (N/D N-None, D-Dynamic allocation)

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. The first panel, Figure 130 on page 529, appears.

**Figure 130: COPYDDN ddname1 JCL panel**

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>COPYDDN ddname1 JCL</th>
<th>Row 1 to 10 of 10</th>
<th>Scroll ===&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Syntax: REORG.DEMO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enter data, then press end.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Blank fields default to the Workid JCL Options.

-------------------------- JCL Options ----------------------------------------

| UNIT . . . . . .          | (Device type / Unit) |
| Dname Prefix . .          | (Prefix for output) |
| RETPD . . . . .           | (1-9999 retention period) |
| EXPDT . . . . .           | (YYYY/DDD or YYDDD Expiration date) |

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.

8 Specify the Sort options.

**Figure 131 on page 529** shows the Sort options.

**Figure 131: Sort options panel**

| Sort JCL . . . . N | (Y/N Y-Put sortwork DD in JCL or N-Set SORTNUM) |
| SORTNUM . . . .    | (0-255 Number of SORTWK DDs or SORTNUM value) |
| SORTDEVT . . . .   | (Device type for temporary sort data sets) |
| PARALLEL . . . .   | (Y/N Rebuild Indexes or partitions in parallel) |

The following table describes the Sort options:
Table 82: Sort options

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort JCL</td>
<td>Type <strong>Y</strong> to generate DD statements for the SORTWK data sets. If you type <strong>N</strong>, the sort allocates the data sets dynamically. If you specify <strong>Y</strong>, 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 <strong>Y</strong> to sort the index keys and rebuild more than one index in parallel, as opposed to sequentially. This approach improves performance.</td>
</tr>
</tbody>
</table>

9 Specify statistics options if you want to collect inline statistics.

The panel below displays 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>
</tbody>
</table>

10 Specify the Index Distribution Statistics.

The panel below displays the Index Distribution Statistics options.

<table>
<thead>
<tr>
<th>Index Distribution Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency Stats:</td>
</tr>
<tr>
<td>NUMCOLS . . . . 1</td>
</tr>
<tr>
<td>Auto Decrement N</td>
</tr>
<tr>
<td>COUNT . . . . . . 10</td>
</tr>
</tbody>
</table>

11 Specify the DB2 Catalog Reporting and Update Options.

The panel below displays the DB2 Catalog Reporting and Update Options.

<table>
<thead>
<tr>
<th>DB2 Catalog Reporting and Update Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT . . . . . N</td>
</tr>
<tr>
<td>UPDATE . . . . . A</td>
</tr>
<tr>
<td>HISTORY . . . . .</td>
</tr>
<tr>
<td>FORCEROLLUP . . .</td>
</tr>
</tbody>
</table>

12 Specify Conditional Reorg options.

The panel below displays the Conditional Reorg options.

<table>
<thead>
<tr>
<th>Conditional Reorg Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFFPOSLIMIT . . . 10</td>
</tr>
<tr>
<td>INDFRFLIMIT . . . 10</td>
</tr>
<tr>
<td>REPORTONLY . . . N</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.

13 Press END until you return to the Edit Action Services panel.

**Figure 132: Edit Action Services panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Action: BMC.SKREORG</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEAE</td>
<td>Edit Action Services ---</td>
</tr>
<tr>
<td></td>
<td>Row 1 to 1 of 1</td>
</tr>
<tr>
<td></td>
<td>Scroll ===&gt; PAGE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Listed is the ordered set of Services contained in the Action. Select one or more options. Then press Enter.</td>
</tr>
<tr>
<td></td>
<td>Action: BMC.SKREORG</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D =Delete I =Insert L =Like A =After B =Before M =Move P =Properties</td>
</tr>
<tr>
<td></td>
<td>S =Syntaxes O =Syntax Options V =View Syntax U =Use default syntax</td>
</tr>
<tr>
<td></td>
<td>OS =Object Set</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>More: &gt;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Act Service</td>
</tr>
<tr>
<td></td>
<td>REORG</td>
</tr>
</tbody>
</table>

14 Generate the BMCTRIG service that refers to this corrective action.

See “Generating an automatic reorganization with BMCTRIG” on page 521.

**To generate the BMCTRIG service that uses the REORG corrective action**

1 Set up a BMCTRIG job as described in “Creating an action with BMCTRIG” on page 477.

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 “Setting up a BMCTRIG job” on page 476.

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 Object Exceptions panel option **Restart a WORKLIST format job that was generated by BMCTRIG**
- Manually
- In batch
- By CLIST

**To run the Restart program online by using ISPF**

1. Run the program from the Object Exceptions panel 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**

1. 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

**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 to, the program assumes that the point of restart advances. Repeated restarts reduce your ability to restart at a previous restart point.

**Note**

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. The table below describes the data set dispositions that the Restart program assigns for Startover processing.

**Table 83: JCL DD statement dispositions for startover**

<table>
<thead>
<tr>
<th>Data set</th>
<th>Disposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nontape</td>
<td>Changed to OLD,KEEP,KEEP</td>
</tr>
<tr>
<td>Tape</td>
<td>Unchanged</td>
</tr>
</tbody>
</table>

Relative generation numbers for GDGs remain unchanged for Startover processing.

**Overview of the restart program**

The Restart program runs against the BMCTRIG-generated job stream, modifying the worklist, execution parameters, and JCL.
The following figure illustrates how the program writes the modifications to a separate JCL output file.

**Figure 133: Online restart**

![Diagram showing online restart process]

**Table 84: JCL DD statement dispositions for restarts**

<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>

**WARNING**

The data set dispositions in this table assume that the initial run of the JCL job cataloged the data sets. If the initial run did not catalog the data sets, you must determine the VOLSER references and the appropriate dispositions. BMCTRIG-generated jobs that contain stacked tape data sets might require manual editing to comply with local Tape Management System (TMS) requirements.

**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 non-tape 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 RSTRIG 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.

Use this procedure to restart a job that BMCTRIG generates when you specify Standard JCL N on the BMCTRIG main menu.

**To restart a BMCTRIG-generated job**

1. On the DASD MANAGER PLUS main menu, select BMCTRIG Generated Job Restart.
Figure 134 on page 536 appears.

**Figure 134: BMCTRIG Generated Job Restart panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>DEAE ---------------------- BMCTRIG Generated Job Restart ---------------------</th>
</tr>
</thead>
</table>

Use this panel to restart jobs generated by the BMCTRIG Service only.

- **Restart:**
  - JCL Type . . . . (1 =Build Restart JCL, 2 =Build Startover JCL)
  - Worklist Seq . .
  - Restart Parm . .

- **Input BMCTRIG Generated Job JCL:**
  - Data Set Name . .

- **Output Restart Job JCL:**
  - Data Set Name . .

Select one or more Processing options. Then press Enter.

S Edit JCL
Submit JCL

2 Specify the restart options as follows:

a In the **JCL Type** field, type 1 or 2:
   - 1 restarts the worklist at a specific command sequence number.
   - 2 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.

c If you specified **JCL Type 1**, use the **Restart Parm** 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 JCL** field, type S to display the output JCL job stream for review or edit.

b In the **Submit JCL** field, type S to submit the JCL automatically.

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 JCL** to S.

   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 135 on page 537. Restart processing follows the algorithms described in the chapter that discusses "Execution" in the *DASD MANAGER PLUS for DB2 Reference Manual*.

**Figure 135: Sample AEXIN_DD input stream for BMCTRIG JCL restart**

```plaintext
//AEXIN DD *
ASU
SSID DB25
DASDTRIG
EVENTS LINES 54 STATS
VCAT DB25CAT
UTILITYID TP263010.S088 RESTART
RESTARTPARM
```

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

The following figure shows sample JCL for batch restart processing of a BMCTRIG-generated job.

For commented RESTART EXEC parameters, see Figure 136 on page 538.

Figure 136: Sample JCL for batch restart processing

```plaintext
//JOB CARD 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          *
//******************************************************************
//STEP01 EXEC PGM=ASUBRST,
//******************************************************************
//* BATCH RESTART OF BMCTRIG GENERATED JOB                         *
//*        PARM='ASUDOPTs 000000 RESTART(PHASE)'                   *
//*               |        |       |                               *
//*    DOPTs MODULE    |       |                               *
//*    RESTART SEQUENCE NUMBER    |                               *
//*    UTILITY RESTART PARAMETER INFORMATION                        *
//*                                                                *
//*                                                                *
//* BATCH STARTOVER OF BMCTRIG GENERATED JOB                       *
//*                                                                *
//*        PARM='ASUDOPTs STARTOVER'                                *
//*               |        |                                       *
//*    DOPTs MODULE    |                                       *
//*    STARTOVER KEYWORD                                          *
//******************************************************************
//STEPLIB DD DISP=SHR, DSN=*** DASD MANAGER PLUS load library
//******************************************************************
//* DASD MANAGER VERSION 6.2 LOAD LIBRARY                          *
//******************************************************************
//* UTILJOB1 DD IS INPUT DATA SET FOR BMCTRIG GENERATED JOB         *
//******************************************************************
//UTILJOB1 DD DISP=SHR, DSN=input.JCL.DataSet
//******************************************************************
//* UTILJOB2 DD IS OUTPUT DATA SET FOR RESTART OF BMCTRIG GENERATED* *
//* JOB                                                            *
//******************************************************************
//UTILJOB2 DD DISP=SHR, DSN=output.JCL.DataSet
//SYSPRINT DD SYSOUT=*                                         
//SYSTERM DD SYSOUT=*                                           
```

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.

**Note**
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 explains how to use the ALTERSEC utility and how to change the secondary quantity.

Prerequisites for changing the secondary quantity

Before you start, consider the following:

- You should be familiar with creating actions and the job generation function. For more information, see “Maintaining and generating Service Actions” on page 223.
- Choose an extent limit that your Service Level Agreement (SLA) can tolerate. The default extent limit is 10.
- 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.

Note

Before running ALTERSEC, choose the table spaces and indexes that you want to monitor, and choose 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 ALTERSEC 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 137 on page 542 describes the workflow of the ALTERSEC utility.

**Figure 137: Altering the secondary quantity**

Table 85 on page 543 describes what happens over time when each execution of ALTERSEC specifies **Percent increase = 100** and **Extent limit = 2**.
Note

Until the BMC or DB2 utility has applied the new SQTY twice, it does not increase the SQTY again.

### Table 85: 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 138 on page 543 shows.

### Figure 138: ALTERSEC -SQL command

```
-SQL 000004 ALTER TABLESPACE ASUDB029.TS29P1 PART 001
SECQTY 3600
```

The -SQL command in the figure 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 139 on page 544 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.

![Flow diagram for ALTERSEC](image)

### 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 a Service Action” on page 241.

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**.

**Figure 140: Alter Secondary Quantity panel**

```
DEDK                   Alter Secondary Quantity                 Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

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)
***************************************************************************** Bottom of data*****************************************************************************
```

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 “Prerequisites for changing the secondary quantity” on page 541.

11 Generate and execute the job as described in “Generating Service Actions” on page 248.
Producing reports

This chapter describes how to produce reports in the DASD MANAGER PLUS product from the Reports option on the main menu. This chapter also describes querying the DASD MANAGER PLUS tables.

Before you begin working with reports

Before using the steps in this section, gather statistics from BMCSTATS or RUNSTATS. Collect statistics on the same objects on a regular basis. 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.

"Customizable report samples" on page 610 discusses the reports in option 0 on the Report Selection Menu. You can run these reports online or in batch.

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.

DEAE ----------------- Report Selection Menu -----------------
COMMAND ===> 
Please type an option and press Enter.
   0 - Customizable Reports
   1 - Event Recording Report
   2 - Statistics Exceptions Report
   3 - Space Allocation Reports
Standard reports descriptions

The following table describes DASD MANAGER PLUS standard reports.

<table>
<thead>
<tr>
<th>Standard DASD MANAGER PLUS Reports</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Event Reporting Report</td>
</tr>
<tr>
<td>2 - Statistics Exceptions Report</td>
</tr>
<tr>
<td>3 - Space Allocation Reports</td>
</tr>
<tr>
<td>4 - Reorg Reports</td>
</tr>
<tr>
<td>5 - Image Copy Reports</td>
</tr>
<tr>
<td>6 - Tablespace Reports</td>
</tr>
<tr>
<td>7 - Index Reports</td>
</tr>
<tr>
<td>8 - Table Reports</td>
</tr>
<tr>
<td>9 - Stogroup Reports</td>
</tr>
</tbody>
</table>

- **Customizable Reports**
- **Event Reporting Report**
- **Statistics Exceptions Report**
- **Space Allocation Reports**
- **Reorg Reports**
- **Image Copy Reports**
- **Tablespace Reports**
- **Index Reports**
- **Table Reports**
- **Stogroup Reports**

**Figure 141: Report Selection Menu panel**

- Tablespace Extent Detail
- Page Update Analysis
- Table Extent Summary
- Tablespace Detail
- Volume Analysis
- Index Key Analysis
- Space Detail Summary
- Tablespace Extent Summary
- Offset RDS
- Tablespace Extent Detail
- Volume Extent Detail
- Clustering Analysis
- Index Combined Statistics
- Extent Summary
- Index Analysis
Table 86: DASD MANAGER PLUS standard reports

<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</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 Job 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</td>
<td>This report provides a quick reference of index definitions by table.</td>
<td>INDXANAL</td>
</tr>
<tr>
<td>Index Clustering Analysis report</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</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</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</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</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>
<tr>
<td>Page Update Analysis report</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</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

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

#### Report short name

**TSEXTENT**

### Tablespace Offset RIDs report

This report helps determine when to reorganize data by showing the number of rows not on their original page.

#### Report short name

**OFFSETRID**

### Tablespace Space Detail report

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.

#### Report short name

**TSSPACAS**

### Volume Analysis report

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.

#### Report short name

**VOLANAL**

### Volume Free Space Detail report

When paired with the Volume Analysis report, this report provides additional information about volumes that are candidates for new table spaces and index spaces.

- This report includes free cylinders, additional free tracks, and the largest contiguous space available in free cylinders and additional free tracks.
- For a detailed description, see the Help panels.

#### Report short name

**VOLFREE**

---

**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 (the Event Recording Report) and Option 2 (the Statistics Exception Report, or simply the Exception Report) represent single reports.

DASD MANAGER PLUS organizes the remaining standard reports by the following groupings: 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.

Generating the Average length of RID chain report

Use this procedure to generate 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 generate 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 is displayed.

   DEAE ----------------------------- AVERAGE LENGTH OF RID CHAIN ----------------
   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
   Average RIDS per entry >=... 3

   Report style.................... 1. Most recent execution
                                 2. Newest and Oldest execution
                                 3. All executions
   Batch Report : N (Y/N)

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:

• The most recent execution of BMCSTATS (allows you to see the latest execution)

• The newest and oldest execution (allows you to compare changes between the oldest and newest executions)

• All executions (allows you to determine if the object has changed over time)

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 Average Length of RID Chain report appears online (Figure 142 on page 552).

*Figure 142: Average Length of RID Chain 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 is displayed.

   DB25 --------------------- Event Recording Request ----------------------------
   COMMAND ===> Type data and press Enter.
   Display Events on DB2 Objects
   Object Type
   Object Name
   Display Events by Function
   Function Name
   Restrict to Action
   Action Name
   Restrict by Timestamp YYYY-MM-DD-HH.MM.SS
   Start Time
   End Time

2. 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 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
BMCCPRS
BMCCOPY
BMCREORG
BMCREORG
BMCTSTATS
BMCTTRIG
BMCTUPRS
BMCTUNLOAD
CHEK DA
CHEK LOB
DSN1COPY
FULLCOPY
INCRCOPY
LOAD
MERGECOPY
MODISTAT
MODIFY
QUIECE
RECOVER
REORC
REPAIR
RESIZE
NGTREORC
REP REORC
REP SET
RUNSTATS
STARC DB
STOP
STOP DB
STOACSE
SYNC
UNLOAD
To report on events for a certain action or group of actions, type the **Action Name**. You can use a wildcard.

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 for a certain type of object, specify the **Object Type**. The **Object Type** can be TS, TB, or IX.

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`.

After you specify all of the search criteria, press **Enter**.

The Event List displays the rows that match your selection criteria.

![Figure 143: Event List](image)

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.

- **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, type `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 144: Index Analysis panel**

   ![Index Analysis panel](image)

   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 <=**
   - **Report style**
   - **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 or 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** 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 is displayed.

---

**Figure 145: Index Analysis report**

<table>
<thead>
<tr>
<th>IXCREATOR</th>
<th>IXNAME</th>
<th>PART</th>
<th>TYPE</th>
<th>CLSTR</th>
<th>UNIQUE</th>
<th>PFREE</th>
<th>PPAGE</th>
<th>LEAF</th>
<th>NLVL</th>
<th>RATIO</th>
<th>FACTOR</th>
<th>EXECU</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB:02U.02UT00_DSC305PF8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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>Value</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>
<tr>
<td>X</td>
<td>Unique and used to enforce the uniqueness of values in a column that is used to identify or find XML values associated with a specific row</td>
</tr>
</tbody>
</table>

- **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.

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.
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>Value</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>
<tr>
<td>X</td>
<td>Unique and used to enforce the uniqueness of values in a column that is used to identify or find XML values associated with a specific row.</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.

   Figure 146: Index Combination Statistics panel

   DECA ----------------------------- INDEX COMBINATION STATISTICS ---------------
   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 ..........
   Where ( Clustering = Y
       AND Clusterratio <=. 96
       OR Nearoffpos/Card * 100.0 >=. 5
       OR Faroffpos/Card * 100.0 >=. 2 )
       OR Average RIDS per entry >=. 3
       OR LEAFDIST >=. 100
       OR Data set extents >=. 3
   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 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
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**.

**Figure 147: 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.
- **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.
   c. On the Index Reports Selection Menu, select Index Key Analysis and press Enter.

   The Figure 148 on page 565 appears.

   Figure 148: Index Keys Analysis panel

   DECA ----------------------------- INDEX KEYS 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
   Database....................
   Database.Tablespace name ....
   Creator.Table name ..........
   Creator.Index 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 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 149: Index Key Analysis report](image)

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**.
   
   b. On the Report Selection Menu, select **Reorg Reports** and press **Enter**.
   
   c. On the Reorg Reports Selection Menu, select **Index Leaf Distribution** and press **Enter**.

   The Index Leaf Distribution panel **Figure 150 on page 567** appears.

### Figure 150: 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 wild card 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*, 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 *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 151: Index Leaf Distribution report](image)

     | INDEX CREATOR | INDEX NAME | AVG PAGES | BETWEEN | LEAFPAGES | EXECUTION TIME |
     |---------------|------------|-----------|---------|-----------|----------------|
     | TB:QZU.QZUT01.DE3S01 | QZU | 0 | 29772931 | 2006-06-20-16.26 |
     | TB:QZU.QZUT01.DE4S01 | QZU | 0 | 29772931 | 2006-06-20-16.26 |
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.
   c. On the Space Allocation Reports Selection menu, select Indexspace Space Detail and press Enter.

   Figure 152 on page 569 appears.

   **Figure 152: Indexspace Space Detail panel**

   ![Indexspace Space Detail panel](image)

   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)

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, **Figure 153 on page 570 appears.**

**Figure 153: Indexspace Space Detail report**

```
BROWSE    SYS11012.T130122.RA000.RBAJKX.S0361881                                               Line 00000000 Col 001 132
Command ===>                                                                                            Scroll ===> PAGE
********************************************************** Top of Data *************************************************
Time. . . .: 01:01:23 PM  Wednesday, January 12, yyyy
DB2 ID. . .: DEDK
DB2 Version: 1015
User. . . .: RDAJXS
Indexspace Space Detail (DB2 Catalog)                MORE DATA ==>
Data set numbers reflect totals of all volumes and extents
Volume name followed by a ‘+’ sign indicates the data set exists on more than one volume
OBJEKTNAME:QZU.QZUKT01.DiOSS010.DSNDBD.QZUX01.RDAJXS.RA001.A001
OBJEKTNAME:QZU.QZUKT01.DiOSS010.DSNDBD.QZUX01.RDAJXS.RA001

OBJEKTNAME:QZU.QZUKT01.DiOSS010.DSNDBD.QZUX01.RDAJXS.RA001

Value | Description
--- | ---
Object Type | Index space
Volser | Volume serial number of the first volume for the object
```

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:
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Devtype</td>
<td>Device type where the object is stored</td>
</tr>
<tr>
<td>Extents</td>
<td>Number of extents that the index or index partition is in</td>
</tr>
<tr>
<td>Kbytes Alloc</td>
<td>Number of kilobytes allocated for the index</td>
</tr>
<tr>
<td>Kbytes Used</td>
<td>Number of kilobytes that the index uses</td>
</tr>
<tr>
<td>Storname</td>
<td>Name of the storage group or ICF catalog used for space allocation</td>
</tr>
<tr>
<td>Tracks Alloc</td>
<td>Number of tracks allocated for the index</td>
</tr>
<tr>
<td>Tracks Used</td>
<td>Number of tracks that the index uses</td>
</tr>
<tr>
<td>Alloc Unit</td>
<td>Allocation unit (T—tracks or C—cylinders)</td>
</tr>
<tr>
<td>Prim Alloc</td>
<td>Primary allocation quantity</td>
</tr>
<tr>
<td>Sec Alloc</td>
<td>Secondary allocation quantity</td>
</tr>
</tbody>
</table>

- **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 Figure 154 on page 572 appears.

**Figure 154: Monthly Detail by Tablespace panel**

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 ..........
- 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, Figure 155 on page 572 appears.

**Figure 155: 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.
- **Ttsname** is the name of the table space.
- **Part #** is the number of the table space 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.
- **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.

<table>
<thead>
<tr>
<th>TSNAME</th>
<th>Part</th>
<th>Month</th>
<th>Card</th>
<th>TYPE</th>
<th>Far Ind Ref</th>
<th>Near Ind Ref</th>
<th>PERC ACTIVE</th>
<th>PERC DROP</th>
<th>NACTIVE</th>
<th>FULL</th>
<th>DIRTY</th>
<th>SPACE</th>
<th>Reorg Space</th>
<th>Pqty</th>
<th>Sqty</th>
<th>XTNTS</th>
<th>PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUS01A1</td>
<td>0</td>
<td>yyyy-12</td>
<td>2036</td>
<td>0</td>
<td>0</td>
<td>84</td>
<td>0</td>
<td>0</td>
<td>84</td>
<td>0</td>
<td>94</td>
<td>0</td>
<td>130</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QZUS02A1</td>
<td>0</td>
<td>yyyy-12</td>
<td>2036</td>
<td>0</td>
<td>0</td>
<td>84</td>
<td>0</td>
<td>0</td>
<td>84</td>
<td>0</td>
<td>94</td>
<td>0</td>
<td>130</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QZUS03A1</td>
<td>0</td>
<td>yyyy-12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>360</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QZUS03A1</td>
<td>1</td>
<td>yyyy-12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>360</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>QZUS03A1</td>
<td>2</td>
<td>yyyy-12</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>360</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>QZUS03A1</td>
<td>3</td>
<td>yyyy-12</td>
<td>350</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>360</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>QZUS03A1</td>
<td>4</td>
<td>yyyy-12</td>
<td>1667</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>360</td>
<td>104</td>
<td>0</td>
<td>360</td>
<td>110</td>
<td>1</td>
<td>1</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>QZUS04A1</td>
<td>0</td>
<td>yyyy-12</td>
<td>7010</td>
<td>0</td>
<td>0</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>540</td>
<td>48</td>
<td>0</td>
<td>100</td>
<td>45</td>
<td>33</td>
<td>9</td>
<td>18</td>
<td>2</td>
</tr>
</tbody>
</table>
■ **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.


   Figure 156 on page 574 appears.

   **Figure 156: Page Update Analysis panel**

   DECA ----------------------------- PAGE UPDATE 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.

   - **Database**
   - **Database.Tablespace name**
   - **Creator.Table name**

   Where
   - **Percent of dirty pages >=** ...

   **Report style**
   
   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 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, Figure 157 on page 575 appears.

**Figure 157: Page Update Analysis report**

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.

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 476.

### 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**.

**Figure 158: Statistics Exception Selection panel**

```
DECA ----------------- Statistics Exception Selection --------------
COMMAND ===> 
Type data and press Enter.
Object Type . . . . . (TB,TS,IX)
Object Name . . . . .
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 (Figure 159 on page 578) and you can select an exception type.

**Figure 159: Statistics Exception Type Selection panel**

```
DEAE -------------- Statistics Exception Type Selection -----------------------
COMMAND ===>
```

Type the number of the exception you wish to select and press Enter.

```
0    1. Card        18. Pctclust    35. Loadstat    52. NoRTSCpy
10. Leafdist    27. Pctdrop     44. Rebdcopy    61. Reorleaf
12. Levels      29. Ixicage     46. Appndins    63. Chkpnd
13. NoStats     30. Ixidrop    47. Arerpend    64. Icppnd
15. Nonuniform  32. LeafFoff    49. Lobfrspc    66. Lobfrspc
17. Pactlo      34. Loadcopy    51. NoRTStat
```

If you have specified **Select Exception type = N**, DASD MANAGER PLUS displays an Exceptions Selection panel that shows all exceptions found for the specified objects (Figure 159 on page 578).

**Figure 160: 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.
```

```
Act Owner/DB Object Name Type Part Exception     Current Value
-----------------------------------------------
CCC XCPPART IX 1 EXTENTS                       1
CCC XCPPART IX 1 EXTENTS                       1
CCC XCPPART IX 1 EXTENTS                       1
CCC XCPPART IX 1 EXTENTS                       1
CCC XCPPART IX 1 EXTENTS                       1
CCC 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
  - 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.

3 To display the exception details for an item in the list, type S next to the item in the **Act field**.

**Figure 161: Exceptions report for a single exception type**

| Object Name  : CALDB.CALTS002                     | Timestamp : yyyy-10-08-09.30 |
|-------------:-------------------------------------------|-------------------------------|
| Type . . .   : TS                              | Type . . . : TS               |
| Action Taken : A447617S GENERATED             | Action Taken : A447617S GENERATED |

<table>
<thead>
<tr>
<th>Part</th>
<th>Exception</th>
<th>Current Value</th>
<th>Compare Value</th>
<th>Trigger Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>COPYAGE</td>
<td>67</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>COPYAGE</td>
<td>67</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>COPYAGE</td>
<td>67</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>COPYAGE</td>
<td>67</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>COPYAGE</td>
<td>67</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>COPYAGE</td>
<td>67</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>COPYAGE</td>
<td>67</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The Exception 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.

The following figure shows the Exceptions Selection panel that appears when you specify Select Exception Type N on the initial Statistics Exception Selection panel.

Figure 162: Exceptions Selection panel

For a description of the fields on this panel, see Figure 162 on page 580. 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 163: Exceptions report for all exception types found for object

For a description of the fields on this panel, see page 580.
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. Figure 164 on page 581 appears.

   **Figure 164: The 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, Figure 165 on page 582 appears.

**Figure 165: Table Column Detail report**

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.

   **Figure 166: Tablespace Extent Summary panel**

   | DECA ----------------------------- TABLESPACE EXTENT SUMMARY ------------------ |
   | 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 ............. |
   | Where                         |
   | Data set extents >=........... |
   | Report style................. |
   | 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 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 Figure 167 on page 584 appears.

**Figure 167: Tablespace Extent Summary report**

The Tablespace Extent Summary report contains the fields described in Table 87 on page 584.

**Table 87: Tablespace Extent Summary report fields**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name/Table Space Name</td>
<td>Fully qualified table space name</td>
</tr>
<tr>
<td>Part Nbr</td>
<td>Number of the table space partition</td>
</tr>
<tr>
<td>TS Type</td>
<td>Type of table space</td>
</tr>
<tr>
<td>Number of Rows</td>
<td>Cardinality of the table space or table space partition</td>
</tr>
<tr>
<td>Xtnts</td>
<td>Number of extents that the table space is in</td>
</tr>
<tr>
<td>Nactive</td>
<td>Number of active pages in the table space</td>
</tr>
<tr>
<td>The product considers pages with row format active even if they currently contain no rows.</td>
<td></td>
</tr>
<tr>
<td>Npages</td>
<td>Number of pages in the table space</td>
</tr>
</tbody>
</table>
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.
   c. On the Reorg Reports Selection Menu, select Offset Rids and press Enter. The Figure 168 on page 585 appears.

   **Figure 168: Tablespace Offset RIDs panel**

   DECA ----------------------------- TABLESPACE OFFSET RIDS ---------------------
   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 ........

   Where
   Percent of rows not on page >=.. 5

   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 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, **Figure 169 on page 586** appears.

**Figure 169: Tablespace Offset RIDs report**

The Tablespace Offset RIDs report contains the fields described in **Table 88 on page 586**.

**Table 88: Tablespace Offset RIDs report fields**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name/Table Space Name</td>
<td>Fully qualified table space name</td>
</tr>
<tr>
<td>Part Nbr</td>
<td>Number of the table space partition</td>
</tr>
<tr>
<td>Near Indref</td>
<td>Number of indirect references to rows that are less than 16 pages from the original page because of an insert into a full page</td>
</tr>
</tbody>
</table>
### Field Name | Description
---|---
**Far Indref** | 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** | Number of rows in the table space or table space partition

**BMCSTATS Execution Time** | BMCSTATS collected the statistics in the report

---

## Tablespace space detail report

Use this procedure to create a report that provides a quick reference of space usage by table space.

### To create a tablespace 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**.

   “Tablespace space detail report Use this procedure to create a report that provides a quick reference of space usage by table space.” appears.

   **Figure 170: Tablespace Space Detail panel**

   ```
   DECA ----------------------------- TABLESPACE 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 ..........
   
   **** WARNING ****
   This report may execute for extended lengths 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, Figure 171 on page 588 appears.

**Figure 171: Tablespace Space Detail report**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN</td>
<td>The fully qualified data set name</td>
</tr>
<tr>
<td>Volser</td>
<td>The volume serial number of the first volume for the table space</td>
</tr>
<tr>
<td>Devtype</td>
<td>The device type where the table space is stored</td>
</tr>
<tr>
<td>Extents</td>
<td>The number of extents that the table space or table space partition is in</td>
</tr>
<tr>
<td>Kbytes Alloc</td>
<td>The number of kilobytes allocated for the table space</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Kbytes Used</td>
<td>The number of kilobytes that the table space uses</td>
</tr>
<tr>
<td>Tracks Alloc</td>
<td>The number of tracks allocated for the table space</td>
</tr>
<tr>
<td>Tracks Used</td>
<td>The number of tracks that the table space uses</td>
</tr>
<tr>
<td>Alloc Unit</td>
<td>The allocation unit (T for tracks or C for cylinders)</td>
</tr>
<tr>
<td>Prim Alloc</td>
<td>The primary allocation quantity</td>
</tr>
<tr>
<td>Sec Alloc</td>
<td>The secondary allocation quantity</td>
</tr>
<tr>
<td>Object Name</td>
<td>The fully qualified name of the table space</td>
</tr>
<tr>
<td>Object Type</td>
<td>Table space</td>
</tr>
<tr>
<td>Part</td>
<td>The number of the table space partition</td>
</tr>
<tr>
<td>Card</td>
<td>The number of rows in the table space</td>
</tr>
<tr>
<td>Page size</td>
<td>The page size: 4, 8, 16, or 32 kilobytes</td>
</tr>
<tr>
<td>TS Type</td>
<td>The type of table space</td>
</tr>
<tr>
<td>Lobdatatype</td>
<td>The type of LOB table space</td>
</tr>
<tr>
<td>Implicit</td>
<td>Indicates whether the database was implicitly created</td>
</tr>
<tr>
<td>Storname</td>
<td>The name of the storage group or ICF catalog used for space allocation</td>
</tr>
<tr>
<td>Primary</td>
<td>The primary allocation quantity</td>
</tr>
<tr>
<td>Secondary</td>
<td>The secondary allocation quantity</td>
</tr>
<tr>
<td>Freepage</td>
<td>The number of free pages</td>
</tr>
<tr>
<td>Percent Free</td>
<td>The percentage of the allocated space that is free</td>
</tr>
<tr>
<td>LOB</td>
<td>Indicates whether the table space is a LOB table space, base, or auxiliary</td>
</tr>
<tr>
<td>Bpool</td>
<td>Is the name of the buffer pool</td>
</tr>
<tr>
<td>Lockmax</td>
<td>Is the maximum number of locks per user to acquire for the table space before escalating</td>
</tr>
<tr>
<td>Compress</td>
<td>Indicates whether the table space is compressed</td>
</tr>
<tr>
<td>Segsize</td>
<td>Is the segment size</td>
</tr>
<tr>
<td>Dsnum</td>
<td>Indicates the data set number</td>
</tr>
<tr>
<td>Dssize</td>
<td>Indicates the size of the data set</td>
</tr>
<tr>
<td>Member Cluster</td>
<td>Indicates whether the table space has a member cluster page set structure</td>
</tr>
<tr>
<td>Lockrule</td>
<td>Is the lock size of the table space</td>
</tr>
<tr>
<td>Log</td>
<td>Indicates whether changes to the table space are to be logged</td>
</tr>
</tbody>
</table>
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 172: Volume Analysis panel**

   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**
   - **Cylinders needed >=**
   - **Report style**
   - **Batch Report**

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 173: Volume Analysis report**

```
BMC SOFTWARE, INC.  ***** DASD MANAGER PLUS FOR DB2 *****  V11.02.00

Time. . . .: 12:30:06 PM Tuesday, August 16, yyyy
DB2 ID. . .: DECA
DB2 Version: 1015
User. . . .: RDAJXS

<table>
<thead>
<tr>
<th>VOLSER</th>
<th>DB2 Tracks Alloc</th>
<th>Non DB2 Data Sets</th>
<th>Total Data Sets</th>
<th>Free Cyl</th>
<th>Free Trk</th>
<th>Free Extents</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZU001</td>
<td>49954</td>
<td>312</td>
<td>314</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>AZU002</td>
<td>41762</td>
<td>1253</td>
<td>1255</td>
<td>545</td>
<td>8182</td>
<td>4</td>
</tr>
<tr>
<td>AZU003</td>
<td>9714</td>
<td>165</td>
<td>167</td>
<td>2682</td>
<td>40240</td>
<td>7</td>
</tr>
<tr>
<td>AZU004</td>
<td>34768</td>
<td>42</td>
<td>44</td>
<td>1006</td>
<td>15187</td>
<td>18</td>
</tr>
</tbody>
</table>
```

The Volume Analysis report contains the following fields:

**Table 90: Tablespace Extent Summary report fields**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLSER</td>
<td>Volume serial number of the DASD volume</td>
</tr>
<tr>
<td>DB2 Tracks Alloc</td>
<td>DB2 Tracks Alloc is the number of tracks allocated for DB2 data sets on the volume</td>
</tr>
<tr>
<td>DB2 Data Sets</td>
<td>Number of DB2 data sets on the volume</td>
</tr>
<tr>
<td>Non DB2 Data Sets</td>
<td>Number of non-DB2 data sets on the volume</td>
</tr>
<tr>
<td>Total Data Sets</td>
<td>Total number of data sets on the volume</td>
</tr>
<tr>
<td>Free Cyl</td>
<td>Number of free cylinders on the volume</td>
</tr>
<tr>
<td>Free Trk</td>
<td>Number of free tracks on the volume</td>
</tr>
<tr>
<td>Free Extents</td>
<td>Number of free extents on the volume</td>
</tr>
<tr>
<td>BMCSTATS Execution Time</td>
<td>When BMCSTATS collected the statistics in the report</td>
</tr>
</tbody>
</table>
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.
   c. On the Space Allocation Reports Selection menu, select Volume Free Space Detail (Real Time) and press Enter.

   **Figure 174: Volume Free Space Detail panel**
   DECA ----------------------------- VOLUME FREESPACE 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
   Stogroup name ............... Volume name ............... Where Cylinders needed >=......... 0 Batch Report : N (Y/N)

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, Figure 175 on page 593 appears.

**Figure 175: Volume Free Space Detail report**

<table>
<thead>
<tr>
<th>Volume Name</th>
<th>Dev Type</th>
<th>Alloc Space</th>
<th>Alloc %</th>
<th>Free Cyls</th>
<th>Free %</th>
<th>Addt'l Tracks</th>
<th>Addt'l %</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>--A300 3390</td>
<td>3390</td>
<td>2 - 1</td>
<td>3336 - 99</td>
<td>14 - 0</td>
<td>3336</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--A301 3390</td>
<td>3390</td>
<td>2 - 1</td>
<td>3336 - 99</td>
<td>14 - 0</td>
<td>3336</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--A302 3390</td>
<td>3390</td>
<td>2 - 1</td>
<td>3336 - 99</td>
<td>14 - 0</td>
<td>3336</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--A303 3390</td>
<td>3390</td>
<td>2 - 1</td>
<td>3336 - 99</td>
<td>14 - 0</td>
<td>3336</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 fields described in Table 91 on page 593.

**Table 91: Volume Free Space Detail report fields**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Volume Name</strong></td>
<td>Volume serial number of the DASD volume</td>
</tr>
<tr>
<td><strong>Dev Type</strong></td>
<td>Type of DASD volume, for example, 3380 or 3390</td>
</tr>
<tr>
<td><strong>Alloc Space</strong></td>
<td>Space that has been allocated on the volume</td>
</tr>
<tr>
<td><strong>%</strong></td>
<td>Percentage of allocated space that has been used</td>
</tr>
<tr>
<td><strong>Free Cyls</strong></td>
<td>Number of unused cylinders on the volume</td>
</tr>
<tr>
<td><strong>%</strong></td>
<td>Percentage of free cylinders that has not been used</td>
</tr>
<tr>
<td><strong>Addt'l Tracks</strong></td>
<td>Total number of free tracks</td>
</tr>
<tr>
<td><strong>%</strong></td>
<td>Percentage of free tracks in cylinders that has not been used</td>
</tr>
<tr>
<td><strong>Largest Extent</strong></td>
<td>Largest free extent, calculated on a cylinder boundary</td>
</tr>
<tr>
<td><strong>Cyls</strong></td>
<td>is the number of cylinders in the largest free extent.</td>
</tr>
<tr>
<td><strong>Addt'l Tracks</strong></td>
<td>is the number of free tracks before or after the last cylinder boundary in the contiguous block of space.</td>
</tr>
</tbody>
</table>
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 "JCL for batch generation" 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 176: JCL for batch generation

```plaintext
//________ JOB (____), 'ASURBAT', MSGCLASS=X, CLASS=A, NOTIFY=_______
//*
//ASURBAT EXEC PGM=ASURBAT, PARM='DB2, ASU812DR, DB2VCAT, ASUDOPTs'
//**---------------------------------------------------------------
//* PARM 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
//*
//SYSTEM DD SYSPUT=*, DCB=BLKSIZE=142
//SYSPRINT DD SYSPUT=*, 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 SYSPUT=*
```

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. The table below shows views that the CNTL library member ASURVIEW provides for use by the QMF reporting procedures.
## Table 92: 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_TSPART 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 (in the table below) to access useful information in the statistics tables.
### Table 93: Descriptions of QMF reports

<table>
<thead>
<tr>
<th>Report</th>
<th>Query</th>
<th>Form</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DBMONTHLY</strong></td>
<td>QDBMONTHLY</td>
<td>FDBMONTHLY</td>
<td>BMCQMF.RS_USER_DBTS</td>
</tr>
<tr>
<td>lists high-level table space statistics for all databases by month.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DBSUMMARY</strong></td>
<td>QDBSUMMARY</td>
<td>FDBSUMMARY</td>
<td>BMCQMF.RS_USER_TBCOUNTS BMCQMF.RS_USER_DBTS</td>
</tr>
<tr>
<td>lists summary information about all databases.</td>
<td></td>
<td></td>
<td>BMCQMF.RS_USER_DBIX</td>
</tr>
<tr>
<td><strong>MONTHCHANGE</strong></td>
<td>QMONTHCHANGE</td>
<td>FMONTHCHANGE</td>
<td>BMCQMF.RS_MO_AV_TSPART</td>
</tr>
<tr>
<td>gives delta information by table space over the last three months.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PARTSTATS3</strong></td>
<td>QPARTSTATS3</td>
<td>FPARTSTATS3</td>
<td>BMCQMF.RS_MO_AV_TSPART</td>
</tr>
<tr>
<td>gives information about all table space partitions over the last three months.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SPACESTATS12</strong></td>
<td>QSPACESTATS12</td>
<td>FSPACESTATS12</td>
<td>BMCQMF.RS_MO_AV_TSPART</td>
</tr>
<tr>
<td>displays the amount of space used by all table spaces over the last twelve months. The report can be modified to display any desired statistic over the last twelve months.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TSBYMOUTH</strong></td>
<td>QTSBYMOUTH</td>
<td>FTSBYMOUTH</td>
<td>BMCQMF.RS_MO_AV_TSPART</td>
</tr>
<tr>
<td>produces detail information by month for every table space in the statistics tables.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TSBYSET</strong></td>
<td>QTSBYSTATSSET</td>
<td>FTSBYSTATSSET</td>
<td>BMCATSnn.RS_TABLEPART</td>
</tr>
<tr>
<td>produces one line for each timestamped entry in the statistics database for a table space partition.</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.
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:
      \[ \text{IMPORT PROC FROM 'HLQ.QMFPROC' (M=LOADPROC)} \]

      \textbf{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.

