REORG PLUS for DB2®
Reference Manual

Supporting

Version 11.1 of REORG PLUS for DB2
Version 11.1 of Database Performance for DB2

January 2014
Contacting BMC Software

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

United States and Canada

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

Telephone  
1 713 918 8800 or  
1 800 841 2031

Fax  
1 713 918 8000

Outside United States and Canada

Telephone  
+01 713 918 8800

Fax  
+01 713 918 8000


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

DB2, FlashCopy, IBM, Language Environment, MVS, RACF, System z, System z10, and z/OS are the trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

Restricted rights legend

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 UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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

Support website

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

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

Support by telephone or e-mail

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

Before contacting BMC

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

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

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

- Send an e-mail message to customer_support@bmc.com.
# Chapter 4  Building and executing REORG PLUS jobs

Building the REORG PLUS job .......................................................... 341
JOB statement ................................................................................. 341
EXEC statement ............................................................................... 342
REGION parameter .......................................................................... 343
Utility parameters on the EXEC statement ........................................ 343
STEPLIB DD statement ...................................................................... 348
REORG PLUS DD statements ............................................................. 348
ANALYZE option for estimating data set allocation ............................. 386

Running REORG PLUS jobs ............................................................... 390
Invoking REORG PLUS ................................................................. 391
Restarting REORG PLUS ................................................................. 391
Terminating or canceling a job .......................................................... 396
Recovering the DB2 object after terminating or canceling a job .......... 397
Recovering from a failure ................................................................. 398
Not completing the reorganization in the UTILTERM phase ............. 400

# Chapter 5  Examples of REORG PLUS jobs

Overview ............................................................................................ 403
Example 1: VCAT-defined segmented table space with SYSIDCIN .......... 407
Example 2: Partitioned table space, single-phase reorganization with
dynamic data set allocation ............................................................... 416
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single
copy of a subset of partitions ............................................................ 441
Example 4: Index-only reorganization ................................................ 463
Example 5: Partial reorganization with UNLOAD PAUSE ..................... 468
Example 6: Restart of a paused REORG job ........................................ 474
Example 7: ANALYZE ONLY to generate space estimates ................. 480
Example 8: Selective unload with discards to archive data set ............. 491
Example 9: ON FAILURE with a user-specified return code ................ 499
Example 10: DSNUTILB reorganization and index that contains keys with
random ordering .............................................................................. 506
Example 11: Partition-by-growth table space ..................................... 517
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning 529
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY .......... 542
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and
partition rebalancing ....................................................................... 553

# Chapter 6  Tuning REORG PLUS jobs

Tuning for performance ...................................................................... 566
Setting installation options for optimal performance .......................... 566
Setting SHRLEVEL CHANGE installation options for optimal
performance ....................................................................................... 567
Specifying command options for optimal performance ....................... 568
Specifying SHRLEVEL CHANGE command options for optimal
performance ....................................................................................... 568
Additional performance information for installation and command options ........................................ 569
Enabling multitasking for performance ........................................ 584
Additional performance tuning recommendations ............................... 589
Additional performance information for SHRLEVEL CHANGE options ........................................ 590
Performance tuning for specific scenarios (any SHRLEVEL) ....................... 590
Performance tuning for specific scenarios (SHRLEVEL CHANGE only) ................. 591
Interpreting performance-related messages ........................................ 592
Tuning to improve memory use ........................................ 597
Memory requirements of the ORDER command option ............................... 597
Tuning sort processing ........................................ 598
Using multitasking to improve memory use ........................................ 598
Tuning copy processing ........................................ 598
Tuning for availability ........................................ 599
Specifying SHRLEVEL ........................................ 599
Renaming or switching staging data sets ........................................ 599
Granting data set authority ........................................ 599
REORG PLUS processing phases ........................................ 600
REORG PLUS architecture ........................................ 600
ANALYZE phase ........................................ 601
UNLOAD phase ........................................ 602
RELOAD phase ........................................ 605
REORG phase for a single-phase reorganization ........................................ 609