3 Press \textbf{F2} 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.

For more information, view the Quick Course "DASD MANAGER PLUS—Customizing Reports."

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.
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.

“Customizable reports” on page 671 describes the DASD MANAGER PLUS external functions, formatting, REXX conventions, and other aspects of the reports. The appendix contains an annotated sample report program.

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 94 on page 598 in the .EXEC library.

Table 94: DASD MANAGER PLUS customizable reports

<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</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>Report title</td>
<td>Description</td>
<td>EXEC name</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Database Space Detail Report</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</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</td>
<td>This report lists exceptions that a BMCTRIG job detected.</td>
<td>ASUEXCPT</td>
</tr>
<tr>
<td>Index Clustering Report</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>ASUCRANANA</td>
</tr>
<tr>
<td>Modified Table Space Pages Report</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</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</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</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</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>
<tr>
<td>Table Space and Index Storage Limits Report</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</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</td>
<td>This report predicts the number of days until DASD volumes will run out of space.</td>
<td>ASUVOLTR</td>
</tr>
</tbody>
</table>

* To run this report, DASD MANAGER PLUS requires BMCSTATS.
All reports have the same general layout:

- Report title and page number
- Product name, version number, and the EXEC name of the report
- Time and date
- DB2 subsystem ID (SSID)
- DB2 Version
- The user ID that requested the report

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**

DASD MANAGER PLUS supplies the DB2 subsystem identifier (SSID) and plan name automatically. You supply the rest of the parameters.

In the online dialog, the Parms input line appears above a line showing the required input.

**Figure 177: Input parameters**

|Parms DBDA ASU812DR ssid plan dbname pctlimit |

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 177 on page 600*, type a database name and pctlimit.

As the following figure shows, the last page of the generated report shows the parameters that you specified on the Parms input line, which identifies the parameters that produced a particular report.
The following figure 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.

```
// PARM='ASUDBLIM DBDA ASU812DR QZUD2% 10'
```

**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.

**Description of input parameters**

The following table describes input parameters that are used to produce the various reports.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>Name of the action Wildcards are allowed.</td>
</tr>
<tr>
<td>cr</td>
<td>Cluster ratio percentage of rows that are in clustering order</td>
</tr>
<tr>
<td>days</td>
<td>Number of days from date of report to compute the reorganized space of a DB2 object (1 to 366)</td>
</tr>
<tr>
<td>dbname</td>
<td>Name of the database Wildcards are valid.</td>
</tr>
</tbody>
</table>
### Name | Description
--- | ---
**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 calculated 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)

**farpc** | 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
**nearpc** | 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
**pctcard** | 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
**pctdirty** | Percentage of dirty pages that a table space must reach or exceed to be included in the report
**pctlimit** | 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
**plan** | 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.
**qualname** | Creator or dbname; wildcards are allowed
**qualtype** | Literal CREATOR or DBNAME
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report-name</td>
<td>Name of the report program, such as ASUNPGR.</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>tsfpg</td>
<td>Number of free pages for table spaces</td>
</tr>
<tr>
<td>tsname</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>

**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 178 on page 603), you can change each report’s units from tracks to KB.

**Figure 178: Internal KB switch**

```sql
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**

Use these procedures to run the customizable reports and 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 179: Customizable Report Selection List**

   COMMAND ===>                                                  SCROLL ===> CSR
   Type S or / to select report(s) and then press ENTER

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASUAMGR</td>
<td>Average Monthly Growth Report</td>
</tr>
<tr>
<td>ASUNPGR</td>
<td>Tables with More than Average Growth Report</td>
</tr>
<tr>
<td>ASUSPCTR</td>
<td>Database Space Trend Report</td>
</tr>
<tr>
<td>ASUVOLTR</td>
<td>Volume Space Trend Report</td>
</tr>
<tr>
<td>ASUDBSPA</td>
<td>Database Storage Utilization Report</td>
</tr>
<tr>
<td>ASUDSPAC</td>
<td>Database Space Detail Report</td>
</tr>
<tr>
<td>ASUDBEXT</td>
<td>Table Space and Index Extent Report</td>
</tr>
<tr>
<td>ASUDBLIM</td>
<td>Table Space and Index Storage Limits Report</td>
</tr>
<tr>
<td>ASUTBSDR</td>
<td>Space Utilization by Table Report</td>
</tr>
<tr>
<td>ASUCRANA</td>
<td>Index Clustering Analysis Report</td>
</tr>
<tr>
<td>ASUDFACT</td>
<td>Default Corrective Action Cross Reference Report</td>
</tr>
<tr>
<td>ASUACTN</td>
<td>Action Report</td>
</tr>
<tr>
<td>ASUSTRSE</td>
<td>Space Estimation Trend Report</td>
</tr>
<tr>
<td>ASUPSE</td>
<td>Space Estimation Report</td>
</tr>
<tr>
<td>ASUZDSLI</td>
<td>Data Set List by Database Name Report</td>
</tr>
<tr>
<td>ASUEXPT</td>
<td>Exceptions Report by BMCTRIG Action</td>
</tr>
<tr>
<td>ASULOBAN</td>
<td>LOB Objects Analysis Report</td>
</tr>
</tbody>
</table>

   2 Type S or slash (/) by the report or reports that you want to produce, and press Enter.

   The first report panel appears.

3 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.

4 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. The input parameters are described in “Description of input parameters” on page 601.

   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 on page 604.
**To Execute batch only**

1. Type S or slash (/) by the **Edit JCL** option and press **Enter**.

2. Review and edit the generated JCL (Figure 180 on page 605).

---

**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 180 on page 605.

---

**Figure 180: 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*
/*SYSTSIN DD *
/*SYSTSPRT DD SYSOUT=* 
/*SYSPRINT DD SYSOUT=*,
// DCB=(DSORG=PS,LRECL=142,BLKSIZE=7100,RECFM=VBA)
/*SYSTERM DD SYSOUT=* 
```

**Figure 181: Generated batch JCL from the online dialog (with long name)**

```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     *
/* 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=* 
```
3 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 “Creating customizable reports” on page 603.

The JCL in **Figure 182** on page 606 runs IKJEFT01, which interprets the DASD MANAGER PLUS report programs.

**Figure 182: Batch JCL to run reports outside the online dialog**

```plaintext
//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
//SYSEXEC  DD DISP=SHR,DSN=DB2*****.DSNLOAD
//SYSPRINT DD SYSOUT=* 
//SYSTERM  DD SYSOUT=*
//SYSTSPRT DD SYSOUT=* 
//SYSTSPRT DD DUMMY
```

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**.

   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.

**Note**
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 “Customizable reports” on page 597 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 183 on page 608 displays the list of input tables from one of the report programs.

**Figure 183: Input tables**

```plaintext
sysdatabase   = "SYSIBM.SYSDATABASE";
sysindexes    = "SYSIBM.SYSINDEXES";
systables     = "SYSIBM.SYSTABLES";
systablespace = "SYSIBM.SYSTABLESPACE";
dmtablepart   = asualias("BMCASU_STABLEPART");
dmindexpart   = asualias("BMCASU_SINDEXPART");
```

- To use the report output as input to another utility, remove the top-of-page formatting (any format definition in which the second node is the word *top*).

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 customizable reports” on page 603.

7 Test and modify the report program until it terminates normally and produces the correct output.

**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 queuing the data to the external data queue.

2 Decide where in the menu to place the new report.
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 184 on page 609).

**Figure 184: Set of report menu code**

```plaintext
:REPORT
ASUAMGR    Average Monthly Growth Report

:PARMS
ssid plan   dbname

  Where:
  ssid       - DB2 subsystem id
  plan       - plan name
  dbname     - database name, may be wildcarded

:DESC
  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.
```

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 603.

**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 SYSTEM 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 currently available customizable reports.

Action report

The Action Report identifies the contents of actions, including the services that they contain, the objects that the services run against, the object type, partition, any object set name, the owner, and any default corrective action.

Figure 185 on page 610 shows an example of an Action report. Only actions that contain the BMCTTRIG utility can contain a corrective action. Actions that show dashes are empty (have no steps).

Figure 185: Action Report: Example 1

<table>
<thead>
<tr>
<th>ACTION REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASD MANAGER PLUS FOR DB2 - ASUACTN</td>
</tr>
<tr>
<td>Time . . . . . : 17:30:39 14 Mar yyyy</td>
</tr>
<tr>
<td>DB2 ID . . . . : DEAE</td>
</tr>
<tr>
<td>DB2 Version. :</td>
</tr>
<tr>
<td>User . . . . . : RDAKPM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTION</th>
<th>SERVICE</th>
<th>OBJECT</th>
<th>TYPE</th>
<th>PART</th>
<th>OBJECT SET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q10409</td>
<td>BMCSTATS</td>
<td>QZUD1%.%</td>
<td>TS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BMCCPRS</td>
<td>QZUD1%.%</td>
<td>TS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BMCPURS</td>
<td>QZUD1%.%</td>
<td>TS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Action report contains the following fields:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Action for which you generate the report. The actions in this column include any corrective actions.</td>
</tr>
<tr>
<td>Service</td>
<td>Service or services that the action specifies. If no service has been specified, this field contains dashes (–).</td>
</tr>
</tbody>
</table>
The Action Report requires one of the following lines of input parameters:

\[
\text{ssid plan action} \\
\text{ssid plan action service}
\]

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.

The report in Figure 185 on page 610 results from specifying only the action parameter. The report in the following figure results from specifying both parameters. You can use wildcards for both action and service.
The Average Monthly Growth report 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 186 on page 612 shows an example of the Average Monthly Growth report.

The Average Monthly Growth report contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBName</td>
<td>Name of the database for which the product reports statistics</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TSNName</td>
<td>Name of the table space</td>
</tr>
<tr>
<td>Part</td>
<td>Number of the partition</td>
</tr>
<tr>
<td>Month</td>
<td>Year and month of the statistics in $YYYY-MM$ format, where $YYYY$ is the year and $MM$ is the month</td>
</tr>
<tr>
<td>Card</td>
<td>Cardinality, or number of rows, in the table space</td>
</tr>
<tr>
<td>%Change</td>
<td>Percentage of change in the cardinality, if any, from the previously reported month</td>
</tr>
<tr>
<td>#Trks</td>
<td>Number of tracks in use that month</td>
</tr>
<tr>
<td>#Pages</td>
<td>Number of pages in use that month</td>
</tr>
<tr>
<td>Average Monthly</td>
<td>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</td>
</tr>
</tbody>
</table>

The Average Monthly Growth report requires one of the following lines of input parameters:

```
ssid plan dbname
```

**Corrective action cross reference report**

The Corrective Action Cross Reference report lists each corrective action in the DB2 subsystem and the actions that specify them.

*Figure 187 on page 613* shows a corrective action cross reference support.

*Figure 187: Corrective Action Cross Reference report*
The Corrective Action Cross Reference report contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Corr Action</td>
<td>Default corrective action</td>
</tr>
<tr>
<td>Action</td>
<td>BMCTRIG action, if any</td>
</tr>
</tbody>
</table>

**Note**
Several actions can use the same corrective action. If an action has multiple BMCTRIG services, the services can reference the same corrective action.

The Corrective Action Cross Reference report requires one of the following lines of input parameters:

- ssid plan

---

**Data Set List by database name report**

The Data Set List by database name report 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, as shown in Figure 188 on page 614, also shows the total tracks for table spaces, total tracks for indexes, and total tracks used in the database.

**Figure 188: Data Set List by database name report**

<table>
<thead>
<tr>
<th>Database Name: QZUD11</th>
<th>Trks</th>
<th>Ext AU</th>
<th>Pqty</th>
<th>Sqty</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0111.I0001.A001</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU026</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0111.I0001.A002</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU015</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0111.I0001.A003</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU064</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0111.I0001.A004</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU003</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0211.I0001.A001</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU005</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0211.I0001.A002</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU037</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0211.I0001.A003</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU037</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0211.I0001.A004</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU037</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0311.I0001.A001</td>
<td>45</td>
<td>1 C</td>
<td>3</td>
<td>2</td>
<td>ASU380</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0311.I0001.A002</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU056</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0311.I0001.A003</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU038</td>
</tr>
<tr>
<td>DEAECAT.DSNDBC.QZUD11.QZUS0311.I0001.A004</td>
<td>30</td>
<td>1 C</td>
<td>2</td>
<td>1</td>
<td>PLU056</td>
</tr>
</tbody>
</table>
### The Data Set List by database name report contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>Name of the specified database</td>
</tr>
<tr>
<td>Data Set Name</td>
<td>Name of each data set in the database</td>
</tr>
<tr>
<td>Trks</td>
<td>Number of tracks used by the data set</td>
</tr>
<tr>
<td>Ext</td>
<td>Number of extents on the data set</td>
</tr>
<tr>
<td>AU</td>
<td>Allocation unit for the data set (T – tracks or C – cylinders)</td>
</tr>
<tr>
<td>Pqty</td>
<td>Primary quantity allocated to the data set</td>
</tr>
<tr>
<td>Sqty</td>
<td>Secondary quantity allocated to the data set</td>
</tr>
<tr>
<td>Volume</td>
<td>Volume where the data set resides</td>
</tr>
</tbody>
</table>

The Data Set List by database name report requires the following line of input parameters:

```
ssid plan dbname
```

### Database space detail report

The Database Space Detail Report provides detailed information about table space and index space allocation, current space utilization, and reorganized space estimation.
Figure 189 on page 616 shows a database detail report.

**Figure 189: Database Space Detail Report**

<table>
<thead>
<tr>
<th>DATABASE OBJECT</th>
<th>PART</th>
<th>TRKS</th>
<th>TRKS</th>
<th>TRKS</th>
<th>TRKS</th>
<th>CUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUD11.QZUS0111</td>
<td>1</td>
<td>30</td>
<td>15</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>30</td>
<td>15</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>30</td>
<td>15</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>30</td>
<td>15</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.02UX01_D11S01T01</td>
<td>0</td>
<td>30</td>
<td>15</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.02UX02_D11S01T01</td>
<td>0</td>
<td>30</td>
<td>15</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZU.02UX03_D11S01T01</td>
<td>1</td>
<td>30</td>
<td>15</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>30</td>
<td>15</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>30</td>
<td>15</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>30</td>
<td>15</td>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Database space detail report contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Object</td>
<td>Name of the table space or index in the specified database</td>
</tr>
<tr>
<td>Part</td>
<td>Partition number</td>
</tr>
<tr>
<td>Trks Pqty</td>
<td>Primary quantity allocation in tracks</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> To change report units to KB, see “Internal KB switch” on page 603.</td>
</tr>
<tr>
<td>Trks Sqty</td>
<td>Secondary quantity allocation in tracks</td>
</tr>
<tr>
<td>Current</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Trks Used</strong> is the number of tracks that the table space or index uses.</td>
</tr>
<tr>
<td></td>
<td><strong>Trks Space</strong> is the number of tracks for the table space or index.</td>
</tr>
<tr>
<td></td>
<td><strong>Ext</strong> is the number of extents that the table space or index is in.</td>
</tr>
<tr>
<td></td>
<td><strong>% Used</strong> is the percentage of the allocated space that is currently used.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reorganized Trks Space</td>
<td>is the number of tracks that would be allocated if you reorganized the object.</td>
</tr>
<tr>
<td>Ext</td>
<td>is the number of extents that the object would span if you reorganized it by using the current primary and secondary quantities.</td>
</tr>
</tbody>
</table>
| %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:  
  \[ 100 \times \frac{\text{Reorganized Space} - \text{Current Space}}{\text{Current Space}} \] |

The Database space detail report requires the following input parameters:

```
ssid plan dbname
```

## Database space trend report

The Database Space Trend Report 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, as shown in Figure 190 on page 617 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.

### Figure 190: Database Space Trend Report

<table>
<thead>
<tr>
<th>DATABASE</th>
<th>NAME</th>
<th>PART</th>
<th>OBJ TYPE</th>
<th>TS/EXT TYPE</th>
<th>DAYS TO FULL</th>
<th>REASON</th>
<th>VOLUME</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUS01W0</td>
<td></td>
<td>70</td>
<td>TP</td>
<td>R</td>
<td>0</td>
<td>MAXSIZE</td>
<td>EMC334</td>
</tr>
<tr>
<td>QZUS02W0</td>
<td></td>
<td>70</td>
<td>TP</td>
<td>R</td>
<td>0</td>
<td>MAXSIZE</td>
<td>EMC422</td>
</tr>
<tr>
<td>QZUS03W0</td>
<td></td>
<td>70</td>
<td>TP</td>
<td>R</td>
<td>0</td>
<td>MAXSIZE</td>
<td>EMC433</td>
</tr>
<tr>
<td>QZUS04W0</td>
<td></td>
<td>70</td>
<td>TP</td>
<td>R</td>
<td>0</td>
<td>VOLUME</td>
<td>EMC327</td>
</tr>
<tr>
<td>QZUS05W0</td>
<td></td>
<td>70</td>
<td>TP</td>
<td>R</td>
<td>0</td>
<td>VOLUME</td>
<td>EMC482</td>
</tr>
<tr>
<td>QZUS06W0</td>
<td></td>
<td>70</td>
<td>TP</td>
<td>R</td>
<td>0</td>
<td>MAXSIZE</td>
<td>EMC36A</td>
</tr>
<tr>
<td>QZUS07W0</td>
<td></td>
<td>70</td>
<td>TP</td>
<td>R</td>
<td>0</td>
<td>VOLUME</td>
<td>EMC447</td>
</tr>
<tr>
<td>QZUS08W0</td>
<td></td>
<td>70</td>
<td>TP</td>
<td>R</td>
<td>0</td>
<td>MAXSIZE</td>
<td>EMC367</td>
</tr>
<tr>
<td>QZUS09W0</td>
<td></td>
<td>70</td>
<td>TP</td>
<td>R</td>
<td>0</td>
<td>VOLUME</td>
<td>EMC37A</td>
</tr>
<tr>
<td>QZUS10W0</td>
<td></td>
<td>70</td>
<td>TP</td>
<td>R</td>
<td>0</td>
<td>VOLUME</td>
<td>EMC336</td>
</tr>
<tr>
<td>QZUX01_DWDS10T01</td>
<td>70</td>
<td>IP</td>
<td>0</td>
<td>MAXSIZE</td>
<td>EMC364</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUX01_DWDS04T01</td>
<td>70</td>
<td>IP</td>
<td>125</td>
<td>EXTENTS</td>
<td>EMC37E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUX01_DWDS05T01</td>
<td>70</td>
<td>IP</td>
<td>126</td>
<td>EXTENTS</td>
<td>EMC45E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUX01_DWDS02T01</td>
<td>70</td>
<td>IP</td>
<td>128</td>
<td>VOLUME</td>
<td>EMC447</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUX01_DWDS06T01</td>
<td>70</td>
<td>IP</td>
<td>136</td>
<td>EXTENTS</td>
<td>EMC367</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUX01_DWDS07T01</td>
<td>70</td>
<td>IP</td>
<td>136</td>
<td>EXTENTS</td>
<td>EMC37A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUX01_DWDS08T01</td>
<td>70</td>
<td>IP</td>
<td>136</td>
<td>EXTENTS</td>
<td>EMC336</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Database space trend report contains the following fields:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Name of the database for which the product generates the report</td>
</tr>
<tr>
<td>Name</td>
<td>Index name or the table space name, depending on the object type</td>
</tr>
<tr>
<td>Part</td>
<td>Number of the partition of the table space partition or index space partition</td>
</tr>
<tr>
<td>Type</td>
<td>Kind of DB2 object that is reported: index partition (IP) or table space partition (TP)</td>
</tr>
<tr>
<td>Days to Full</td>
<td>Predicted number of days until the DB2 object runs out of space</td>
</tr>
<tr>
<td>Reason</td>
<td>Reason that the object is expected to be full in the number of days shown</td>
</tr>
<tr>
<td>Volume</td>
<td>Direct Access Storage Device (DASD) volume is approaching capacity.</td>
</tr>
<tr>
<td>Extents</td>
<td>Volume is approaching the maximum number of extents.</td>
</tr>
<tr>
<td>Max Size</td>
<td>Object is approaching the maximum size that DB2 allows.</td>
</tr>
<tr>
<td>Volume</td>
<td>DASD volume that stores the DB2 object</td>
</tr>
</tbody>
</table>

The Database space trend report requires the following input parameter:

```
ssid plan dbname
```

**Database storage utilization report**

The Database Storage Utilization Report lists the following items:

- 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
- The name of the first volume for each table space and index of a DB2 database
- Totals by indexes

The report also displays, table spaces, table spaces and indexes, database—and by report if you specify multiple databases.

**Figure 191 on page 619** shows a Database Storage Utilization Report

---

**The Database storage utilization report contains the following fields:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database Object</strong></td>
<td>Name of the table space or index for which the product generates the report</td>
</tr>
<tr>
<td><strong>Part</strong></td>
<td>Number of the partition in the table space or index</td>
</tr>
<tr>
<td><strong>Trks Space</strong></td>
<td>Number of tracks that are allocated for the object</td>
</tr>
<tr>
<td><strong>Note</strong>:</td>
<td>To change report units to KB, see “Internal KB switch” on page 603.</td>
</tr>
<tr>
<td><strong>Trks Used</strong></td>
<td>Tracks that the object uses</td>
</tr>
<tr>
<td><strong>% Used</strong></td>
<td>Percentage of the allocated space that the object uses</td>
</tr>
<tr>
<td><strong>Ext</strong></td>
<td>Number of extents that the object is in</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNS</td>
<td>Number of data sets in the object</td>
</tr>
<tr>
<td>Qty</td>
<td>Primary allocation quantity</td>
</tr>
<tr>
<td>Sqty</td>
<td>Secondary allocation quantity</td>
</tr>
<tr>
<td>AU</td>
<td>Allocation unit (T—tracks or C—cylinders)</td>
</tr>
<tr>
<td>Vols</td>
<td>Number of volumes for the object</td>
</tr>
<tr>
<td>Volser</td>
<td>Volume serial number of the first volume for the object</td>
</tr>
<tr>
<td>Total – Tableparts</td>
<td>Total number of tracks allocated and used for the tables pace partitions</td>
</tr>
<tr>
<td>Total – Indexparts</td>
<td>Total number of tracks allocated and used for the index space partitions</td>
</tr>
<tr>
<td>Total – Indexes</td>
<td>Total number of tracks allocated and used for the indexes</td>
</tr>
<tr>
<td>Total – Table Space</td>
<td>Total number of tracks allocated and used for the table spaces</td>
</tr>
<tr>
<td>Total – Table Space and Indexes</td>
<td>Total number of tracks allocated and used for the table spaces and indexes</td>
</tr>
<tr>
<td>Total – Database</td>
<td>Total number of tracks allocated and used for the database</td>
</tr>
</tbody>
</table>

The Database storage utilization requires the following lines of input parameters:

```
ssid plan dbname vcatname
```

**Exception report for BMCTRIG action**

The Exception report by BMCTRIG Action 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 192 on page 620** shows an Exception report for BMCTRIG Action.

**Figure 192: Exception report for BMCTRIG Action**

```
Exception by BMCTRIG Action Report - E361499T

DASD MANAGER PLUS FOR DB2        - ASUEXCPT
Time . . . . : 11:30:39    15 Mar yyyy
DB2 ID . . . : DEAE       
DB2 Version. :           
User . . . . : RDAPKM2  

<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>INDEX</td>
<td>QZU.QZUX02_DSC30S01T</td>
<td>0</td>
<td>IXDIRTY</td>
<td>98</td>
<td>0</td>
</tr>
<tr>
<td>INDEX</td>
<td>QZU.QZUX02_DSC30S02T</td>
<td>0</td>
<td>IXDIRTY</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>INDEX</td>
<td>QZU.QZUX02_DSC30S03T</td>
<td>0</td>
<td>IXDIRTY</td>
<td>95</td>
<td>0</td>
</tr>
</tbody>
</table>
```
The Exception report for BMCTRIG action report contains the following fields:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj Type</td>
<td>Type of object</td>
</tr>
<tr>
<td>Name</td>
<td>Object name</td>
</tr>
<tr>
<td>Part</td>
<td>Partition number</td>
</tr>
<tr>
<td>Exception</td>
<td>Exception name</td>
</tr>
<tr>
<td>Current</td>
<td>Current value</td>
</tr>
<tr>
<td>Compare</td>
<td>Compare value</td>
</tr>
</tbody>
</table>

The Exception report for BMCTRIG action Report requires the following line of input parameters:

```
ssid plan  action
```

**Index clustering analysis report**

The Index Clustering Analysis Report 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
The Index Clustering Analysis report contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB.TS/Table/Index</td>
<td>Fully qualified table space name, the table name, and the index name that the product is reporting</td>
</tr>
<tr>
<td>Part</td>
<td>Number of the partition in the index</td>
</tr>
<tr>
<td>UR</td>
<td>Unique rule, indicating whether the index is unique. Valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td>■ C—Unique and used to enforce a unique constraint</td>
</tr>
<tr>
<td></td>
<td>■ D—Duplicates allowed</td>
</tr>
<tr>
<td></td>
<td>■ G—Unique and used to enforce the uniqueness of values in the column defined as ROWID GENERATED BY DEFAULT</td>
</tr>
<tr>
<td></td>
<td>■ N —Unique and defined with UNIQUE WHERE NOT NULL</td>
</tr>
<tr>
<td></td>
<td>■ P—Unique and primary index</td>
</tr>
<tr>
<td></td>
<td>■ R—Unique and used to enforce the uniqueness of a nonprimary parent key</td>
</tr>
<tr>
<td></td>
<td>■ U—Unique</td>
</tr>
<tr>
<td></td>
<td>■ X—Unique and used to enforce the uniqueness of values in a column that is used to identify or find XML values that are associated with a specific row</td>
</tr>
<tr>
<td>CR</td>
<td>Clustering ratio, the percentage of rows in clustering order</td>
</tr>
<tr>
<td>Data Repeat Factor</td>
<td>Number of pages read while following an index key order</td>
</tr>
<tr>
<td>Nearoff</td>
<td>Number of referenced rows near, but not in, optimal position because of an insert into a full page</td>
</tr>
<tr>
<td>Faroff</td>
<td>Number of referenced rows far from optimal position because of an insert into a full page</td>
</tr>
<tr>
<td>Card</td>
<td>Number of columns in the referenced table</td>
</tr>
</tbody>
</table>
The Index Clustering Analysis Report requires the following line of input parameters:

```plaintext
ssid plan dbname cr nearpct farpct
```

### Modified table space pages report

The Modified Table Space Pages 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.

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 `pctdirty` that the input parameters specify.

#### Figure 194: Modified Table Space Pages Report

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSNAME</th>
<th>PART</th>
<th>CARD</th>
<th>NACTIVE</th>
<th>DIRTY</th>
<th>DIRTY</th>
<th>COPY</th>
<th>STATTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUDDL</td>
<td>QZUS02DL</td>
<td>0</td>
<td>284</td>
<td>209</td>
<td>206</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS03DL</td>
<td>0</td>
<td>845</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS04DL</td>
<td>0</td>
<td>845</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS05DL</td>
<td>0</td>
<td>564</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS06DL</td>
<td>0</td>
<td>1125</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS07DL</td>
<td>0</td>
<td>2250</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS08DL</td>
<td>1</td>
<td>284</td>
<td>209</td>
<td>206</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS08DL</td>
<td>2</td>
<td>284</td>
<td>209</td>
<td>206</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS08DL</td>
<td>3</td>
<td>284</td>
<td>209</td>
<td>206</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS08DL</td>
<td>4</td>
<td>284</td>
<td>209</td>
<td>206</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS09DL</td>
<td>1</td>
<td>284</td>
<td>209</td>
<td>206</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS09DL</td>
<td>2</td>
<td>563</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS09DL</td>
<td>3</td>
<td>563</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS09DL</td>
<td>4</td>
<td>563</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS09DL</td>
<td>5</td>
<td>563</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS09DL</td>
<td>6</td>
<td>563</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS10DL</td>
<td>1</td>
<td>2250</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS10DL</td>
<td>2</td>
<td>845</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS10DL</td>
<td>3</td>
<td>563</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS10DL</td>
<td>4</td>
<td>845</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS10DL</td>
<td>5</td>
<td>845</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS10DL</td>
<td>6</td>
<td>284</td>
<td>209</td>
<td>206</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS11DL</td>
<td>1</td>
<td>2250</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS11DL</td>
<td>2</td>
<td>2250</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS11DL</td>
<td>3</td>
<td>2250</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
<tr>
<td>QZUDDL</td>
<td>QZUS11DL</td>
<td>4</td>
<td>2250</td>
<td>207</td>
<td>205</td>
<td>99</td>
<td>FULL</td>
<td>yyyy-10-28-12.50</td>
</tr>
</tbody>
</table>
The Modified Table Space Pages report contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBname/TSname</td>
<td>Fully qualified table space name</td>
</tr>
<tr>
<td>Part</td>
<td>Number of the partition in the table space</td>
</tr>
<tr>
<td>Card</td>
<td>Cardinality, or number of rows, in the table space</td>
</tr>
<tr>
<td>Nactive</td>
<td>Number of active pages in the table space</td>
</tr>
<tr>
<td>Dirty</td>
<td>Modified, or dirty, pages in the table space</td>
</tr>
<tr>
<td>%Dirty</td>
<td>Percentage of modified, or dirty, pages in the table space</td>
</tr>
<tr>
<td>Copy</td>
<td>Recommended type of image copy: full or incremental</td>
</tr>
<tr>
<td>Statstime</td>
<td>Date and time when the product collected statistics</td>
</tr>
</tbody>
</table>

The Modified Table Space Pages report requires the following line of input parameters:

`ssid plan dbname pctdirty`
Space estimation report

The Space Estimation 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.

The report (Figure 195 on page 625) 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 195: Space Estimation Report

The Space estimation report contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Object</td>
<td>Fully qualified DB2 object name</td>
</tr>
<tr>
<td>Part</td>
<td>Partition number</td>
</tr>
<tr>
<td>Pqty</td>
<td>Primary allocation quantity</td>
</tr>
<tr>
<td>Sqty</td>
<td>Secondary allocation quantity</td>
</tr>
<tr>
<td>AU</td>
<td>Allocation unit (T – tracks or C – cylinders)</td>
</tr>
<tr>
<td>% Free</td>
<td>Percentage of the allocated space that is free</td>
</tr>
<tr>
<td>Free Page</td>
<td>Number of free pages</td>
</tr>
<tr>
<td>Computed Card</td>
<td>Cardinality of the object, taken from the historical database, with the pctcard percentage applied.</td>
</tr>
<tr>
<td>KB</td>
<td>Estimated number of kilobytes after the reorganization</td>
</tr>
</tbody>
</table>
### Space estimation trend report

The Space Estimation Trend Report 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 196 on page 626 is an example of a Space estimation trend report.
The Space estimation trend report contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Object</td>
<td>Fully qualified table space or index name</td>
</tr>
<tr>
<td>Part</td>
<td>Partition number</td>
</tr>
<tr>
<td>Pqty</td>
<td>Primary allocation quantity</td>
</tr>
<tr>
<td>Sqty</td>
<td>Secondary allocation quantity</td>
</tr>
<tr>
<td>AU</td>
<td>Allocation unit (T – tracks or C – cylinders)</td>
</tr>
<tr>
<td>% Free</td>
<td>Percentage of the a percellocated space that is free</td>
</tr>
<tr>
<td>Free Page</td>
<td>Number of free pages</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change in Card per Day</td>
<td>Average increase or decrease in the number of rows per day over the specified number of days</td>
</tr>
<tr>
<td>Projected Card</td>
<td>Estimated number of rows in the object after the specified number of days, based on the change in cardinality per day</td>
</tr>
</tbody>
</table>
| Reorg Space mnn Days from report-date | 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. |

The Space estimation trend report requires the following lines of input parameters:

```
ssid plan qualtype qualname observations days
```

## Space utilization by table report

The Space Utilization by Table 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.

If you do not specify a table space name, the product retrieves all table spaces.

Figure 197 on page 628 is an example of a Space utilization by table report.
The Space utilization by table report contains the following fields:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBname/TSname</td>
<td>Fully qualified name of the table space</td>
</tr>
<tr>
<td>Type</td>
<td>Type of table space</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the table for which statistics appear</td>
</tr>
<tr>
<td>NPages</td>
<td>Number of pages that contain rows of this table</td>
</tr>
<tr>
<td>PctPgs</td>
<td>Percentage of table space pages that contain rows from this table</td>
</tr>
<tr>
<td>Card</td>
<td>Cardinality, or number of rows, in the table</td>
</tr>
</tbody>
</table>

The Space utilization by table report requires the following line of input parameters:

```
ssid plan dbname tsname
```
The Table Space and Index Storage Limits Report 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 (Figure 198 on page 630) prints a line for objects whose %GB is greater than or equal to the pctlimit. The report sorts the objects by the largest percentage of possible space used.

**Figure 198: 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 STAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUDVL</td>
<td>QZUS02VL</td>
<td>1</td>
<td>G</td>
<td>59904392</td>
<td>64</td>
<td>89.26</td>
<td>2.2E+09</td>
<td>264585582</td>
<td>01.16.2014</td>
</tr>
<tr>
<td>QZUDVL</td>
<td>QZUS03VL</td>
<td>0</td>
<td></td>
<td>59904392</td>
<td>64</td>
<td>89.26</td>
<td>2.2E+09</td>
<td>264585582</td>
<td>03.06.2014</td>
</tr>
<tr>
<td>QZUDIC</td>
<td>QZUS06IC</td>
<td>2</td>
<td>L</td>
<td>3473280</td>
<td>4</td>
<td>82.81</td>
<td>52645202</td>
<td>10992705</td>
<td>04.27.2014</td>
</tr>
<tr>
<td>QZUDIC</td>
<td>QZUS06IC</td>
<td>3</td>
<td>L</td>
<td>3473280</td>
<td>4</td>
<td>82.81</td>
<td>52642925</td>
<td>10928233</td>
<td>04.27.2014</td>
</tr>
<tr>
<td>QZUDIC</td>
<td>QZUS06IC</td>
<td>4</td>
<td>L</td>
<td>3473280</td>
<td>4</td>
<td>82.81</td>
<td>52609084</td>
<td>10921290</td>
<td>04.27.2014</td>
</tr>
<tr>
<td>QZUDIC</td>
<td>QZUS06IC</td>
<td>5</td>
<td>L</td>
<td>3473280</td>
<td>4</td>
<td>82.81</td>
<td>52668283</td>
<td>10933497</td>
<td>04.27.2014</td>
</tr>
<tr>
<td>QZUDJA</td>
<td>QZUS01JA</td>
<td>1</td>
<td>L</td>
<td>831196</td>
<td>1</td>
<td>79.27</td>
<td>1620678</td>
<td>423851</td>
<td>10.27.2015</td>
</tr>
<tr>
<td>QZUDXM</td>
<td>QZUS25XM</td>
<td>&lt;=</td>
<td>G</td>
<td>400508</td>
<td>1</td>
<td>38.20</td>
<td>50050</td>
<td>80987</td>
<td>04.27.2014</td>
</tr>
</tbody>
</table>

The Table space and index storage limits report contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBName/TSName</td>
<td>Fully qualified table space name</td>
</tr>
<tr>
<td>DBName/Creator/IXName</td>
<td>Fully qualified index name</td>
</tr>
<tr>
<td>Part</td>
<td>Partition number</td>
</tr>
<tr>
<td>Type</td>
<td>Type of table space</td>
</tr>
</tbody>
</table>
### Field and Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used Space (KB)</td>
<td>Space that the DB2 object currently uses, in kilobytes</td>
</tr>
<tr>
<td>Max GB</td>
<td>Maximum amount of space to be allocated for the object in gigabytes</td>
</tr>
<tr>
<td>% GB</td>
<td>Percentage of the maximum space that the object currently uses</td>
</tr>
<tr>
<td>Rows</td>
<td>Number of rows in the object</td>
</tr>
<tr>
<td>Estimated New Rows</td>
<td>Estimated number of rows still available for the object.</td>
</tr>
<tr>
<td>Date Stats</td>
<td>Date that you last ran BMCSTATS on the object</td>
</tr>
</tbody>
</table>

The Table space and index storage limits report requires the following line of input parameters:

```
ssid plan dbname pctlimit
```

### 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 (Figure 199 on page 631) 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 a message that there is nothing to report.

#### Figure 199: Tables with More than Average Growth report

```
TABLES WITH MORE THAN AVERAGE GROWTH REPORT

DASD MANAGER PLUS FOR DB2 - ASUNPGR
Time . . . .: 12:22:51  15 Mar yyyy
DB2 ID . . .: DEAE
DB2 Version.: 
User . . . .: RDAPKM2
AVERAGE GROWTH (PAGES/DAY): 15

------------- TABLE-------------  GROWTH     OBSERVATION
CREATOR.NAME  (PAGES/DAY)     AGE IN DAYS
QZU.QZUT01_DSC30S22  15         559

TABLES WITH MORE THAN AVERAGE GROWTH REPORT

DASD MANAGER PLUS FOR DB2 - ASUNPGR
Time . . . .: 12:22:51  15 Mar yyyy
DB2 ID . . .: DEAE
DB2 Version.: 
User . . . .: RDAPKM2
Input Parameters
ssid plan  table-creator
DEAE ASUB2DDR %
```

The Tables with more than average growth report contains the following fields:
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Growth</td>
<td>Average increase in the size of the tables per day, expressed in number of pages/day</td>
</tr>
<tr>
<td>Table Creator/Name</td>
<td>Fully qualified table name. The report lists only the tables that have grown more than the value of the Average Growth field.</td>
</tr>
<tr>
<td>Growth (Pages/Day)</td>
<td>Average number of pages per day that have been added to the table as determined by the last two instances of captured statistics</td>
</tr>
<tr>
<td>Observation Age in Days</td>
<td>Number of days since the last statistics collection. An 0 age indicates that you collected statistics on the current date</td>
</tr>
</tbody>
</table>

The Tables with more than average growth report requires one of the following lines of input parameters:

```
ssid plan table-creator
```

**Table space and index extents report**

The Table Space and Index Extents Report 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 200 on page 632* is an example of a Table space and index extents report.

---

**Figure 200: Table Space and Index Extents Report**

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSNAME</th>
<th>PART</th>
<th>PQTY</th>
<th>SQTY</th>
<th>A</th>
<th>SPACE</th>
<th>EXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUDB8</td>
<td>QZU00014</td>
<td>0</td>
<td>11</td>
<td>3</td>
<td>C</td>
<td>1179375</td>
<td>427</td>
</tr>
<tr>
<td>QZUDXT</td>
<td>QZUS01XT</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>T</td>
<td>226</td>
<td>226</td>
</tr>
<tr>
<td>QZUSDSC31</td>
<td>QZUS0131</td>
<td>0</td>
<td>112</td>
<td>70</td>
<td>C</td>
<td>117420</td>
<td>186</td>
</tr>
<tr>
<td>QZUD94</td>
<td>QZUS0294</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>C</td>
<td>4380</td>
<td>182</td>
</tr>
<tr>
<td>QZUSDSC31</td>
<td>QZUS0231</td>
<td>0</td>
<td>112</td>
<td>70</td>
<td>C</td>
<td>117420</td>
<td>146</td>
</tr>
<tr>
<td>QZUDTS</td>
<td>QZUS01TS</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>T</td>
<td>106</td>
<td>106</td>
</tr>
<tr>
<td>QZUD60</td>
<td>QZUS0160</td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>T</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>QZUS56</td>
<td>QZUS5665C</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>T</td>
<td>82</td>
<td>82</td>
</tr>
<tr>
<td>QZUD97</td>
<td>QZUS0197</td>
<td>4</td>
<td>54</td>
<td>6</td>
<td>T</td>
<td>534</td>
<td>81</td>
</tr>
<tr>
<td>QZUDB6</td>
<td>QZUS01B6</td>
<td>4</td>
<td>56</td>
<td>7</td>
<td>T</td>
<td>553</td>
<td>72</td>
</tr>
<tr>
<td>QZUS02B6</td>
<td></td>
<td>4</td>
<td>56</td>
<td>7</td>
<td>T</td>
<td>539</td>
<td></td>
</tr>
</tbody>
</table>

---

*632  DASD MANAGER PLUS for DB2 User Guide*
### TABLE SPACE AND INDEX EXTENTS REPORT

DASD MANAGER PLUS FOR DB2 11.02.00 - ASUDBEXT

- **Time**: 12:27:48 15 Mar
- **DB2 ID**: DEAE
- **DB2 Version**: 1015
- **User**: RDAPKM2

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>CREATOR</th>
<th>IXNAME</th>
<th>PART</th>
<th>PQTY</th>
<th>SQTY</th>
<th>U</th>
<th>SPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUD17</td>
<td>QZU</td>
<td>QZUX03_D17S07T01</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>C</td>
<td>1860</td>
</tr>
<tr>
<td>QZUD17</td>
<td>QZU</td>
<td>QZUX01_D17S06T01</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>C</td>
<td>1860</td>
</tr>
<tr>
<td>QZUD17</td>
<td>QZU</td>
<td>QZUX01_D17S07T01</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>C</td>
<td>1860</td>
</tr>
<tr>
<td>QZUD17</td>
<td>QZU</td>
<td>QZUX03_D17S05T01</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>C</td>
<td>1860</td>
</tr>
<tr>
<td>QZUD17</td>
<td>QZU</td>
<td>QZUX03_D17S06T01</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>C</td>
<td>1860</td>
</tr>
<tr>
<td>QZUD17</td>
<td>QZU</td>
<td>QZUX03_D17S07T01</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>C</td>
<td>1860</td>
</tr>
<tr>
<td>QZUD17</td>
<td>QZU</td>
<td>QZUX03_D17S07T01</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>C</td>
<td>1860</td>
</tr>
<tr>
<td>QZUD17</td>
<td>QZU</td>
<td>QZUX03_D17S07T01</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>C</td>
<td>1860</td>
</tr>
<tr>
<td>QZUD94</td>
<td>QZU</td>
<td>QZUX01_D94S02T01</td>
<td>0</td>
<td>42</td>
<td>1</td>
<td>T</td>
<td>136</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>336</td>
</tr>
<tr>
<td>QZUD17</td>
<td>QZU</td>
<td>QZUX01_D17S05T01</td>
<td>0</td>
<td>32</td>
<td>1</td>
<td>C</td>
<td>1860</td>
</tr>
<tr>
<td>QZUDVL</td>
<td>QZU</td>
<td>QZUX02_DVLS01T01</td>
<td>0</td>
<td>729</td>
<td>183</td>
<td>C</td>
<td>716625</td>
</tr>
</tbody>
</table>

### TABLE SPACE AND INDEX EXTENTS REPORT
The Table space and index extents report contains the following fields:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBName/TSName</td>
<td>fully qualified name of the table space</td>
</tr>
<tr>
<td>Part</td>
<td>partition number</td>
</tr>
<tr>
<td>Pqty</td>
<td>primary allocation quantity</td>
</tr>
<tr>
<td>Sqty</td>
<td>secondary allocation quantity</td>
</tr>
<tr>
<td>AU</td>
<td>allocation unit (T – tracks or C – cylinders)</td>
</tr>
<tr>
<td>Space</td>
<td>allocated space in tracks or cylinders</td>
</tr>
<tr>
<td>Ext</td>
<td>number of extents for the object</td>
</tr>
<tr>
<td>DBName/Creator/IXName</td>
<td>fully qualified index name</td>
</tr>
</tbody>
</table>

The Table space and index extents report requires the following lines of input parameters:

```
ssid plan dbname extents
```

**Volume space trend report**

The Volume Space Trend Report 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 201: Volume Space Trend Report**
The Volume space trend report contains the following fields:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume</td>
<td>volume serial number of the DASD volume</td>
</tr>
<tr>
<td>Days to Full</td>
<td>estimated number of days until the volume is full</td>
</tr>
<tr>
<td>Free Tracks</td>
<td>number of free tracks on the volume</td>
</tr>
</tbody>
</table>

The Volume space trend report requires the following line of input parameters:

```plaintext
$ssid plan volume
```
Exporting and deploying product definitions

This chapter discusses exporting object definitions. The chapter also explains how to set up and implement the DASD MANAGER PLUS Export utility.

Overview of exporting object definitions

The Export utility is available to users who have a license and password for the 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 you cannot delete definitions. (You must delete the definitions manually by logging into IBM z/OS images and accessing DASD MANAGER PLUS.)

How the Export utility works

When deploying definitions, the Export utility 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 utility 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.
If you explicitly select object Sets A, B, and C to be exported, the Export utility 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, the Export utility overwrites object Sets A, B, and C with the definitions from the source (if you selected **Overwrite existing rows** on the Export Deployment panel).

The Export utility lists dependent objects and their parent objects as follows:

- Services
  - Service syntax

- Exception definitions
  - Thresholds
  - Corrective actions

The Export utility also validates remote object definitions that a selected definition references (called referenced objects).

**Example**

When exporting an action, the Export utility validates that the referenced service is being exported (because you selected it), or that the referenced service already exists on the destination.

Table 96 on page 638 lists the types of definitions and how the Export utility analyzes them.

**Table 96: Definition validation**

<table>
<thead>
<tr>
<th>Exported definition</th>
<th>Dependent objects included implicitly</th>
<th>Referenced objects that the Export utility validates</th>
<th>Referenced objects that the Export utility does not validate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td></td>
<td>Service</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Service syntax</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Object sets</td>
<td></td>
</tr>
<tr>
<td>Corrective action</td>
<td></td>
<td>Exception definition</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Action</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Object sets</td>
<td></td>
</tr>
<tr>
<td>Exception definition</td>
<td>Corrective action</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exception</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Threshold</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Exported definition

<table>
<thead>
<tr>
<th>Dependent objects included implicitly</th>
<th>Referenced objects that the Export utility validates</th>
<th>Referenced objects that the Export utility does not validate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object-action priorities</td>
<td></td>
<td>Object sets Action</td>
</tr>
<tr>
<td>Object sets</td>
<td></td>
<td>Object sets</td>
</tr>
<tr>
<td>Service</td>
<td>Service syntax</td>
<td></td>
</tr>
<tr>
<td>Exception thresholds</td>
<td>Exception definition</td>
<td>Object sets</td>
</tr>
</tbody>
</table>

*The Export utility 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 object types that you must include in the export if the referenced objects do not already exist on the destination.

#### 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 as well (if they do not already exist on the destination). Otherwise, you will receive an error.

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.
- Therefore, 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 treated as 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.

#### Exporting a definition that has referenced objects not referentially related to the definition

You must explicitly specify the referenced definition in the export if all of the following conditions are true:

- You are exporting an object that references another definition.
- The object is not referentially related to the other definition.
- The referenced definition does not already exist on the destination.
If you do not specify the referenced definition for export in this case, the product does not issue an error message but the jobs on the destination system do not operate as expected. This outcome applies to object actions, object sets, thresholds, and corrective actions that reference object sets as well as actions whose names can include wildcards.

**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, the product does not issue an error, but you might encounter unexpected results.

**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.

**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 diverge from 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 the service. 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 Syntax B. Export includes Syntax A and Syntax B definitions in the export. If Service A exists with Syntax A, Syntax B, and Syntax C on the destination, Export deletes the Syntax C definition from the destination and updates Service A with Syntax A and Syntax 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.

These recommendations help 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.

Connection list types

When you launch Export, you must define at least one host connection. You can then add and work with DB2 data sources. Defined host connections remain available each time you start Export and log in. All host connections reside on the UIM server.

There are three types of connection lists:

- **Enterprise connection list (ECL)—**a shared list used to identify the host connections that you define (via host and port information)
  
  The ECL contains host and port information for one or more UIM servers.

  **Administration:** One or more administrators maintain the ECL. Users with appropriate authority can add, delete and edit ECL connection information.

  **DB2 subsystem information:** The ECL lacks DB2 subsystem information. Consequently, the ECL cannot be used as an Export destination. The ECL is more commonly used for IMS connections. From a DB2 perspective, you can use ECL as a seed value to create a PCL or CCL. The PCL and CCL do contain DB2 subsystem information and can be used as Export destinations.

- **Personal Connection List (PCL)—**a host connection for personal use
  
  Separating personal connections from host connections makes it simple to isolate activities in different environments (such as testing systems and production systems or different groups of application systems).

  The PCL cannot be shared.
Common Connection List (CCL)—similar to PCL, but can be shared

You can obtain all destinations from your PCL or CCL. To define PCL or ECL, enter the connection information (host name, port and DB2 subsystem ID). If you administrator has set up an ECL, you can create a PCL or CCL from an ECL by adding the DB2 subsystem ID information.

*Note*

To delete definitions from multiple data sources, you must manually delete the definitions from each data source.

**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 and IP and UIM Server access**

Export uses existing login credentials for the definition phase. Export prompts you for login credentials:
- When you specify a UIM connection for the primary UIM Server, and
- For any other UIM Servers participating 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 IBM RACF security administrator must perform the following tasks:
- Define an Open Multiple Virtual Storage (OMVS) segment for the UIM Server started task in order to enable TCP/IP access
- 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 that stores PCLs and ECLs. Communication to the primary UIM Server uses POF values for host name and port number. The security administrator usually specifies this information during installation.

When you launch Export, DASD MANAGER PLUS accesses the POF to retrieve the primary UIM host name and port number. Export then does the following:
- Prompts you for a TSO user ID and password
- 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 the administrator does
not set the option, Export uses the system where you are logged in as the primary
UIM server.

Table 97 on page 643 lists the POF keywords that are associated with Export.

**Table 97: POF keywords for specifying the UIM host definition**

<table>
<thead>
<tr>
<th>POF keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ASU_XP_LOGD_DATAC</strong>=</td>
<td>Specifies the SMS data class and the allocation attributes of the Export</td>
</tr>
<tr>
<td></td>
<td>log file</td>
</tr>
<tr>
<td><strong>ASU_XP_LOGD_MGMTC</strong>=</td>
<td>Specifies the SMS management class that defines the migration,</td>
</tr>
<tr>
<td></td>
<td>retention, and backup requirements of the Export log file</td>
</tr>
<tr>
<td><strong>ASU_XP_LOGD_PRIQTY</strong>=10</td>
<td>Defines the primary allocation for the Export log file</td>
</tr>
<tr>
<td><strong>ASU_XP_LOGD_SECQTY</strong>=2</td>
<td>Defines the secondary allocation for the Export log file</td>
</tr>
<tr>
<td><strong>ASU_XP_LOGD_STORC</strong>=</td>
<td>Specifies the SMS storage class that defines the processing</td>
</tr>
<tr>
<td></td>
<td>requirements of the Export log file</td>
</tr>
<tr>
<td><strong>ASU_XP_LOGD_UNIT</strong>=SYSDA</td>
<td>Specifies the unit for the Export log file</td>
</tr>
<tr>
<td><strong>ASU_XP_LOGDSN</strong>=&amp;PREFIX..XPORT.LOG(R)</td>
<td>Specifies the Export log file</td>
</tr>
<tr>
<td><strong>ASU_XP_UIMSRVPORT</strong>=</td>
<td>Specifies the port number of the primary UIM server that contains the</td>
</tr>
<tr>
<td></td>
<td>host definitions repository for the Export utility</td>
</tr>
<tr>
<td><strong>ASU_XP_UIMSRVHOST</strong>=</td>
<td>Specifies the host name of the primary UIM server for the Export utility</td>
</tr>
<tr>
<td><strong>ASU_XP_UIMSRVTIMEOUT</strong>=300</td>
<td>Specifies the UIM timeout parameter that determines how long the Export</td>
</tr>
<tr>
<td></td>
<td>utility should wait for a response from the UIM server before timing out</td>
</tr>
</tbody>
</table>

**TCP and IP and user access**

You must define an OMVS segment in IBM RACF for each Export user ID. You must
have an OMVS segment 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 CCL

When defining Export destinations, one or more administrators can set up and manage a shared list of connections. If you make the shared list available to other users, they do not have to define their own PCLs.

Users with appropriate authority can modify information in the ECL and CCL. The security administrator sets the authority levels that enable users’ access to and editing privileges for these connections. Table 98 on page 644 displays the available authority levels and their explanations.

Table 98: Authorization to edit the ECL or CCL

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>You must set up your own connections. You cannot view or edit the ECL or CCL.</td>
</tr>
<tr>
<td>READ</td>
<td>You can view and select connection definitions from the ECL or CCL.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>You can view, edit, delete, and add connection definitions in the ECL or CCL.</td>
</tr>
</tbody>
</table>

In addition, the SAF provides an interface to your security product (for example, Computer Associates ACF, eTrust CA ACF2 Security for DB2, or eTrust CA Top Secret for DB2). SAF determines who can access z/OS resources, and at what levels, by using security rules. You can use SAF to define who can read or maintain the ECL based on:

- User ID
- Product function or feature

Protection of access controls

During installation of console-enabled products, the product does not prompt you to implement access controls. The product does prompt you to define address spaces for the UIM server and any other servers that are installed. Files in the server address space determine server configuration.

UIM server configuration

The UIM server configuration file SMF$DHSP is an XML configuration file that is used to implement access controls. The file is installed in the UIM server configuration data set (default data set name HLQ.CONFIG) on the mainframe.

To protect access control policy from unauthorized changes, you must secure update access to the XML configuration file. Doing so protects defined resource names from being substituted.
Figure 202 on page 645 displayed the format that is used in the XML configuration file for specifying access controls. The format includes the name of the rule, the name of the resource, and the SAF class.

**Figure 202: Console configuration file excerpt**

```xml
<rule name="SDBA_DNA_EDIT_CONNECTION_LIST"
     resource="BBM.SDBA.DNA.ECL"
     class="FACILITY" />
<rule name="BBM.SDBA_DNA_EDIT_CCLCONNECTION_LIST"
     resource="BBM.SDBA.DNA.CCL"
     class="FACILITY" />
<rule name="SDBA_DNA_SPOOL"
     resource="BBM.SDBA.DNA.%FUNCTION%.SPOOL"
     class="FACILITY" />
```

**Creation of SAF resources**

You can base the SAF resources that you create on the following elements:

- User ID
- Product function or feature

The following access types are predefined:

- **ACL_Full**—Access Control List Read/Write/Alter
  Users can:
  - Reference the CCL in the Select Export Destinations dialog
  - Manage Destinations
  - Administer the CCL (add, delete and update connections)

- **ACL_Write**—Access Control List Read/Write
  Users have the same access as ACL_Full.

- **ACL_Read**—Access Control List Read only
  - Users can reference the CCL in the Select Export Destination dialog.
  - Users cannot manage Destinations.

- **ACL_None**—Access Control List none
  CCL is not used. Export reverts to the PCL. Individual users manage their own connection lists.

  The rule name SDBA_DNA_COMMON_CONNECTION_LIST was not found in the SMF$DHSP member.

SAF uses classes to simplify security access implementation. When you create SAF resources, BMC recommends that you use the predefined FACILITY class. To use a different class, you must edit the XML configuration file HLQ.CONFIG(SMF$DHSP) attributes. The class name that is defined to SAF must be the same as the class name.
attribute in the XML configuration file. The XML configuration file should be protected from UPDATE access after installation to prevent a user from bypassing access controls.

**SAF resource definition for the console**

SAF resource definitions for the console use the naming pattern BBM.SDBA.*.*.*

BBM.SDBA is the high-level qualifier to define resources. The asterisks represent additional product-specific qualifiers. The number of low-level qualifiers is product specific. Table 99 on page 646 The table below shows how to access resource types.

<table>
<thead>
<tr>
<th>Resource type</th>
<th>Specify this definition</th>
<th>To create a SAF resource that controls access to this resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECL</td>
<td>BBM.SDBA.DNA.ECL</td>
<td>ECL</td>
</tr>
<tr>
<td>CCL</td>
<td>BBM.SDBA.DNA.CCL</td>
<td>CCL</td>
</tr>
<tr>
<td>SPOOL</td>
<td>BBM.SDBA.DNA.JOBVIEW.SPOOL</td>
<td>JES queue viewer</td>
</tr>
<tr>
<td>SPOOL</td>
<td>BBM.SDBA.DNA.JOBMASK.SPOOL</td>
<td>Filter for the jobs that are displayed in the JES viewer</td>
</tr>
</tbody>
</table>

**Note**

If any of these resources (for example, RACF or ACF2) are not defined to your security system, all users have unrestricted access to the function that invokes the undefined resource.

**Task summary for exporting object definitions**

The following table 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>“GUID-C76581A1-E208-4A99-BA73-AC4D86E945F0.xml”</td>
</tr>
<tr>
<td></td>
<td>Review the tasks for preparing your environment</td>
<td></td>
</tr>
<tr>
<td>Access Export</td>
<td>Access the Export utility.</td>
<td>“Accessing the Export utility” on page 648</td>
</tr>
<tr>
<td>Task</td>
<td>Subtasks</td>
<td>Reference</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</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 649</td>
</tr>
<tr>
<td>Select definitions to export</td>
<td>Select actions to export</td>
<td>“Specifying action definitions” on page 654</td>
</tr>
<tr>
<td></td>
<td>Select services to export</td>
<td>“Specifying service definitions” on page 655</td>
</tr>
<tr>
<td></td>
<td>Select object sets to export</td>
<td>“Specifying object set definitions” on page 656</td>
</tr>
<tr>
<td></td>
<td>Select object-action priorities</td>
<td>“Specifying object-action priority definitions” on page 656</td>
</tr>
<tr>
<td></td>
<td>Select exceptions to export</td>
<td>“Specifying exceptions definitions” on page 657</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 658</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 659</td>
</tr>
<tr>
<td>Deploy definitions</td>
<td>Execute the request to deploy the definitions to other DB2 subsystems.</td>
<td>“Deploying the definitions” on page 659</td>
</tr>
<tr>
<td>Review status</td>
<td>Review the results</td>
<td>“Reviewing the deployment report” on page 661</td>
</tr>
</tbody>
</table>

### Preparing your environment for exporting

Use this procedure 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 “Using the UIM server” on page 717.
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 JESMSGLG 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 this 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

Use the following procedure to access Export. When you launch Export, you must define at least one host connection.

Before using this procedure, ensure that you have completed all tasks in “Preparing your environment for exporting” on page 647.

Note

To use the Export utility from DASD MANAGER PLUS, you must have a license and password for Database Performance for DB2 solution.

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  

(\textit{optional}) To use these credentials for all logins throughout your session, choose \textbf{Select to use these credentials for all logins} at the bottom of the panel.

c  

Press \textbf{Enter}.

The required steps are automatically selected as indicated by the letter \textit{S} on the panel. After you press \textbf{Enter}, additional dialogs for the selected steps are displayed in sequential order. When you complete a step, the \textit{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 649).

- Select definitions to be exported (as described in “Specifying definitions to export” on page 654).

- Select destinations for the exported definitions (as described in “Selecting the destinations for an export” on page 659).

- Execute the request to deploy definitions (as described in “Deploying the definitions” on page 659).

\section*{Setting up connections}

The Manage destinations panels help you manage connection information for your export sessions. From this panel you can set up and manage the following connections:

- Enterprise connections (ECL)
- Personal (destination) connections (PCL)

\section*{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 a</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 a</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 a</td>
<td>1. From the Delete Confirmation panel, confirm the deletion of the host by specifying option 1.</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 one of 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.

   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 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.

4. 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 Enterprise List.  
  ■ Choose 2 to enter host 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. |
<table>
<thead>
<tr>
<th>If you enter this in the Act column</th>
<th>Location</th>
<th>Complete this action</th>
</tr>
</thead>
</table>
| S                                   | SSID     | 1 From the Export Destination Subsystems panel, choose one of the following options:  
2 Choose 1 to select an available connection from the subsystem discovery list.  
3 Choose 2 to enter subsystem information manually.  
4 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.  
5 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. |
| 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. |
If you enter this in the Act column | Location | Complete this action
---|---|---
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

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.

### 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 637.

**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.
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 637.

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.
If you chose L in Step 3 on page 655, 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.
Typing A in the Act column next to Object-Action Priorities selects all object-action priorities.

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 637.

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 637.

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 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.

3 Press END to save the specifications.

Deploying the 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.

*Note*
As an alternative, you can choose **Select to use these credentials for all logins** to use the user ID and password for any remaining destinations that require login information.

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.

      *Note*
      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**.

      *Note*
      After the deployment is complete, you can view the Deployment Report to check status. For more information, see “Reviewing the deployment report” on page 661.

5 Press **Enter**.

When you run deployment, the Export Destination Login panel appears. From this panel, specify a login to each destination. As an alternative, you can choose **Select to use these credentials for all logins** to use the User ID and password for any remaining destinations that require login information.
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 101: Report Statuses in Deployment reports

<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>RIErrror</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 section describes the compiler, how to test SLIBs before compiling them, and the compiler’s associated runtime unit.

BMC has improved the performance of JCL construction by using the BMC skeleton library (SLIB) compiler. The SLIB compiler is a tool that is supplied with the Administrative Products for DB2. The use of compiled SLIBs with JCL Generation’s runtime unit, which was designed to handle the compiled SLIBs, can eliminate the ISPF file tailoring process. Figure 203 on page 664 illustrates the processing flow of the SLIB compiler.
Figure 203: Processing flow of the SLIB compiler
SLIB compilation

The SLIB compiler is written in REXX and runs as a batch TSO job.

Each SLIB is compiled, assembled, and linked into a separately executable load module. The separation of load modules provides maximum flexibility and ease of use because no other dependencies between SLIBs or other object modules exist. You can change one SLIB without having to recompile or relink any other SLIBs or modules.

The SLIB compiler examines each line of the interpretive SLIB language and translates it into assembler source code with commands and instructions that directly interact with the runtime unit. The runtime unit processes the requests and builds the JCL. Like other compilers, the SLIB compiler translates a higher-level language into assembler instructions. Like compilers for C, C++, COBOL, and other languages, the SLIB compiler has its own runtime unit.

Note

The SLIB compiler does not support all options provided with ISPF file tailoring, but it does support all features that the Administrative products currently use.

The SLIB compiler assumes that the SLIB adheres to standard ISPF file tailoring rules and constructs. Because the compiler does little verification of SLIB syntax, BMC recommends that you verify SLIB changes by using standard ISPF file tailoring before you compile the SLIBs. For details, see “SLIB verification using ISPF file tailoring” on page 666.

SLIB changes

To change an SLIB, code the changes to your SLIB source.

Before you compile the SLIB, you should use JCL Generation to test the changes by using ISPF standard file tailoring. Testing the changes ensures that the SLIB is coded correctly and that no ISPF-related errors exist.

You must recompile an SLIB each time that you change its source. Sample JCL for the SLIB compiler is in member AJXCOMPS in the HLQ.BMCCNTL data set that BMC provides at installation. To customize the JCL to your shop’s standards, follow the directions provided in this member.
WARNING
BMC uses SMP/E to package and deliver the SLIB members. If you must make a change to an SLIB, consider copying the SLIB member or members to be changed into a separate library. You can then make your changes to the SLIB members in that library without the risk of applying SMP/E maintenance that overlays your changes. Note that PTFs or GA releases that BMC distributes might change the SLIB source. You will need to determine whether the SLIB source was changed. If the SLIB source was changed, you will need to copy the new version of the SLIB source to your separate library, and reapply any changes that you made.

SLIB verification using ISPF file tailoring

The runtime unit first attempts to process compiled SLIBs. If the runtime unit cannot process a compiled SLIB, the unit reverts to standard ISPF file tailoring.

Note
If standard file tailoring is required, the SLIB that is being processed must exist in the ISPSLIB data set.

Any of the following criteria force the runtime unit to use standard ISPF file tailoring for an SLIB:

- A compiled SLIB cannot be loaded from either STEPLIB or ISPLLIB.
  Remove or rename the compiled version of the SLIB in the STEPLIB or ISPLLIB.
  Removing the compiled version forces the runtime unit to process the uncompiled SLIB, using standard ISPF file tailoring.

- The first four bytes of the compiled SLIB contain binary zeros.
  Insert a new line 1 in the SLIB. Beginning in column 1, enter:

  )CM NO-COMPILE.

  Then compile the SLIB.
  The compiler recognizes this SLIB as non- compilable and builds a load module with binary zeros in the first four bytes. The binary zeros force the runtime unit to process this SLIB by using standard ISPF file tailoring.
The ddname, $USESTFT, is allocated as DUMMY to either the TSO session or the batch job that is being executed. Add the ddname $USESTFT to the JCL stream for batch jobs, or allocate it to your TSO session with the TSO ALLOC command. This option forces all SLIBs to be processed by standard ISPF file tailoring. For example, in Batch Execution JCL Generation, add the following JCL:

```
//$USESTFT DD DUMMY
```

The option remains in effect for the duration of the batch job. In foreground processing, issue the following command before entering the product:

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

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

```
TSO FREE FI($USESTFT)
```

### Compilation of changed SLIBs

BMC strongly recommends that you process all SLIBs as compiled SLIBs, because the runtime performance can be adversely affected by processing non-compiled SLIBs.

**Note**

Before you test the compiled SLIB, turn off or remove any options that you used, such as $USESTFT and )CM NO-COMPILE.

You can use the report that the runtime unit generates to verify that you are running with compiled SLIBs and that the compile date on the SLIB is what you expect. For more information about the runtime report, see “Generating the SLIB report” on page 668.

After you successfully test the SLIB using standard file tailoring, compile the SLIB into your production HLQ.UBMCLINK library.

### SLIB processing

The runtime unit replaces the ISPF file tailoring interface. In most cases, interfaces to the runtime unit are the same as interfaces to ISPF file tailoring. To improve runtime performance, the runtime unit offers special logic that is designed specifically for JCL Generation.

The runtime unit handles mixed mode processing of compiled and noncompiled SLIBs. However, mixed mode processing is not recommended because it
compromises the improved performance that the runtime unit was designed to provide.

All JCL processing flows through the runtime unit. The runtime unit does the following:

- Resolves all variables
- Provides numeric data padding
- Handles I/O
- Processes standard file tailoring requests, when necessary
- Builds a report about the JCL generation process

### Generating the SLIB report

The reporting feature of the runtime unit helps you determine which SLIBs were processed, how they were processed, and when they were last assembled.

**To generate the SLIB report**

To use the SLIB reporting feature in Batch Execution JCL Generation, perform the following steps:

1. Add the ddname JGENSRPT to your batch JCL stream in the step that executes AJXBMAIN with a DCB of the following parameters:

   ```
   //JGENSRPT DD SYSOUT=*,
   // DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB,DSORG=PS)
   ```

2. Resubmit your job.

   If you need to produce this report in the foreground, you can use the TSO ALLOC command to allocate the *ddname* to any data set with RECFM=FB and LRECL=80. An example follows:

   ```
   TSO ALLOC FI(JGENSRPT) DA('dataSet.name')SHR
   ```

   In this example, *dataSet.name* is an existing sequential data set of RECFM=FB and LRECL=80.

**Figure 204 on page 668** shows a sample runtime report.

**Figure 204: Sample runtime report**

<table>
<thead>
<tr>
<th>Skelname</th>
<th>Usage</th>
<th>Compile Date</th>
<th>Compile Time</th>
<th>Usage Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJX$ACMX</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.41</td>
<td>1</td>
</tr>
</tbody>
</table>
The report summary at the end of Figure 204 on page 668 provides the information shown in Table 102 on page 669.

Table 102: Runtime report statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of FTINCLs</td>
<td>Number of file tailoring FTINCL requests</td>
</tr>
<tr>
<td>Number of )IMs</td>
<td>Number of imbeds that are encountered when FTINCLs are processed</td>
</tr>
<tr>
<td>SLIBs processed</td>
<td>43</td>
</tr>
<tr>
<td>Number of JCLRECs</td>
<td>166</td>
</tr>
<tr>
<td>Runtime units lastcc</td>
<td>0</td>
</tr>
<tr>
<td>Runtime units maxrc</td>
<td>0</td>
</tr>
</tbody>
</table>

Totals

Number of FTINCLs: 20
Number of )IMs: 55
SLIBs processed: 43
Number of JCLRECs: 166
Runtime units lastcc: 0
Runtime units maxrc: 0
<table>
<thead>
<tr>
<th><strong>Statistic</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>SLIBs processed</td>
<td>Number of SLIBs</td>
</tr>
<tr>
<td>Number of JCLRECs</td>
<td>Number of JCL records</td>
</tr>
<tr>
<td>Runtime units lastcc</td>
<td>Last condition code encountered</td>
</tr>
<tr>
<td>Runtime units maxrc</td>
<td>Highest return code encountered</td>
</tr>
</tbody>
</table>
Customizable reports

This section describes the report programs and emphasizes customization.

Overview of customizable reports

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 z/OS:

- 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 103 on page 672.
Table 103: External functions not available in REXX

<table>
<thead>
<tr>
<th>External function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASURXSQL (REXX SQL)</td>
<td>ASURXSQL runs the following database attachment and SQL statements: ▫ CONNECT ▫ DISCONNECT ▫ OPEN cursor ▫ FETCH cursor ▫ CLOSE cursor ▫ SELECT ▫ COMMIT ▫ ROLLBACK</td>
</tr>
<tr>
<td></td>
<td>ASURXSQL functionality goes beyond reporting. Using ASURXSQL, you can run any SQL statement including the following statements: ▫ CREATE ▫ INSERT ▫ UPDATE ▫ DELETE ▫ GRANT</td>
</tr>
<tr>
<td></td>
<td>The ability to modify the SQL or substitute new SQL gives the report programs their great flexibility.</td>
</tr>
<tr>
<td>ASURXRPT (REXX Report)</td>
<td>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.</td>
</tr>
<tr>
<td>ASURXLSP (REXX LSpace)</td>
<td>ASURXLSP reads the volume table of contents of direct access storage devices (DASD VTOCs) to collect real-time information about space on volumes.</td>
</tr>
<tr>
<td>ASURXSE (REXX Space</td>
<td>ASURXSE returns the reorganized space for the specified partitioned table space, nonpartitioned table space, table or index.</td>
</tr>
<tr>
<td>Estimation)</td>
<td></td>
</tr>
<tr>
<td>ASURXLOC (REXX Locate)</td>
<td>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.</td>
</tr>
</tbody>
</table>

Customizable report programs

This section describes the DASD MANAGER PLUS report programs and the external functions that the programs call, and provides brief notes on REXX as it applies to the report programs.

This section illustrates report layout, variables, and SQL using a sample program. The customizable report programs are in the HLQ.CLIST library. (For more information, see “Producing reports” on page 547.)
Report program structure

DASD MANAGER PLUS customizable report programs share the same basic structure.

**Figure 205: Basic report program structure**

```plaintext
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 the following figure is based on one of the report programs.

The adaptation produces the Volume free space report 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. 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 682
- “ASURXRPT external function” on page 684
- “ASURXLSP external function” on page 686
- “ASURXSE external function” on page 686
- “ASURXLOC external function” on page 691

**Figure 206: Sample REXX report program**

```rexx
/* rexx **************************************************************/
/* detail-line format definition                                             */
01 format. = ;
04 format.1 = "@<<<< @>>>>>>>";
05 format.2 = "vl.volid vl.freetrk";
06 format.3 = " . " /* end of format definition */
07 /* top of page format definition                                             */
08 format.top.1 = " ;
09 format.top.2 = " @| | | | | | | | | | | | | | | | | PAGE @<<<<";
10 format.top.3 = " title rpt.page# ";
11 format.top.4 = " . ";
12 format.top.5 = " VOLUME FREE TRK";
13 format.top.6 = " ------ --------";
14 format.top.7 = " . " /* end of format definition */
```
Sample report 1

The following report is produced from the sample report program. The report programs use the data in the DASD MANAGER PLUS historical database, for example, from ATSV nn.RS_VOLUMES. However, you can also specify input from the DB2 catalog or your own databases.

**Figure 207: Volume free space report**

<table>
<thead>
<tr>
<th>VOLUME</th>
<th>FREE TRK</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV115</td>
<td>31</td>
</tr>
<tr>
<td>DEV100</td>
<td>106</td>
</tr>
<tr>
<td>DEV591</td>
<td>165</td>
</tr>
<tr>
<td>DEV055</td>
<td>436</td>
</tr>
<tr>
<td>DEV020</td>
<td>782</td>
</tr>
<tr>
<td>DEV592</td>
<td>794</td>
</tr>
<tr>
<td>DEV070</td>
<td>919</td>
</tr>
<tr>
<td>DEV075</td>
<td>956</td>
</tr>
<tr>
<td>DEV060</td>
<td>1059</td>
</tr>
<tr>
<td>DEV095</td>
<td>1118</td>
</tr>
<tr>
<td>DEV320</td>
<td>1223</td>
</tr>
<tr>
<td>DEV035</td>
<td>1244</td>
</tr>
</tbody>
</table>

The input that created the report requires the following input parameters:

**ssid plan volume**

The example uses the following values for the input parameters: DBDA ASU620DC DEV%. See also “Report-dependent variables” on page 678.

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

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.

Figure 208 on page 675 contains the layout for the sample report in “Sample report 1” on page 674.

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

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 and the WRITE instruction uses them.

![Figure 208: Sample report layout](image)

02 /* detail-line format definition */
03 format. = :
04 format.1 = " @<<<<<<< @>>>>>>>";
05 format.2 = " vl.volid vl.freetrk";
06 format.3 = " . " ; /* end of format definition */
07 /* top of page format definition */
08 format.top.1 = " ";
Table 104: 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.number</td>
<td>format.1</td>
</tr>
<tr>
<td>Top of page</td>
<td>stem.top.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.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Lines</td>
<td>A field line can contain any of the following items:</td>
</tr>
<tr>
<td></td>
<td>■ Fixed text (for example, lines 12 and 13)</td>
</tr>
<tr>
<td></td>
<td>■ Field holders for variable text (line 4)</td>
</tr>
<tr>
<td></td>
<td>■ A combination of fixed text and field holders (line 9)</td>
</tr>
<tr>
<td>Field Holder</td>
<td>A <em>field holder</em> 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:</td>
</tr>
<tr>
<td></td>
<td>04 format.1 = &quot; @&lt;&lt;&lt;&lt;&lt;&lt;&lt;@@@@&gt;&gt;&gt;&quot;; (field line)</td>
</tr>
<tr>
<td></td>
<td>05 format.2 = &quot; vl.volid vl.freetrk&quot;; (value line)</td>
</tr>
<tr>
<td></td>
<td>In a <em>field holder</em>, the characters that follow the @ indicate field justification. The number of characters (including the @) indicate the field length. For example, @&gt;&gt;&gt; indicates a four-character, right-justified field. Left angle brackets (&lt;) indicate a left-justified field, and vertical bars after an @ symbol (,@</td>
</tr>
<tr>
<td>Value Line</td>
<td>The value line specifies the variables that plug into the field holders—one variable for each field holder.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>
Variables

This topic describes the use of variables in report programs.

You can edit or add variables. The number of variables will differ in each report program but the use of variables will always be the same. Table 105 on page 677 shows the use of variables in report programs. (Numbers in parentheses represent line numbers in the sample program on “Sample report program” on page 673.)

Table 105: How to use variables

<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>$dmtable$</td>
<td>Setting input table name (16), SQL (26, 29)</td>
</tr>
<tr>
<td></td>
<td>$tablepart$</td>
<td></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>
<tr>
<td>line number</td>
<td>$rpt.line#$</td>
<td>Line number (not shown)</td>
</tr>
<tr>
<td>report-dependent parameters</td>
<td>$volume$</td>
<td>Title’s input identifier (20), SQL(27)</td>
</tr>
<tr>
<td>format identifier</td>
<td>$format$</td>
<td>Format definitions (3-6, 8-14), Report initialization (18), write (38)</td>
</tr>
<tr>
<td>value line fields</td>
<td>$vl.volid$</td>
<td>Value line (5)</td>
</tr>
<tr>
<td>REXX EXEC PARMS</td>
<td>$ssid$, $plan$ and $dbname$, $tbname$, $volume$, $workid$</td>
<td>Parse arg (15), CONNECT (17)</td>
</tr>
</tbody>
</table>

Input table-name

The asualias REXX EXEC returns the corresponding alias 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 alias and qualifier on the plan.

```
tablepart = asualias('BMCASU_STABLEPART')
indexpart = asualias('BMCASU_SINDEXPART')
```
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.

```
* USAGE NOTES
* Parms
* ssid plan action service
```

The PARMS input line automatically supplies the current SSID and report plan. 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. Figure 209 on page 678 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.

```
Figure 209: Wildcards in report-dependent variables
```

```
// PARM='ASUWKID DBDA ASU620DC V% %COPY'
```

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 ." ,
```

{678} DASD MANAGER PLUS for DB2 User Guide
REXX EXEC PARM

The REXX EXEC PARM in the JCL passes information to the report programs.

“Report-dependent variables” on page 678 provides an example of input parameters. “Sample report 1” on page 674 describes the PARM values.

SQL

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.

Figure 210: 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 210 on page 679, 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.

For an example of the variables, see “SQL as argument” on page 679. In unmodified report programs, at a minimum, the variables represent the input table name, `dmtable`, and EXEC PARM input data, such as `volume`.

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 shown 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.

```
" ' " variable " ' "
```

The first set of double quotation marks can enclose code other than the single quotation mark, as shown below:

```
" "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"" 
```

JCL

The JCL 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.
The EXEC PARM

The REXX EXEC PARM passes information in the following table to the program.

### Table 106: 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 “JCL” on page 680, you must supply all parameters shown in Table 106 on page 681. 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

The following table describes the Data Definition (DD) statements in the JCL that execute the customizable report programs.

### Table 107: DD Statements for Report Programs

<table>
<thead>
<tr>
<th>ddname</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSEXEC</td>
<td>Partitioned data set (PDS) library that contains the report program; the EXEC PARM references the member name</td>
</tr>
<tr>
<td>SYSTSPRT</td>
<td>Data set that contains SAY output of return codes and error messages</td>
</tr>
<tr>
<td>SYSTSIN</td>
<td>DD dummy</td>
</tr>
</tbody>
</table>
**ASURXSQL external function**

The following table describes the SQL statements and commands that the ASURXSQL external function provides.

In each example in Table 108 on page 682, the argument in parentheses is an SQL statement or command, such as CONNECT or DISCONNECT. When ASURXSQL 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 ASURXSQL function, you can perform any SQL statement.

**Table 108: ASURXSQL external function**

<table>
<thead>
<tr>
<th>SQL statement</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT</td>
<td><code>rc = asurxsql(&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 ASURXSQL plan DBRM is bound.</td>
</tr>
<tr>
<td>DISCONNECT</td>
<td><code>rc = asurxsql(&quot;DISCONNECT&quot;)</code></td>
<td>Commit, free all storage that ASURXSQL obtains, close the plan, disconnect from the DB2 subsystem, delete ASURXSQL from memory.</td>
</tr>
<tr>
<td>OPEN CURSOR</td>
<td><code>rc = asurxsql(&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>
| FETCH CURSOR        | `rc = asurxsql("FETCH cursor_name")`                                    | 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 @, #, $, €, , !, ?, _.
|                     | `rc = asurxsql("FETCH cursor_name INTO v1, v2, . . . vn")`                |                                                                                                       |
| CLOSE CURSOR        | `rc = asurxsql("CLOSE cursor_name")`                                    | Close the specified cursor.                                                                           |
EXECUTE
rc = asurxsql(sql_statement)
Prepare and run the SQL statement.
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.