Chapter 7 Online reorganization ................................. 615
Introduction to online reorganizations ........................................ 616
Overview of SHRLEVEL CHANGE ........................................ 616
How SHRLEVEL CHANGE works ........................................ 618
How SHRLEVEL CHANGE differs in REORG PLUS and IBM DB2 REORG ........................................ 629
SHRLEVEL CHANGE processing differences from DB2 REORG ........................................ 629
SHRLEVEL CHANGE administrative differences from DB2 REORG ........................................ 630
Running a SHRLEVEL CHANGE reorganization ........................................ 631
Control of the log apply process ........................................ 633
Log apply control options ........................................ 633
Hierarchy of log apply control options ........................................ 636
Using XBM to view and dynamically control the log apply process ........................................ 639
Log apply control option scenarios ........................................ 644
Serialization and concurrency for SHRLEVEL CHANGE ........................................ 646
Concurrency with other applications ........................................ 646
Object status for SHRLEVEL CHANGE ........................................ 647
Operational considerations for online reorganizations ........................................ 647
Interacting with applications ........................................ 648
Incompatible REORG PLUS options ........................................ 648
Allocation of spill data sets ........................................ 648
SHRLEVEL CHANGE considerations for using SELECT or DELETE ........................................ 649
SHRLEVEL CHANGE statistics considerations ........................................ 649
Support for APPLICATION RESTART CONTROL (AR/CTL) ........................................ 650
Recoverability of the reorganized table space ........................................ 651
Copy data sets for SHRLEVEL CHANGE .............................................. 652
Incremental copy data sets for SHRLEVEL CHANGE .............................. 653
Copy registration failure during a SHRLEVEL CHANGE reorganization .... 656
Restart considerations for a SHRLEVEL CHANGE reorganization .......... 657
  Restarting before the UTILTERM phase ............................................ 657
  Not completing the reorganization before the UTILTERM phase .......... 657
  Restarting in the UTILTERM phase ................................................ 658
  Not completing the reorganization in the UTILTERM phase .............. 658
Performance considerations ............................................................... 659
  Sizing memory for the RID translation map ..................................... 660
  Sizing memory for log records ...................................................... 662
  Sizing the spill data sets ............................................................ 664
  Calculating storage requirements for log data ................................. 664
  Copying nonpartitioned indexes during a partial table space reorganization ......................................................... 665
  Making inline copies during a SHRLEVEL CHANGE reorganization ... 665

Appendix A  REORG PLUS installation options 667
Overview .................................................................................... 667
Basic REORG PLUS installation options ........................................... 668
Dynamic allocation installation options ............................................. 740
DYNALOC installation option .......................................................... 761

Appendix B  Common utility tables 765
Overview ....................................................................................... 765
  Considerations and warnings for common utility tables .................. 766
  Managing common utility tables ................................................. 767
BMCDICT table ................................................................................. 769
  Considerations for the BMCDICT table ....................................... 769
  Maintaining the BMCDICT table ............................................... 770
BMCHIST table ................................................................................. 770
  Maintaining the BMCHIST table ................................................. 772
BMCLGRNX table .............................................................................. 772
BMCSYNC table ................................................................................ 773
  Considerations for the BMCSYNC table .................................... 775
  Maintaining the BMCSYNC table ............................................. 776
BMCTRANS table ............................................................................... 777
BMCUTIL table .................................................................................. 778
  Maintaining the BMCUTIL table .............................................. 780
BMXCOPY table .................................................................................. 781
  Maintaining the BMXCOPY table ............................................. 785

Appendix C  REORG PLUS user exits 787
Overview ....................................................................................... 788
Accessing the sample user exits ......................................................... 788
Using DSNUEXIT to construct data set name patterns......................... 789
DSNUEXIT requirements and considerations........................................ 789
DSNUEXIT return codes...................................................................... 790
DSNUEXIT user-defined variables......................................................... 791
DSNUEXIT assembler user exit.............................................................. 791
DSNUEXIT COBOL II and LE COBOL user exit....................................... 805
DSNUEXIT C user exit......................................................................... 814
DSNUEXIT LE C user exit.................................................................... 823
Using a DSRSEXIT, TERMEXIT, or MAPTEXIT user exit...................... 832
DSRSEXIT, TERMEXIT, and MAPTEXIT common restrictions................. 832
DSRSEXIT, TERMEXIT, and MAPTEXIT common variables and
return codes..................................................................................... 832
Using DSRSEXIT to manage VSAM data set redefinition....................... 834
Resizing DB2 objects........................................................................... 835
Setting REDEFINE NO......................................................................... 835
Using the DSRSEXIT user exit to order storage group volumes.............. 836
Using the DSRSEXIT user exit to add SMS classes............................... 836
DSRSEXIT requirements and restrictions.............................................. 836
Running the DSRSEXIT exit................................................................. 837
DSRSEXIT variables............................................................................ 838
Modifying DSRSEXIT variables........................................................... 841
Sample DSRSEXIT REXX user exit....................................................... 844
Using TERMEXIT to control BMCHIST and statistics updates.............. 849
Additional TERMEXIT restrictions....................................................... 850
Running the TERMEXIT exit................................................................. 850
TERMEXIT variables........................................................................... 851
Sample TERMEXIT REXX user exit....................................................... 852
Using MAPTEXIT to create mapping objects........................................ 857
Additional MAPTEXIT requirements and restrictions............................. 858
Supplying MAPTEXIT in your STEPLIB............................................... 858
Supplying MAPTEXIT in a SYSEXEC DD statement............................... 859
MAPTEXIT variables........................................................................... 859
Sample MAPTEXIT REXX exit............................................................... 861