COMMIT
rc = asurxsql("COMMIT")
Commit uncommitted work, and close all open
cursors.

ROLLBACK
rc = asurxsql("ROLLBACK")
Back out relational database changes made since the
last commit.

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 is shown in Table 109 on page 683.

Table 109: ASURXSQL return codes

<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>
<tr>
<td>37</td>
<td>SQL failure; the variable SQLCODE will be set and the variable SQLERRM 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>

Printing error messages

The following code formats and prints SQL error messages to the SYSTEM data set:
if sqlcode ^= 0 then do i = 1 to 1000 by 80
   m = substr(sqlerrm,i+1,79)
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 SQLCODE=0 so that the row values will be available.

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
Table 110 on page 685 describes ASURXRPT functionality.

Table 110: 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 init, you can optionally specify the number of lines per page. The default is 66. Suppress 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 111 on page 685 shows.

Table 111: 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 RPTERRM</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 RPTERRM contains a printable diagnostic message.
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 112 on page 686 shows.

### Table 112: ASURXLSP 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>

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 `SERESULT` as blank-separated values. Table 113 on page 687 describes ASURXSE functionality.
Table 113: ASURXSE external function

<table>
<thead>
<tr>
<th>Function</th>
<th>Sample input</th>
<th>Sample output</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>( rc = \text{ASURXSE}(&quot;TP&quot; \ db2ver \ card \ pqty \ sqty \ partitions \ pctfree \ freepage \ pgsz \ dssz \ rowavg \ maxrows \ compress \ devtype \ allocunit \ type \ segsz); ) where:</td>
<td>SEREDULT = ( \text{TP rpp pages space trks cyls extents dsns, where:} )</td>
</tr>
<tr>
<td></td>
<td>( TP = ) the literal ‘TP’</td>
<td>( TP = ) the literal ‘TP’</td>
</tr>
<tr>
<td></td>
<td>( db2ver = ) DB2 version</td>
<td>( rpp = ) rows per page</td>
</tr>
<tr>
<td></td>
<td>( card = ) number of rows in partition</td>
<td>( pages = ) number of pages in partition</td>
</tr>
<tr>
<td></td>
<td>( pqty = ) primary quantity in allocation units</td>
<td>( space = ) space in KB</td>
</tr>
<tr>
<td></td>
<td>( sqty = ) secondary quantity in allocation units</td>
<td>( trks = ) space in tracks</td>
</tr>
<tr>
<td></td>
<td>( partitions = ) number of partitions</td>
<td>( cyls = ) space in cylinders</td>
</tr>
<tr>
<td></td>
<td>( pctfree = ) percentage of free spacefree page = free pages</td>
<td>( extents = ) number of extents</td>
</tr>
<tr>
<td></td>
<td>( pgsz = ) size of page (4, 8, 16, or 32)</td>
<td>( dsns = ) number of data sets</td>
</tr>
<tr>
<td></td>
<td>( dssz = ) maximum size of a data set in kilobytes</td>
<td>Note: If you omit the type and segsize parameters, the product assumes the type is a simple table and the segment size is 0. Specifying 1 for the type parameter is the equivalent of specifying a blank in the online interface. You must specify a non-zero value for the segsize parameter if you set the type parameter to R. If the type parameter is R and you specify a 0 value for segsize or omit the segsize parameter, DASD MANAGER sets the default segment size to 4.</td>
</tr>
<tr>
<td></td>
<td>( rowavg = ) average row length</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( maxrows = ) maximum rows per page for table space</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( compress = ) data compression (Y or N)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( devtype = ) device type (for example, 3390)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( allocunit = ) allocation unit (C or T)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( type = ) type of table space (L, I, K, R, or 1 where 1 indicates blank or not specified)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( segsz = ) number of pages in the segment</td>
<td></td>
</tr>
</tbody>
</table>

Note: If you omit the type and segsize parameters, the product assumes the type is a simple table and the segment size is 0. Specifying 1 for the type parameter is the equivalent of specifying a blank in the online interface. You must specify a non-zero value for the segsize parameter if you set the type parameter to R. If the type parameter is R and you specify a 0 value for segsize or omit the segsize parameter, DASD MANAGER sets the default segment size to 4.
<table>
<thead>
<tr>
<th>Function</th>
<th>Sample input</th>
<th>Sample output</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TB</strong></td>
<td>( rc = \text{ASURXSE(&quot;TB&quot; ( db2ver ) ( card ) ( pctfree ) ( freepage ) ( pgsiz) ( rowavg ) ( compress ) ( segsiz) ( maxrows)); where} )</td>
<td><strong>SERESULT = TB ( rpp ) ( pages); where:</strong></td>
</tr>
<tr>
<td><strong>Definition:</strong> Non-partitioned table</td>
<td>( TB = \text{the literal 'TB'} ) ( db2ver = DB2 \text{ version} ) ( card = \text{number of rows in table} ) ( pctfree = \text{percentage of free space} ) ( freepage = \text{free pages} ) ( pgsiz) ( = \text{size of page} (4, 8, 16, \text{or} 32) ) ( rowavg = \text{average row length} ) ( compress = \text{data compression} (Y \text{ or} N) ) ( segsiz) ( = \text{segment size} ) ( maxrows = \text{maximum rows per page} ) ( \text{for table space} )</td>
<td>( TB = \text{the literal 'TB'} ) ( rpp = \text{number of rows per page} ) ( pages = \text{number of pages in the table} )</td>
</tr>
<tr>
<td><strong>Description:</strong> Estimates the reorganized space for a nonpartitioned table</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TS</strong></td>
<td>( rc = \text{ASURXSE(&quot;TS&quot; ( db2ver ) ( pages ) ( pqty ) ( sqty ) ( freepage ) ( pgsiz) ( compress ) ( segsiz) ( devtyp) ( alocunit ) ( type)); where} )</td>
<td><strong>SERESULT = TS ( pages ) ( space ) ( trks ) ( cyls ) ( extents ) ( dsns); where:</strong></td>
</tr>
<tr>
<td><strong>Definition:</strong> Non-partitioned table space</td>
<td>( TS = \text{the literal 'TS'} ) ( db2ver = DB2 \text{ version} ) ( pages = \text{total pages for all tables} ) ( pqty = \text{primary quantity in allocation units} ) ( sqty = \text{secondary quantity in allocation units} ) ( freepage = \text{free pages} ) ( pgsiz) ( = \text{size of page} (4, 8, 16, \text{or} 32) ) ( compress = \text{data compression} (Y \text{ or} N) ) ( segsiz) ( = \text{segment size} ) ( devtyp) ( = \text{device type} (\text{for example,} 3390) ) ( alocunit = \text{allocation unit} (C \text{ or} T) ) ( type = \text{type of table space} (L, I, K, \text{or} \text{blank}) )</td>
<td>( TS = \text{the literal 'TS'} ) ( pages = \text{total pages for all tables} ) ( space = \text{space in KB} ) ( trks = \text{space in tracks} ) ( cyls = \text{space in cylinders} ) ( extents = \text{number of extents} ) ( dsns = \text{number of data sets} )</td>
</tr>
<tr>
<td><strong>Description:</strong> Estimate the reorganized space for a nonpartitioned table space</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Sample input</td>
<td>Sample output</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>TS</td>
<td>( rc = \text{ASURXSE} (&quot;TS&quot; \ db2ver \ pages \ pqty \ sqty \ freepage \ pgsizy \ compress \ segsize \ devtype \ alocunit \ G \ rpp \ dsize \ card) ); where ( TS ) = the literal ‘TS’ ( db2ver ) = DB2 version ( pages ) = total pages for all tables ( pqty ) = primary quantity in allocation units ( sqty ) = secondary quantity in allocation units ( freepage ) = free pages ( pgsizy ) = size of page (4, 8, 16, or 32) ( compress ) = data compression (Y or N) ( segsize ) = segment size ( devtype ) = device type (for example, 3390) ( alocunit ) = allocation unit (C or T) ( type ) = type of table space (G) ( rpp ) = rows per page ( dsize ) = maximum size of a data set in kilobytes ( card ) = number of rows in table</td>
<td>( SERESULT = TS \ parts \ pages \ space \ trks \ cylys \ extents \ dsns \ pages2 \ space2 \ trks2 \ cylys2 \ extents2 \ dsns2 \ card \ card2 ); where: ( TS ) = the literal ‘TS’ ( parts ) = total number of partitions ( pages ) = total pages per partition ( space ) = KBs per partition ( trks ) = tracks per partition ( cylys ) = cylinders per partition ( extents ) = extents per partition ( dsns ) = data sets per partition ( pages2 ) = pages in the last partition ( space2 ) = KBs in the last partition ( trks2 ) = tracks in the last partition ( cylys2 ) = cylinders in the last partition ( extents2 ) = extents in the last partition ( dsns2 ) = data sets in the last partition ( card ) = rows per partition ( card2 ) = rows in the last partition</td>
</tr>
<tr>
<td>Function</td>
<td>Sample input</td>
<td>Sample output</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
</tbody>
</table>
| IP index | \[ rc = \text{ASURXSE}("IP" \ db2ver\ card\ rpk\ pqty\ sqty\ parts\ avgkeylength\ nlavgkeylen\ pctfree\ freepage\ piecesize\ uniquerule\ indextype\ devtype\ allocunit\ typets\ pgsizer) \); where \]
| **Definition:** Estimates the reorganized space for an index | \[ \text{SERESULT} = \text{IP pages space trks cyls nlevels extents dsns}; \]
| | \[ IP = \text{the literal 'IP'} \]
| | \[ db2ver = \text{DB2 version} \]
| | \[ card = \text{number of rows in table} \]
| | \[ rpk = \text{rows per key} \]
| | \[ pqty = \text{primary quantity in allocation units} \]
| | \[ sqty = \text{secondary quantity in allocation units} \]
| | \[ parts = \text{number of partitions} \]
| | \[ avgkeylength = \text{average length of key} \]
| | \[ nlavgkeylen = \text{average length for nonleaf page key} \]
| | \[ subpages = \text{number of subpages} \]
| | \[ pctfree = \text{percentage of free space} \]
| | \[ freepage = \text{free pages} \]
| | \[ piecesize = \text{file size for ix} \]
| | \[ uniquerule = \text{unique rule} \]
| | \[ indextype = \text{index type (2)} \]
| | \[ devtype = \text{device type (for example, 3390)} \]
| | \[ allocunit = \text{allocation unit (C or T)} \]
| | \[ typets = \text{type of table space (G, R, L, 1 (1 indicates blank or not specified))} \]
| | \[ pgsizer = \text{size of eaf pages in the index (4, 8, 16, or 32)} \]
| | **Note:** If you omit the typets and pgsizer parameters, the product assumes that the type is a nonlarge table space and the page size is 4. Specifying 1 for the typets parameter is the equivalent of specifying a blank in the online interface. | |
Function | Sample input | Sample output
--- | --- | ---
a | If you omit the type and segsize parameters, the product assumes the type is a simple table and the segment size is 0. Specifying 1 for the type parameter is the equivalent of specifying a blank in the online interface. | 
b | You must specify a non-zero value for the segsize parameter if you set the type parameter to R. If the type parameter is R and you specify a 0 value for segsize or omit the segsize parameter, DASD MANAGER sets the default segment size to 4. | 
c | If you omit the typets and pgsze parameters, the product assumes the type is a nonlarge table space and the page size is 4. Specifying 1 for the typets parameter is the equivalent of specifying a blank in the online interface. | 
d | Excluding the last partition. | 

ASURXSE produces the return codes that Table 114 on page 691 shows.

**Table 114: ASURXSE 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>51</td>
<td>Error obtaining a REXX variable</td>
</tr>
<tr>
<td>54</td>
<td>Argument error (too many, too few, invalid value)</td>
</tr>
<tr>
<td>58</td>
<td>Error loading module IRXEXCOM</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 212 on page 692 shows a sample REXX program that uses the ASURXLOC external function.

**Figure 212: Sample REXX program using ASURXLOC**

/* 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:

devtype = word(VolList.i,1)
volserno = word(VolList.i,2)
volseqno = word(VolList.i,3)

ASURXLOC produces the return codes that Table 115 on page 692 shows.

**Table 115: 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>

**Notes on REXX**

This topic discusses comments, REXX delimiters, case, continuation characters, and the concatenation operator.

For more information on SQL within REXX code, see “Quotation marks” on page 680.
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" eqorlike " 'dbnamex' '",
"      AND TIMESTAMP = (SELECT MAX(TIMESTAMP) ",
```

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"||,
```

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:
The complete sample program is at “Sample report 2” on page 707.

Specify the DB2 table and report title

Lines 1 through 22 of the following figure document the program and specify the name of the DB2 table that contains the input data and the report title.

Figure 213: Specifying the DB2 object and report title

```rexx
/* rexx *************************************************************/
/*                                                               */
/*  SYNOPSIS                                                          */
/*      TSSPACE - 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_STABLEPART');
/* title = "DASD SPACE FOR TABLESPACE";                             */
```

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-17</td>
<td>These lines are comments.</td>
</tr>
<tr>
<td>1</td>
<td>This line identifies REXX as the programming language. The word REXX is required in the first line of each REXX program.</td>
</tr>
<tr>
<td>3-4</td>
<td>These lines identify the name of the report and its subject: space utilization for DB2 table spaces.</td>
</tr>
<tr>
<td>6-8</td>
<td>These lines provide a brief description of the report.</td>
</tr>
</tbody>
</table>
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.

Example:

```
//REXX EXEC PGM=IKJEFT01,PARM='TSSPACE ssid plan dbname-wildcard'
```

These lines are comments.

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:

Example:

```
"BMCASU_STABLEPART BMCASU_STABLEPART ",
```

This line assigns the report title, DASD SPACE FOR TABLESPACE, to the variable title.

Table 116 on page 695 describes the information passed by the EXEC PARM.

### Table 116: 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 ASURSQL 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>

### Define report formats

Lines 23 through 38 of the following figure 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 214 on page 696)
- 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 214: Defining report formats**

<table>
<thead>
<tr>
<th>Line number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>This line is comments.</td>
</tr>
<tr>
<td>25-28</td>
<td>These lines define the format for the table partition detail lines.</td>
</tr>
<tr>
<td>25</td>
<td>This line is a null assignment to stem variable <code>tpfd</code>, 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.)</td>
</tr>
<tr>
<td>26</td>
<td>This line uses four <em>field holders</em> to define the length, justification, and spacing of fields in the detail line (<em>Table 117 on page 697</em>). Line 26 assigns the detail line format definition to variable <code>tpfd.1</code>. The line contains spaces that appear in the output.)</td>
</tr>
<tr>
<td>27</td>
<td>This line identifies the four variables to plug into the format definitions in line 26 and assigns them to variable <code>tpfd.2</code>. These detail-line variables correspond to the column headings defined in line 36. <em>Table 118 on page 697</em> describes the variables in the detail line. The last three variables use the following format: <code>cursormame.columnname</code></td>
</tr>
<tr>
<td>28</td>
<td>This line ends the format definition of the detail line by assigning a single period to variable <code>tpfd.3</code>.</td>
</tr>
<tr>
<td>30</td>
<td>This line is a comment.</td>
</tr>
<tr>
<td>32-38</td>
<td>These lines define the format for the top-of-page. This definition includes the report title, page number, and column headings.</td>
</tr>
<tr>
<td>32</td>
<td>This line assigns a space to variable <code>tpfd.top.1</code>. The variable creates a blank line at the top of each page.</td>
</tr>
<tr>
<td>34</td>
<td>This line assigns the report title defined in line 21 to variable <code>tpfd.top.3</code>, and assigns the page number to variable <code>rpt.page#</code>, which is an internal variable set in ASURXRPT.</td>
</tr>
<tr>
<td>35</td>
<td>This line assigns a space to variable <code>tpfd.top.4</code>. The variable creates a blank line under the report title.</td>
</tr>
<tr>
<td>36</td>
<td>This line specifies the column headings and assigns them to <code>tpfd.top.5</code>.</td>
</tr>
</tbody>
</table>
Table 117: 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;</td>
<td>right</td>
<td>9 (1@ + 8 &gt;)</td>
</tr>
<tr>
<td>4</td>
<td>@&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 675.

The cursor TP is declared in the OPEN CURSOR statement in line 67. Table 118 on page 697 describes the variables in the detail line.

Table 118: 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;&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;</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>

**Note**

The word top must be the second node of each top-of-page format name.

line 33 This line specifies dashed lines under the column headings and assigns them to variable *tpfd.top.6*. Line 33 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 119 on page 698). Spaces separate the field definitions that appear in the output.
Table 119: 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></td>
<td></td>
<td></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 684 ASURXRPT external function”.

**Connect to the DB2 subsystem and open the plan**

Lines 41 through 46 of the following figure obtain the DB2 subsystem ID, plan name, and database name, then connect to the subsystem and open the plan.

**Figure 215: Connecting to the DB2 subsystem and opening the plan**

```
41 parse upper arg ssid plan dbnamex ; /* get input data */
42 rc = asurxsql("CONNECT" ssid plan); /* connect and open plan */
44 if (rc ≠ 0) then /* if connect or open plan error */
46   signal caferror; /* go output caf error message */
```

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>This line obtains the SSID, plan name, and database names—information passed to the program by the PARM of the EXEC statement.</td>
</tr>
<tr>
<td></td>
<td>■ After retrieving the information, this instruction converts the information to uppercase before saving it into the variables <code>ssid</code>, <code>plan</code>, and <code>dbnamex</code>.</td>
</tr>
<tr>
<td></td>
<td>■ The period at the end of the instruction discards information (such as the sequence number) that follows the SSID, plan, and database name.</td>
</tr>
<tr>
<td>43</td>
<td>This line issues the SQL CONNECT instruction, using the SSID and plan name obtained in line 41, and assigns the return code to the <code>rc</code> variable.</td>
</tr>
<tr>
<td>45-46</td>
<td>If the CONNECT or open fails—if the return code is not 0—these lines trap the CAF error condition (<code>caferror</code>) and issue a CAF error message (lines 118 through 126 on “The CAF error procedure” on page 703).</td>
</tr>
</tbody>
</table>
**Initialize the report**

Lines 48 through 51 of the following figure 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 216: Initializing the report**

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>This line runs external function ASURXRPT with an <em>init</em> argument string. The return code from the function is assigned to the <em>rc</em> variable. The product sends the arguments displayed in Table 120 on page 699 to ASURXRPT.</td>
</tr>
<tr>
<td>50-51</td>
<td>If the return code is not 0—if initialization fails—these lines trap the <em>rpterror</em> error condition and issue a report error message (lines 111 through 115 in “Fetch and write the table space data” on page 702).</td>
</tr>
</tbody>
</table>

**Table 120: ASURXRPT arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>init</td>
<td>Initializes the report</td>
</tr>
<tr>
<td>60</td>
<td>Defines the page length as 60 lines. The default is 66.</td>
</tr>
<tr>
<td>suppress</td>
<td>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.</td>
</tr>
<tr>
<td>tpfd</td>
<td>Names the stem variable that identifies the format definition. (See lines 25 through 38 in “Define report formats” on page 695.)</td>
</tr>
</tbody>
</table>

**Begin reading input and get database names**

Lines 53 through 58 of the following figure read and process the input data from the EXEC PARM.

The input contains one or more database names, which can contain wildcards.

**Figure 217: Getting the input and testing for wildcards**

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>**change * to % in dbname */</td>
</tr>
<tr>
<td>54</td>
<td>**default to equal */</td>
</tr>
<tr>
<td>55</td>
<td>**if wildcard in dbname */</td>
</tr>
<tr>
<td>56</td>
<td>**use like */</td>
</tr>
<tr>
<td>57</td>
<td>**use like */</td>
</tr>
<tr>
<td>Line Number</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>53-54</td>
<td>These lines substitute the % sign for any asterisks (*) encountered in input database names and set the variable <code>eqorlike</code> to equal (=).</td>
</tr>
<tr>
<td>53</td>
<td>This line changes any * in a database name to a %.</td>
</tr>
<tr>
<td>54</td>
<td>This line assigns an equal sign (=) to the variable <code>eqorlike</code>. The equal sign tells REXX to interpret the database name exactly as it is written.</td>
</tr>
</tbody>
</table>
| 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. |
| 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:  
- 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. |
| 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:  
- 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. |
| 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. |
Open cursor for SELECT

Lines 60 through 79 of the following figure prepare to obtain table partition data by opening a cursor for SELECT FOR FETCH ONLY processing.

Figure 218: Fetching the data

```
/* obtain tablepart data */

rc = asurxsql("OPEN ", /* open cursor */
               "TP CURSOR FOR SELECT ",
               "A.DBNAME ",
               "A.TSNAME ",
               "A.PARTITION ",
               "A.SPACE ",
               "A.EXENTS ",
               "FROM " dmtablepart "A ",
               "WHERE A.DBNAME eqorlike "'dbnamex' " ,
               "AND TIMESTMP = (SELECT MAX(TIMESTMP) ",
               "FROM " dmtablepart "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 */
```

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>This line is a comment.</td>
</tr>
<tr>
<td>62-79</td>
<td>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). <strong>Note:</strong> 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.</td>
</tr>
<tr>
<td>81-82</td>
<td>If the OPEN fails—if the return code is not zero—these lines trap the SQL error condition (sqlerror) (lines 135 through 141).</td>
</tr>
</tbody>
</table>
Fetch and write the table space data

Lines 84 through 102 of the following figure prepare the data, write it, close the cursor, and execute the cleanup procedure.

Figure 219: Preparing and writing the data

```
84 do forever                        /* loop for all tablepart rows */
  rc = asurxsql("FETCH TP ");      /* fetch next row */
86
87 if (rc ¬= 0) then               /* if fetch failed */
  signal sqLError;              /* go output sql error message */
89 if (sqlcode = 100) then         /* if end of data */
  leave;                        /* leave forever loop */
90
92 dbn = strip(tp.dbname, 'T');    /* remove trailing blanks */
93 tsn = strip(tp.tsname, 'T');    /* remove trailing blanks */
94 tsobj = dbn."tsn";              /* dbname.tsname */
95 rc = asurxrpt("WRITE tpfd");   /* output tablepart data */
96 if (rc ¬= 0) then               /* if an error */
  signal rptError;              /* go output report error message */
98 end                             /* end do forever */
99
100 rcx = asurxsql("CLOSE TP");   /* close cursor */
101
102 signal cleanup;                   /* cleanup */
103
```

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>84-98</td>
<td>These lines are a do-forever loop to fetch the table partition rows.</td>
</tr>
<tr>
<td>85</td>
<td>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.</td>
</tr>
<tr>
<td>87-88</td>
<td>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).</td>
</tr>
<tr>
<td>89-90</td>
<td>These lines exit the loop if all rows have been fetched (if the return SQLCODE = 100).</td>
</tr>
<tr>
<td>92-93</td>
<td>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.</td>
</tr>
<tr>
<td>94</td>
<td>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.</td>
</tr>
<tr>
<td>95</td>
<td>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.</td>
</tr>
<tr>
<td>96-97</td>
<td>If the return code is not 0, this line trap the report error condition and output the report error message (lines 111 through 115).</td>
</tr>
<tr>
<td>98</td>
<td>This line ends the do-forever loop begun in line 84.</td>
</tr>
<tr>
<td>100</td>
<td>This line closes the cursor opened in lines 62 and 63.</td>
</tr>
<tr>
<td>102</td>
<td>This line executes the cleanup procedure.</td>
</tr>
</tbody>
</table>
The report error procedure

Lines 105 through 115 of the following figure 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 220: 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                         */
```

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>105-109</td>
<td>These lines are comments.</td>
</tr>
<tr>
<td>111</td>
<td>This line marks the beginning of the report error procedure, which runs during report initialization or while it writes the data.</td>
</tr>
<tr>
<td>113</td>
<td>This line outputs the report return code to the SYSTSPRT DD data set, using the format &quot;ASURXRPT error code = rptcode&quot;. The ASURXRPT error code is anything other than zero (0).</td>
</tr>
<tr>
<td>114</td>
<td>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.</td>
</tr>
<tr>
<td>115</td>
<td>This line directs REXX to go to the cleanup label and begin executing the cleanup procedure.</td>
</tr>
</tbody>
</table>

The CAF error procedure

Lines 118 through 126 of the following figure 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 221: Reporting CAF error code and message

```plaintext
117 /*-------------------------------------------------------------------*/
118 caferror - output caf error msg
119 /*-------------------------------------------------------------------*/
120 caferror:
121 say 'Connect failed, rc = 'rc'. cafreason = 'cafreason;
122 exit 8;
```

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>117</td>
<td>A comment indicating the beginning of the CAF error procedure.</td>
</tr>
<tr>
<td>118</td>
<td>This line marks the beginning of the CAF error procedure, which runs when encountering a CAF processing error.</td>
</tr>
<tr>
<td>120</td>
<td>This line outputs a printable diagnostic message to the SYSTSPRT DD data set by using the variable rc and cafreason.</td>
</tr>
<tr>
<td>121</td>
<td>This line directs REXX to exit with a status of 8, indicating failure.</td>
</tr>
</tbody>
</table>
These lines are comments.

This line marks the beginning of the CAF error procedure, which the product runs if the CONNECT fails (line 46).

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 684.

This line exits the program and sets the return code to 8.

The SQL error procedure

Lines 129 through 142 of the following figure 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 222: The SQL error procedure

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)
- SQL errors (lines 81 and 82)
- FETCH row fails (line 88)

This line returns the SQL error return code in the format ASURXSQL return code = 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.
Lines 144 through 147 of the following figure are the cleanup procedure to terminate the report, disconnect from DB2, and issue a return code.

**Figure 223: Invoking the cleanup procedure**

144 cleanup:
145  rcx = asurxrpt("TERM");           /* terminate report */
146  rcx = asurxsql("DISCONNECT");     /* disconnect from db2 */
147  exit rc;

**JCL and execution**

The JCL in the following figure executes program IKJEFT01 to interpret the REXX program which is used to create reports.
The lines are numbered for easy reference.

**Figure 224: JCL for customized REXX report**

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>This line runs the IKJEFT01 program to run the REXX program and supplies PARM information. In the annotated program, the report-dependent parameter was a database name that contained a wildcard.d.</td>
</tr>
<tr>
<td>3-8</td>
<td>These lines are comments.</td>
</tr>
<tr>
<td>4-6</td>
<td>These lines identify the required step libraries.</td>
</tr>
<tr>
<td>11</td>
<td>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.</td>
</tr>
<tr>
<td>12-13</td>
<td>These lines identify the data sets that contain the report and the error messages that the ASURXRPT and ASURXSQL functions generate.</td>
</tr>
<tr>
<td>12</td>
<td>This line identifies the data set that contains the report.</td>
</tr>
<tr>
<td>13</td>
<td>This line identifies the data set that contains error messages from the functions ASURXRPT and ASURXSQL and the SAS/C runtime library.</td>
</tr>
<tr>
<td>14</td>
<td>This line identifies the data set that contains <code>SAY</code> output of return codes and error messages (lines 113, 114, 125, 136, and 141).</td>
</tr>
</tbody>
</table>
Sample report 2

The annotated report program is displayed in the following figure.

**Figure 225: A sample customizable report**

<table>
<thead>
<tr>
<th>TABLESPACE</th>
<th>PART</th>
<th>SPACE</th>
<th>EXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUD20.QZUS0120</td>
<td>0</td>
<td>4472224</td>
<td>40</td>
</tr>
<tr>
<td>QZUD21.QZUS0121</td>
<td>1</td>
<td>303840</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>292424</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>142440</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>244800</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>171572</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>181172</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>243360</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>142440</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>298080</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>853388</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>400320</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>175298</td>
<td>1</td>
</tr>
<tr>
<td>QZUD23.QZUS0123</td>
<td>1</td>
<td>325745</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>87375</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>309756</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>19125</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>317745</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>56280</td>
<td>9</td>
</tr>
<tr>
<td>QZUD25.QZUS0125</td>
<td>1</td>
<td>120</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>60</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>465</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>120</td>
<td>1</td>
</tr>
</tbody>
</table>

Complete sample program

The complete report program annotated in this appendix is in the following figure.

**Figure 226: Complete sample program**

```rexx
/* rexx **************************************************************************/
/* * SYNSOPSIS * TSSPACE - 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 */
dmttablepart = asualias('BMCASU_STABLEPART');
title = "DASD SPACE FOR TABLESPACE";
/* */
/* tablepart format definition */```
tpfd. =
tpfd.1 = "@<<<<<@s@>>>@s@>>>":
tpfd.2 = "tsobj tp.partition tp.space tp.extents":
tpfd.3 = ": /* end of format definition */
/* tablepart top of page format definition */
tpfd.top.1 = :
tpfd.top.2 = "@|||PAGE@<<<":
tpfd.top.3 = "title rpt.page#":
tpfd.top.4 = :
tpfd.top.5 = "TABLESPACE PART SPACE EXT":
tpfd.top.6 = ": /* 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)) 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 ",
  "WHERE A.DBNAME" eqorlike ":"dbnamex"":
    AND TIMESTMP = (SELECT MAX(TIMESTMP)"
    FROM " dmtablepart "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 */
tsn = strip(tp.tsnname, 'T');
tsobj = dbn"."tsn; /* dbname.tsnname */
rc = asurxrpt("WRITE tpfd"); /* output tablepart data */
if (rc = 0) then /* if an error */
signal rpterror; /* go output report error message */
end /* end do forever */
rcx = asurxsql("CLOSE TP"); /* close cursor */
signal cleanup; /* cleanup */

rpterror - output error code and message from asurxrpt

rpterror:
say "ASURXRPT return code = "rptcode; /* error code */
say rpterrm; /* output report error message */
signal cleanup; /* cleanup */

caferror - output caf error msg

caferror:
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
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 describes an example of the use of Utility automation.

Overview of utility automation

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.

In addition, BMCSTATS calculates REORGSIZE, 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 REORGSIZE 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 the utility-automation 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.

You can set up the automation by modifying the corrective actions and optionally modifying 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
- The name of your ISPF library
- The DB2 subsystem ID or group attach name
- The 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
- A creator ID for the installed DASD MANAGER PLUS tables
- The default options module name installed for DASD MANAGER PLUS

Each sample contains instructions about modifying the file.
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.

You can use the DASD MANAGER PLUS online interface to view and modify the installed thresholds and corrective actions. For more information about modifying thresholds and corrective actions, see "Analyzing objects by using BMCTRIG."
Modifying the default corrective actions

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.

To modify the default corrective actions

1  *(optional)* Using the DASD MANAGER PLUS online interface, update the default actions named in "Adding corrective action definitions" to select utility options for your site.

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.

---

**Example**

You could add a QUIESCE or a BMCCOPY before the BMCREORG in the BMC_BMCREORG_DEFAULT.

---

For more information about maintaining actions, see “Maintaining and generating Service Actions” on page 223.

Running the BMCSTATS utility

Use this procedure to run the BMCSTATS utility to populate the DASD MANAGER PLUS database with statistics. This procedure uses the ASUTSYST sample JCL. The procedure collects current statistics for evaluation by BMCTRIG.

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 “Using object sets” on page 279.
For more information about setting up BMCTRIG jobs, see the DASD MANAGER PLUS for DB2 Reference Manual.

### Execute or schedule 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 “Analyzing objects by using BMCTRIG” on page 425.
Using the UIM server

This appendix covers defining the UIM server for use with the Export utility.

Overview of the UIM server

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.

If you need to start and stop the UIM server, you must issue IBM MVSTM 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**

1. Issue the following MVS operator command:

   ```
   S uimServerName
   ```

   *uimServerName* is the name of the UIM server started task.
To stop the UIM server

1 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.

You should verify that the UIM server is running, and then 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 121: 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 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 719.

To access the BMC UIM server Commands web page

1. 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:
**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 718.

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 718.

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 718.

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.

- 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 718.
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 718.

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 722.

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:
Resolving UIM Server problems

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.

You can solve this problem by changing 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

1 Perform one of the following tasks:

UIM server does not start and displays message LSCX902 in SYSTERM DD
- 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 SYSTERM 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.

You can solve this problem by inserting 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:

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSCX904</td>
<td>**** WARNING **** ERRNO = EACCES</td>
</tr>
<tr>
<td>Generated in BIND called from line</td>
<td>Unexpected failure in bind, reason code 744C7246</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Port Number</th>
<th>Protocol</th>
<th>Job Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8300</td>
<td>TCP</td>
<td>STFTUIM</td>
<td>BMC UIM server</td>
</tr>
</tbody>
</table>

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.
Enhanced database performance evaluation using SQL Performance

This chapter describes how you can obtain more robust database performance evaluations by combining native DASD MANAGER PLUS functionality with a feature in the SQL Performance for DB2 solution.

Overview of using Performance Advisor with DASD MANAGER PLUS

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.

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, BMCTTRIG 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 735.

To generate a reorganization job based on the recommended objects list from the REORG candidate table

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. Change and execute the corrective actions.
4. Change the sample BMCTTRIG JCL.
5. Schedule or manually run the BMCTTRIG TRIGFORC job.

TRIGFORC runs BMCTTRIG against the object sets and automatically generates reorganization job steps for each object that was identified by the object set. BMCTTRIG 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 BMCTTRIG evaluations

In this scenario, BMCTTRIG generates a reorganization job only if the objects meet both criteria (recommended by the REORG candidate table and triggered by BMCTTRIG-specified exception thresholds).

The following steps represent a high-level summary of the process. For detailed instructions, see “Setting up automated reorganizations” on page 735.

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 for DB2 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 SQL Performance for DB2 and you want to use the feature, complete the procedures provided in the following sections:

- “Preparing to use REORG Advisor” on page 732
- “Setting up automated reorganizations” on page 735

### 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)
- Obtain sample JCL files

### 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 ASUQDYDYN 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(ASUQDYDYN) +
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) +
CACHESIZE(1024) CURRENTDATA(YES) +
SQLRULES(DB2) ACQUIRE(USE) +
EXPLAIN(NO) DYNAMICRULES(BIND) +
KEEPDYNAMIC(NO) IMMEDWRITE(NO) +
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) +
CACHESIZE(1024) CURRENTDATA(YES) +
SQLRULES(DB2) ACQUIRE(USE) +
EXPLAIN(NO) DYNAMICRULES(BIND) +
KEEPDYNAMIC(NO) IMMEDWRITE(NO) +
ENCODING(37) DISCONNECT(EXPLICIT)+
```
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.

Before you begin

Collect the following information:

- DASD MANAGER PLUS installed library names
- ISPF library name
- DB2 subsystem ID or group attach name of the system where the DASD MANAGER PLUS tables are installed
- DB2 subsystem ID (SSID) or group attach name of the system where the Performance Advisor REORG candidate table is installed
- Creator ID and name of the Performance Advisor REORG candidate table
- 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
- Creator ID for the installed DASD MANAGER tables
- Default options module name installed for DASD MANAGER PLUS

To obtain the sample files

1 Obtain the sample files from the LLQSAMP library (where LLQ is DB, XX, BB, and UBB) of the SQL Performance for DB2 files. The following listing shows the portion of the library that contains the sample JCL:

```sql
PKLIST(BMU430_D_MAIN.*,*
   *ASU930_D_MAIN.*,*
   ACS930_D_MAIN.*,+
   ATS011_D_MAIN:*+)
)
```
Setting up automated reorganizations

After enabling REORG Advisor, complete one of the procedures here for automating your reorganizations:

- “To automate reorganizations based only on REORG candidate table recommendations” on page 735.
- “To automate reorganizations based on REORG candidate table recommendations and BMCTRIG evaluations” on page 737

**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 732.

**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 INSOBJS 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 uses 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 “Using object sets” on page 279.

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
The sample files use the pound sign (#) to identify lines that require changes. 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 “Maintaining and generating Service Actions” on page 223.

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 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 are 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.

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 “Analyzing objects by using BMCTRIG” on page 425.

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 the INSOBJS member as detailed in the instructions within the sample file. Following is a summary of the steps to edit this file.
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 uses to query the REORG candidate table by populating the DASD MANAGER PLUS control tables.

a. Edit the INSOBJS 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.

For more information about maintaining object sets, see “Using object sets” on page 279.

3. Change and execute the INSTHRSH member to insert thresholds that BMCTRIG uses 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 is 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 "Analyzing objects by using BMCTRIG."
Threshold name | Value
---|---
APPNDINS | 10
DSEXTENT | 50
EXTENTS | 50
FARIND | 15
LEAFFOFF | 10
LEAFTOFF | 30
PSEUDODL | 10
REORMDEL | 1
REORMODS | 10
REORPEND | 1
REORSPAC | 30
SPACE | 30
TOTALIND | 30
UNCLUST | 10

4 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.

5 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.
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 on page 738.

a. Change the job card to meet your installation standards.

b. Change the ASUSSID in SYSTSIN to the subsystem ID where your DASD MANAGER tables are located and where this job is executed.

c. Change the creator name used with the DASD MANAGER PLUS tables to match your site’s installation.

d. Submit the job.

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.

TRIGEXCP contains a sample job that
- Runs BMCSTATS against the object set
- Runs BMCTRIG to further evaluate the objects that are using the exceptions
- 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. (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.

b. Change the member to include the DB2 and DASD MANAGER PLUS data set names that are installed at your site.

c. Change the job card to meet your installation standards.

d. Change the parameters for ASUSMAIN and ASUSTRIG to specify your DB2 subsystem and the DASD MANAGER PLUS installation options module.

e. Change the BMCTRIG command syntax to identify the data sets that are used to generate the BMC REORG PLUS jobs.
f 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.

9 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 “Analyzing objects by using BMCTRIG” on page 425.
# #DOMZAP

Member of the CNTL data set that contains sample JCL that invokes the IBM AMASPZAP utility to apply a SUPERZAP.

9DEFAULT record

A record shipped with the System and SQL Performance products that contains a default User Profile. It is composed of a 9DEFAULT security record and a 9DEFAULT user record that together contain the default values for all parameters in the User Profile.

A

abend

See abnormal end of task.

abend reason code

A 4-byte hexadecimal code that uniquely identifies a problem with DB2. A complete list of DB2 abend reason codes and their explanations is contained in DB2 for z/OS Codes and DB2 for z/OS Messages.

abnormal end of task

Termination of a task, a job, or a subsystem because of an error condition that cannot be resolved during execution by recovery facilities.

ABOUT command
A basic panel command that displays copyright and trademark information about the product.

access method

A technique for moving data between main storage and input/output devices.

ACK

The product code that BMC uses to identify the CHECK PLUS for DB2 product.

ACKESQL

An SQL batch execution program provided with the CHECK PLUS product that you can use to execute the DELETE statements in the discard data set that CHECK PLUS creates.

ACT

The code used to identify the CATALOGMANAGER 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 (rerunnable) 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 an 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.

**active log**

The portion of the DB2 log to which log records are written as they are generated. The active log always contains the most recent log records, whereas the archive log holds those records that are older and no longer will fit on the active log. For recovery purposes, this is the log in use when a failure occurs.

**active log**

The DB2 log data set that is used to record incoming updates to tables and other events. For recovery purposes, this is the log data set in use when a failure occurs. See also log and archive log.

**ADDD2B command**

A Data Collector command that temporarily adds a DB2 subsystem to the list of DB2s that can be monitored by a Data Collector.

**ADDP2D command**

A Data Collector command that dynamically activates a System and SQL Performance product that was previously deactivated using the DELP2D command.
address space

A range of virtual storage pages identified by a number (ASID) and a collection of segment and page tables which map the virtual pages to real pages of the computer's memory.

ADMIN command

A basic panel command that displays the Administration menu.

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.

ADU

The product code that BMC uses to identify the UNLOAD PLUS for DB2 product.

advisor

A collection of information (mainly accessed from reports) that displays key values, pinpoints problems detected by Pool Advisor or System Performance, and recommends actions you can take to correct those problems. See also background advisors and system advisor.

AES

The BMC Software Application Enhancement Series, which includes the APPLICATION RESTART Control products for IMS, DB2, and VSAM and the BATCH Control FACILITY products.

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.
agent

A submitted job or started task that communicates information about the DB2 subsystems on a particular system for SHRLEVEL CHANGE copy jobs in a data sharing environment. There must one agent per system with an active DB2 data sharing member. The agent is used to determine the registration point for copies.

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 ALTERproduct in BMC Software database naming conventions.

AMU

The product code that BMC uses to identify the LOADPLUS for DB2 product.

anomaly

An update transaction that was not in error and needs to be preserved even though it affects the same rows as a problem transaction that was in error. The Log Master for DB2 product provides backout integrity checking features to detect anomalies and enable customers to take appropriate action.

AP command

A basic panel command that navigates from any System and SQL Performance panels or reports to APPTUNE Workload Analysis or the DB2 Status report in APPTUNE.

APF

Authorized program facility. An IBM facility that allows you to establish authorization for programs to use certain functions that are otherwise restricted.

API

See application programming interface (API).

APPFILT command

An APPTUNE Data Collector command used to dynamically change the APPTUNE filter for a DB2 without stopping the Data Collector.

application

A program or set of programs that perform a task (for example, a payroll application).
application group

A logical collections of plans, programs, and users that represent the workload associated with a particular business function or individual.

application plan

In DB2, the control structure produced during the bind process and used to process SQL statements encountered during statement execution.

application profile

A collection of one or more application groups.

application program interface

A functional interface supplied by the operating system or by a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or licensed program.

apply agent

A high-speed Apply Engine task that updates a target table.

apply processing

The process of updating the target tables by using the information provided in the configuration file and apply request.

apply request

The batch job or command that is used to start and run high-speed Apply Engine. The apply request can contain job processing information and overriding values for some of the configuration parameters.

APPOFF command

A Data Collector command used to stop APPTUNE data collection and free data reduction storage.
APPON command

A Data Collector command used to start data collection for APPTUNE after the Data Collector is started (when APPTUNE data collection is not started automatically) or after data collection has been stopped by an APPOFF command.

APPREAD statement/parameter

A control statement (or parameter of the REPORT statement) used to specify the application group that will be used for group reporting when one of the APPTUNE group reports is specified or the APGRP qualifier is used.

APPRESET command

An APPTUNE Data Collector command used to write the data in the reduction table to the trace data set.

APPSTAT command

An APPTUNE Data Collector command used to display the current status of the APPTUNE intercept.

APPTUNE

A BMC product used to analyze DB2 application performance (a component of SQL Performance for DB2).

archive directory

A data set that contains a directory of archived data sets, used to select the data sets to use for batch reporting.

archive log

The portion of the DB2 log that contains log records that have been copied from the active log. The archive log is retrievable in the event that it is needed for recovery purposes.

archive log

A DB2 active log data set that is no longer in use and has been archived. The log data set may be retrievable for recovery purposes. See also log and active log.
ARU

The product code that BMC uses to identify the REORG PLUS for DB2 product.

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.

audit

A search for log records that indicate data that has changed within DB2 tables. In Log Master, you can narrow the search for changed data to, for example, specific users, tables, or plans.

authorization ID

A string that can be verified for connection to DB2 and to which a set of privileges are allowed. It can represent an individual, an organizational group, or a function, but DB2 does not determine this representation.

authorized program facility

A facility that permits identification of programs authorized to use restricted functions.

auto-refresh

A feature that causes the values displayed for DB2 instrumentation data to be updated automatically at an interval specified by the user.

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.

auxiliary table

A table that stores columns outside the table in which they are defined.

B

background advisors

Advisors that work in the background monitoring DB2, detecting problems, and making recommendations for corrective actions.

backout

A backward recovery (versus a forward recovery).

backout integrity checking

A process that compares changes of interest to subsequent updates and allows you to determine the impact of correcting a problem transaction on other activity.

backout recovery
A recovery method exclusive to RECOVER PLUS that backs out the log records to undo or redo the changes that occurred between the selected point in time and the current point. This method returns the spaces and indexes to the required state without the overhead of restoring image copies, or rebuilding or restoring indexes.

**backup copy**

A duplicate of the primary image copy of one or more table spaces or partitions. A backup copy will be used in an attempted recovery of those spaces when an attempted recovery that uses the primary copy fails. A backup copy may be designated for a local site or for a recovery site. See also primary copy.

**BACKWARD command**

A basic panel command that moves the display of scrollable lists back in the direction of the top of the list. Synonymous with the UP command.

**base table**

A table that is created by the SQL CREATE TABLE statement and that holds persistent data, or a table that contains a LOB or XML column.

**base table**

When used in the context of LOBs (large objects) and LOB table spaces, a base table is a DB2 table containing a LOB column definition. The actual LOB column data is not stored with the base table. The base table contains a row identifier for each row and an indicator column for each of its LOB columns.

**base table space**

A table space that contains base tables.

**basic sequential access method**

An access method for storing or retrieving data blocks in a continuous sequence, by using either a sequential access or a direct access device.

**batch reporting**
The process used to produce printed reports from instantaneous or historical data. All reports can be processed in batch (APPTUNE AND SQL Performance only).

**batch reports**

Reports designed in a format suitable for printing. These reports can also be displayed online (APPTUNE and SQL Performance only).

**BCSS**

BMC Consolidated System Subsystem.

**BEEP command**

A basic panel command that alternately disables and enables the terminal bell that rings whenever a message is displayed and when an exception condition is displayed.

**bind**

The process by which the output from the DB2 precompiler is converted to a usable control structure called a package or an application plan. During the process, access paths to the data are selected and some authorization checking is performed.

**BLOB (Binary Large Object)**

A sequence of bytes within DB2 where the size of the sequence ranges from 0 bytes to 2 GB minus one byte. Such a string does not have an associated CCSID.

**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.

BMCP

BMC Primary Subsystem.

BMCREORG

A command that runs the BMC Software REORG PLUS utility to perform a reorganization. The short form of the command is BMCR.

BMCSORT

A BMC sort routine that is installed with and used by the BMC Utility products for DB2.

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.

**bootstrap data set (BSDS)**

A VSAM data set that contains DB2 name and status information. It also contains RBA range specifications for active and archive log data sets, catalog and directory passwords, and lists of conditional restart and checkpoint records.

**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.

**BSAM**

See basic sequential access method.

**BSDS (Bootstrap Data Set)**

A VSAM data set that administers the DB2 log data sets. The BSDS is a component of DB2 that controls the log data sets and manages an inventory of the logs.

**buffer pool**

Main storage reserved to satisfy the buffering requirements for one or more table spaces or indexes.

**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.

**C**

**cache**

In LOADPLUS, a set of identity column values that LOADPLUS reserves when updating the MAXASSIGNEDVAL field of the SYSIBM.SYSEQUENCES table. LOADPLUS uses these values when generating values for an identity column.

In REORG PLUS, a range of values that REORG PLUS reserves when reorganizing a base table space that contains an XML column. If the document ID (DOCID) column of the base table space is missing, REORG PLUS adds the column and populates it with these cached values.

**CAF**

*See* call attachment facility.

**call attachment facility**

A DB2 attachment facility for application programs running in TSO or batch. The CAF is an alternative to the DSN command processor and provides greater control over the execution environment.

**CANCEL command**

There are two versions of the CANCEL command:

- **User Session:** Causes the last panel displayed before the current panel to be redisplayed, discarding any data entry in the process.

- **Data Collector:** Cancels a product user session.
There are two versions of the CANCEL command:

- **User Session**: Causes the last panel displayed before the current panel to be redisplayed, discarding any data entry in the process.

- **Data Collector**: Cancels a product user session.

**CAPS command**

A basic panel command that alternately changes the text displayed on all panels to uppercase or mixed case.

**CART**

Abbreviation for cartridge, a storage device that consists of magnetic tape, on supply and take-up reels, in a protective housing.

**catalog**

The OS/390 data set catalog or the DB2 catalog, depending on the context of the reference.

**catalog**

In DB2, a collection of tables that contains descriptions of objects such as tables, views, and indexes.

**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 uses aliases that point 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.
catalog table

Any table in the DB2 catalog.

c change accumulation file

A change accumulation file accumulates, extracts, and stores log record data for a specified object or group of objects. The BMC R+/CHANGE ACCUM product produces change accumulation files.

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.

CHANGES command

A basic panel command used to display the current Summary of Changes.

CHANGES command

A basic panel command used to display the current Summary of Changes.

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.
CHECK

Check pending (a DB2 space status).

check pending

A state of a table space or partition that prevents its use by some utilities and some SQL statements, because it can contain rows that violate referential constraints, table check constraints, or both.

CHECK PLUS for DB2

A BMC product that provides high-performance structural and data integrity checking of your DB2 databases.

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.

checkpoint

A point at which DB2 records internal status information about the DB2 log that would be used in the recovery process if DB2 should abend.

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.

cipher text

Data that has been converted to mask its meaning from an unauthorized recipient. Cipher text is the same as encrypted text.
clause

In SQL, a distinct part of a statement, such as a SELECT clause or a WHERE clause.

CLEAR command

A panel-specific command that erases the contents of the command text buffer on the Command Interface input panels.

CLEAR command

A panel-specific command that erases the contents of the command text buffer on the Command Interface input panels.

clear text

Data in normal, readable form. COPY PLUS standard image copies are in clear text. Clear text is the same as plaintext.

CLIST

Command list. A language for performing TSO tasks.

CLOB (Character Large Object)

A sequence of bytes within DB2 representing single-byte characters or a mixture of single and double-byte characters where the size can be up to 2 GB minus one byte. Although the size of CLOB values can be anywhere up to 2 GB minus one byte, in general, they are used whenever a character string might exceed the limits of the VARCHAR type.

clone table

A table that is structurally identical to a base table. The base and clone table each have separate underlying VSAM data sets, which are identified by their data set instance numbers.

CLUSTERING

An attribute of an index indicating whether CLUSTER was specified when the index was created; reported by BMCSTATS and RUNSTATS.
clustering index

An index that determines how rows are physically ordered (clustered) in a table space. If a clustering index on a partitioned table is not a partitioning index, the rows are ordered in cluster sequence within each data partition instead of spanning partitions. Prior to DB2 Version 8, the partitioning index was required to be the clustering index.

CMD command

A basic panel command that displays the Command Interface Menu, from which you can choose one of the Command Interface input panels to issue commands to the Data Collector, MVS, DB2 and other BMC products. Synonymous with the CMDMENU command.

CMDBOT command

A basic panel command that causes the Command line to be displayed at the bottom of all product panels.

CMDDDB2 command

A basic panel command that displays the DB2 Command Interface input panel, from which you can issue commands to DB2.

CMDDDC command

A basic panel command that displays the Data Collector Command Interface input panel, from which you can issue commands to the Data Collector.

CMDMENU command

A basic panel command that displays the Command Interface Menu, from which you can choose one of the Command Interface input panels to issue commands to the Data Collector, MVS, DB2 and other BMC products. Synonymous with the CMD command.

CMDOPT command

A basic panel command that displays the OPERTUNE Command Interface input panel, from which you can issue commands to OPERTUNE.

CMDTOP command
A basic panel command that causes the **Command** line to be displayed at the top of all product panels.

**cold start**

A process by which DB2 restarts without processing any log records.

**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.

**COLOR command**

A basic panel command that alternately enables and disables the color feature used to build product panels in color on terminals using native TSO without ISPF.

**column**

The vertical component of a table. A column has a name and a particular data type (for example, character, decimal, or integer).

**column condition rules**

Rules that you define within CHECK PLUS for checking column-level conditions.

**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.

**comma-delimited data**

*See* comma-separated-value data.
comma-separated-value (CSV) data

Data for which each value in a row is delimited by a comma. Also called comma-delimited or delimited data.

In LOADPLUS and UNLOAD PLUS, the delimiting character for data identified with FORMAT CSV can be any character.

command

A RECOVER PLUS command such as OPTIONS or REBUILD INDEX, DB2 operator command, or a DSN subcommand. Distinct from an SQL statement.

Command Interfaces

A set of panels used to issue and view the results of Data Collector, DB2, MVS, z/OS, and OPERTUNE commands and to receive DB2 messages.

commit

The operation that ends a unit of work so that the database changes made during that unit of work can be used by other processes. Contrast with rollback.

commit point

A point in time when data is considered consistent.

common service area

In OS/390, a part of the common area that contains data areas that can be addressed by all address spaces.

compatibility profile

A method of classifying data objects by performance characteristics to assist in the segregation of objects into buffer pools with other objects with similar characteristics.

compensated log record

A record in the DB2 log that reflects a database change that DB2 subsequently "reverses" or "compensates for." DB2 reverses changes in response to several situations, including
ROLLBACK statements, ROLLBACK TO SAVEPOINT statements, or some negative SQL return codes. *Contrast with* compensation log record and noncompensated log record.

**compensation log record**

A record in the DB2 log that reflects a database change made by DB2 to “reverse” or “compensate for” a previous change. DB2 creates compensation log records in response to several situations, including ROLLBACK statements, ROLLBACK TO SAVEPOINT statements, or some negative SQL return codes. *Contrast with* compensated log record and noncompensated log record.

**completion processing**

*See* row completion processing.

**component**

The term "component" refers to a product in its role as part of a solution.

**compression**

A process whereby a given amount data is compressed into a smaller number of bytes or characters without losing any of the information represented by the data.

**concurrency**

The shared use of resources by multiple processes at the same time.

**concurrent copy**

A feature that allows a copy to be made of a data set while concurrent read and write activity is occurring against that data set.

**condition**

A combination of predicates on a Utility product’s WHEN or WHERE statement. Conditions use operators such as AND, OR, NOT, and ().

**conditional restart**
A DB2 restart that is directed by a user-defined conditional restart control record (CRCR).

**conditional restart control record**

A control record that is used to direct a user-defined DB2 conditional restart.

**configuration**

Information about the database, input files, and processing options used during apply processing. The configuration information can be specified in the configuration file or the apply request.

**Configuration Advisor**

The product component that analyzes the buffer pool configuration and recommends the grouping of page sets into buffer pools based on performance attributes.

**configuration file**

The file that contains configuration information for an apply request.

**configuration manager**

The part of High-speed Apply Engine that validates and processes the configuration information.

**conflict**

The situation that occurs when High-speed Apply Engine attempts to update a target table, but the target database prevents the update. Conflicts can occur if the target table definition does not match the source table definition, if resources are unavailable, or if a time out occurs during apply processing.

**conflict manager**

The part of High-speed Apply Engine that receives conflict information from the apply agent, and processes conflicts according to the resolution rules defined in the configuration.

**conflict resolution rule**
a set of configuration parameters that describes how High-speed Apply Engine responds to a particular type of conflict. Each rule consists of a conflict type (specified with the Code parameter) and one or more actions (specified with the Action parameter).

**consistent copy**

A copy that does not contain any uncommitted data.

**conversion**

The process during a LOADPLUS or UNLOAD PLUS job where the utility changes the data type of the input data to a different data type for the output data (for example, from INTEGER to CHAR). *Contrast with* translation.

**coordinated recovery**

A method of recovery in which backups, recoveries, and associated tasks are coordinated across multiple DBMSs and applications.

**copies**

A DASD MANAGER PLUS printing option that specifies the number of image copies to make.

**copy pending**

A DB2 table space status.

**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 on 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.

coupling facility

An operating system hardware feature that facilitates communication between multiple DB2s in a data sharing group and the shared data storage.

CRCR

See conditional restart control record.

cross-system coupling facility

A facility that provides a communications link across systems.

CSA

See common service area.

CSV data

See comma-separated-value data.

D

DAC

The BMC Software DATA ACCELERATOR Compression product.

Daily History reports

Reports that provide a summary of all data in the trace data set, divided into 24-hour intervals.
See direct access storage device.

**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).

**Data Capture Changes (DCC)**

An attribute of a DB2 table that determines how much data DB2 logs when a row in the table is changed using an UPDATE statement. If the DCC attribute is set, and a table row is changed, the DB2 log contains at least one image of the entire row. If a table is defined with the DCC attribute, Log Master does not need to perform row completion processing on the log records related to that table. *Contrast with Data Capture None (DCN).*

**data capture flag**

A REORG PLUS flag that tells DB2 to log the entire row rather than only the changed portion of the row during a SHRLEVEL CHANGE reorganization. REORG PLUS turns on this flag only when the DATACAP option is YES and at least one table space partition is compressed.

**Data Capture None (DCN)**

An attribute of a DB2 table that determines how much data DB2 logs when a row in the table is changed using an UPDATE statement. If the DCN attribute is set, and a table row is changed, DB2 logs the changed portion of that row (from the first changed byte to either the last changed byte or to the end of the row). If a table is defined with the DCN attribute, Log Master performs row completion processing on the log records related to that table (to attempt to obtain the entire table row). *Contrast with Data Capture Changes (DCC).*

**data class**

A collection of DB2 or BMC trace records (IFCIDs) that can be specified in an output group. Only the data classes specified will be collected and stored in the trace data sets for that output group.
A collection of DB2 or BMC trace records (IFCIDs) that can be specified in an output group. Only the data classes specified will be collected and stored in the LOGSET log files for that output group.

**data collection information**

The recovery data collected during actual, simulated, and estimated recoveries for comparison and analysis.

**Data Collector**

The component that coordinates requests for data from all product and solution users and retrieves data from DB2.

**data control block**

A control block that is used by access method routines in storing and retrieving data.

**data definition name**

In JCL, the name of the data definition (DD) statement that names a data set and provides all of the required information for that data set to be processed.

**data definition name (ddname)**

The name of a data definition (DD) statement that corresponds to a data control block that contains the same name.

**Data Facility Storage Management Subsystem**

An operating environment that helps automate and centralize the management of storage.

**data migration**

the process of moving data from one database environment (table space or instance) to another

**data partition**

A VSAM data set that is contained within a partitioned table space.
data set name pattern user exit

A user exit routine that you can use with LOADPLUS or REORG PLUS to define a pattern for dynamically allocated data sets.

data set redefine user exit

A REXX-based user exit routine, supported by the DSRSEXIT option of REORG PLUS, that enables you to redefine certain aspects of your VSAM data sets.

data sharing

A DB2 mode of operation in which multiple DB2s can share data resources (such as table spaces).

data sharing

The ability of two or more DB2 subsystems to directly access and change a single set of data.

data-partitioned secondary index (DPSI)

A secondary index that is partitioned. The index is partitioned according to the underlying data.

data-sorting index

An index that is used to sort data. For traditional table spaces, this is a clustering index. For table-controlled partitioned table spaces, this is either a clustering index or, if there is no clustering index, a partitioning index.

database

A collection of tables, or a collection of table spaces and index spaces.

Database Administration for DB2

A BMC solution that contains the CATALOG MANAGER, CHANGE MANAGER, COPY PLUS, LOADPLUS, SNAPSHOT UPGRADE FEATURE, and UNLOAD PLUS products for DB2. This solution is designed to help database administrators, system programmers, and application developers automate the tasks associated with the implementation and administration of a DB2 database.
Users with a solution password can take advantage of additional features of the solution that are available when one component can rely on the presence of other components.

database administrator

An individual responsible for the design, development, operation, safeguarding, maintenance, and use of a database.

database descriptor

An internal representation of a DB2 database definition which reflects the data definition found in the DB2 catalog. The objects defined in a database descriptor are table spaces, tables, indexes, index spaces, and relationships.

database management system

A software system that controls the creation, organization, and modification of a database and access to the data stored within it.

Database Performance for DB2

A BMC solution that contains the DASD MANAGER PLUS, REORG PLUS, and, SNAPSHOT UPGRADE FEATURE products for DB2. This solution is designed to help database administrators, system programmers, and application developers automate the tasks associated with maintaining the performance of a DB2 database.

Users with a solution password can take advantage of additional features of the solution that are available when one component can rely on the presence of other components.

DATASOURCE statement

A control statement used to select the source of data for batch reporting (data set or Data Collector).

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.
**DB2 command**

An instruction to the DB2 subsystem that enables you to start or stop DB2, to display information about current users, to start or stop databases, to display information about databases, and so on.

**DB2 Component Services (DBC)**

DBC provides a persistent z/OS subsystem address space into which BMC products can dynamically initialize their own product services.

**DB2 Product Configuration**

The DB2 Product Configuration technology separates product (or solution) installation from configuration. Through its online interface, DB2 Product Configuration simplifies configuration and deployment by setting default option values for you.

**DB2LOAD statement/parameter**

A control statement (or parameter of the REPORT statement) used to produce a data set (from the data gathered to generate a batch report) in a format that can be loaded into DB2 tables using the DB2 Load utility.

**DBA**

See database administrator.

**DBADM**

Database administration privileges.

**DBCLOB (Double-Byte Character Large Object)**

A sequence of bytes representing double-byte characters where the size can be up to 2 GB minus one byte. Although the size of DBCLOB values can be anywhere up to 2 GB minus one byte, in general, they are used whenever a double-byte character string might exceed the limits of the VARGRAPHIC type.

**DBCS**

double-byte character set
**DBCS command**

A basic panel command that alternately enables and disables DBCS support when you are using native TSO without ISPF.

**DBCTRL**

Database control privileges.

**DBD**

*See* database descriptor.

**DBID**

Database identifier.

**DBMS**

*See* database management system.

**DBNAME**

A statistic collected by BMCSTATS and RUNSTATS that specifies a database name.

**DBRM**

database request module (DB2 only). High-speed Apply Engine creates a DBRM that contains all unique SQL statements in an apply request. This DBRM facilitates static SQL processing for DB2 targets.

**DCB**

Data control block.

**DD name**

*See* data definition name.
DD statement

Data definition statement.

DDL

data definition language. A subset of SQL statements that define data objects, such as the CREATE statements.

ddname

See data definition name.

DEBUG command

A Data Collector command that enables or disables various diagnostic functions useful in problem determination.

default options module

See installation options module.

default value

A predetermined value, attribute, or option that is assumed when no other is explicitly specified.

DELDB2 command

A Data Collector command that dynamically deletes a DB2 subsystem from the list of DB2s that can be monitored by a Data Collector.

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.

delimited data
See comma-separated-value data.

DELPROD command

A Data Collector command that dynamically deactivates an active System and SQL Performance product.

delta value

A value derived by subtracting the value in one record from the same value in another record.

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.

DFSMS Concurrent Copy

An IBM DFSMS feature that allows a copy to be made of a data set while concurrent read and write activity is occurring against that data set.

direct access storage device (DASD)

A device in which access time is effectively independent of the location of the data. Used in reference to fixed disk drives.

directory

The system database that contains internal objects such as database descriptors.

disaster recovery

The recovery of data that has been lost due to a major catastrophe at a computer installation. The recovery is performed at a remote recovery site using copies of the data made before the catastrophe.
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.

distributed system

For purposes of this document, a computer system other than a mainframe. A distributed system can be a minicomputer, a LAN server, or a personal computer. This document assumes that the operating system of a distributed system is UNIX or Windows.

DML

data manipulation language. A subset of SQL statements that manipulate data, such as the INSERT, UPDATE and DELETE statements.

DOMBLOD1

The utility that creates the appropriate CREATE TABLE DDL and LOAD utility control statements for the DB2LOAD data set written by the DOMBRPT1 utility.

DOMBPLOG

The utility that prints all entries from a report log data set.

DOMBRPT1

The utility that produces printed reports from a batch job.

DOMBSWIT

The utility that invokes the Data Collector SWITCH command from a batch job.

DOMEXIT1

A user exit that sets or changes the subsystem ID for security processing.

DOMEXIT2
A user exit that overrides selected user security attributes specified in the SECURITY data set.

DOMEXIT4

A user exit that validates a user’s authority to invoke a DB2 function.

DOMEXIT6

A user exit that assigns the user session ID and allows group session IDs to be used.

DOMPLEX option set

A collection of attributes that define one or more Data Collectors and their components (for example, the DB2s that can be monitored and the trace data sets that are used).

DOPTs

See installation options module.

double-byte character set

A set of characters that is used by national languages such as Japanese and Chinese that has more symbols than can be represented by a single byte. Each character is two bytes in length, and therefore requires special hardware to be displayed or printed.

DOWN command

A basic panel command that moves the display of scrollable lists forward in the direction of the bottom of the list. Synonymous with the FORWARD command.

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.

DUMP command

A Data Collector command that enables or disables the diagnostic dump that occurs at abend retry, or takes a diagnostic dump of a specific job.

dynamic allocation

The process by which the Utility products allocate work files and copy data sets during the utility job instead of requiring that the data sets be allocated by the user before running the job. Dynamic allocation usually provides the optimal number and size of data sets that a particular job needs.

E

ECKD device

Extended count-key-data device. A type of disk storage device.

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.

EDITPROC

In DB2, an SQL edit procedure typically used for the compression or encryption of data.

enclosure characters
The characters that surround a field in a delimited data file.

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.

encrypted text

Data that has been converted to mask its meaning from an unauthorized recipient. Encrypted text is the same as cipher text.

END command

A basic panel command that causes the last panel displayed before the current panel to be redisplayed, saving or committing any data entry in the process. Synonymous with the EXIT command.

Engine Level Mask (ELM)

An identification number of the compression engine used by the DATA ACCELERATOR Compression product. A new ELM is provided when a new compression engine becomes available or when changes are made to existing compression engines.

ENQS command

A Data Collector command that displays enqueue conflicts or all enqueues held or waited for by a specific job or a specific DB2 subsystem.

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.
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 BMCASU nn.EVENTS.

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.

exception processing

An APPTUNE feature that allows you to collect SQL text only for SQL calls that exceed thresholds you set for elapsed and/or CPU times.

exception status

In DB2, an abnormal table space or partition status (for example, check pending, copy pending, or recover pending).

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 BMCATS nn. RS_EXCEPTIONS2.
EXCP

  Execute channel program.

Execution

  ABMC 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.

execution mode

  Indicates whether a recommendation will be issued when a rule is triggered (manual) or the command associated with the rule will be issued (automatic).

EXIT command

  A basic panel command that causes the last panel displayed before the current panel to be redisplayed, saving or committing any data entry in the process. Synonymous with the END command.

exit routine

  A user-written (or IBM-provided default) program that receives control from DB2 to perform specific functions. Exit routines run as extensions of DB2.

EXPAND command

  A command used to move from summary data to more detailed data within a report.

EXPAND statement/parameter

  A control statement (or parameter of the REPORT statement) used to print batch reports in their expanded format.

Explain

  A powerful function that creates a set of unique, historical baseline snapshots containing key DB2 catalog statistics for an application.
EXPORT command

A command used to copy the selected records for the current report to an external DASD data set.

EXPORT files

Sequential data sets used to save the data collected to generate a report. Can be used as input to online or batch reporting.

F

fallback

The process of returning to a previous image copy of a space after attempting unsuccessfully to use the most recent image copy of a space for recovery.

filter

A set of tailored options for the collection of data based on specific programs, plans, DB2 subsystems, and user IDs (APPTUNE and SQL Performance only).

filter

The definition of selection criteria used to extract information from the logical log. Filters may include unit of recovery fields (for example, plan name or authorization ID), log record header fields (for example, DBID or PSID), and the contents of the columns within the data rows. A filter is used in conjunction with a time frame. Filters may be created through the Structured interface or through the Free Form interface in Log Master. See also Structured filter and Free Form filter.

filter association

The relationship of a filter to one or several work IDs.

filter option set

A set of tailored options for the collection of data based on specific programs, plans, DB2 subsystems, and user IDs (APPTUNE and SQL Performance only).
filter predicate

Search criteria of a filter definition. To define a predicate, you must define the field, conditional operator, and value. You can specify multiple predicates by using the online interface.

FIND command

A basic panel command that finds a specified string of text and moves the cursor to that text.

FKA command

A basic panel command that changes the display of function keys, alternating between the primary keys, alternate keys, and no display. Synonymous with the KEYS and PFSHOW commands.

FKEYS command

A basic panel command that displays the User Function Key Values panel (used to change the default values for function keys). Synonymous with the PFKEYS and PFKS commands.

FORWARD command

A basic panel command that moves the display of scrollable lists forward in the direction of the bottom of the list. Synonymous with the DOWN command.

forward recovery

The process of reconstructing a data set from a particular point by restoring a saved version of the data set, then applying changes to it in the same order in which they were originally made.

Free Form filter

A filter created through the Free Form interface in Log Master. You create the filter by using a format similar to an SQL WHERE clause with the predicates restricted to those supported by Log Master.

free space

The total unused space in a page, that is, the space not used to store records or control information.
full copy

In DB2, a complete copy of a specified space or spaces. This can be an image copy or a DSN1COPY-type copy.

FULLCOPY

A utility command that runs a full image copy using the IBM COPY utility. The short form of the command is FULL.

G

GB

Gigabyte (1,073,741,824 bytes).

GDG (Generation Data Group)

A collection of data sets kept in chronological order. Each data set is a generation data set. Each group of data sets has a qualifier that contains an integer value. Each time the system generates a GDG data set, the integer value increases by one, which keeps the data sets in chronological order.

generation data group (GDG)

A collection of data sets kept in chronological order. Each data set is a generation data set.

global options

A set of values that determines how the System and SQL Performance products are configured.

group attachment name

An alternative to subsystem ID for data sharing that allows the application to attach to any member in the group.

group buffer pool

DB2 buffer pool used for sharing access to pages between members of a data sharing group.
**group member name**

A unique name given to each data sharing member, which is specified in DSNZPARM macro DSN6GRP, parameter MEMBNAME.

**H**

**HELP command**

A basic panel command that causes informational text to be displayed, the topic of which is determined by the parameter specified, the point at which the command is issued, or by the position of the cursor.

**HILITE command**

A basic panel command that enables highlighting support when you are using native TSO without ISPF.

**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.

**historical reports**

Reports used to view data gathered from recent and/or archived trace data. See also Recent Trends reports and Daily History reports.

**HOME command**

A basic panel command that moves the cursor to the first input field on the current panel.

**hyperlink**

A navigational feature whereby you can move from reports to other reports or advisors with just one keystroke.
I

I/O

Input/output.

ICF

See integrated catalog facility.

IDCAMS

An IBM access method services program for data set and catalog structure maintenance.

IEDIT command

A command that exports command text to an ISPF edit session (when you are operating under ISPF).

IFCID

Instrumentation Facility Component identifier. The identifier assigned to a traceable DB2 event and to the associated trace record produced by DB2. This term is also used for records created by the System and SQL Performance products. IFCIDs generated by DB2 are preceded by "DB2" and records generated by the products are preceded by “BMC.”

ignore status

Indicates whether a recommendation or a command will be issued when a rule is triggered (no) or the rule will be ignored and no action will be taken (yes).

image copy

In DB2, a replica of a physical object such as a table space or data set.

Imagecopy

A BMCSTATS parameter that specifies whether to use the latest image copy as input for collecting statistics.
IN-DB2 time

The time interval used to measure the execution time of an entire SQL statement. IN-DB2 time includes the time spent on associated housekeeping tasks. See also IN-SQL time.

IN-SQL time

The time interval used by APPTUNE to measure the execution time of an SQL statement. IN-SQL measurement excludes the time spent on associated housekeeping tasks. See also IN-DB2 time.

INCRCOPY

A utility command that runs an incremental image copy using the IBM COPY utility. The short form of the command is `INCR`.

incremental copy

In DB2, a copy of only the changes that have been made to a specified space since the previous full or incremental copy was made.

independent filter

A Log Master filter created and saved as a Log Master object outside of a work ID.

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 key

The set of columns in a table used to determine the order of index entries.

Index Pages

ADASD 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

A VSAM data set that is contained within a partitioned index space.

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 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.
index record

A type of record written to a DB2 log that contains index information. This type of record is not required for recovery purposes.

index space

A page set used to store the entries of one index.

index-controlled partitioning

A type of partitioning in which partition boundaries for a partitioned table are controlled by values that are specified on the CREATE INDEX statement. Partition limits are saved in the LIMITKEY column of the SYSIBM.SYSINDEXPART catalog table.

indoubt

A status of a unit of recovery. If DB2 fails after it has finished its phase 1 commit processing and before it has started phase 2, only the commit coordinator knows if this unit of recovery is to be committed or rolled back. At emergency restart, if DB2 does not have the information needed to make this decision, its unit of recovery is indoubt until DB2 obtains this information from the coordinator.

indoubt resolution

The process of resolving the status of an indoubt logical unit of work to either the committed or the rollback state.

inflight

A status of a unit of recovery. If DB2 fails before its unit of recovery completes phase 1 of the commit process, it merely backs out the updates of its unit of recovery when it is restarted. These units of recovery are termed inflight.

inflight resolution recovery

A consistent recovery to any user-specified logpoint or timestamp without the need for a quiet point, available only with the Recovery Management for DB2 solution. Inflight resolution technology provides the ability to resolve inflight units of work at any point in time or to any log point and completely eliminates the need to perform quiesces to establish consistent recovery points during application execution.
inline copy

A copy produced by a LOAD or REORG utility. The data set produced by the inline copy is logically equivalent to a full image copy produced by running a COPY utility with read-only access (SHRLEVEL REFERENCE).

input file

the file that contains the updates for the target tables. This file can be a Log Master for DB2 logical log file or an SQL file.

installation options module

A module that contains the option values that control the default operating environment.

installation SYSADM

The person having the highest level of authority within DB2. This authority includes SYSADM authority and the privilege of granting SYSADM authority to others. This authority is assigned at installation time.

Instant Snapshot

An Instant Snapshot is a non-standard, data set level copy that is made by using intelligent storage systems with COPY PLUS and XBM. Instant Snapshots may be used for recovery by RECOVER PLUS or RECOVERY MANAGER but by not DB2 RECOVER. Instant Snapshots are registered only in the BMCXCOPY table.