Index

871
# Figures

- Two-phase table space reorganization ........................................... 54
- Two-phase index reorganization ...................................................... 55
- Single-phase table space reorganization .......................................... 56
- Single-phase index reorganization .................................................... 57
- Processing phases when invoking DSNUTILB ...................................... 58
- Data set rename process when STAGEDSN=BMC .................................. 105
- Data set name FASTSWITCH process ............................................... 107
- REORG PLUS command syntax diagram .......................................... 160
- Detail syntax diagrams ..................................................................... 165
- THRESHLD example 1 ......................................................................... 313
- THRESHLD example 2 ......................................................................... 314
- THRESHLD example 3 ......................................................................... 314
- THRESHLD example 4 ......................................................................... 314
- JCL for example 1 ........................................................................... 408
- SYSPRINT for example 1 ................................................................... 410
- JCL for example 2, case 1 ................................................................. 418
- JCL for example 2 ........................................................................... 420
- SYSPRINT for example 2, case 1 ....................................................... 421
- SYSPRINT for example 2, case 2 ....................................................... 428
- JCL for example 3, case 1 ................................................................. 443
- JCL for example 3, case 2 ................................................................. 444
- SYSPRINT for example 3, case 1 ....................................................... 445
- SYSPRINT for example 3, case 2 ....................................................... 452
- ASUSRPT for example 3, case 2 ........................................................ 460
- JCL for example 4 ........................................................................... 463
- SYSPRINT for example 4 ................................................................... 464
- JCL for example 5 ........................................................................... 469
- SYSPRINT for example 5 ................................................................... 470
- JCL for example 6 ........................................................................... 475
- SYSPRINT for example 6 ................................................................... 476
- JCL for example 7, case 1 ................................................................. 481
- JCL for example 7, case 2 ................................................................. 482
- SYSPRINT for example 7, case 1 ....................................................... 482
- SYSPRINT for example 7, case 2 ....................................................... 487
- JCL for example 8 ........................................................................... 493
- SYSPRINT for example 8 ................................................................... 494
- JCL for example 9 ........................................................................... 500
- SYSPRINT for example 9 ................................................................... 501
- JCL for example 10 ........................................................................... 508
- SYSPRINT for example 10 ................................................................. 509
<table>
<thead>
<tr>
<th>Tables</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG PLUS functional enhancements</td>
<td>46</td>
</tr>
<tr>
<td>Functional and operational differences between REORG PLUS and IBM DB2 REORG</td>
<td>48</td>
</tr>
<tr>
<td>SHRLEVEL CHANGE differences between REORG PLUS and IBM DB2 REORG</td>
<td>50</td>
</tr>
<tr>
<td>REORG PLUS processing phases</td>
<td>52</td>
</tr>
<tr>
<td>REORG PLUS data sets</td>
<td>59</td>
</tr>
<tr>
<td>Products and components that REORG PLUS uses</td>
<td>61</td>
</tr>
<tr>
<td>Installation options that enable DSNUTILB features</td>
<td>76</td>
</tr>
<tr>
<td>Dynamic allocation options considerations for a DSNUTILB reorganization</td>
<td>78</td>
</tr>
<tr>
<td>Options that are incompatible or translated for a DSNUTILB reorganization</td>
<td>80</td>
</tr>
<tr>
<td>Executing BMC utilities concurrently</td>
<td>83</td>
</tr>
<tr>
<td>Database initial status requirements for SHRLEVEL NONE</td>
<td>84</td>
</tr>
<tr>
<td>Restrictive statuses that are not permitted for SHRLEVEL NONE</td>
<td>84</td>
</tr>
<tr>
<td>Status changes during a SHRLEVEL NONE reorganization</td>
<td>86</td>
</tr>
<tr>
<td>Database initial status requirements for SHRLEVEL NONE</td>
<td>86</td>
</tr>
<tr>
<td>Statuses that are not permitted for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE</td>
<td>87</td>
</tr>
<tr>
<td>Status changes during a SHRLEVEL REFERENCE UNLOADONLY reorganization</td>
<td>88</td>
</tr>
<tr>
<td>Status changes during a SHRLEVEL REFERENCE reorganization</td>
<td>89</td>
</tr>
<tr>
<td>Status changes during a SHRLEVEL CHANGE reorganization</td>
<td>90</td>
</tr>
<tr>
<td>Staging data set actions</td>
<td>102</td>
</tr>
<tr>
<td>Naming convention for the rename process with STAGEDSN=BMC</td>
<td>103</td>
</tr>
<tr>
<td>Naming convention for the rename process with STAGEDSN=DSN</td>
<td>103</td>
</tr>
<tr>
<td>Naming convention for the FASTSWITCH process</td>
<td>106</td>
</tr>
<tr>
<td>Options that are incompatible when reorganizing catalog objects</td>
<td>112</td>
</tr>
<tr>
<td>Command options that are incompatible when reorganizing LOB table spaces</td>
<td>120</td>
</tr>
<tr>
<td>SIXSNAP and Instant Snapshot</td>
<td>123</td>
</tr>
<tr>
<td>REDEFINE and SIXSNAP</td>
<td>124</td>
</tr>
<tr>
<td>Using the CONDEXEC command and installation options</td>
<td>137</td>
</tr>
<tr>
<td>Using the limit command and installation options</td>
<td>137</td>
</tr>
<tr>
<td>Reorganization run and exceptions reset by type of DASD MANAGER PLUS exception</td>
<td>139</td>
</tr>
<tr>
<td>Exceptions that initiate a reorganization</td>
<td>140</td>
</tr>
<tr>
<td>Real-time statistics updates</td>
<td>143</td>
</tr>
<tr>
<td>REORG PLUS command options</td>
<td>156</td>
</tr>
<tr>
<td>SORTNUM values</td>
<td>188</td>
</tr>
<tr>
<td>Effects of MAXSORTS and SMAX options on REORG PLUS processing</td>
<td>189</td>
</tr>
</tbody>
</table>
Action REORG PLUS takes when memory resources are constrained during sort processing ........................................... 196
Effects of ANALYZE options on cardinality and average row length estimation .......................................................... 200
Description of the TERMINATE UTILITY option .............................................................. 225
Return code hierarchy for the TIMEOUT option ............................................................... 238
Program language keywords .................................................................................. 240
Values for SET ........................................................................................................ 286
Truth table for AND, OR, NOT ................................................................................. 288
Rules for constants .................................................................................................. 291
DDTYPE keywords .................................................................................................. 294
Symbolic variables for the DSNPAT command option .............................................. 316
Valid special delimiter characters for utility ID ......................................................... 319
Variable concatenation examples ............................................................................. 320
Symbolic variables for the SPILLDSNPAT command option ...................................... 332
Valid special delimiter characters for utility ID ......................................................... 333
Variable concatenation examples ............................................................................. 334
Data set type descriptions and quick command reference .......................................... 349
Default copy data set names .................................................................................... 353
DD statements required when allocating copy data sets in your JCL ....................... 356
Calculations for allocating copy data sets ................................................................ 357
Factors that affect BMCSORT dynamic allocation of sort work data sets ................. 368
SYSREC usage ........................................................................................................ 379
Number of SYSREC data sets to allocate ................................................................. 380
SYSUT1 usage .......................................................................................................... 383
Number of SYSUT1 data sets to allocate ................................................................. 384
Estimates provided by the ANALYZE option ............................................................ 388
Record size for SYSREC data sets .......................................................................... 389
Recopying of data sets for restart processing ........................................................... 394
Recovering objects after terminating or canceling jobs .............................................. 397
Recovering from a reorganization failure ................................................................. 398
Cross-reference of examples by function ................................................................ 404
Example 1 key command options and DD statements .............................................. 407
Key command options and DD statements used in example 2 .................................. 417
Key command options used in example 3 .................................................................. 442
Key command options used in example 4 .................................................................. 463
Key command options used in example 5 .................................................................. 469
Key command options used in example 6 .................................................................. 475
Key command options used in example 7 .................................................................. 481
Command options and DD statements used in example 8 ....................................... 492
Key command options and DD statements used in example 9 .................................. 499
Key command options used in example 10 ............................................................... 507
Key command options used in example 11 ............................................................... 517
Command options, installation options, and DD statements used in
example 12 ............................................................................................................. 530
Command options used in example 13 ...................................................................... 542
Key command options used in example 14 ............................................................... 553
Installation default option changes for performance ................................................. 566
Major fields for the DSNUEXIT COBOL II and LE COBOL variable mapping record ................................................................. 808
Major structure fields for the DSNUEXIT C exit parameter ......................... 815
Major variable mapping structure fields for the DSNUEXIT COBOL II and LE COBOL user exit ........................................ 816
Major structure fields for the DSNUEXIT LE C user exit parameter ............ 824
Major variable mapping structure fields for the DSNUEXIT LE C user exit ...... 825
Variables common to all REXX exits .................................................. 833
Variables that REORG PLUS passes to DSRSEXIT ......................... 839
Variables that REORG PLUS passes to TERMEXIT .............................. 851
Variables that REORG PLUS passes to MAPTEXIT .............................. 860
About this book