instantaneous reports

Reports used to view data obtained instantaneously from DB2.

integrated catalog facility

A facility that provides for integrated catalog facility catalogs.

Interactive System Productivity Facility

An IBM licensed program that provides interactive dialog services.
internal resource lock manager

A subsystem that is used by DB2 to control communication and database locking.

INTERVAL statement/parameter

A control statement (or a parameter of the REPORT statement) used to specify a time interval that applies to the data in batch reports.

IPL

Initial Program Load.

IRLM

See internal resource lock manager.

ISPF

Interactive System Productivity Facility. A mainframe interactive interface that allows users to access mainframe utilities, tools, and applications.

ISPF skeletons

DD statement templates used by JCL Generation. Descriptions of these can be found in $AJXDOC.

J

JCL - job control language

A control language used to identify a job to an operating system and to describe the requirements of the job.

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.

JES

Job Entry Subsystem.

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 library

A set of user-identified partitioned data sets used as the primary source of load modules for 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 BMCASUxn.UT_JOB.

JOBLIB

See job library.

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

K

Kilobyte (1024 bytes).

KB

Kilobyte (1024 bytes).

key

A column or an ordered collection of columns identified in the description of a table, index, or referential constraint.

key data set

A data set that contains essential encryption key information and that is required for COPY PLUS encryption.
key format

The format of an output data set produced by CHECK PLUS that contains the key for each row that has a referential integrity violation.

key length

An information field showing the length of the index key as stored, specifically, the number of bytes (1 - 254).

key stores

Internal temporary working storage areas that Log Master backs with DASD data sets, as needed. Log Master uses the log record (LR), forward completion (FC), and backward completion (BC) key stores to process log records that require row completion processing.

KEYS command

A basic panel command that changes the display of function keys, alternating between the primary keys, alternate keys, and no display. Synonymous with the FKA and PFSHOW commands.

L

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.

LEFT command

A basic panel command that shifts the display of data to the left when a wide-mode panel or report (132 columns) is displayed on a screen with a width of 80 columns.

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.

limit key

The highest value of the index key for a partition.

lines

ADASD MANAGER PLUS graphic display option that specifies whether to draw lines on statistical graphs: LINE (lines) or NOLI (no lines).

LINESPP statement/parameter

A control statement (or a parameter of the REPORT statement) used to specify the number of lines to print on each page.

LOAD

A command that runs the IBM LOAD utility. The short form of the command is LOAD.
load module

A program unit that is suitable for loading into main storage for execution.

LOADPLUS for DB2

A BMC product that provides a flexible, high-performance way to load your DB2 data. LOADPLUS provides a feature-rich set of options, including options for maintaining the availability of your data during the load process and options for loading data in a variety of formats.

LOB (Large Object)

A sequence of bytes within DB2, representing bit data, single-byte characters, double-byte characters, or a mixture of single- and double-byte characters. A LOB can be up to 2GB minus one byte in length. See also BLOB, CLOB, and DBCLOB.

LOB table space

Analogous to table space, a LOB table space contains all data for a particular LOB column in the related base table.

local

Refers to any object maintained by the local DB2 subsystem.

local site

A physical DB2 system installation where DB2 applications are installed and in use. Compare with recovery site.

LOCATE command

A basic panel command that finds a specified list item and move the row containing that item to the top of the display area.

lock

A means of controlling concurrent events or access to data. DB2 locking is performed by the IRLM.
lock duration

The interval over which a DB2 lock is held.

locking

The process by which the integrity of data is ensured. Locking prevents concurrent users from accessing inconsistent data.

log

A collection of records that describes the events that occur during DB2 execution and their sequence. The information recorded is used for recovery in the event of a failure during DB2 execution. In particular, the log records are used to reconstruct damaged tables and indexes.

log apply

Refers to either a REORG PLUS SHRLEVEL CHANGE reorganization or the process that occurs in the LOGAPPLY phase of this type of reorganization. The log apply process applies log records to the reorganized object.

log inventory

The contents of the BSDS.

log mark

A name that you assign to a designated point on the log by using Log Master. When you specify a log mark, you can later refer to that point on the log using a name or term; you do not have to know the hexadecimal value for a particular relative byte address (RBA).

log record sequence number (LRSN)

A number DB2 generates and associates with each log record. DB2 also uses the LRSN for page versioning. The LRSNs generated by a given DB2 data sharing group form a strictly increasing sequence for each DB2 log and a strictly increasing sequence for each page across the DB2 group.

log scan
A process whereby information from DB2 logs is extracted based on your criteria and converted into a logical log.

**logging environment modeling tool**

An interactive component of PACLOG that helps you to represent a multi-variable process mathematically with the goal of finding the optimum values of one or more of those variables.

**logical log**

A human readable representation of the DB2 log containing a before image and an after image of specific changes based on your selection criteria. The logical log has two components: a control file that describes the format of the logical log and a data file that includes the actual data. (The logical log is platform independent.)

**logical log control file**

One of the two files that compose a logical log. This file contains information about the format and contents of the logical log.

**logical log data file**

One of the two files that compose a logical log. This file contains the actual data for each transaction (the update, delete, insert, and exchange actions).

**logical page list**

In DB2, a list of entries for pages that are logically in error. One of the types of exception statuses that RECOVERY MANAGER for DB2 is capable of recovering.

**logical partition**

A set of key or RID pairs in a nonpartitioning index that are associated with a particular partition.

**LOGSET**

A LOGSET is a group of z/OS linear data sets (or log files) in which the Next Generation Logger (NGL) stores data records. Each instance of NGL can support multiple LOGSETs.
LOGSORT

The RMGR strategy that allows you to use the LOGSORT option of RECOVER PLUS. This RECOVER PLUS option specifies that the appropriate range of log records be sorted and merged for use with a table space recovery.

LPL

See logical page list.

LVL

The number of levels in an index.

M

MainView for DB2

A BMC Software product used to monitor DB2 activity in real time and historically.

MainView for DB2 - Data Collector (DC)

A component of the MainView for DB2 product that allows MainView users to share some functions of the System and SQL Performance products Data Collector.

margin

ADASD 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

ADASD MANAGER PLUS graphic display option that specifies whether to use place markers on statistical graphs: **MARK** (markers) or **NOMA** (no markers).

Master Profile

A User Profile from which profile values can be extracted and assigned to other User Profiles.
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.

MB

Megabyte (1,048,576 bytes).

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.

MENU command

A basic panel command that displays the Available Commands panel. Synonymous with the SHOWCMDS command.

MERGECOPY

An IBM DB2 utility that is used to merge incremental image copies with previous incremental image copies or with a full image copy.

MERGECOPY

A command that runs the IBM MERGECOPY utility. The short form of the command is MERG.

migration

The application of data changes to another DB2 table. Log Master accomplishes migration by processing log records to produce SQL statements or load files and control cards to bring shadow tables up to date.
modeling tool

See logging environment modeling tool.

modified-page indicator

A flag bit in a space map page that is set if the corresponding data page has been updated since the last copy was made of this table space.

MODIFY

A command that runs the IBM MODIFY utility. The short form of the command is MOD1.

MODIFY command

A Data Collector command that displays or modifies the product timeout limit and the APPTUNE collection interval.

Most Frequent Value display

A BMCSTATS column statistics display that shows the ten most frequent values found in the column and their sequence.

multiple virtual storage (MVS)

The mainframe operating system under which DB2, IMS, and VSAM run. In PACLOG documentation, MVS means MVS/XA, MVS/ESA, MVS/370, and OS/390, unless otherwise noted.

MVS

See multiple virtual storage (MVS).

MVS/ESA

Multiple Virtual Storage/Enterprise Systems Architecture.

MVS/XA

Multiple Virtual Storage/Extended Architecture product.
N

NEXT command

A command that retrieves and displays report data in segments when the data collected for a report is too large to fit in the report output buffer.

Next Generation Logger (NGL)

NGL is a logging facility that logs and retrieves data based on application-defined keys and a time span. NGL runs as a service within the DB2 Component Services (DBC) subsystem and relies on the Runtime Component System (RTCS) for registry services.

nonclustering index

A table index in which the sequence of records in the index is defined independently of how records are physically ordered in the table.

noncompensated log record

A record in the DB2 log that reflects a database change that DB2 does not “reverse” or “compensate for.” In normal processing, most log records are noncompensated log records. Contrast with compensated log record and compensation log record.

nonpartitioned index

An index that is not physically partitioned. Both partitioning indexes and secondary indexes can be nonpartitioned.

nonpartitioned table space

In DB2, a simple table space (one that is not partitioned).

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.

0

OBID

Identifier of the table space file descriptor.

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

The smallest unit that RMGR recognizes and performs functions against.

object set

An extension of the DB2 object wildcarding facility in DASD MANAGER PLUS.

ongoing processing

A type of processing defined within Log Master that is designed to be run repeatedly. The start point of each run of an ongoing log scan depends on the end of the previous run. This type of processing enables you to repeatedly scan the DB2 logs for data without changing the SYSIN syntax of a job. With ongoing processing, the product ensures that

- any transactions that are open at the end of the current log scan will be included in the next log scan (during the next run of the job)
- any transactions that were completed within the previous log scan are not processed twice, even though the product might scan part of the same log range again.

online reporting
The process used to view DB2 data on the screen. Current DB2 activity can be monitored while the system is operating, or historical data can be viewed that has been stored in the trace data sets.

online utility

A type of utility execution that allows the object to remain available to other users in either read-only status (by using SHRLEVEL REFERENCE) or read/write status (by using SHRLEVEL CHANGE).

OPERTUNE for DB2

A BMC product used to dynamically modify DB2 installation parameters. Pool Advisor and System Performance have an interface to OPERTUNE that allows users to issue commands to OPERTUNE either via the OPERTUNE Command Interface panel or as a result of an advisor request.

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.

OPTIONS command

A basic panel command that displays the User Options Menu (offering a selection of panels used to modify the User Profile).

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).

OS/390
An IBM product that bundles MVS, TCP/IP, TSO, and VTAM. The first version of OS/390 (version 1.1, or R1) was equivalent to MVS version 5.2.2. In addition to earlier versions of MVS, the PACLOG products support OS/390 version 1.1 and later.

OUTLIM statement/parameter

A control statement (or parameter of the REPORT statement) used to specify the number of report groups to include in reports.

output group

The product component used to buffer trace records and to define and allocate the trace data sets to which records will be written from the output groups.

overtime mode

A mode of processing within Log Master that allows you to read log records associated with DB2 objects that are no longer defined in the DB2 catalog (called old objects). In overtime mode, the product obtains structure definitions for the previous instances of DB2 objects and uses the definitions to read log records related to those instances. The product can use any of several sources for the old object structure definitions.

P

package

An object containing a set of SQL statements that have been bound statically and that are available for processing. A package is sometimes also called an application package.

page

A unit of storage within a table space (4 KB, 8 KB, 16 KB, or 32 KB) or index space (4 KB). In a table space, a page contains one or more rows of a table.

page fixing

In virtual storage systems, marking a page as non-pageable so that it remains in real storage.

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.

**PANELID command**

A basic panel command that alternately displays or suppresses display of the panel ID in the upper left corner of all product panels.

**Parameter**

an element of an High-speed Apply Engine configuration file, also referred to as a configuration parameter. Parameters define processing information for an apply request.

**Parameters**

Specifications that set limits on DB2 resources and ZPARM settings and that specify increments by which to alter the settings, based on current performance conditions.

**Parse**

In RECOVER PLUS, to analyze the options entered with the RECOVER PLUS commands and use the information to create a parameter list for the command processor.

**Partial recovery**

Refers to a table space recovery to a prior point in time. The recovery can be to a prior copy, a prior quiesce point, or to a prior RBA.

**Partition**

Partition is defined as either of the following:

- A portion of a page set. Each partition corresponds to a single, independently extendable data set. The maximum size of a partition depends on the number of partitions in the partitioned page set. All partitions of a given page set have the same maximum size.

- A BMCSTATS and RUNSTATS statistic indicating the partition number.
partition-by-growth table space

Partition-by-growth table spaces are a type of universal table space (UTS). The usage of partition-by-growth table spaces is similar to a single table DB2 managed segmented table space. Partition-by-growth table space are managed by DB2. DB2 automatically adds a new data set when the database need more space to satisfy an insert. The table space begins as a single-partition table space and automatically grows additional partitions as needed to accommodate data growth. Partition-by-growth table spaces can grow up to 128 TB. The maximum size is determined by MAXPARTITIONS or DSSIZE value that you specified in DB2.

partitioned data set (PDS)

An MVS data set in direct access storage that is divided into portions, called members. Each member can contain a program, part of a program, or data. PDS is synonymous with library.

partitioned index

An index that is physically partitioned. Both partitioning indexes and secondary indexes can be partitioned.

partitioned table space

A table space subdivided into parts (based on index key range), each of which may be independently processed by utilities.

partitioning index

An index in which the leftmost columns are the partitioning columns of the table. The index can be partitioned or nonpartitioned.

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.

password activation deadline

The date (either the 15th or the last day of the month) through which a password will be accepted (installed) by the product authorization interface. When this date has passed, BMC Software Contracts Administration must issue another password.
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.

PDS

partitioned data set

performance analysis

Displays of statistics on log activity, information about the frequency of checkpoints, conditional restart records, and log command histories, as well as information about commit, rollback, and image copy frequency.

PFKEYS command

A basic panel command that displays the User Function Key Values panel (used to change the default values for function keys). Synonymous with the FKEYS and PFKS commands.

PFKS command

A basic panel command that displays the User Function Key Values panel (used to change the default values for function keys). Synonymous with the FKEYS and PFKEYS commands.

PFSHOW command

A basic panel command that changes the display of function keys, alternating between the primary keys, alternate keys, and no display. Synonymous with the KEYS and FKA commands.
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.

phase

In RECOVER PLUS, a distinct part of the total process required for recovery. RECOVER PLUS phases are UTILINIT/ANALYZE, LOG INPUT, MERGE, RESTORE, UNLOAD, BUILD, SNAP, and UTILTERM.

physical log

The DB2 log.

plaintext

Data that is in normal, readable form. (COPY PLUS standard image copies are plaintext.) Plaintext is the same as clear text.

plan

See application plan.

point of consistency

An RBA or LRSN to which a recovery can be made without jeopardizing the integrity of the data being recovered. Also called a quiesce point.

point of consistency

A time when all recoverable data that an application program accesses is consistent with other data.

point of discovery

The point at which you realize that a problem exists in the database. You might want to perform a redo or undo action relative to this point.
point-in-time recovery

Refers to a table space recovery to a prior point in time. The recovery can be to a prior copy, a prior quiesce point, or to a prior RBA.

Pool Advisor for DB2

A BMC Software product used to monitor and manage DB2 storage resources.

PQTY

An attribute indicating the primary space allocation (in tracks); reported by BMCSTATS.

pre-filtering

The process of filtering out unneeded log records at an earlier point in processing to improve overall Log Master performance. The product automatically performs pre-filtering, but you can increase the number of pre-filtered records (and improve performance) by including URID-level predicates in your filter (like authorization ID or plan name).

predicate

A conditional clause that Log Master uses to select log records (part of a filter).

primary authorization ID

The authorization ID used to identify the application process to DB2.

primary copy

An image copy (either full or incremental) of one or more table spaces or partitions. This is the first copy to be used (when specified) in an attempted recovery of those table spaces. A primary copy may be designated for a local site or for a recovery site. Contrast with backup copy.

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.

PROC

In TSO, a command procedure.

product administrator

An individual designated to control internal security and determine whether users should be restricted from performing tasks such as issuing MVS or DB2 commands.

product option file (POF)

A file that contains options you can specify to generate the JCL for individual data sets.

PRODUCTS command

A Data Collector command that displays a list of all the currently active BMC Software Performance products for DB2, which comprises the following products:

- Pool Advisor
- APPTUNE
- OPERTUNE
- System Performance for DB2
- SQL Performance for DB2
- MainView for DB2 - Data Collector
profile

See User Profile and application profile.

profile

See User Profile, DOMPLEX option set.

PSID

Identifier of the table space page set descriptor.

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.

QSAM

Queued sequential access method.

QUALIFIER statement/parameter

A control statement (or parameter of the REPORT statement) used to filter the data included in batch reports by identifier values.

Quiesce

A DB2 utility program that establishes a quiesce point (or quiet point) for a set of table spaces and records that point in the SYSIBM.SYSCOPY table of the DB2 catalog. The quiet point can be established for a table space, a partition, a table space set, a list of table spaces, or a list of table space sets.

The command that runs Quiesce utility, QUIESCE can also be used in the short form QUI.
quiesce point

See point of consistency.

quiet point

A point on the log during which no transactions were in flight for a set of table spaces. This information can be used to assist in recovering from problem transactions with a conventional recovery (and possibly in generation of REDO SQL). These points can also be used as targets for data migration.

QUIT command

A basic panel command that causes a quick, complete exit from the product in a manner equivalent to repeated executions of the CANCEL command.

R

RACF

See resource access control facility.

range-partitioned table spaces

Range-partitioned table spaces are a type of universal table space (UTS). Range-partitioned table spaces are based on partitioning ranges. The maximum size of a range-partitioned UTS is 128 TB. Range-partitioned UTS uses the segmented space map page organization. However, it contains a single table, which makes it similar to the regular partitioned table space.

RBA

The position of a byte in the DB2 log, expressed as an offset from the beginning of the log. A log record is identifiable by the RBA of the first byte of its header. That RBA is called the relative byte address of the record. The log record’s RBA is like a timestamp because it uniquely identifies a record that starts at a particular point on the continuing log.

reallocation

The process of respecifying the size of the space that is allocated to a data set.
rebind

To create a new application plan for an application program that has been bound previously. If, for example, you have added an index for a table accessed by your application, you must rebind the application to take advantage of that index.

Recall

A BMCSTATS parameter that enables you to recall archived data sets when collecting statistics.

Recent Trends reports

Reports that contain data from recent intervals (up to 48 hours).

recover pending

A DB2 table space status. Denotes that the table space has not yet been recovered to the current state.

recovery

In DB2 installations, the process of rebuilding DB2 table spaces after a failure of those spaces.

recovery

The process of rebuilding databases after a system failure.

recovery log

See log.

recovery pending

A status that prevents SQL access to a table space or index space that may need to be recovered. This status denotes that the table space has not yet been recovered to the current state.

recovery site
A physical DB2 system installation that is designated as the backup system for a local site in the event that a disaster recovery is necessary.

RECP

See recover pending.

referenced files

Files to which your input (LOADPLUS) or unload (UNLOAD PLUS) data set references. In LOADPLUS, you can load LOB or XML data from referenced files. In UNLOAD PLUS, you can unload LOB or XML data to referenced files.

referential integrity

The condition that exists when all intended references from data in one column of a table to data in another column of the same or a different table are valid. Maintaining referential integrity requires enforcing referential constraints on all LOAD, RECOVER, INSERT, UPDATE, and DELETE operations.

REGISET

The product registration data set.

register

In DB2, to store information (critical to recovery) about an image copy in the SYSIBM.SYSCOPY table.

registration point

The registration point is the log point (relative byte address (RBA) or log record sequence number (LRSN)) that is registered in the START_RBA column of SYSIBM.SYSCOPY or BMCXCOPY for an image copy. If a recovery is performed using the image copy, it is usually necessary to apply log records staring at the START_RBA.

relative byte address (RBA)

A hex address (offset) that uniquely identifies a DB2 log record. The RBA is with respect to the start of the log (offset = zero).
**remote**

Refers to any object maintained by a remote DB2 subsystem; that is, by a DB2 subsystem other than the local one. A remote view, for instance, is a view maintained by a remote DB2 subsystem.

**REORG**

A command that runs the IBM REORG utility. The short form of the command is REOR.

**REORG pending**

A condition that restricts SQL access and most utility access to an object that must be reorganized.

**REORG PLUS for DB2**

A BMC product that provides a flexible, high-performance way to reorganize your DB2 data. REORG PLUS provides a feature-rich set of options, including options for maintaining the availability of your data during the reorganization process.

**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.

**REORP**

See REORG pending).

**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`.

**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.

**report**

A panel or printed output used to display DB2 data.

**report log**

A data set used to store images of screens and reports produced using the LOG command.

**report logging**

The process used to send a copy of a report or screen image to the user’s report log data set for subsequent viewing or printing.

**Report Manager**

The System and SQL Performance products component that provides the user interface—the visible portion of the product. Through the Report Manager, users request that data be collected and measurements be taken. After the Data Collector gathers the data, the Report Manager sorts and formats the data into reports.

**report output buffer**

The buffer (allocated in extended private storage) that holds formatted report data.

**report set**

A comprehensive set of predefined reports that comes with all System and SQL Performance products.
The control statement used to produce batch reports.

REPORT_DD statement/parameter

A control statement (or parameter of the REPORT statement) used to designate a default ddname to which report output should be directed.

Repository

A set of DB2 tables that defines the content and format of information extracted from the DB2 log. The Repository retains the work ID definitions and, after a run, the work ID execution history data. The Repository is specifically created to be used by Log Master.

RESET command

A panel-specific command that resets the default values for function keys to the values shipped with this product or resets the contents of the command text on the Command Interface panels to its initial value.

resolution rule

See conflict resolution rule.

resource access control facility (RACF)

An IBM-licensed program that provides for access control by identifying and by verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources.

RESTART command

A command issued from only the product’s main menu that reprocesses the startup options in the User Profile as if an initial entry to the product was in progress.

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 \texttt{RESTARTPARM \textit{RESTARTPARMstring}} where \textit{RESTARTPARMstring} is a list of \textit{lineoneparms}.

restart processing

the processing that High-speed Apply Engine performs to start an apply request at an appropriate point after an interruption.

restart table

the table that High-speed Apply Engine uses to store restart information about apply requests. You must create one or more restart tables before High-speed Apply Engine can perform restart processing.

RETRIEVE command

A basic panel command that redisplays the last command issued on the \texttt{Command} line (excluding navigational commands like \texttt{EXIT}).

RETURN command

A basic panel command that causes a series of \texttt{EXIT} commands to be issued until the product’s main menu is displayed.

RFIND command

A basic panel command that reissues the previous \texttt{FIND} command (including the direction).

RI

\textit{See} referential integrity.

RID translation map
A temporary in-storage map that REORG PLUS uses to translate the RIDs of the original objects to the RIDs of the reorganized objects. REORG PLUS allocates and maintains this map when performing a SHRLEVEL CHANGE reorganization. The DB2 REORG utility uses a mapping table to perform this translation, and it consists of a DB2 table and index.

**RIGHT command**

A basic panel command that shifts the display of data to the right when a wide-mode panel or report (132 columns) is displayed on a screen with a width of 80 columns.

**RMGR**

RECOVERY MANAGER for DB2 product from BMC.

**RMGR group**

A group of objects defined to a specific RMGR product where all those objects correspond to a single DBMS.

**RO**

Read-only access. In DB2, a table space status that allows only read access to the table space.

**rollback**

The process of restoring data changed by SQL statements to the state at its last commit point. All locks are freed.

**row completion processing**

The additional processing that Log Master performs to obtain a complete image of the table row reflected in the log record of an update action. The log records of insert and delete actions contain a complete image of the table row involved in the action, but the log records of update actions might not contain a complete image of the table row unless the table is defined with Data Capture Changes (DCC). Log Master uses additional sources to obtain a complete image, including the current table space, available image copies, or other log records. Jobs that perform row completion processing require extra time and system resources.

**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**.

**Runtime Component System (RTCS)**

RTCS runs as a started task and provides programming services to various BMC mainframe products. RTCS is designed for continuous operation and seldom, if ever, needs to be stopped.

**RW**

Read-write access. In DB2, a table space status that allows both read and write access to the table space.

**S**

**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.
sampling

A method used to reduce overhead by collecting data for one four-second period out of each 16 seconds of elapsed time, providing a representative sample of data with greatly-reduced overhead (APPTUNE and SQL Performance only).

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.

scan range

The set of log records that will be read by Log Master, including the selection range and any other log information required for reporting purposes (such as backout integrity checking).

secondary authorization ID

An authorization ID that is associated with a primary authorization ID by the authorization exit routine.

secondary index

A nonpartitioning index on a partitioned table.
section name

the label that identifies a group of parameters. Section names are delimited by slash characters (/SectionName/) or square brackets ([SectionName]).

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)

segmented table space

A table space that is divided into equal-sized groups of pages called segments. Segments are assigned to tables so that rows of different tables are never stored in the same segment.

segmented viewing

A method used to retrieve and display report data in segments when the data collected for a report is too large to fit in the report output buffer.

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.

selection range
The set of log records (defined by the time frame or RBA) that will be output to the logical log.

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.

sequential data set

A non-DB2 data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape. Several of the DB2 database utilities require sequential data sets.

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.

severity indicators, message

The letter following a BMC message number that indicates whether a message provides information, provides a warning, indicates an error, or indicates that you must take an action. CHECK PLUS provides an option to change the severity indicator for a range of CHECK PLUS messages.

shadow tables

A copy of tables.

SHOWCMDS command

A basic panel command used to display the Available Commands panel, which lists all commands that can be issued from the panel.

SHRLEVEL REFERENCE

A term that is applied to a DB2 table space or partition to indicate that only read access is allowed to the space.
SHUTDOWN command

A Data Collector command used to stop the Data Collector subsystem in a normal manner.

SID

system identifier (Oracle only). A unique name that identifies an Oracle instance.

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.

simple table space

A table space that is neither partitioned nor segmented.

simulation

In PACLOG, the process of creating processed copies of archive logs without performing any actions against the original logs.

SMF

See system management facility.

SMS

See Storage Management Subsystem.

SMF

See Storage Management Subsystem.

snapshot utility

An execution of a Utility product that uses the SNAPSHOT UPGRADE FEATURE (SUF) technology of EXTENDED BUFFER MANAGER (XBM) to perform an online execution.
SORT command

A basic panel command that rearranges the order of data on scrollable panels or the order of repeating groups of data in reports.

SORT/SORT2 statements/parameters

The control statements (or parameters of the REPORT statement) used to sort the data included in batch reports.

source table

for data migration or transaction recovery, the table that originally contained the data that will be migrated or recovered.

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.

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).

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.

spawner

See automation spawner.

SPFOFF command
A basic panel command that disables the use of ISPF as the dialog display mechanism when it has been temporarily enabled using the SPFON command.

**SPFON command**

A basic panel command that enables the use of ISPF as the dialog display mechanism (SPFOFF is the default when you begin your session).

**spill data sets**

Data sets that REORG PLUS uses to store RID maps or log records when the amount of space needed to store these maps or records is greater than the space originally specified.

**SPUFI**

SQL Processor Using File Input. The IBM utility that lets you execute SQL statements without embedding them in an application program.

**SQL**

Structured Query Language. A language that can be used within programs and interactively to request information from a DB2 subsystem.

**SQL Explorer for DB2**

The SQL Explorer for DB2 product is an SQL analysis tool that enables you to solve performance problems that result from inefficient SQL statements (component of SQL Performance for DB2).

**SQL Generator**

The part of Log Master that generates UNDO, REDO, and MIGRATE SQL with WHERE clauses based on available index information. Another part of the product (called the SQL Processor) executes the SQL statements to undo or redo specific transactions, or to migrate data to another location in a database.
SQL ID

The authorization ID that is used as an implicit qualifier of table and index names in dynamic SQL statements. In RMGR, the SQL ID defaults to the user ID.

SQL output data set

The data set containing SQL generated by Log Master. This data set can be reviewed and then executed in batch.

SQL Performance for DB2

A BMC solution that combines the features and functions of APPTUNE and SQL Explorer with additional index capabilities.

SQL processor

The part of Log Master that processes the UNDO, REDO, or MIGRATE SQL generated by the SQL Generator. Log Master currently uses the High-speed Apply Engine to execute SQL statements.

SQL statistics collection interval

The interval (in minutes) or multiple intervals (in one-hour increments) at which data is written to the LOGSET log files (APPTUNE for DB2 and SQL Performance for DB2 only).

SQL template data set

A data set that contains descriptions of all distinct types of SQL statements that are contained in the SQL output data set. The SQL template information is optional, but recommended because it improves performance when the High-speed Apply Engine executes the generated SQL.

SQTY

An attribute indicating the secondary space allocation (in tracks); reported by BMCSTATS.

SSID

subsystem identifier (DB2 only). A unique name that identifies a DB2 subsystem.
stack tapes

A DASD MANAGER PLUS user option that specifies whether to stack image copy tapes.

stacked tape

In backup and recovery, a tape on which successive image copies are stored.

stacked tape

A tape on which a series of data sets are stored in succession. In PACLOG, these are copies of processed archive logs.

staging data sets

Data set shadow copies to which LOADPLUS and REORG PLUS write during online operations (SHRLEVEL CHANGE or SHRLEVEL REFERENCE). After the load or reorganization is complete, the utility replaces the original data sets with the staging data sets.

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.

STARTDB

Start database privileges.
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.

STATUS command

There are two versions of the STATUS command:

- **Report Manager:** When issued from a report, STATUS displays the Report Status panel. When issued from any other product panel, STATUS displays the Session Status panel.

- **Data Collector:** Displays status information about the Data Collector subsystem and its associated DB2 subsystems.

STATUSR command

A report-specific command that displays the Report Status panel containing a summary of information about the current report.

STATUSS command

A basic panel command that displays the Session Status panel containing a summary of information about the current user session.

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 command

There are two versions of the STOP command:

- **Report Manager**: Cancels auto mode (under which values on the current panel are automatically updated and redisplayed at a specified interval).

- **Data Collector**: Stops the Data Collector subsystem in a normal manner.

STOP DB

A utility command that runs the DB2 STOP DATABASE command. The short form of the command is STO DB.

STORAGE command

A Data Collector command that displays details of storage usage for a specific job or task on the system.

storage group (stogroup)

A named set of DASD volumes on which DB2 data can be stored.

storage management mode

A state of storage constraint that determines whether increases and decreases in storage will be allowed based on the available storage in the DBM1 address space and the MVS paging rate.

Storage Management Subsystem

An operating system component that is used to automate and centralize the management of storage by providing the storage administrator with control over data class, storage class, management class, storage group, and ACS routine definitions.

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.

**Structured filter**

A filter created through the Structured interface in Log Master. You are prompted to select a field, conditional operator, and a value through a set of linked panels to create a predicate.

**Structured Query Language**

A database sublanguage used in querying, updating, and managing relational databases.

**submit job**

A DASD MANAGER PLUS job generation option that specifies whether to submit the current job for execution.

**subsystem**

A distinct instance of a RDBMS.

**subsystem groups**

A balanced set of groups that contain all objects on a DB2 subsystem. Subsystem groups are created automatically by RECOVERY MANAGER.

**SUF**

BMC Software’s SNAPSHOT UPGRADE FEATURE for DB2 product.
supervisor call (SVC)

A request that serves as the interface into operating system functions, such as allocating storage. The SVC protects the operating system from inappropriate user entry. All operating system requests must be handled by SVCs.

SVC

See supervisor call.

SWITCH command

A Data Collector command that generates an archive from the currently active log file for the specified output group.

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.

Symbolic variables should not be confused with global job variables (AJX-type), which have their values set for all users and all sessions.

symbolic variables

In RMGR, a symbol that represents a variable node in a data set name. The symbol is a string of one or more characters that are prefaced by an ampersand (&).

SYNC

A utility command that runs a checkpoint to use for restart processing.

sync 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{n}.UT_SYNC.

**synchronization point (sync point)**

A point in time when data is considered consistent and from which an application can restart if a failure occurs.

**SYSADM**

System administration privileges.

**SYSCTRL**

System control privileges.

**SYSIBM.SYSCOPY**

In DB2, a catalog table that stores (registers) information about each image copy made. The information stored is required for recovery purposes and includes the names of the subject database and spaces, the date and time of the copy, and the SHRLEVEL type.

**SYSIBM.SYSLGRNX**

In DB2, a table used to store log ranges associated with table space transactions.

**SYSPLEX command**

A Data Collector command that establishes or terminates sysplex communication between the local Data Collector and other members of its DOMPLEX or displays information about the status of the DOMPLEX and the DB2s it monitors.
A Data Collector command that establishes or terminates sysplex communication between the local Data Collector and other members of its DOMPLEX or displays information about the status of the DOMPLEX and the DB2s it monitors.

system administrator

The person having the second highest level of authority within DB2. System administrators make decisions about how DB2 is to be used and implement those decisions by choosing system parameters. They monitor the system and change its characteristics to meet changing requirements and new data processing goals.

system advisor

The advisor that evaluates recommendations from individual component advisors and accepts or rejects them, based on its analysis of all system requirements and available resources.

System and SQL Performance products for DB2

An integrated family of products that share common DB2 data collection facilities and a common interface. The System and SQL Performance products comprise the following products and solutions:

- APPTUNE
- MainView for DB2 - Data Collector
- Pool Advisor
- SQL Explorer
- SQL Performance for DB2
- System Performance for DB2

system management facility

An optional control program feature of OS/VS that provides the means for gathering and recording information that can be used to evaluate system usage.

System Performance for DB2

A BMC solution that combines the features and functions of MainView for DB2, Pool Advisor, and OPERTUNE for DB2 with additional reporting capabilities.
system resources

In DB2, those items that are controlled by DB2, such as the BSDSs, the logs, and the catalog and directory.

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

A named data object consisting of a specific number of columns and some number of unordered rows. Synonymous with base table.

Table Pages

A DASD MANAGER PLUS statistical graph that plots the number of pages in a table over time.

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 check constraint

A user-defined constraint that specifies the values that specific columns of a base table can contain.
table constraints

Application-type criteria that must be satisfied before access to that table is allowed.

Table Percent Pages

A DASD MANAGER PLUS statistical graph that plots the percentage of pages in a table over time.

table space

A page set used to store the records of one or more tables.

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.

table-controlled partitioning

A type of partitioning in which partition boundaries for a partitioned table are controlled by values that are defined in the CREATE TABLE statement. Partition limits are saved in the LIMITKEY_INTERNAL column of the SYSIBM.SYSTABLEPART catalog table.

Tables

A BMCSTATS parameter that displays a panel from which to select tables and columns for collecting statistics.

target table

any table that High-speed Apply Engine updates during an apply request

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.

template file

a separate file generated by the Log Master for DB2 product that High-speed Apply Engine can use to efficiently process SQL input as static SQL against DB2 for mainframe or DB2 UDB targets. The template file contains one entry for each distinct statement type in the SQL input. If an apply request includes a template file, High-speed Apply Engine processes the template file before it begins processing SQL input.
TERSE command

A command used to suppress blank lines and nonessential data on the display screen in order to increase the area used to display report output data.

thread

A DB2 mechanism that provides a path for an application to connect to (and perform work in) DB2 and that determines access to DB2 resources and services.

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.

time frame

A period of time that determines from which specific log files (or sections of log files) the product extracts log records. Time frames are defined by specifying a start and end point and by specifying the source of the log records to be scanned.

time-sharing option

Provides interactive time sharing from remote terminals.

timestamp

A seven-part value that consists of a date and time expressed in years, months, days, hours, minutes, seconds, and microseconds.
timestamp recovery

A consistent recovery to any user-specified timestamp without the need for a quiet point, available only with the Recovery Management for DB2 solution. See "inflight resolution recovery" on page 276.

TMM

Tape Mount Management.

TOP command

A command used after one or more NEXT commands to redisplay the first segment of a report.

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.

trace data sets

The data sets allocated in the DOMPLEX option set for the storage of DB2 trace records. Two data sets are allocated for each specification—one to store the trace records and one to store the index entries used to keep track of trace data.

transaction recovery

An application recovery whereby the effects of some or all of the transactions in a specified time frame are removed from the database.

translation

The process during a LOADPLUS or UNLOAD PLUS job where the utility changes the encoding scheme of the input data to a different encoding scheme for the output data (for example, from ASCII SBCS to EBCDIC SBCS). Contrast with conversion.
trigger

A condition or set of conditions in the DASD MANAGER PLUS exceptions table that cause REORG PLUS to perform a reorganization when the CONDEXEC option is BMC.

TSO

See time-sharing option.

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.

**unit of recovery**

A transaction (such as an insert, update, or delete action) that might need to be recovered. (A sequence of operations within a unit of work between commit points.)

**universal table space (UTS)**

A table space that is both segmented and partitioned. A combination of partitioned and segmented table space schemes provides better space management as it relates to varying-length rows and improved mass delete performance. Universal table space types include range-partitioned and partition-by-growth table spaces.

**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.

**UP command**

A basic panel command that moves the display of scrollable lists back in the direction of the top of the list. Synonymous with the BACKWARD command.
UPDATEDB2

A historical parameter indicating whether the DB2 catalog was updated with the statistics collected by the BMCSTATS utility.

User Profile

A collection of attributes that define a user’s access to product functions and reports.

USERS command

A Data Collector command that displays a list of users in session with a specified Data Collector.

UT

Utility-only access. In DB2, a table space status that allows only utilities to access the space.

UTILCODE

An 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.

utility ID

The 1- to 16-character identifier that gives a unique name to a utility job.

Utility Products for DB2

A collection of high-performance products from BMC that includes CHECK PLUS, LOADPLUS, REORG PLUS, and UNLOAD PLUS. These products are designed to help database administrators, system programmers, and application developers perform the tasks associated with administration of a DB2 database.
UTRO

Utility/read-only. In DB2, a table space status that indicates a DB2 utility is executing against the table space but allows only read access by others to the table space.

UTRW

Utility/read-write. In DB2, a table space status that indicates a DB2 utility is executing against the table space but allows read and write access by others to the table space.

UTUT

Utility/utility. In DB2, a table space status that indicates a DB2 utility is executing against the table space and allows only other utilities to read and write to the table space.

V

validation

In RMGR, the process of verifying that all objects in a group are recoverable.

variable

See symbolic variable and global job variable.

variable repository

A storage mechanism of System Performance for DB2 and Pool Advisor for DB2 where the user can view the parameter variables and modify them.

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.

version code

See submodel.

Virtual Storage Access Method (VSAM)

An IBM data-access method that allows efficient sequential and direct access to MVS data sets.

cvolser

The name of a volume.

colume

A single unit of storage that is referenced by a serial number and is used for data set allocation by DASD MANAGER PLUS.

VSAM

See Virtual Storage Access Method.

W

warm start

The normal DB2 restart process which involves reading and processing log records so that data under the control of DB2 is consistent. Contrast with cold start.

What-If Edit SQL Explain

This process edits the SQL and a dynamic Explain is performed.

What-If Index
This process simulates the effects of adding or dropping an index or updating statistics for an index using cloned structures.

**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:

```plaintext
Table space: databaseName.TS%Stogroup: SYS%Volume: V%
```

**work file**

A temporary file that is used for the temporary storage of data that is being processed.

**WORKIDS table**

A DASD MANAGER PLUS object definition table that stores actions and task IDs. This table is named BMCASUnn.DO_WORKIDS.

**working group**

In RMGR, the current group; the object group that is currently being processed.
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.

X

XBM

EXTENDED BUFFER MANAGER, a BMC product.
Index

-BR14 command 499
-NEWJ command 499
-UDSJ command 499
(DB2), supported versions 45
#DOMZAP
definition 743
$AJXDOC data set 115

2MEGSQL POF keyword 134
9DEFAULT record
definition 743

A

abend
definition 743
abend reason code
definition 743
abnormal end of task
definition 743
ABOUT command
definition 743
access method
definition 744
accessing
actions 278
functions 55
Object Sets 311
other databases 73
plans 70
product database 73
statistics function 334
ACK
definition 744
ACKESQL
definition 744
ACM_AMS POF keyword 135
ACM_ANALYSIS_SYSTOUT POF keyword 134
ACM_BASDIAG POF keyword 135
ACM_BRPTDIAG POF keyword 135
ACM_BRPTDSN POF keyword 135
ACM_CDLDSDN POF keyword 135
ACM_CDLS PS keyword 136
ACM_CDLS S POF keyword 136
ACM_CDLSU POF keyword 136
ACM_CPLDIAG POF keyword 136
ACM_CMPDIAG POF keyword 136
ACM_CPLCDLO POF keyword 136
ACM_CPLCDLIG POF keyword 136
ACM_CPLWDSN POF keyword 136
ACM_CPLWDSNO POF keyword 137
ACM_DBRM1 POF keyword 137
ACM_DBRM2 POF keyword 137
ACM_DBRM3 POF keyword 137
ACM_DYNSORTW_NUM POF keyword 137
ACM_DYNSORTW_UNIT POF keyword 137
ACM_GLID POF keyword 137
ACM_IBMR_MAP_REQ POF keyword 137
ACM_IMPDIAG POF keyword 138
ACM_JDSN POF keyword 138
ACM_JDSNB POF keyword 138
ACM_JDSNBG POF keyword 138
ACM_JDSNC POF keyword 139
ACM_JDSNCPL POF keyword 139
ACM_JDSNCPLO POF keyword 139
ACM_JDSNE POF keyword 139
ACM_JDSNI POF keyword 139
ACM_JDSNE POF keyword 139
ACM_PARALLEL_MAXINIT POF keyword 140
ACM_PARALLEL_MININIT POF keyword 140
ACM_PARALLEL_WORKLST POF keyword 140
ACM_PARALLEL_XIMGRP POF keyword 140
ACM_PARALLEL_XIMPROC POF keyword 140
ACM_PARALLEL_XIMSTRT POF keyword 141
ACM_PARALLEL_XIMTRCE POF keyword 141
ACM_PIC POF keyword 141
ACM_SBSTN POF keyword 141
ACM_SDSNE POF keyword 141

Index  849
ACM_WDSN POF keyword 141
ACM_WLORDER POF keyword 142
ACM_WLORDERMSG POF keyword 142
ACM_WLPS POF keyword 142
ACM_WLSS POF keyword 142
ACM_WLU POF keyword 142
ACT
  definition 744
action
  definition 744
action definitions, specifying to Export 654
action name
  definition 744
action owner
  definition 744
action POF
  definition 744
action POF, associating with actions 265
Action Report 610
action status
  definition 745
action type
  definition 745
actions
  associating a POF 265
  controlling access to 278
corrective, for REORG 524
creating Object Set separately 291, 294
eliminating duplicates 433
generating 248
  identifying associations with object sets 300
JCL Options file (POF) 265
  POF, defining 125
  reports 610
Actions
  creating 241
active log
  definition 745
ACTWRK_DATACLASS 134
ACTWRK_MGMTCLASS 134
ACTWRK_PRIQTY 134
ACTWRK_STORCLASS 134
ACTWRK_UNIT 134
ADDDB2 command
  definition 745
adding
  threshold 462
  adding a customizable report to the menu 608
ADDLOAD1 POF keyword 143
ADDLOAD2 POF keyword 143
ADDPROD command
  definition 745
definition 745
address space
  definition 746
ADMIN command
  definition 746
administrative products for (DB2) 47
Administrative Products for DB2
  definition 746
ADU
  definition 746
advantages of using object sets 280
advisor
  definition 746
AES
  definition 746
AEXAUNLD package 71
AEXESTDL package 71
AEXPRINT
  definition 746
AEXSQLIO package 71
AEXvrHA Execution Monitor Entry plan 71
AEXvrHM Execution Monitor plan 71
agent
  definition 747
AJX variables
  definition 747
AJXIN input stream 114
AJXPODAT edit macro 104
AJXPOFER message file 107
AJXPOFIN input stream 114
AJXPOFVL
  definition 747
AJXPOFVL message file 107
allocating data sets 69
allocation unit
  definition 747
ALLOCUNIT
  definition 747
ALLOW_NETCMD 718
ALTDD
  definition 747
ALTER
  administrative product 47
  Execution plans 71
  fast path navigation commands 58
ALTER for DB2
  definition 747
altering and re-creating objects 376
ALTERSEC
altering SQTY 544
command description (ALT2) 225
considerations 541
definition 541
flow diagram 541
how it works 541
introduction 541
SQL command 541
ALU
definition 748
ampersands in installation options module 204
AMU
definition 748
analyzing
column statistics 388
column value statistics 390
index partition page group statistics 394
index partition statistics 392
index statistics 390
statistics 379
storage group statistics 395
table space partition page group statistics 385
table space partition statistics 382
table space statistics 381
table statistics 386
volume statistics 396
Analyzing Statistical Trends 379
anomaly
definition 748
AP command
definition 748
APF
definition 748
API
definition 748
Append insert trigger 449
APPFILT command
definition 748
application
definition 748
application group
definition 749
application plan
definition 749
application profile
definition 749
application program interface
definition 749
apply agent
definition 749
apply processing
definition 749
apply request
definition 749
APPNDINS exception 449, 517
APPOFF command
definition 749
APPON command
definition 750
APPPROF statement/parameter
definition 750
APPRESET command
definition 750
APPSTAT command
definition 750
APPTUNE
definition 750
ARCH_DATACLASS POF keyword 143
ARCH_DATACLASS_ALT POF keyword 143
ARCH_EXPDT 143
ARCH_MGMTCLASS POF keyword 143
ARCH_MGMTCLASS_ALT 143
ARCH_PREFIX POF keyword 143
ARCH_PRIQTY 144
ARCH_RETPD POF keyword 144
ARCH_SECQTY 144
ARCH_STACK POF keyword 144
ARCH_STORCLASS POF keyword 144
ARCH_STORCLASS_ALT POF keyword 144
ARCH_THRESH POF keyword 144
ARCH_UNIT POF keyword 144
ARCH_UNIT_ALT POF keyword 144
archive directory
definition 750
archive log
definition 750
AREOPEND exception 449, 517
ARERPEND exception 449, 517
ARU
definition 751
ASCII
definition 751
ASU
definition 751
ASU_XP_LOGD_DATAC POF keyword 145
ASU_XP_LOGD_LOGDSN POF keyword 145
ASU_XP_LOGD_MGMTC POF keyword 145
ASU_XP_LOGD_PRIQTY POF keyword 145
ASU_XP_LOGD_SECQTY POF keyword 145
ASU_XP_LOGD_STORC POF keyword 145
ASU_XP_LOGD_UNIT POF keyword 145
ASU_XP_UIMSRVHOST POF keyword 145, 642
ASU_XP_UIMSRVPORT POF keyword 145, 642
ASU_XP_UIMSRVTIMEOUT POF keyword 146
ASUDOPTS installation listing 205
ASURXLOC external function 691
ASURXLSP external function 686
ASURXRPT external function 684
ASURXSE external function 686
ASURXSQL external function 682
ASUvrDJ BMCTRIG Utility Job Generation plan 70
ASUvrDR Report Display plan 70
ASUvrDS Statistics Collection plan 70
ASUvrDZ Browse Statistics plan 70
ATBWORKAREA installation option 206
audit
  definition 751
authorization
  actions 278
  collecting statistics 356
  controlled through security exits 72
  DASD MANAGER PLUS database 73
  for Export 642
  Object Sets 311
  to access the BMC UIM server Commands web page 718
  to other databases 73
authorization ID
  definition 751
authorized program facility
  definition 751
auto-refresh
  definition 751
automation modes
  definition 751
automation spawner
  definition 752
AutoSub
  definition 752
auxiliary table
  definition 752
Average Length of RID Chain Report 551
Average Monthly Growth Report 612

B

background advisors
  definition 752
backout
BMC UIM Server Commands web page 718
BMC_CHECK_LOAD POF keyword 148
BMC_CHECK_OPTS POF keyword 148
BMC_COPY_LOAD POF keyword 148
BMC_COPY_OPTS POF keyword 148
BMC_LOAD_LOAD POF keyword 148
BMC_LOAD_OPTS POF keyword 148
BMC_RECOVER_LOAD POF keyword 149
BMC_RECOVER_OPTS POF keyword 149
BMC_REORG_LOAD POF keyword 149
BMC_REORG_OPTS POF keyword 149
BMC_REORG_XBMID POF keyword 150
BMC_UNLOAD_LOAD POF keyword 150
BMC_UNLOAD_OPTS 150
BMCCOPY
  command description (BMCC) 225
  Copy Options panel 120
  definition 754
  grouped services 436
  specifying GDGs on the COPYDDN options panel 120
  WEEKLYREORG service 224
BMCCOPYI (BMCC) command description 225
BMCCPRS
  command description 225
  definition 754
  grouped services 436
  updating database 374
BMCLOAD
  definition 754
BMCLOAD (BMCL) command description 225
BMCMOD command description 225
BMCP
  definition 755
BMCREORG
  command description (BMCR) 225
  definition 755
  grouped services 436
  mapping table variables for REORG 524
  specifying GDGs on the COPYDDN options panel 120
  WEEKLYREORG service 224
BMCSORT
  definition 755
BMCSSTATS
  accessing 334
  Age trigger 453
  benefits 317
  browsing 341
  command description (BMCS) 225
  database tables 318
  definition 755
  deleting statistics 376
  displaying for analysis 380
  grouped services 436
  identifying trigger values 469
  options 320
  sample JCL 363
  sampling index parameter 320
  sampling tablespace parameter 320
  See also statistic[BMCSTATS:aa] 313
  table columns updated 357
  table space/index columns updated 357
  updating DB2 catalog 363
  WEEKLYREORG service 224
BMCSYNC installation option 206
BMCTRIG
and commands 425
and utilities 425
bypassing evaluations 432
command description (BMCT) 225
definition 755
evaluating exceptions, overview 442
exceptions 432
exceptions job 476
excluding items 521
flow diagrams 442
generated standard JCL example 502
information sources 203
interim worklist 499
introduction 425
JCL generation options 521
job card exit in interim worklist 501
job generation options 521
order of using information 203
processing options, introduction 440
Rebind option 485
REORG corrective action 524
REORG job 521
Resize Down option 486
restart/start over a job 535
setting thresholds 445
syntax options 479
taking corrective action 469
thresholds 445
worklist JCL 507
BMCUNLOD
definition 755
BMCUNLOD (BMCUN) command description 225
BMCUPRS
command description (BMCU) 225
definition 755
grouped services 436
updating DB2 catalog 370
BMCZOOM ISPF command 54
bootstrap data set (BSDS)
definition 756
box
definition 756
browse function
definition 756
browsing
column statistics 345
DB2 catalog and BMCSTATS 341
index partition statistics 348
index statistics 346
table space partition statistics 344
table space statistics 342
table statistics 343
BSAM
definition 756
BSDS (Bootstrap Data Set)
definition 756
BSTATAGE exception 453
buffer pool
definition 756
Buffers
definition 756
build JCL
definition 756
build worklist
definition 757
building a Service Action 265
Bypass Exceptions option 480

cache
definition 757
CAF
definition 757
call attachment facility
definition 757
call attachment return codes 684
CANCEL command
definition 757
CANCEL ISPF command 54
CAPS command
definition 758
CARD exception 449
Card trigger 449
CART
definition 758
CAT_LOAD POF keyword 150
catalog
definition 758
Catalog Indirection
definition 758
CATALOG MANAGER
fast path navigation commands 58
CATALOG MANAGER for DB2
definition 758
catalog table
definition 759
Catalog Upd Age trigger 453
CDL (Change Definition Language)
data set, allocating 69
change accumulation file
definition 759
CHANGE MANAGER
administrative product 47
Execution plans 71
fast path navigation commands 58
CHANGE MANAGER for DB2
definition 759
CHANGES command
definition 759
changes to the product 19
char H/W
definition 759
char size
definition 759
CHECK
definition 760
check pending
definition 760
CHECK PLUS for DB2
definition 760
CHECK+_LOAD POF keyword 150
CHECKDOPT POF keyword 151
checkpoint
definition 760
CHEK DA
definition 760
CHEK DA (CKDA) command description 225
CHEK LOB
definition 760
CHEK LOB (CKEK) command description 225
CHGMAN_LOAD POF keyword 151
cipher text
definition 760
clause
definition 761
CLEANUP_RC POF keyword 151
cleanup, data set 79
CLEAR command
definition 761
clear text
definition 761
CLIST
definition 761
CLOB (Character Large Object)
definition 761
close table
definition 761
CLOSE cursor command in ASURXSQL 682
CLUSTERING
definition 761
clustering index
definition 762
CMD command
definition 762
CMDBOT command
definition 762
CMDDDB2 command
definition 762
CMDDDC command
definition 762
CMDMENU command
definition 762
CMDDOPT command
definition 762
CMDTOP command
definition 762
CNTL_DATACLASS POF keyword 151
CNTL_EXPDT POF keyword 151
CNTL_MGMTCLASS 151
CNTL_PREFIX POF keyword 151
CNTL_PRIQTY POF keyword 152
CNTL_RETPD POF keyword 152
CNTL_SECQTY POF keyword 152
CNTL_STORCLASS POF keyword 152
CNTL_UNIT POF keyword 152
CNTLMOUT_DSN POF keyword 152
CNTLMSCH_DSN POF keyword 152
cold start
definition 763
color
definition 763
COLOR command
definition 763
column
definition 763
column condition rules
definition 763
column group statistics
Select COLGROUP field 326
columns
analyze statistics 388
analyze value statistics 390
browse statistics in DB2 catalog 345
definition 763
format of worklists 233, 235
renaming using RENAME COLUMN 376
sampling table 333
update statistics in DB2 catalog 352
updating the column names 376
comma-delimited data
definition 763
comma-separated-value (CSV) data
definition 764
command
definition 764
Command Interfaces
definition 764
commands

- BR14 499
- NEWJ 499
- UD SJ 499
ALTERSEC 225
BMCCOPY 225
BMCCOPYI 225
BMCCPRS 225
BMCDLOAD 225
BMCMOD 225
BMCREORG 225
BMCS STATS 225
BMCTRIG 225
BMCU NLOAD 225
BMCPUR S 225
CHEK DA 225
CHEK LOB 225
descriptions, utilities 225
DSN1COPY 225
fast path navigation 58
FULLCOPY 225
INCRCOPY 225
ISPF 54
LOAD 225
MERGECOPY 225
MODICOPY 225
QUIESCE 225
RECOVER 225
REORG 225
REP REC 225
REP SET 225
REPAIR 225
RESIZE 225
RUNSTATS 225
SQL ALTER SECQTY 541
START DB 225
STOP 225
STOP DB 225
STOP SPACE 225
submitted through BMCTRIG 425
SYNC 225
use in utility jobs 225
worklist 233
comment lines in worklists 233, 237
commit
definition 764
COMMIT command in ASURXSQL 682
commit point
definition 764
common service area
definition 764
compatibility profile  
definition 764  
compatibility with (DB2) utilities 49  
compensated log record  
definition 764  
compensation log record  
definition 765  
compiling SLIBs 667  
completion processing  
definition 765  
component  
definition 765  
compression  
definition 765  
concurrency  
definition 765  
concurrent copy  
definition 765  
condition  
definition 765  
conditional restart  
definition 765  
conditional restart control record  
definition 766  
configuration  
definition 766  
Configuration Advisor  
definition 766  
configuration file  
definition 766  
configuration manager  
definition 766  
confirmation panel, definition 53  
conflict  
definition 766  
conflict manager  
definition 766  
conflict resolution rule  
definition 766  
connection  
Enterprise List 641  
personal list 641  
consistent copy  
definition 767  
conventions, documentation 16  
conversion  
definition 767  
coordinated recovery  
definition 767  
copies  
definition 767  
copy  
object sets 302  
Service Actions 246  
Copy an object set 289  
copy exceptions 455  
COPY exceptions 432  
Copy Pend trigger 455  
copy pending  
definition 767  
COPY PLUS for DB2  
definition 767  
COPY PLUS options, unsupported 49  
COPY#IIC exception 455  
COPY+_LOAD POF keyword 152  
COPYAGE exception 455  
COPYDOPT POF keyword 152  
copying  
threshold 465  
COPYMODS exception 455  
CopyPend  
definition 767  
COPYPEND exception 455  
corrective action  
definition 767  
corrective actions  
Action report 610  
adding 470  
ALTERSEC utility 541  
automating 440  
Cross Reference Report 613  
defining system-level 470  
definition 41  
deleting 470  
editing 470  
Exceptions Report by BMCTRIG Action 620  
for exceptions 431  
generating 442  
in BMCTRIG 469  
managing 469  
setting in BMCTRIG syntax 470  
setting up for REORG 524  
specifying default actions 435  
using BMCTRIG 425  
using priorities to limit 428  
when to use 432  
coupling facility  
definition 768  
CPLAN installation option 206  
CPYEXP_DATACLASS POF keyword 153
Create an object set 289
creating
  actions 265
  Actions 241
  Object Sets outside actions 291, 294
  services 265
  statistics graphs 397
Creating dynamic SQL in a specification 289
cross-system coupling facility
definition 768
CSA
definition 768
CSTATAGE exception 453
CSV data
definition 768
current environment, displaying 60
customizable reports
ASURXLOC external function 691
ASURXLOC return codes 691
ASURXLSP external function 686
ASURXLSP return codes 686
ASURXRPT external function 684
ASURXRPT return codes 684
ASURXSE external function 686
ASURXSE return codes 686
ASURXSQL external function 682
ASURXSQL return codes 683
call attachment return codes 684
call complete sample program 707
customization 674
definitions of external functions 671
descriptions 598
EXEC parms 681
field holders 676
field justification 676
field lines 676
format definition 695
formatting 675
JCL DD statements 681
line numbers 678
MAX(TIMESTMP) 680
number of cursors supported 684
overview 44, 597
page numbers 678
plan variable 677, 679
printing SQL error messages 683
program structure 673
quotation marks in SQL 680
report layout 675
reporting tasks 684
reserved word 676
retrieving the most recent data 680
REXX notes 692
rpt.line# variable 677
rpt.page# variable 677
rpterrm variable 684
running 603
sample program 673, 693, 694
sample report 674
SQL support and limitations 684
ssid variable 677, 679
SYSTERM data set 683
timestamp use 680
value lines 676
variables 677, 680
customizing
  report programs 607
D

D2U (DB2 Utilities Common Code) 48
DAC
  definition 768
Daily History reports
  definition 768
DASD
  definition 768
DASD MANAGER PLUS
  Execution plans 71
DASD MANAGER PLUS (for DB2) 35
DASD MANAGER PLUS database
  definition 769
DASD MANAGER PLUS online utility 39
DASD MANAGER PLUS plans
  ASUvrmDJ 70
  ASUvrmDR 70
  ASUvrmDS 70
  ASUvrmDZ 70
DASD_LOAD POF keyword 154
DASDDOPT POF keyword 154
Data Capture Changes (DCC)
  definition 769
data capture flag
  definition 769
Data Capture None (DCN)
  definition 769
data class
  definition 769
data collection information
  definition 770
Data Collector
  definition 770
data control block
  definition 770
data definition name
  definition 770
data definition name (ddname)
  definition 770
Data Facility Storage Management Subsystem
  definition 770
data migration
  definition 770
data partition
  definition 770
data set
  map 79
Data Set List by Database Name Report 614
data set name pattern user exit
  definition 771
data set redefine user exit
  definition 771
data sets
  $AJXDOC 115
  CDL 69
cleanup 79
  DATAWORK 86
deleting automatically 79
diagnostic output 69
discard 79
error 79
image copy 120
JCL 69
listing by database 338
LOGSORT 86
names specified for worklist JCL 509
permanent work 88
prefixes 88
punch 79
RECV 118
report 69
resolving names 88, 118
sample VSAM object 79
setting default processing options 88
sizing 79, 114, 116
SORTOnnn 79
SORTOUT 79, 88
SORTPnnn 79
SORTWORK 79, 86
SUTnnn 79
SYSC 118
SYSCOPY 88
SYSDISC 88
SYSERR 88
SYSMAP 88
SYSPUNCH 88
SYSREC 88
SYSUT 88
SYSUTnnn 79
unload 79
work 79
worklist 69
WRKnnn 79
data sharing
  definition 771
DATA_PACKER_LOAD POF keyword 154
data-partitioned secondary index (DPSI)
column statistics 352
index partition statistics 355
index statistics 353
interactively 348
overview 340
table columns 357
table space partition statistics 351
table space statistics 349
table space/index columns 357
table statistics 350
using BMCSTATS 363
using BMCUPRS 370
DB2 command
definition 773
DB2 Component Services (DBC)
definition 773
DB2 objects
analyzing trends 379
creating a statistics graph 397
customizable reports 598
deleting BMC Software statistics 377
displaying 335
displaying DB2 catalog and BMCTSTATS on
the same panel 341
including or excluding from service generation
279
listing 335
monitoring with BMCTRIG 429
running BMCSTATS to populate 313
simulating changes 405
simulating changes to
405
statistics tools 313
updating the DB2 catalog 348
wildcard facility 279
DB2 Product Configuration
definition 773
DB2 RTS option 480
DB2 Solution Common Code (SCC) 48
DB2 utilities
ALTERSEC 541
DB2CAT installation option 206
DB2EXIT POF keyword 155
DB2LOAD POF keyword 155
DB2LOAD statement/parameter
definition 773
DB2NTRY installation option 206
DB2WAIT installation option 206
DBA
definition 773
DBADM
DEF_GDG_LIMIT POF keyword 155
DEF_GDG_NOSCR POF keyword 155
DEF_GDG2_LIMIT POF keyword 155
default options module
definition 775
default options. See installation options 124
default value
definition 775
defaults, setting product options
data sets 88
debugging 95
generation data groups 93
JCL jobcard 75
JCL static data sets 79
JCL STEPLIBs 78
product options file 103
SORTWORK data sets 86
tapes 84
utility module names 97
defaults, setting user options
BMCTRIG Control-M options 101
data sets 88
debugging 93
general 74
generation data groups 93
non-worklist JCL 98
user variables 102
definitions
definition 773
DBCLOB (Double-Byte Character Large Object)
definition 773
DBCS
definition 773
DBCS command
definition 774
DBCTRL
definition 774
DBD
definition 774
DBID
definition 774
DBMS
definition 774
DBNAME
definition 774
DBRM
definition 774
DCB
definition 774
DD name
definition 774
DD statement
definition 775
DD statements
adding for BMC required services 260
DD Statement List panel, services 260
for report programs 681
JCL DD statement dispositions for restarts 533
JCL DD statement dispositions for startover 533
JCL Generation component 114
JCL processing 534
omitting in JCL 535
preparing a customizable report in batch 597
processing of GDG data sets 534
user-defined programs 228
Utility Copy 120
Utility Dual Copy 120
Utility Dual Recovery Site Image Copy 120
Utility Recovery Site Image Copy 120
definition 775
ddname
definition 775
DEADLINE Shrlevel option 524
DEBUG command
definition 775
debugging, setting default processing options 95
DEF_GDG_BASE POF keyword 155
action POF 125
Action Report 598
ALTERSEC 225, 541
Append inserts trigger 449
Average Length of RID Chain Report 548
Average Monthly Growth Report 598
BMCCOPY 225
BMCCOPYI 225
BMCCPRS 225
BMCMLOAD 225
BMCMOD 225
BMCREORG 225
BMCCSTATS 225, 317
BMCCSTATS Age trigger 453
BMCTRG 225
BMCULOD 225
BMCUPRS 225
Card trigger 449
Catalog Upd Age trigger 453
CHEK DA 225
CHEK LOB 225
Copy Pend trigger 455
Corrective Action Cross Reference Report 598
Data Set List by Database Name Report 598
Database Space Detail Report 598
Database Space Trend Report 598
Database Storage Utilization Report 598
Dataset Extents trigger 449
Dirty trigger 455
DSEXTENTS 517
DSN1COPY 225
DtyIncremt trigger 455
Event Recording Report 548
Exceptions Report by BMCTRIG Action 598
Extents trigger 449
FARINDREF 517
Faroff trigger 449
FAROFFPOS 517
Full Copy Age trigger 455
FULLCOPY 225
Image Copy Age trigger 455
INCRCOPY 225
Index Analysis Report 548
Index Clustering Analysis Report 548
Index Clustering Report 598
Index Combination Statistics Report 548
Index Key Analysis Report 548
Index Leaf Distribution Report 548
Indexspace Space Detail Report 548
LEAFDIST 517
Leafdist trigger 449
LEVELMIN 517
Levels trigger 449
LOAD 225
Load after copy trigger 455
Load after stats trigger 453
LOB Freespace trigger 449
LOB ORGRATIO trigger 449
Mass del reorg trigger 449
Mass del stats trigger 453
MERGCOPY 225
MODICOPY 225
Modified Table Space Pages Report 598
Mods since copy trigger 455
Mods since reorg trigger 449
Monthly Detail by Tablespace Report 548
NEARINDREF 517
NEAROFFPOS 517
Nostats trigger 453
NumIncremt trigger 455
Page Update Analysis Report 548
Pct Dropped Rows trigger 449
PctActivHi trigger 449
PctActivLo trigger 449
PCTCLUS 517
PctClus trigger 449
processing options for BMCTRIG 440
QUIESCE 225
Rebd after copy trigger 455
Rebd after stats trigger 453
RECOVER 225
REORDEL 517
REORG 225
Reorg after copy trigger 455
Reorg after stats trigger 453
Reorg Pend trigger 449
Reorgspace trigger 449
REORINS 517
REORLEAF 517
REORMODS 517
REORPEND 517
REORSPACE 517
REP REC 225
REP SET 225
REPAIR 225
RESIZE 225
RUNSTATS 225, 317
Service Action generation 231
Space Estimation Report 598
Index
DOMBLOD1
definition 777

DOMBPLOG
definition 777

DOMBRPT1
definition 777

DOMBSWIT
definition 777

DOMEXIT1
definition 777

DOMEXIT2
definition 777

DOMEXIT4
definition 778

DOMEXIT6
definition 778

DOMPLEX option set
definition 778

DOPTs
definition 778
double-byte character set
definition 778

DOWN command
definition 778

DPNAM installation option 206
DRAIN Shrlevel option 524
DRAIN_WAIT Shrlevel option 524
dropping and re-creating objects 376
DROPR_NOIC POF keyword 158
DSETPASS
definition 778

DSEXTENT exception 449
DSEXTENTS exception 517

DSN1COPY
definition 779

DSN1COPY (DSN1) command description 225
DSNCHECK44 POF keyword 159
DSNTIAD_PLAN POF keyword 159
DtyIncremt trigger 455

DUMP command
definition 779
dynamic allocation
data set prefixes 88
definition 779
diagnostic output data set 69
of output for (UNLOAD PLUS) 49
temporary work files 524

E

EAP installation option 206
ECKD device
definition 779
edit JCL
definition 779
edit list, definition 53
Edit object set 289
edit worklist
definition 779
editing
Action Services 265
actions 265
corrective action 470
information-only fields 51
input fields 51
JCL generation options 124
object sets 298, 301
object-action priority 473
secondary quantity 544
Service Actions 244
services 265
threshold 463

EDITPROC
definition 779

EIP installation option 206
ELIMDUPACTS option 433
ELIMDUPOPTS option 433

enclosure characters
definition 779

ENCODING
definition 780

encrypted text
definition 780

END command
definition 780

END ISPF command 54
Engine Level Mask (ELM)
definition 780

ENQS command
definition 780

ENTER ISPF command 54
Enterprise Connection List (ECL) 641
enterprise connections, setting up in Export 649
entry field delimiter
definition 780

environment, displaying current 60
EPP installation option 206
ERR_DATACLASS POF keyword 159
ERR_DATACLASS_ALT POF keyword 159
ERR_EXPDT POF keyword 159
ERR_MGMTCLASS POF keyword 159
ERR_MGMTCLASS_ALT POF keyword 159
ERR_PREFIX POF keyword 160
ERR_PRIQTY POF keyword 160
ERR_RETPD POF keyword 160
ERR_SECQTY POF keyword 160
ERR_STORCLASS POF keyword 160
ERR_STORCLASS_ALT POF keyword 160
ERR_THRESH POF keyword 160
ERR_UNIT POF keyword 160
ERR_UNIT_ALT POF keyword 160
error data set 79
error messages
  ASURXLOC 691
  ASURXLSP 686
  ASURXRPT 684
  ASURXSE 686
  ASURXSQL 683
estimate source
definition 780
evaluating BMCTRIG exceptions, overview 442
Event Recording Report 553
events log
definition 781
EVENTS table
definition 781
exception
definition 781
exception definitions, specifying to Export 657
exception processing
definition 781
exception status
definition 781
exceptions
APPNDINS 449
AREOPEND 449
ARERPEND 449
BSTATAGE 453
CARD 449
copy 455
COPY#IIC 455
COPYAGE 455
COPYMODS 455
COPYPEND 455
CSTATAGE 453
DIRTY 455
DIRTYIC 455
DSEXTENT 449
EXTENTS 449
FARIND 449
FAROFF 449
FICAGE 455
IXDIRTY 455
IXICAGE 455
LEAFDIST 449
LEAFFOFF 449
LEVELINC 449
LEVELMIN 449
LEVELS 449
LOADCOPY 455
LOADSTAT 453
LOBFRSPC 449
LOBORGR 449
MINSPACE 449
NONUNIFORM 453
NORTSCPY 455
NORTSSTAT 453
NOSTATS 453
PACTHI 449
PACTLO 449
PACTCLUS 449
PCTDROP 449
PSEUDODL 449
REBDSTAT 453
REORCOPY 455
REORDELS 449
REORDSPC 449
reorganization 449
REORGINS 449
REORGLOB 449
REORLEAF 449
REORMDEL 449
REORMODS 449
REORPEND 449
exceptions report
   definition 781
EXCEPTIONS2 table
   definition 781
excluding objects
    from sets 298
    from trigger job 521
EXCP
   definition 782
EXEC_LOAD POF keyword 161
EXECUTE command in ASURSQL 682
Execution
   definition 782
Execution component
    packages 71
    plans, list of 71
execution mode
   definition 782
execution, worklist
    cleanup job step for data sets 79
    deleting data sets automatically 79
EXIT command
   definition 782
exit routine
   definition 782
EXPAND command
   definition 782
EXPAND statement/parameter
   definition 782
Explain
   definition 782
Export
   accessing 648
   action definitions 654
   allowing default service syntax changes 659
   ASU_XP_UIMSRVHOST POF keyword 642
   ASU_XP_UIMSRVPORT POF keyword 642
   authorizations 642
   connection recommendations 641
   dependent objects examples 640
   deploying the definitions 659
   enterprise connection list (ECL) 641
   exception definitions 657
   how it works 637
   launching 648
   object set definitions 656
   object-action priority definitions 656
   overview 45
   overwriting existing rows 659
   personal connection list (PCL) 641
   POF keywrods 642
   preparing your environment 647
   referenced objects examples 639
   report, reviewing statuses 661
   SAF authority 644
   selecting destinations to export 659
   service definitions 655
   setting the primary UIM server host name 642
   setting the primary UIM server port 642
   setting up destination connections 651
   setting up enterprise connections 649
   TCP/IP access 642
   UIM Server access 642
   verifying the list to export 658
   viewing status of export job 659
EXPORT command
   definition 783
EXPORT files
   definition 783
exporting
   action definitions 654
   exception definitions 657
   object definitions 637
   object set definitions 656
   object-action definitions 656
   service definitions 655
EXTENTS exception 449
Extents trigger 449
external functions
F

F2 (SPLIT) 54
F9 (SWAP) 54
fallback
definition 783
FARIND exception 449
FARINDREF exception 517
FAROFF exception 449
Faroff trigger 449
FAROFFPOS exception 517
fast path navigation commands 58
FASTSWITCH SHRLEVEL option 524
FCPY_DATACLASS POF keyword 161
FCPY_EXPDT POF keyword 161
FCPY_MGMTCLASS POF keyword 161
FCPY_PREFIX POF keyword 161
FCPY_PRIQTY POF keyword 161
FCPY_RETPD POF keyword 161
FCPY_SECQTY POF keyword 162
FCPY_STORCLASS POF keyword 162
FCPY_UNIT POF keyword 162
features
analyzing trends 44
collecting and managing statistics 39
customizing reports 44
deploying object definitions 45
estimating space 39, 44
generating utilities 39
setting and analyzing thresholds 41
unsupported 45
FICAGE exception 455
field
holder 676
information-only 51
input 51
justification in customizable reports 676
line 676
file structure in worklists 233, 235
file tailoring 114, 115, 666
FILT_DATACLASS POF keyword 162
FILT_EXPDT POF keyword 162
FILT_MGMTCLASS POF keyword 162
FILT_PREFIX POF keyword 162
FILT_PRIQTY POF keyword 163
FILT_RETPD POF keyword 163
FILT_SECQTY POF keyword 163
FILT_STORCLASS POF keyword 163
FILT_UNIT POF keyword 163
filter
definition 783
filter association
definition 783
filter option set
definition 783
filter predicate
definition 784
FIND command
definition 784
FKA command
definition 784
FKEY0MSG option 481
FKEYs command
definition 784
flow diagrams
ALTERSEC 541
evaluate BMCTRIG exceptions 443
evaluate exceptions and generate utilities in BMCTRIG 442
evaluate UTA exceptions in BMCTRIG 444
first-time utility job 231
generate BMCTRIG jobs without evaluating exceptions 443
generating standard JCL 497
regenerated utility job 231
foreign key, data set sizing 231
FORWARD command
definition 784
forward recovery
definition 784
Free Form filter
definition 784
free space
definition 784
FREQVAL installation option 206
full copy
definition 785
Full Copy Age trigger 455
FULLCOPY
definition 785
FULLCOPY (FULL) command description 225
functions on main menu 55

G

GB
  definition 785
GDG (generation data group)
  and user options 118
  setting default processing options 93
  specifying in copy DD options panel 120
  symbolic variable 118
GDG (Generation Data Group)
  definition 785
GDG symbolic variable 88
GDG_MODEL POF keyword 163
GDG(generation data group)
  specifying 88
  symbolic variable 88
  using for data sets 88
GDGDEF installation option 206
GDGLIM installation option 206
Generate priority option 483
generating
  JCL for GDGs 117
  standard JCL, flow diagrams 497
  trigger job with exceptions 476
  utility job 248
generation data group. See GDG
generation data group (GDG)
  definition 785
generation data group. See GDG (generation data group), 118
Generation Data Groups 114
global options
  definition 785
graphs
  creating 397
  display options 400
  set options 400
  statistics 397
  types 397
group attachment name
  definition 785
group buffer pool
  definition 785
group member name
  definition 786
Group objects option 483
grouped services
examples 436
  general 436
  not logged objects 436
  object-action priorities 434
  processing 436
  guidelines for triggering REORG 517

H

HASHFAIL POF keyword 163
HASHWARNRC POF keyword 163
HELP command
  definition 786
HELP ISPF command 54
HILITE command
  definition 786
historical database 318
  definition 786
historical reports
  definition 786
HLQ.UDBCNTL members
  product options file 125
HOME command
  definition 786
hyperlink
  definition 786

I

I/O
  definition 787
ICF
  definition 787
IDCAMAS
  definition 787
Identify actions associated with an object set 289
identifying actions associated with object sets 300
IEDIT command
  definition 787
IFCID
  definition 787
ignore status
  definition 787
image copy
  definition 787
Image Copy Age trigger 455
image copy data sets 120
Imagecopy
  definition 787
IN-DB2 time
  definition 788
IN-SQL time
  definition 788
INCLUDE_SYSPRIN2 POF keyword 164
including objects in sets 298
INCRCOPY
  definition 788
INCRCOPY (INCR) command description 225
incremental copy
  definition 788
independent filter
  definition 788
index
  analyze partition page group statistics 394
  analyze partition statistics 392
  analyze statistics 390
  browse partition statistics in DB2 catalog 348
  browse statistics in DB2 catalog 346
  graph summary 397
  reorganizing 517
  sampled statistics, updating the DB2 catalog 206
  simulating changes 405
  update partition statistics in DB2 catalog 355
  update statistics in DB2 catalog 353
Index
  definition 788
  Index Analysis Report 556
  Index Cardinality
    definition 788
  index columns, updated in the DB2 catalog 357
  Index Combination Statistics Report 562
  index key
    definition 789
  Index Key Analysis Report 565
  Index Leaf Distribution Report 567
  Index Pages
    definition 789
  index partition
    definition 789
  Index Partition Cardinality
    definition 789
  Index Partition Extents
    definition 789
  Index Partition Leaf Distribution
    definition 789
  Index Partition Page Group
    definition 789
  Index Partition Pages
ATBWORKAREA 206
BMCSYNC 206
CPLAN 206
DATACLASS 206
DATE 206
DB2CAT 206
DB2NTRY 206
DB2WAIT 206
DPNAM 206
EAP 206
EIP 206
EPP 206
FREQVAL 206
GDGDEF 206
GDGLIM 206
graphic explanation 124
ISPTLIB 206
JC1, JC2, JC3, JC4, JC5 206
JCLCLEAN 206
JDSNE 206
JPLAN 206
listing 205
LOCATION 206
MGMTCLAS 206
notation 204
OPNDB2ID 206
OPTIMIZECOMMIT 206
OSAUTHCHK 206
overriding default values 124
POFDS 206
PRODUCT 206
QUIESCEINTERVAL 206
RECVMAX 206
RECVMAXU 206
RECVPREF 206
RECVPS 206
RECVSS 206
RECVUNIT 206
RPLAN 206
SL1 206
SL2 206
SL3 206
SL4 206
SL5 206
SPLAN 206
STATAUTH 206
STORCLAS 206
SWPS 206
SWSS 206
SWU 206
SYSCMAX 206
SYSCMAXU 206
SYSCPS 206
SYSCSS 206
SYSCUNIT 206
SYSRMAX 206
SYSRMAXU 206
SYSRPREF 206
SYSRPS 206
SYRSRS 206
SYSRUNIT 206
SYSTRICS 206
SYSTYPE 206
SZDEVT 206
TAPE1, TAPE2, TAPE3 206
UPDCATIXS 206
UPDNUCAT 206
used in utilities 203
WDC 206
WDSN 206
WMC 206
WPS 206
WSC 206
WSS 206
WU 206
ZPLAN 206
installation options module
ASUDOPTS 205
definition 791
using ampersands 204
installation SYSADM
definition 791
Instant Snapshot
definition 791
instantaneous reports
definition 791
integrated catalog facility
definition 791
Interactive System Productivity Facility
definition 791
interface
commands 54
field types 51
panel types 53
interim worklist 499
internal KB switch 603
internal resource lock manager
definition 792
INTERVAL statement/parameter
definition 792
JCL Generation
  data set sizing 116
  RUNSTATS utility 116
  VSAM object sampling 116
JCL Generation (JCLGEN)
  definition 793
JCL Generation component 114
JCL job card, setting default processing options 75
JCL job statement, defaults 75
JCL static data sets, setting default processing options 79
JCL STEPLIBs, setting default processing options 78
JCL variable display
  definition 793
JCLCLEAN installation option 206
JCLCLEANUP POF keyword 165
JCLGEN
  definition 793
JCLLIB POF keyword 165
JDSNE installation option 206
JES
  definition 793
JES3 POF keyword 165
  job
    definition 793
  job card exit in interim worklist 501
job control language (JCL)
  definition 792
job library
  definition 793
job profile
  definition 793
job status
  definition 794
JOB table
  definition 794
JOB_INCLUDE_MEMBER POF keyword 165
JOBCARD1 POF keyword 165
JOBCARD2 POF keyword 165
JOBCARD3 POF keyword 165
JOBCARD4 POF keyword 165
JOBCARD5 POF keyword 165
JOBLIB
  definition 794
JPLAN installation option 206
justify
  definition 794

J

J1 installation option 206
J2 installation option 206
J3 installation option 206
J4 installation option 206
J5 installation option 206
JCL
  data set, allocating 69
  DD processing, GDG data sets 534
  DD statement, restarts 533
  DD statement, startovers 533
  flow diagram for standard 497
  for sample BMCSTATS job 363
  generated standard, example 502
  standard benefits 497
  worklist benefits 507
  worklist format 507
JCL DSN
  definition 792
JCL generation
  and restarts 532
  data set names required for worklist 509
  GDGs 117
  ISPF file tailoring 114
  options for worklist 509
  options in BMCTRIG 521
  overview 237
  worklist 237

I

IOALOAD1 POF keyword 164
IOALOAD2 POF keyword 164
IPL
  definition 792
IRLM
  definition 792
ISPF
  AJXPODAT macro 104
  commands, common 54
  definition 792
  file tailoring 666
  file tailoring, used by JCL Generation 114, 115
ISPF skeletons
  definition 792
ISPTLIB installation option 206
IX Copy option 482
IX Escalate option 482
IXDIRTY exception 455
IXICAGE exception 455
JC1 installation option 206
JC2 installation option 206
JC3 installation option 206
JC4 installation option 206
JC5 installation option 206
JCL
  data set, allocating 69
  DD processing, GDG data sets 534
  DD statement, restarts 533
  DD statement, startovers 533
  flow diagram for standard 497
  for sample BMCSTATS job 363
  generated standard, example 502
  standard benefits 497
  worklist benefits 507
  worklist format 507
JCL DSN
  definition 792
JCL generation
  and restarts 532
  data set names required for worklist 509
  GDGs 117
  ISPF file tailoring 114
  options for worklist 509
  options in BMCTRIG 521
  overview 237
  worklist 237

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Index 871
K

K
  definition 794
KB
  definition 794
KEEPIUNCTIONARY Reorg option 524
key
  definition 794
key data set
  definition 794
key format
  definition 795
key length
  definition 795
key stores
  definition 795
KEYS command
  definition 795
keywords

ALTERSEC 225
BMCCOPY 225
BMCCOPYI 225
BMCCPRS 225
BMCCLOAD 225
BMCMDT 225
BMCREORG 225
BMCSATION 225, 320
BMCCUNLOD 225
BMCCUPRSA 225
CHEK DA 225
CHEK LOB 225
DSIICOPY 225
FULLCOPY 225
INCRCOPY 225
LOAD 225
MERGECOPY 225
MODICOPY 225
QUIESCE 225
RECOVER 225
REORG 225
REP REC 225
REP SET 225
REPAIR 225
RESIZE 225
RUNSTATS 225
START DB 225
STOP 225
STOP DB 225
STOSPACE 225
SYNC 225
use in utility jobs 225

L

layout of reports 598
Leaf distribution
  definition 795
LEAFDIST exception 449, 517
Leafdist trigger 449
LEAFFOFF exception 449
LEFT command
  definition 795
LEFT ISPF command 54
length
  definition 795
Level
  definition 796
<table>
<thead>
<tr>
<th>Definition</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVELINC</td>
<td>796</td>
</tr>
<tr>
<td>LEVELINC exception</td>
<td>449</td>
</tr>
<tr>
<td>LEVELMIN</td>
<td>796</td>
</tr>
<tr>
<td>LEVELMIN exception</td>
<td>449, 517</td>
</tr>
<tr>
<td>LEVELS</td>
<td>796</td>
</tr>
<tr>
<td>LEVELS exception</td>
<td>449</td>
</tr>
<tr>
<td>Levels trigger</td>
<td>449</td>
</tr>
<tr>
<td>limit key</td>
<td>796</td>
</tr>
<tr>
<td>lines</td>
<td>796</td>
</tr>
<tr>
<td>LINESPP statement/parameter</td>
<td>796</td>
</tr>
<tr>
<td>list of utilities and commands</td>
<td>225</td>
</tr>
<tr>
<td>LISTDEF_DSN POF keyword</td>
<td>165</td>
</tr>
<tr>
<td>listing data sets by database</td>
<td>338</td>
</tr>
<tr>
<td>LL_CLIB POF keyword</td>
<td>165</td>
</tr>
<tr>
<td>LL_CLIB2 POF keyword</td>
<td>165</td>
</tr>
<tr>
<td>LL_CLIB3 POF keyword</td>
<td>165</td>
</tr>
<tr>
<td>LL_CLIB4 POF keyword</td>
<td>165</td>
</tr>
<tr>
<td>LL_CLIB5 POF keyword</td>
<td>165</td>
</tr>
<tr>