This book contains detailed information about the REORG PLUS for DB2® product and is intended for IBM® DB2 system administrators, DB2 database administrators, and DB2 application programmers.

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.

Related publications

From the BMC Support Central website (http://www.bmc.com/support), you can

- download a zipped set of documentation PDFs from each product’s EPD page

- link to the BMC Documentation Center (https://webapps.bmc.com/infocenter/index.jsp) to browse documentation sets, or to view video demos (short overviews of selected product concepts, tasks, or features)

- view individual product documents (books and notices) within the “A – Z Supported Product List”
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.

**Conventions**

This book uses the following special conventions:

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

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

  testsys/instance/fileName

- Change bars signify changes that clarify or correct existing information, or that provide new information corresponding to product changes. This book does not use change bars to denote editorial or formatting changes, unless these updates significantly affect your use of the information.
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.)

---
Summary of changes

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 MVS™ commands, keywords, clauses, and data types are displayed in uppercase letters. However, if an item can be shortened, the minimum portion of the MVS command or keyword might be displayed in uppercase letters with the remainder of the word in lowercase letters (for example, CANcel).

- The following conventions apply to variables in syntax diagrams:

  - Variables typically are 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 11.1.00 January 2014

This version fixes known problems in the product. In addition, this version includes the following enhancements and changes:

DB2 Version 11 support - native support

REORG PLUS supports the following features of DB2 Version 11 natively:

- catalog and directory changes and restructuring
■ extended relative byte addresses (RBAs) and log record sequence numbers (LRSNs)

As part of this support, the new RBALRSN_CONVERSION command option enables you to tell REORG PLUS which format you want the reorganized object to be in after the reorganization. This conversion is supported natively except in certain cases when you specify SHRLEVEL CHANGE. For more information, see “RBALRSN_CONVERSION” on page 228.

■ tables that are defined with ENABLE ARCHIVE

The exception to this support is that REORG PLUS terminates when you specify any of the following options for a reorganization of a table space that contains an archive-enabled table:

— SELECT
— UPDATE
— DELETE

■ persistent read only (PRO) status

REORG PLUS terminates when the object that you are reorganizing is in PRO status.

DB2 Version 11 support - invoking DSNUTILB

REORG PLUS supports the following reorganizations by invoking DSNUTILB. For more information about requirements and restrictions when using features that invoke DSNUTILB, see “Reorganization jobs that invoke DSNUTILB” on page 72.

■ reorganizations of table spaces that contain indexes that are defined with EXCLUDE NULL KEYS (supported by a new installation option, UTILB_NULLIX, discussed on page 736)

REORG PLUS natively reorganizes these types of indexes.

■ reorganizations of table spaces or indexes that contain columns with a CCSID specification (supported by a new installation option, UTILB_COLCCSID, discussed on page 736)

■ table space reorganizations on DB2 Version 11 when you specify REBALANCE and the table space is defined with table-controlled partitioning
Summary of changes

Enhanced support for reorganization jobs that invoke DSNUTILB

REORG PLUS has enhanced DSNUTILB reorganization functionality with the following updates:

- New message BMC50182I reports the reason for invoking DSNUTILB.

- When running on DB2 Version 11 or later, REORG PLUS now provides the MAPPINGDATABASE syntax option for DSNUTILB reorganizations, enabling you to override the DB2 subsystem parameter REORG_MAPPING_DATABASE. When invoking DSNUTILB for a SHRLEVEL CHANGE table space reorganization, REORG PLUS passes the value of this option to the IBM DB2 REORG utility. For more information, see “MAPPINGDATABASE” on page 340.

- When running on versions earlier than DB2 Version 11, REORG PLUS now provides an automated method for supplying a mapping table when invoking DSNUTILB. This method uses a REXX exit that you specify in the new MAPTEXIT installation or command option. For more information about the options, see page 243 and page 709. For more information about the REXX exit, see Appendix C, “REORG PLUS user exits.”

Additional enhancements and changes

- In most cases, REORG PLUS natively supports inline LOB data. REORG PLUS invokes DSNUTILB when either of the following conditions exists:

  — The inline length definition was altered before the reorganization.
  — An index exists on an inline LOB column.

  These conditions are covered by the LOB and IXONEX installation options, respectively. Therefore, the INLOB installation option is now obsolete. If your installation options module contains the INLOB option, the module compiles successfully but completes with return code 4 and a message indicating that the option is obsolete.

- To reorganize LOB table spaces when reorganizing the base table space, you must now use the following specifications:

  — AUX YES on the REORG command
  — LOB=YES and DSNUTILB=YES in your installation options module

  **NOTE**
  The LOB=YES requirement is a change from earlier versions and might affect existing jobs. Review your installation options module to ensure that the LOB option is set appropriately for your needs.
REORG PLUS no longer uses or requires the AUXREORG installation option. If your installation options module contains the AUXREORG option, the module compiles successfully but completes with return code 4 and a message indicating that the option is obsolete.

- REORG PLUS no longer requires that you specify an EXTENDED BUFFER MANAGER (XBM) subsystem for reorganization jobs that use snapshot technology. When the following conditions exist, REORG PLUS searches for an XBM subsystem that is at the correct maintenance and enablement level:
  - You are performing a function that requires snapshot technology.
  - A value is not in effect for the XBMID option.

- This version of REORG PLUS enhances the locking technique for sequential file access to reduce the possibility of user 3900 abends.

- This version of REORG PLUS has the following changes to minimum requirements. For full requirement information, see “System setup” on page 65.
  - IBM System z10® processor
  - DB2 Solution Common Code (SCC) version 11.1.00 with PTF BPJ0689
  - DB2 Utilities Common Code (D2U) version 11.1.00
  - XBM or SUF version 6.1.00

**Version 10.2.00  June 2013**

This version fixes known problems in the product. In addition, this version includes the following enhancements and changes:

**Native support for objects for which REORG PLUS invoked DSNUTILB in earlier releases**

**NOTE**

REORG PLUS ignores a DSNUTILB YES specification for natively supported features (unless some other feature requires it). BMC recommends that you review existing jobs to ensure that you obtain expected results. For example, REORG PLUS ignores certain keywords when invoking IBM DSNUTILB, but will use those keywords now that DSNUTILB is not being invoked. For more information, see “Reorganization jobs that invoke DSNUTILB” on page 72.
Summary of changes

The following enhancements provide native support for which REORG PLUS invoked DSNUTILB in earlier releases:

- REORG PLUS provides native support for reorganizing all temporal objects.

  **NOTE**
  You cannot specify SELECT, UPDATE, or DELETE when reorganizing a table space that contains a system-period temporal table that is enabled for versioning.