<td>LL_LINK POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_LINK2 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_LINK3 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_LINK4 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_LINK5 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_MLIB POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_MLIB2 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_MLIB3 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_MLIB4 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_MLIB5 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_PLIB POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_PLIB2 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_PLIB3 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_PLIB4 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_PLIB5 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_SLIB POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_SLIB2 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_SLIB3 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_SLIB4 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_SLIB5 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_TLIB POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_TLIB2 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_TLIB3 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_TLIB4 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_TLIB5 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_XML POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_XML2 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_XML3 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_XML4 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LL_XML5 POF keyword</td>
<td>166</td>
</tr>
<tr>
<td>LLQ POF keyword</td>
<td>167</td>
</tr>
<tr>
<td>LOAD</td>
<td>796</td>
</tr>
<tr>
<td>Load after copy trigger</td>
<td>455</td>
</tr>
<tr>
<td>Load after stats trigger</td>
<td>453</td>
</tr>
<tr>
<td>LOAD command description</td>
<td>225</td>
</tr>
<tr>
<td>LOAD+ _LOAD POF keyword</td>
<td>167</td>
</tr>
<tr>
<td>LOADCOPY POF exception</td>
<td>455</td>
</tr>
<tr>
<td>LOADDOPT POF keyword</td>
<td>167</td>
</tr>
<tr>
<td>LOADPLUS for DB2</td>
<td>797</td>
</tr>
<tr>
<td>LOADSTAT exception</td>
<td>453</td>
</tr>
<tr>
<td>LOB (Large Object)</td>
<td>797</td>
</tr>
<tr>
<td>LOB Freespace trigger</td>
<td>449</td>
</tr>
<tr>
<td>LOB ORGRATIO trigger</td>
<td>449</td>
</tr>
<tr>
<td>LOB SYSREC data set</td>
<td>88</td>
</tr>
<tr>
<td>LOB table space</td>
<td>797</td>
</tr>
<tr>
<td>LOBFRSPC exception</td>
<td>449, 517</td>
</tr>
<tr>
<td>LOBORGR exception</td>
<td>449</td>
</tr>
<tr>
<td>local</td>
<td>797</td>
</tr>
<tr>
<td>local site</td>
<td>797</td>
</tr>
<tr>
<td>LOCATE command</td>
<td>797</td>
</tr>
<tr>
<td>LOCATION installation option</td>
<td>206</td>
</tr>
<tr>
<td>lock</td>
<td>797</td>
</tr>
<tr>
<td>lock duration</td>
<td>798</td>
</tr>
<tr>
<td>locking</td>
<td>798</td>
</tr>
<tr>
<td>log</td>
<td>798</td>
</tr>
<tr>
<td>log apply</td>
<td>798</td>
</tr>
<tr>
<td>log inventory</td>
<td>798</td>
</tr>
<tr>
<td>log mark</td>
<td>798</td>
</tr>
<tr>
<td>log record sequence number (LRSN)</td>
<td>798</td>
</tr>
</tbody>
</table>
LOG Reorg option 524
log scan
definition 798
logging environment modeling tool
definition 799
logical log
definition 799
logical log control file
definition 799
logical log data file
definition 799
logical page list
definition 799
logical partition
definition 799
LOGSET
definition 799
LOGSORT
definition 800
LOGSORT data set 86
LOGWK_NBR POF keyword 167
LOGWK_UNIT POF keyword 167
long object name fields 59
LONGLOG Shrlevel option 524
loop counter 704
LPL
definition 800
LVL
definition 800

M
macros
AJXPODAT 104
main menu
functions 55
Main Menu
description 51
fields 51
MainView for DB2
definition 800
MainView for DB2 - Data Collector (DC)
definition 800
managing
Service Actions 231
statistics 355
map data set 79
MAP_DATACLASS POF keyword 167
MAP_DATACLASS_ALT POF keyword 167
MAP_EXPDT POF keyword 168
MAP_MGMTCLASS POF keyword 168
MAP_MGMTCLASS_ALT POF keyword 168
MAP_PREFIX POF keyword 168
MAP_PRIQTY POF keyword 168
MAP_RETPD POF keyword 168
MAP_SECQTY POF keyword 168
MAP_STORCLASS POF keyword 168
MAP_STORCLASS_ALT POF keyword 168
MAP_THRESH POF keyword 169
MAP_UNIT POF keyword 169
MAP_UNIT_ALT POF keyword 169
MAPPINGTABLE Shrlevel option 524
margin
definition 800
markers
definition 800
Mass del reorg trigger 449
Mass del stats trigger 453
Master Profile
definition 800
Max reorg parts option 483
max row length
definition 801
MAX_CYL POF keyword 169
MAX_PRIQTY POF keyword 169
MAX_SECQTY POF keyword 169
MAX_UNITCNT POF keyword 169
Maximum pqty option 486
MAXRO Shrlevel option 524
MAXROWS
definition 801
MB
definition 801
MEMLIMIT POF keyword 169
MEMLIMIT system parameter 222
memory requirements 222
menu
definition 801
MENU command
definition 801
MERCOPY (MERG) command description 225
MERCOPY
definition 801
message files 107
MGMTCLAS installation option 206
migration
definition 801
MINSPACE exception 449
modeling tool
definition 802
MODICOPY (MODI) command description 225
Modified Table Space Pages Report 623
modified-page indicator
definition 802
MODIFY
definition 802
MODIFY command
definition 802
Mods since copy trigger 455
Mods since reorg trigger 449
Monthly Detail by Tablespace Report 571
Most Frequent Value display
definition 802
MSGLEVEL option 481
multiple virtual storage (MVS)
definition 802
multiple-line commands in worklists 233, 236
multitasking, using tape stacking 84
MVS
definition 802
MVS/ESA
definition 802
MVS/XA
definition 802

naming objects 69
navigating the panels 55
NEARINDREF exception 517
NEAROFFPOS exception 517
NEXT command
definition 803
Next Generation Logger (NGL)
definition 803
NEXT ISPF command 54
non-worklist JCL 98
nonclustering index
definition 803
noncompensated log record
definition 803
nonpartitioned index
definition 803
nonpartitioned table space
definition 803
NonUniform
definition 803
NONUNIFORM exception 453
NORTSCPY exception 455
NORTSSTAT exception 453
NOSTATS exception 453
Nostats trigger 453
NOTIFY option 481
NumIncremt
definition 804
NumIncremt trigger 455

O

OBID
definition 804
object
definition 804
Object
definition 804
object list
using DB2 335
object resolution 279
object set
definition 804
object set definitions, specifying to Export 656
object set patterns
resulting object sets 281
types 281
object sets
advantages 280
copying 302
deleting 303
example 285
identifying associated actions 300
include/exclude objects 298
renaming 301
specification panel 285
using wildcard 280
Object Sets
controlling access to 311
creating outside of actions 291, 294
object-action definitions, specifying to Export 656
objects
adding a new priority 473
altering and re-creating 376
BMCSTATS for 380
changing the pattern 265
collecting statistics on when changed 220
deleting priorities 473
displaying DB2 catalog and BMCSTATS statistics for 341
displaying DB2 lists 335
dropping and re-creating 376
editing priority 473
establishing priorities for when a corrective action is triggered 434
evaluating 440
grouping 265
minimum object size, specifying 332
naming 69
preventing from running out of extents 541
restricting sampling by type 333
sampling large 332
security 69
simulating changes to 405
sizing 332
specifying which ones to monitor 476
utilities operating concurrently on 313
viewing exceptions by 514
wildcard characters, using 59
ongoing processing
definition 804
online Help 59
online reporting
definition 804
online utility 39
definition 805
online utility generation 39
OPEN cursor command in ASURSQL 682
OPERTUNE for DB2
definition 805
OPNDB2ID
definition 805
OPNDB2ID installation option 206
OPTIMIZECOMMIT installation option 206
options
BMCSTATS 320
BMCTRIG Control-M 101
BMCTRIG processing, introduction 440
data sets 88
debugging and display 95
ELIMDUPACTS 433
ELIMDUPOPTS 433
generating worklist JCL 509
generation data group 93
graphic display 400
JCL job cards 75
JCL static data sets 79
JCL STEPLIBs 78
non-worklist JCL 98
panel display 74
printing graphs 400
product options file 103
setting general 74
SORTWORK data sets 86
system information 74
tapes 84
user variables 102
utility generation and processing 74
utility module names 97
OPTIONS command
definition 805
order
definition 805
order of use for utility information sources 203
organization of product panels 55
ORGRATIO exception 517
orient
definition 805
ORTPARM_DSN POF keyword 170
OS/390
definition 805
OSAUTHCHK installation option 206
OUTLIM statement/parameter
definition 806
output data sets for reports 609
output group
definition 806
Overriding options 114
overtime mode
definition 806

P
package
packages, Execution component 71
PACTHI exception 449
PACTLO exception 449
page  definition 806
page fixing  definition 806
PageGroup  definition 806
PANELID command  definition 807
panels
Action List 244
BMCSTATS 320
common elements 53
Main Menu 51
navigating 55
object set specification 285
organization 55
types 53
Parallel index Sort option 524
parameter  definition 807
parameters  definition 807
parse  definition 807
partial recovery  definition 807
PARTITION  definition 807
partition-by-growth table space  definition 808
partitioned data set (PDS)  definition 808
partitioned index  definition 808
partitioned table space  definition 808
partitioning index  definition 808
PartLvl  definition 808
PARTLVL option 483
password activation deadline  definition 808
PCPY1_DATACLASS POF keyword 170
PCPY1_DATACLASS_ALT POF keyword 170
PCPY1_EXPDT POF keyword 170
PCPY1_MGMTCLASS POF keyword 170
PCPY1_MGMTCLASS_ALT POF keyword 170
PCPY1_PRIQTY POF keyword 170
PCPY1_RETPD POF keyword 170
PCPY1_SECQTY POF keyword 170
PCPY1_STACK POF keyword 171
PCPY1_STORCLASS POF keyword 171
PCPY1_STORCLASS_ALT POF keyword 171
PCPY1_SUPPRESS_SUFF POF keyword 88, 171
PCPY1_THRESH POF keyword 171
PCPY1_UNIT POF keyword 171
PCPY1_UNIT_ALT POF keyword 171
PCPY2_DATACLASS POF keyword 172
PCPY2_DATACLASS_ALT POF keyword 172
PCPY2_EXPDT POF keyword 172
PCPY2_MGMTCLASS POF keyword 172
PCPY2_MGMTCLASS_ALT POF keyword 172
PCPY2_PRIQTY POF keyword 172
PCPY2_RETPD POF keyword 172
PCPY2_SECQTY POF keyword 172
PCPY2_STACK POF keyword 173
PCPY2_STORCLASS POF keyword 173
PCPY2_STORCLASS_ALT POF keyword 173
PCPY2_SUPPRESS_SUFF POF keyword 88, 173
PCPY2_THRESH POF keyword 173
PCPY2_UNIT POF keyword 173
PCPY2_UNIT_ALT POF keyword 173
Pct Dropped Rows trigger 449
PctActivHi  definition 809
PctActivHi trigger 449
PctActivLo  definition 809
PctActivLo trigger 449
PCTCLUS exception 449, 517
PctClus trigger 449
PCTDROP exception 449
PCTFREE  definition 809
PDS  definition 809
percentages, specifying sample 333
Performance Advisor 729
Performance Advisor, how it works 731
performance analysis  definition 809
permanent work data sets 88
Personal Connection List (PCL) 641
point-in-time recovery
definition 811
Pool Advisor for DB2
definition 811
PQTY
definition 811
Pqty % if max option 486
PRE_JOBSTEP_INCLUDE POF keyword 174
pre-filtering
definition 811
predicate
definition 811
prefixes 88
PREFORMAT Reorg option 524
PREV ISPF command 54
primary authorization ID
definition 811
primary copy
definition 811
primary index
definition 811
primary key
definition 812
priorities
accessing system object-action 473
adding a new object-action 473
defining system-level object-action 473
deleting object-action 473
editing object-action 473
limiting corrective actions 428
object-action 434
ordering objects when generating maintenance jobs 428
threshold 430, 434
PRIQTY
definition 812
PROC
definition 812
PROC_BMCHECK_NAME POF keyword 174
PROC_BMCHECK_STEP POF keyword 174
PROC_BMCCOPY_NAME POF keyword 174
PROC_BMCCOPY_STEP POF keyword 174
PROC_BMCCPRS_NAME POF keyword 174
PROC_BMCCPRS_STEP POF keyword 175
PROC_BMCLOAD_NAME POF keyword 175
PROC_BMCLOAD_STEP POF keyword 175
PROC_BMCRECOVER_NAME POF keyword 175
PROC_BMCRECOVER_STEP POF keyword 175
PROC_BMCREORG_NAME POF keyword 175
PROC_BMCREORG_STEP POF keyword 176
proc_bmcstats_name pof keyword
proc_bmcstats_step pof keyword
proc_bmcstop_name pof keyword
proc_bmcstop_step pof keyword
proc_bmctrig_name pof keyword
proc_bmctrig_step pof keyword
proc_bmcunload_name pof keyword
proc_bmcunload_step pof keyword
proc_bcmcuprs_name pof keyword
proc_bcmcuprs_step pof keyword
proc_dsncopy_name pof keyword
proc_dsncopy_step pof keyword
proc_dsnullib_name pof keyword
proc_dsnullib_step pof keyword
proc_gen_set_var pof keyword
proc_idcams_name pof keyword
proc_idcams_step pof keyword
proc_iefbr14_name pof keyword
proc_iefbr14_step pof keyword
proc_tso_name pof keyword
proc_tso_step pof keyword
proc_user_def_step pof keyword
proc_user_defined pof keyword
product administrator
definition 812
product changes 19
product fixes, displaying 60
product functionality 35
product installation option 206
product integration
administrative products for (DB2) 47
release level compatibility 49
product option file (POF)
definition 812
product options file. See POF
product options file. See POF 125
products command
definition 812
profile
definition 813
Pseudo Deleted Key trigger 449
PSEUDODL exception 449
PSID
definition 813
publications, related 15
punch data set 79
punch_dataclass 180
punch_dataclass 180
punch_expdt pof keyword 180
punch_mgmtclass pof keyword 180
punch_managementclass pof keyword 180
punch_prefix pof keyword 180
punch_priqty pof keyword 180
punch_retpd pof keyword 180
punch_seqqty pof keyword 181
punch_storclass pof keyword 181
punch_unit pof keyword 181
Q
QMF
definition 813
QMF reports 594
QSAM
definition 813
QUALIFIER statement/parameter
definition 813
Quiesce
definition 813
QUIESCE (QUI) command description 225
quiet point
definition 814
QUIESCEINTERVAL installation option 206
quiet point
definition 814
QUIT command
definition 814
R
RACF
definition 814
range-partitioned table spaces
definition 814
RBA
definition 814
RCPY1_DATACLASS POF keyword 181
RCPY1_DATACLASS_ALT POF keyword 181
RCPY1_EXPDT POF keyword 181
RCPY1_MGMTCLASS POF keyword 181
RCPY1_MGMTCLASS_ALT POF keyword 181
RCPY1PREFIX POF keyword 181
RCPY1_PRIQTY POF keyword 182
RCPY1_RETPD POF keyword 182
RCPY1_SECQTY POF keyword 182
RCPY1_STACK POF keyword 182
RCPY1_STORCLASS POF keyword 182
RCPY1_STORCLASS_ALT POF keyword 182
RCPY1_SUPPRESS_SUFF POF keyword 88, 182
RCPY1_THRESH POF keyword 182
RCPY1_UNIT POF keyword 183
RCPY1_UNIT_ALT POF keyword 183
RCPY2_DATACLASS POF keyword 183
RCPY2_DATACLASS_ALT POF keyword 183
RCPY2_EXPDT POF keyword 183
RCPY2_MGMTCLASS POF keyword 183
RCPY2_MGMTCLASS_ALT POF keyword 183
RCPY2_PREFIX= POF keyword 183
RCPY2_PRIQTY POF keyword 183
RCPY2_RETPD POF keyword 184
RCPY2_SECQTY POF keyword 184
RCPY2_STACK POF keyword 184
RCPY2_STORCLASS POF keyword 184
RCPY2_STORCLASS_ALT POF keyword 184
real-time statistics (RTS)
SYSIBM.SYSINDEXSPACESTATS 363
SYSIBM.SYSTABLESPACESTATS 363
updating
reallocation
definition 814
REBALANCE Reorg option 524
Rebd after copy trigger 455
Rebd after stats trigger 453
REBDSTAT exception 453
rebind
definition 815
Rebind option 485
Rebind Version option 485
REBINDFAIL POF keyword 185
REBINDRC POF keyword 185
Recall
definition 815
Recent Trends reports
definition 815
RECOVER (RECV) command description 225
recover pending
definition 815
RECOVER+_LOAD POF keyword 185
RECOVERDOPT POF keyword 185
recovery
definition 815
recovery log
definition 815
recovery pending
definition 815
recovery site
definition 815
RECP
definition 816
RECV data set 118
RECVMAX installation option 206
RECVMAXU installation option 206
RECVPREF installation option 206
RECVPS installation option 206
RECVSS installation option 206
RECVUNIT installation option 206
referenced files
definition 816
referenced objects, Export examples 639
referential integrity
definition 816
refreshing
content, UIM Server 723
REGION POF keyword 185
REGISET
definition 816
register
definition 816
registration point
definition 816
related publications 15
relative byte address (RBA)
definition 816
remote
definition 817
Remove dup actions option 482
Rename an object set 289
renaming a column 376
renaming object sets 301
REORCOPY exception 455
REORDEL exception 517
REORDELS exception 449
REORDSPC exception 449, 517
REORG
command description (REOR) 225
corrective action, setting up 524
definition 817
guidelines for triggering 517
REORG Advisor
automating reorganizations based on REORG candidate table recommendations and BMCTRIG evaluations 735

automating reorganizations based only on REORG candidate table recommendations 735

populating the REORG candidate table 732

preparing to use 731

setting up automated reorganizations 735

Reorg after copy trigger 455

Reorg after stats trigger 453

REORG candidate table

making BIND modifications to retrieve data from 733

REORG exceptions 432

Reorg options

KEEPDICTIONARY 524

LOG 524

PREFORMAT 524

REBALANCE 524

Resize 524

Restart 524

REUSE 524

SCOPE 524

UNLOAD 524

Reorg Pend trigger 449

REORG pending

definition 817

REORG PLUS for DB2

definition 817

REORG PLUS parameters, unsupported 49

REORG_MAPDB POF keyword 185

REORG_MAPTAB POF keyword 186

REORG+_LOAD POF keyword 186

reorganization exceptions 449

REORGDOPT POF keyword 186

REORGINS exception 449

REORGLOB exception 449

REORINS exception 449, 517

REORMDEL exception 449, 517

REORMODS exception 449, 517

REORP

definition 817

REORPEND exception 449, 517

REORSPEC exception 449, 517

REORSTAT exception 453

REP REC

definition 817

REP REC (REPC) command description 225

REP SET

definition 817

REP SET (REPS) command description 225

REPAIR command description 225

report

definition 818

Report

definition 818

report data set 69

report layout for customizable report 675

report log

definition 818

report logging

definition 818

Report Manager

definition 818

REPORT option 481

report output buffer

definition 818

report set

definition 818

REPORT statement

definition 818

REPORT DD statement/parameter

definition 819

reports
action 610
adding a customizable report to the menu 608
Average Length of RID Chain 551
Average Monthly Growth 612
batch customizable 606
batch generation 594
Corrective Action Cross Reference 613
customizable 598
customizing a program 607
Data Set List by Database Name 614
Database Space Detail 615
Database Space Trend 617
Database Storage Utilization 618
Event Recording 553
Exceptions Report by BMCTRIG Action 620
finding standard 547
from historical database 594
Index Analysis 556
Index Combination Statistics 562
Index Key Analysis 565
Index Leaf Distribution 567
Indexspace Space Detail 569
input parameters for customizable 601
internal KB switch 603
layout 598
locating 547
menu 547
Modified Table Space Pages 623
Monthly Detail by Tablespace 571
output 609
QMF descriptions 594
redundancy 551
running customizable 603
Space Estimation 625
Space Estimation Trend 626
Space Utilization by Table 628
standard 548
Statistics Exception 576
Table Level Space Utilization 628
Table Space and Index Extents 632
Table Space and Index Storage Limits 630
Tables with More than Average Growth 631
TablespacE Extent Summary 583
Tablespace Offset RIDs 585
Tablespace Space Detail 587
Update Analysis 574
Volume Analysis 590
Volume Free Space Detail 592
reports, generating POF 107
Repository
restarting
  BMCTRIG-generated job 535
  with batch JCL generation 532
restricting access
  other databases 73
  plans 70
  product database 73
RETRIEVE command
definition 820
RETRY Shlevel option 524
RETRY_DELAY Shlevel option 524
return codes
  ASURXLOC 691
  ASURXLRPT 684
  ASURXSE 686
  ASURXSQL 683
  batch programs 277
RETURN command
definition 820
REUSE Reorg option 524
REXX
  comments 693
  concatenation operator 693
  continuation characters 693
  creating user-defined exceptions 466
  customizable reports, listing 608
  delimiters 693
  EXEC PARM 679, 681
  in customizable reports 44, 692
  JCL to run customized report 681
  quotation marks, SQL 680
  sample program 673
  sample program using ASURXLOC 691
  upper and lower case 693
  variables 691
RFIND command
definition 820
RI
definition 820
RID translation map
definition 820
RIGHT command
definition 821
RIGHT ISPF command 54
RMGR
definition 821
RMGR group
definition 821
RO
definition 821
rollback
definition 821
ROLLBACK command in ASURXSQL 682
Round qty by option 486
row completion processing
definition 821
ROWID SYSREC data set 88
rows
definition 821
rows/key
definition 822
ROWS/KEYS exception 453
RPLAN installation option 206
rpt.page# variable in customizable reports 677
rptcode variable 684
RUNSTATS
  (RUNS) command description 225
definition 317, 822
  grouped services 436
RUNSTATS utility 116
Runtime Component System (RTCS)
definition 822
RUNTIME_HLQ POF keyword 188
RW
definition 822
S
SAF authority for Export 644
sample
  annotated program 693, 694
  complete program 707
  customizable report 674
  customizable report program 673
  files for Performance Advisor 734
Sample
definition 822
sampling
definition 823
  in BMCSTATS 320
  restricting by object type 333
  table columns 333
SAMPLING
definition 822
Save
definition 823
save last used
definition 823
Save option 481
SaveStats
  definition 823
SAVESTATS
  definition 823
scan range
  definition 823
SCC (DB2 Solution Common Code) 48
SCHED_TRIG_CNTM_JOBS POF keyword 188
SCOPE Reorg option 524
secondary authorization ID
  definition 823
secondary index
  definition 823
secondary quantity. See ALTERSEC 541
section name
  definition 824
security
  accessing plans 70
  BMC object 69
  controlling access 278, 311
  other databases 73
  product database 73
security exit
  definition 824
segmented table space
  definition 824
segmented viewing
  definition 824
SEGSIZE
  definition 824
selection list
  definition 824
selection list, definition 53
selection range
  definition 824
SEQTY
  definition 825
sequential data set
  definition 825
server configuration
  UIM 644
service
  definition 825
Service Action generation
  build/edit a Service Action 265
  definition 231
Service Actions 239, 252
Action List panel 244
  deleting 248
  editing 244
  managing 231
service definitions, specifying to Export 655
service syntax, controlling access to 278
services
  BMCTRIG information sources 203
  command descriptions 225
  controlling access to 278
  creating 265
  non-BMCTRIG information sources 203
  rebind option 485
  submitted through BMCTRIG 425
  use in utility jobs 225
  using installation options 203
setting up
  customizable report 603
  QMF reports feature 594
  REORG corrective action 524
  trigger job for REORG 521
severity indicators, message
  definition 825
shadow tables
  definition 825
SHOWCMDS command
  definition 825
Shrlevel options
  DEADLINE 524
  DELAY 524
  DRAIN 524
  DRAIN_WAIT 524
  FASTSWITCH 524
  LONGLOG 524
  MAPPINGTABLE 524
  MAXRO 524
  RETRY 524
  RETRY_DELAY 524
  TIMEOUT 524
SHRLEVEL REFERENCE
  definition 825
SHUTDOWN command
  definition 825
SID
  definition 826
Simple Space Estimation (SSE)
  definition 826
Simple Space Estimation (SSE) feature 54
  simple table space
  definition 826
simulating changes to DB2 objects 405
simulation
definition 826
sizing objects 332
skeleton library compiler. See SLIB (skeleton library) compiler
skeleton library, overriding POF values 109
SL1 installation option 206
SL2 installation option 206
SL3 installation option 206
SL4 installation option 206
SL5 installation option 206
SLIB (skeleton library) compiler
changing an ISPF skeleton 665
compiling 665, 667
ISPF file tailoring 666
processing 667
runtime report 668
runtime report summary 668
runtime unit 667
testing changes 666
SMF
definition 826
SMF$DHSP 644
SMS
definition 826
SMS(Storage Management Subsystem) 88
snapshot utility
definition 826
SORT command
definition 827
Sort JCL Sort option 524
Sort options
Parallel index 524
Sort JCL 524
SORTDEVT 524
SORTNUM 524
WORKDDN1 template 524
SORT/SORT2 statements/parameters
definition 827
SORTDEVT Sort option 524
SORTNUM Sort option 524
SORTOnnn data set 79
SORTOUT data set 79, 88
SORTPnnn data set 79
SORTWK_NBR POF keyword 188
SORTWK_PRIQTY POF keyword 188
SORTWK_SECQTY POF keyword 189
SORTWK_UNIT POF keyword 189
SORTWORK data set 86
source table
definition 827
space
definition 827
space estimation
introduction 405
overview 39, 44
simulating changes to DB2 objects 405
table space 406, 417
space estimation function
definition 827
Space Estimation Report 625
Space Estimation Trend Report 626
SPACE exceptions 449
Space trigger 449
Space Utilization by Table Report 628
SpaceOnly
definition 827
spawner
definition 827
specifying
data set names for worklist JCL 509
exclusions in trigger job 521
GDGs in copy DD options panel 120
JCL generation options in BMCTRIG 521
job generation options in BMCTRIG 521
sampling percentages 333
trigger values 469
SPFOFF command
definition 827
SPFON command
definition 828
spill data sets
definition 828
SPLAN installation option 206
SPLIT ISPF command 54
SPUFI
definition 828
SQL
definition 828
SQL Explorer for DB2
definition 828
SQL Generator
definition 828
SQL ID
definition 829
SQL output data set
definition 829
SQL Performance for DB2
<table>
<thead>
<tr>
<th>SQL processor</th>
<th>SQL statistics collection interval</th>
<th>SQLEXP_LOAD POF keyword</th>
<th>SQTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>definition</td>
<td>SQTY</td>
<td></td>
</tr>
<tr>
<td>SQL statistics</td>
<td>collection interval</td>
<td>SQTY</td>
<td></td>
</tr>
<tr>
<td>SQLEXP_LOAD</td>
<td>POF keyword</td>
<td>SQTY</td>
<td></td>
</tr>
<tr>
<td>POF keyword</td>
<td>SQTY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQTY</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SQL processor**

- SQL processor definition
- SQL statistics collection interval definition
- SQLEXP_LOAD POF keyword definition
- SQTY definition

**SQLEXP_LOAD POF keyword**

- SQLEXP_LOAD POF keyword definition

**SQL statistics collection interval**

- SQL statistics collection interval definition

**SQTY**

- SQTY definition
accessing function 334
analyzing 379
analyzing column 388
analyzing index 390
analyzing index partition 392
analyzing index partition page group 394
analyzing table 386
analyzing table space 349, 381
analyzing table space partition 382
analyzing table space partition page group 385
authorization required 356
benefits of utility 317
BMCCPRS 225
browsing column 345
browsing DB2 and BMCSTATS 341
browsing index 346
browsing index partition 348
browsing table 343
browsing table space 342
browsing table space partition 344
collecting frequency 206
column value analysis 390
database tables 318
deleting 377
deleting after altering or dropping objects 376
Exception Report 576
graphs 397
introduction 313
managing 355
not collected for some objects 315
overview 39
RUNSTATS, definition 317
storage group analysis 395
table space/index columns updated 357
tips for collecting 315
updating column 352
updating database using BMCCPRS 374
updating DB2 catalog interactively 348
updating DB2 catalog, overview 340
updating index 353
updating index partition 355
updating table 350
updating table space partition 351
using BMCSTATS to update DB2 catalog 363
using BMCUPRS to update DB2 catalog 370
volume analysis 396
Statistics Exception Report 576
statistics exceptions 453
STATMDEL exceptions 453
STATMODS exception 453
STATS exceptions 432
status codes
definition 831
STATUS command
definition 831
STATUSR command
definition 831
STATUSS command
definition 831
STEP table
definition 831
STEP.Include_MEMBER POF keyword 191
STEPLIB libraries, setting default values 75
stogroup allocation
definition 831
STOP
definition 832
STOP command
definition 832
STOP command description 225
STOP DB
definition 832
STOP DB (STO DB) command description 225
stopping the UIM Server 717
STOPWAIT POF keyword 191
STOPWTSECS POF keyword 191
STORAGE command
definition 832
storage group (stogroup)
definition 832
storage group statistics analysis 395
storage management mode
definition 832
Storage Management Subsystem
definition 832
STORCLAS installation option 206
STORNAME
definition 832
STORTYPE
definition 833
STOSPACE
definition 833
STOSPACE (STOS) command description 225
Structured filter
definition 833
Structured Query Language
definition 833
submit job
definition 833
submitting a utility job 248
subsystem definition 833
subsystem groups definition 833
SUF definition 833
summary of changes 19
supervisor call (SVC) definition 834
supported DB2 versions 45
SUPPRESS_COMMENTS POF keyword 191
SUTnnndata set 79
SVC definition 834
SWAP ISPF command 54
SWITCH command definition 834
SWPS installation option 206
SWSS installation option 206
SWU installation option 206
symbolic variable definition 834
symbolic variables 74 definition 834
GDG 88 supported 228
SYNC definition 834
SYNC command description 225
sync point definition 834
SYNC table definition 835
SYNCDELETE POF keyword 191
synchronization point (sync point) definition 835
synonyms, finding 60, 681
syntax options
Bypass Exceptions 480
DB2 RTS 480
DELETEAGE 481
Detect inline opts 482
FKEY0MSG 481
Generate priority 483
Group objects 483
Indexes 480
IX Copy 482
IX Escalate 482
Max reorg parts 483
Maximum pqty 486
MSGLEVEL 481
NOTIFY 481
PARTLVL 483
Pqty % if max 486
Rebind 485
Rebind Version 485
Remove dup actions 482
REPORT 481
Resize % pqty 486
Resize % sqty 486
Resize Down 486
Resize jobs 486
Round qty by 486
Save 481
System Triggers 480
syntax options, BMCTRIG 479
syntax statement conventions 16
SYSADM definition 835
SYSC data set 118
SYSCMAX installation option 206
SYSCMAXU installation option 206
SYSCOPY data set, setting JCL options 88
SYSCPREF installation option 206
SYSCPS installation option 206
SYSCSS installation option 206
SYSCCTRL definition 835
SYSCUNIT installation option 206
SYSDISC data set 88
SYSSERR data set 88
SYSEXEC POF keyword 191
SYSIBM.SYSCOPY definition 835
SYSIBM.SYSLGRNX definition 835
SYSSMAPdata set 88
SYSPLEX command
table
  analyze statistics 386
  browse statistics in DB2 catalog 343
  columns updated in DB2 catalog 357
  definition 837
  graph summary 397
  update statistics in DB2 catalog 350
Table Average Row Length
definition 837
Table Cardinality
definition 837
Table check constraint
definition 837
Table Column Detail panel 581
Table column detail report 581
table constraints
definition 838
TABLE option, collecting frequency statistics 206
Table Pages
definition 837
Table Percent Pages
definition 838

table space
  analyze partition page group statistics 385
  analyze partition statistics 382
  analyze statistics 381
  browse partition statistics in DB2 catalog 344
  browse statistics in DB2 catalog 342
  definition 838
  graph summary 397
  reorganizing 517
  simulating changes 405
  space estimation 406, 417
  update partition statistics in DB2 catalog 351
  update statistics in DB2 catalog 349
  updated in DB2 catalog 357
Table Space and Index Extents Report 632
Table Space and Index Storage Limits Report 630
Table Space Pages
definition 838
Table Space Partition Cardinality
definition 838
Table Space Partition Extents
definition 838
Table Space Partition Page Group
definition 838
Table Space Partition Pages
definition 838
Table Space Partition Percent Active/Drop
definition 839
table space set
definition 839
table-controlled partitioning
definition 839
Tables
definition 839
Tables with More than Average Growth report 631
Tablespace Extent Summary Report 583
Tablespace Offset RIDs Report 585
Tablespace Space Detail Report 587
TAPE_EXPDT POF keyword 193
TAPE_RETPD POF keyword 193
TAPE_VOLCNT POF keyword 193
TAPE1 installation option 206
TAPE1 POF keyword 193
TAPE2 installation option 206
TAPE2 POF keyword 193
TAPE3 installation option 206
TAPE3 POF keyword 193
tapes
  setting default processing options 84
  stacking, disabling 84
  using 88
target table
definition 839
Tasks
definition 839
TCP/IP access to Export 642
template file
definition 839
TEMPLATE_DSN POF keyword 194
temporary work data sets 86
TEMPUNIT POF keyword 194
TERSE command
definition 840
thread
definition 840
three-part name
definition 840
threshold
definition 840
threshold, for alternate unit 88
thresholds
accessing system 459
copying 465
deleting 464
editing 463
for exceptions 445
in BMCTRIG command syntax 428
installation of a set of common ones 428
object-action priorities, precedence of 434
priorities 430
reorganization guidelines for 517
setting 428
setting exception 445
setting in the BMCTRIG syntax 457
system-level 428, 459
user-defined 427
time frame
definition 840
time-sharing option
definition 840
TIMEOUT Shrlevel option 524
TIMEPARM POF keyword 194
timestamp
definition 840
timestamp recovery
definition 841
tips
collecting statistics 315
column analysis 389
column value analysis 390
index analysis 392
index partition analysis 394
storage group analysis 396
table analysis 387
tablespace analysis 382
table space analysis 382
table space partition analysis 384
table space partition page group analysis 386
volume analysis 397
TMM
definition 841
To view object set results 304
TOP command
definition 841
TOTALIND
definition 841
TOTALIND exception 449
TOTALOFF
definition 841
TOTALOFF exception 449
Totaloff trigger 449
trace data sets
access to Export 642
accessing BMC UIM Server Commands web page 718
accessing the server commands web page 718
authorization to access the BMC UIM server commands web page 718
changing the security authorization timeout feature temporarily 721
enabling or disabling specific tracing options temporarily 722
enabling or disabling the tracing option temporarily 721
refreshing content 723
refreshing the content 723
starting 724
starting and stopping 717
stopping 717
troubleshooting problems 724
verifying that the server is running 718
viewing active users 720
UIM server configuration 644
ULLQ POF keyword 194
UNCLUST exception 449, 517
Unclust inserts trigger 449
Unicode
data in SYSIN streams 45
UNIQUE
definition 842
UNIQUERULE
definition 843
unit name, ROWID SYSREC data set 88
unit of recovery
definition 843
universal table space (UTS)
definition 843
UNLD_FREF_DATACLASS POF keyword 195
UNLD_FREF_DIRBLOCK POF keyword 195
UNLD_FREF_MGMTCLASS POF keyword 195
UNLD_FREF_PREFIX POF keyword 195
UNLD_FREF_PRIQTY POF keyword 195
UNLD_FREF_SECQTY POF keyword 195
UNLD_FREF_STORCLASS POF keyword 195
UNLD_FREF_SUPPR_SUFF POF keyword 195
UNLD_FREF_UNIT POF keyword 195
UNLD1_DATACLASS POF keyword
UNLD1_DATACLASS_ALT POF keyword
UNLD1_EXPDT POF keyword 196
UNLD1_MGMTCLASS POF keyword 196
UNLD1_MGMTCLASS_ALT POF keyword 196
UNLD1_PREFIX POF keyword 196

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Index 891

tracing
TRCLOGDD 721
tracing, TRCLOGDD 722
trend analysis, overview 39, 44
trigger
definition 842
triggers
automating REORG job 521
detecting exceptions job 476
determining values 469
evaluating exceptions, overview 442
rebind 485
resizing 486
viewing exceptions 495
TRIGNOCOPY keyword 432
TRIGNOREORG keyword 432
TRIGNOSTATS keyword 432
troubleshooting
Export statuses 661
IBM TCP/IP 726
reversed port number 726
started task 724, 725
TRTCH POF keyword 194
Truncation location field 59
TSO
definition 842
TSO submit exit
definition 842
TSOPROGRAM POF keyword 194
TSOSUBEXIT POF keyword 194
Type
definition 842
TYPE
definition 842
TYPE TS
definition 842
types of graphs 397

U
UIM Server
UNLD1_PRIQTY POF keyword 196
UNLD1_RETPD POF keyword 197
UNLD1_SECQTY= POF keyword 197
UNLD1_STACK POF keyword 197
UNLD1_STORCLASS POF keyword 197
UNLD1_STORCLASS_ALT POF keyword 197
UNLD1_SUPPRESS_SUFF POF keyword 88, 197
UNLD1_THRESH POF keyword 197
UNLD1_UNIT POF keyword 197
UNLD1_UNIT_ALT POF keyword 198
UNLD2_DATACLASS POF keyword 198
UNLD2_DATACLASS_ALT POF keyword 198
UNLD2_EXPRDT POF keyword 198
UNLD2_MGMTCLASS POF keyword 198
UNLD2_MGMTCLASS_ALT POF keyword 198
UNLD2_PRED POF keyword 198
UNLD2_SECQTY POF keyword 199
UNLD2_STACK POF keyword 199
UNLD2_STORCLASS POF keyword 199
UNLD2_STORCLASS_ALT POF keyword 199
UNLD2_SUPPRESS_SUFF POF keyword 88, 199
UNLD2_THRESH POF keyword 199
UNLD2_UNIT POF keyword 199
UNLD2_UNIT_ALT POF keyword 199
UNLD3_DATACLASS POF keyword 200
UNLD3_EXPRDT POF keyword 200
UNLD3_MGMTCLASS POF keyword 200
UNLD3_MGMTCLASS_ALT POF keyword 200
UNLD3_PRED POF keyword 200
UNLD3_SUPPRESS_SUFF POF keyword 200
UNLD3_THRESH POF keyword 200
UNLD3_UNIT POF keyword 200
UNLD4_DATACLASS POF keyword 201
UNLD4_MGMTCLASS POF keyword 201
UNLD4_PRED POF keyword 201
UNLD4_STORCLASS POF keyword 201
UNLD4_SUPPRESS_SUFF POF keyword 201
UNLD4_UNIT POF keyword 201
UNLOAD
  definition 843
unload data set, used by utilities 79
UNLOAD PLUS for DB2
  definition 843
UNLOAD PLUS unsupported options 49
UNLOAD Reorg option 524
UNLOAD+, LOAD POF keyword 201
UNLOADDOPT POF keyword 202
unsupported
COPY PLUS options 49
DB2 versions 45
features of (DB2) 45
REORG PLUS parameters 49
UNLOAD PLUS options 49
UP command
  definition 843
UP ISPF command 54
Update Analysis Report 574
UPDATEDB2
  definition 844
updating
  column statistics 352
database using BMCCPRS 374
DB2 catalog interactively 348
index partition interactively 348
index statistics 353
table space partition statistics 351
table space statistics 349
table statistics 350
UPDCATIXS installation option 206
UPDNUCAT installation option 206
User Interface Middleware (UIM) Server 48
user options
  and GDGs 118
  panel display 74
  setting general 74
  system information 74
  utility generation and processing 74
user POF
  creating 103
  updating directly 104
  updating in options panels 104
  using multiple 106
User Profile
  definition 844
user variables 102
USER_HLQ POF keyword 202
USER_VAR1_CHAR POF keyword 202
USER_VAR2_CHAR POF keyword 202
USER_VAR3_CHAR 202
USER_VAR4_CHAR POF keyword 202
USER_VAR5_CHAR POF keyword 202
user-defined programs 228
user-defined services
element of creating 260
JCL elements for generating services 228
symbolic variables used in syntax 228
USERS command
  definition 844
using
  ampersands in installation options module 204
  BMCTRIG, introduction 425
  BMCUPRS instead of BMCSTATS 370
  DB2 object lists 335
  stand-alone path 290
  wildcards in object sets 280
UT
  definition 844
UTILCODE
  definition 844
Utilities for DB2
  definition 844
utility automation, example of 717
utility generation
  and processing options 74
  command descriptions 225
  copying Service Actions 246
  first job, flow diagrams 231
  generate/submit a job 248
  introduction 223
  list of utilities and commands 225
  overview 39
  regenerated job, flow diagram 231
  sample JCL for BMCSTATS job 363
  worklist 233
utility ID
  definition 844
utility module names, setting default processing
  options 97
utility products for (DB2) 48
Utility Products for DB2
  definition 844
UTRO
  definition 845
UTRW
  definition 845
UTUT
  definition 845

V
validation
  definition 845
value line 676
variable
  definition 845
variable option notation 204
variable repository
  definition 845
variables 120
  in customizable reports 677
  in the SQL 680
  job sequence 501
  JOBSEQ 501
  option notation 204
  REORG mapping table 524
  report-dependent 678
  REXX 691
  symbolic, supported 228
VCAT allocation
  definition 845
VCATNAME
  definition 845
VCATPASS
  definition 846
version code
  definition 846
version support, (DB2) 45
View objects 289
View Objects panel
  view object set results 304
viewing
  Service Actions 239, 252
Virtual Storage Access Method (VSAM)
  definition 846
volser
  definition 846
volume
  definition 846
Volume Analysis Report 590
Volume Free Space Detail Report 592
volume statistics analysis 396
VSAM
  definition 846
VSAM object
  data set sizing 79
  JCL generation 116
  sampling 116

W
warm start
  definition 846
WDC installation option 206
WDSN installation option 206
What-If Edit SQL Explain
  definition 846
What-If Index
definition 846
wildcard
definition 847
characters 59
definition 847
expansion in worklists 233, 235
using in object set 280
WMC installation option 206
work data sets
permanent 88
temporary 86
used by utilities 79
work file
definition 847
Work with object sets 289
WORK_DATACLASS POF keyword 202
WORK_MGMTCLASS POF keyword 202
WORK_STORCLASS POF keyword 202
WORKDDN1 template Sort option 524
WORKIDS table
definition 847
working group
definition 847
worklist
column format 233, 235
commands 233
comment lines 233, 237
definition 233, 848
file structure 233, 235
interim, commands 499
interim, structure 499
JCL 507
JCL generation 237
multiple-line commands 233, 236
wildcard expansion 233, 235
worklist DSN
definition 848
worklists
data set 69
workload balancing 439
WPS installation option 206
WRKnnn data set 79
WSC installation option 206
WSS installation option 206
WU installation option 206
X
XBM
definition 848
Z
zaps, displaying 60
zoom, ISPF command (BMCZOOM) 54
ZPLAN installation option 206