This change makes the TEMPRALDATA installation option obsolete. If your installation options module contains the TEMPRALDATA option, the module compiles successfully but completes with return code 4 and a message indicating that the option is obsolete.

- REORG PLUS natively reorganizes the following XML objects:
  - table spaces that contain XML columns that support XML versions
  - associated XML table spaces
  - associated node ID indexes

  **NOTE**
  REORG PLUS terminates if you attempt to reorganize a versioned XML table space with SHRLEVEL CHANGE and an update occurs on the XML table space.

This change makes the XML installation option obsolete. If your installation options module contains the XML option, the module compiles successfully but completes with return code 4 and a message indicating that the option is obsolete.

- REORG PLUS natively supports indexes that contain non-key columns.

  This change makes the IXINCLCOL installation option obsolete. If your installation options module contains the IXINCLCOL option, the module compiles successfully but completes with return code 4 and a message indicating that the option is obsolete.

- REORG PLUS natively reorganizes clone objects and base objects that participate (or have participated) in a clone relationship. Base table spaces are now supported regardless of instance number.

  This change makes the CLONE installation option obsolete. If your installation options module contains the CLONE option, the module compiles successfully but completes with return code 4 and a message indicating that the option is obsolete. (The CLONE command option remains valid.)
REORG PLUS natively supports indexes that contain a key that is derived from certain expressions. For more information, see “Indexes on expression” on page 148.

— REORG PLUS currently supports expressions that use the following functions:

- UPPER
- LOWER
- SUBSTR
- DATE
- DAY
- MONTH
- YEAR

— REORG PLUS reorganizes indexes on these expressions and reorganizes table spaces that contain these indexes.

NOTE

PTFs BPU3652, BPJ0470, and BPJ0507 provided this functionality for version 10.1.00. For more information, see the technical bulletin dated December 23, 2011.

REORG PLUS natively reorganizes universal table spaces that are defined with MEMBER CLUSTER.

This change makes the UTSMEM installation option obsolete. If your installation options module contains the UTSMEM option, the module compiles successfully but completes with return code 4 and a message indicating that the option is obsolete.

REORG PLUS natively supports timestamp columns that are defined with a precision (number of microseconds) other than 6.

As part of this support, REORG PLUS enables you to specify a precision other than 6 when including a timestamp in the following specifications:

- WHERE clause on a SELECT, UPDATE, or DELETE statement
- SET clause on an UPDATE statement
- DEADLINE
- LOGFINAL
- ALTER statement in a DDLIN data set

This change makes the TSPREC installation option obsolete. If your installation options module contains the TSPREC option, the module compiles successfully but completes with return code 4 and a message indicating that the option is obsolete.
Summary of changes

- REORG PLUS natively supports timestamp columns that are defined as TIMESTAMP WITH TIME ZONE.

In addition to supporting TIMESTAMP WITH TIME ZONE columns, this support includes the following functionality:

— You can specify a time zone when including a timestamp in the following specifications:
  - WHERE clause on a SELECT, UPDATE, or DELETE statement
  - SET clause on an UPDATE statement
  - DEADLINE
  - LOGFINAL
  - ALTER statement in a DDLIN data set

— You can now specify WITH TIME ZONE on CURRENT_TIMESTAMP for the following specifications:
  - WHERE clause on a SELECT, UPDATE, or DELETE statement
  - SET clause on an UPDATE statement

— You can use the new IMPLICIT_TZ command option to specify a time zone to use in the following cases:
  - You specify CURRENT_TIMESTAMP WITH TIME ZONE.
  - Your WHERE clause includes a TIMESTAMP WITH TIME ZONE column and is comparing column values to a constant that does not contain a time zone.

For more information, see “IMPLICIT_TZ” on page 281.

This change makes the TSTZ installation option obsolete. If your installation options module contains the TSTZ option, the module compiles successfully but completes with return code 4 and a message indicating that the option is obsolete.

Additional enhancements and changes

- You can now reorganize DB2 catalog (database DSNDB06) table spaces and indexes when running in a DB2 Version 10 new-function mode environment or later. For more information, see “Catalog objects” on page 111.

- You can now have REORG PLUS write discards to your archive file (SYSARC) in a format that is compatible with the FORMAT UNLOAD option of the IBM DB2 LOAD utility. New ARCHFORMAT command and installation options enable this functionality. For more information, see page 185 or page 675.
You can now tell REORG PLUS to cancel threads in DB2 that might prevent a successful drain. A new command option, FORCE, enables this functionality; corresponding installation options FORCE, FORCE_AT, and FORCE_RPT provide default values for this option. For more information, see “FORCE” on page 248 or the installation options that begin on page 693.

This option requires additional DB2 authorization. For more information, see “Additional authorizations for canceling threads” on page 69.

You can also specify a new DD statement, BMCFORCE DD, to contain the thread cancelation report. For more information, see “BMCFORCE data sets” on page 352.

NOTE
PTFs BPU3948 and BPJ0472 provided some of this functionality for version 10.1.00. For more information, see the technical bulletin dated May 30, 2012.

REORG PLUS has enhanced statistics collection:

— For UPDATEDB2STATS YES, BMCSTATS YES, and BMCSTATS REPORT, REORG PLUS now collects statistics for the following objects:
  - LOB table spaces
  - partition-by-growth table space partitions that are added during the reorganization

— For UPDATEDB2STATS YES, REORG PLUS now updates additional DB2 catalog statistics by passing the following options to the Common Statistics component:
  - UPDATEDB2 ALL (instead of UPDATEDB2 ACCESSPATH)
  - HISTORY ALL (instead of HISTORY ACCESSPATH)
  - KEYCARD Y (instead of KEYCARD N)

For information about how these options affect which statistics are updated in the DB2 catalog, see the DASD MANAGER PLUS documentation.

NOTE
You cannot change these statistics options.
Summary of changes

- The following new features provide you with additional control over how REORG PLUS handles your staging data sets:
  - You can now have REORG PLUS use rename processing if FASTSWITCH processing fails. The new FSFALLBACK installation and command options tell REORG PLUS whether to enable this capability. For more information, see page 220 or page 695.
  - You can now set a threshold number of staging data sets beyond which REORG PLUS performs FASTSWITCH processing. For data sets under this threshold, REORG PLUS performs rename processing. The new FSTHRESHOLD installation and command options let you set the threshold. For more information, see page 221 or page 695.

- You can now send REORG PLUS output messages to a second output data set by specifying a SYSPRIN2 DD statement in your JCL. In a worklist environment, you can use SYSPRIN2 to view REORG PLUS output in real time. For more information, see “SYSPRIN2 data set” on page 377.

  **NOTE**
  PTF BPU3886 provided this functionality for version 10.1.00.

- REORG PLUS fully supports compressed indexes that have been versioned.

  **NOTE**
  PTFs BPU3853 and BPJ0506 provided this functionality for version 10.1.00. For more information, see the technical bulletin dated December 23, 2011.

- You can now specify the number of columns to use in the limit key when specifying the REBALANCE option. For more information, see “ON n COLUMNS” on page 181.

- REORG PLUS has enhanced DSNUTILB functionality with the following updates:
  - When you specify the &JDATE variable with the DSNPAT option and you are running a DSNUTILB job, REORG PLUS now translates this variable to the IBM &JDATE(3,5) variable.
  - When running under a version of the IBM DB2 REORG utility that allows it, you can now specify noncontiguous partitions for a REORG PLUS table space reorganization that invokes DSNUTILB.
You can now specify the type of data set that you want REORG PLUS to create during dynamic allocation. For more information, see the DSNTYPE installation option or command option on page 750 and page 309.

If you specify an ENDING AT or VALUES clause on an ALTER statement in your DDLIN data set, you can now use the MAXVALUE and MINVALUE keywords.

You can now specify noncontiguous partitions when all of the following conditions exist:

— You are reorganizing a table space whose partition limit keys were altered before the reorganization.

— You are not rebalancing partitions during the reorganization.

— Your version of DB2 supports the ability to specify noncontiguous partitions.

REORG PLUS invokes DSNUTILB to enable this capability. For more information, see “Reorganization jobs that invoke DSNUTILB” on page 72.

This release removes the ACFORTSS option. Now, if the following conditions exist, you must ensure that the DB2 external security exit (DSNX@XAC) from Computer Technologies is implemented:

— You use the Computer Technologies CA-ACF2 or CA-Top Secret security product for DB2.

— You previously used ACFORTSS=Y.

This version of REORG PLUS has the following changes to minimum requirements. For full requirement information, see “System setup” on page 65.

— BMCSORT version 2.4.01
— DB2 Solution Common Code (SCC) version 11.1.00
— DB2 Utilities Common Code (D2U) version 10.2.00
— BMC Common Statistics component version 11.1.00 (to update DB2 catalog statistics or the DASD MANAGER PLUS database statistics)

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

Future releases of REORG PLUS will not support the following modes:

— DB2 Version 9 CM
— DB2 Version 10 CM8
This release includes the following documentation changes:

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

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

**Version 10.1.00   April 2011**

This version fixes known problems in the product. In addition, this version includes the following enhancements and changes:

**DB2 Version 10 support - native support**

REORG PLUS supports the following features of DB2 Version 10 natively:

- catalog and directory changes and restructuring
- new system and database authorities
- migration to DB2 Version 10 from either DB2 Version 8 or DB2 Version 9
- compression dictionaries that were created during DB2 SQL INSERT processing
- reorganizations of indexes on tables that are defined as ORGANIZE BY HASH
- reorganizations of indexes on universal table spaces that are defined with MEMBER CLUSTER
- reorganizations of user-defined XML indexes for which the indexed values are stored as SQL DATE or TIMESTAMP values

REORG PLUS also natively reorganizes XML table spaces that contain these indexes, unless the table space was created in versioning format.

- reorganizations of an index on a table space that contains inline LOB data, unless the index contains a key derived from an expression on an inline LOB column
- reorganizations of table spaces containing an application-period temporal table, unless an index that is defined with BUSINESS_TIME WITHOUT OVERLAPS exists on the table
- reorganizations of indexes on temporal tables, unless the indexes are defined with BUSINESS_TIME WITHOUT OVERLAPS

- reorganizations of table spaces or indexes that contain pending DDL changes in those cases where the IBM DB2 REORG utility would not materialize the DDL changes

  REORG PLUS does not materialize the DDL changes.

**DB2 Version 10 support - invoking DSNUTILB**

REORG PLUS provides the following support by invoking DSNUTILB. For more information about requirements and restrictions when using features that invoke DSNUTILB, see “Reorganization jobs that invoke DSNUTILB” on page 72.

- the ability to reorganize LOB table spaces when reorganizing the base table space (supported by a new installation option, AUXREORG, discussed on page 676)

  You can also specify a new command option, AUX, that REORG PLUS passes to the IBM DB2 REORG utility. For more information, see “AUX” on page 256.

- reorganizations of table spaces that contain tables that are defined as ORGANIZE BY HASH (supported by a new installation option, HASHAX, discussed on page 696)

  You can also specify a new command option, AUTOESTSPACE, to indicate how to determine the hash space size. REORG PLUS passes this option to the DB2 REORG utility. For more information, see “AUTOESTSPACE” on page 256.

- indexes that contain non-key columns (supported by a new installation option, IXINCLCOL, discussed on page 701)

- reorganizations of universal table spaces that are defined as MEMBER CLUSTER (supported by a new installation option, UTSMEM, discussed on page 737)

- reorganizations of table spaces or indexes that contain pending DDL changes in cases when the IBM DB2 REORG utility would materialize the changes (supported by a new installation option, PENDDDL, discussed on page 715)

  If the DB2 REORG utility would not materialize the pending changes, REORG PLUS reorganizes the object natively (but does not materialize the changes). For information about the conditions under which the DB2 REORG utility would not materialize the pending changes, see the documentation for the DB2 REORG utilities.
Summary of changes

- the following types of temporal tables and associated objects (supported by a new installation option, TEMPRLDATA, discussed on page 730):
  - system-period temporal tables
  - history tables that are associated with system-period temporal tables
  - indexes that are defined with BUSINESS_TIME WITHOUT OVERLAPS
  - application-period temporal tables that have an index that is defined with BUSINESS_TIME WITHOUT OVERLAPS

- the following types of reorganizations that include inline LOB data:
  - any table space reorganization (supported by a new installation option, INLOB, discussed on page 701)
  - an index reorganization when the index is on an inline LOB column, or a table space reorganization when an index on an inline LOB column exists on the table space

    Because an index on an inline LOB column contains a key that is derived from an expression, the existing IXONEX installation option supports this feature. For more information, see page 702.

- LOB table spaces when SHRLEVEL CHANGE is in effect (supported by the LOB installation option, discussed on page 704)

- table spaces that contain XML columns that support XML versions, and their associated XML table spaces and node ID indexes (supported by the XML installation option, discussed on page 739)

- timestamp columns that are defined with a precision other than 6 (supported by a new installation option, TSPREC, discussed on page 733)

- timestamp columns that are defined as TIMESTAMP WITH TIME ZONE (supported by a new installation option, TSTZ, discussed on page 734)

Enhanced support for reorganization jobs that invoke DSNUTILB

- REORG PLUS now invokes DSNUTILB without using the DSNUTILS stored procedure. This change provides the following enhancements:

  - You no longer need to install and activate the IBM DSNUTILS stored procedure.

    Consequently, you no longer need to meet the system requirements or authorization requirements for the DSNUTILS stored procedure.
— All messages are now displayed in the REORG PLUS SYSPRINT or job log.

You no longer have to go to the message log of the started task for the
DSNUTILS stored procedure to find execution messages. This change requires
that you do not allocate your SYSPRINT with FREE=CLOSE.

— REORG PLUS no longer requires an extra DB2 thread to invoke DSNUTILB.

REORG PLUS now supports DSNUTILB reorganizations that discard rows by
supporting the following data sets.

**NOTE**

Support for these data sets applies to rows that the DB2 REORG utility discards. The
SELECT and DELETE options of REORG PLUS are not valid for a DSNUTILB
reorganization.

— discard data sets

REORG PLUS uses the dynamic allocation information for DDTYPE ARCHIVE
to create the template for the discard data set that the DB2 REORG utility uses.
For more information, see “SYSARC data set” on page 370.

— LOAD control card data sets

A new DDTYPE, SYSPUNCH, enables you to specify dynamic allocation
information for this data set. REORG PLUS uses this information to create the
template for the SYSPUNCH data set that the DB2 REORG utility uses. For more
information, see “DDTYPE” on page 294 and “SYSPUNCH data set” on
page 378.

When invoking DSNUTILB, REORG PLUS now passes the FASTSWITCH value
that you specify in REORG PLUS instead of using the value in the FASTSWITCH
DSNZPARM.

**Enhanced statistics processing**

REORG PLUS now uses the BMC Common Statistics component of DASD
MANAGER PLUS to update and report statistics from your reorganization job.
This component provides statistics processing for the BMCSTATS and
UPDATEDB2STATS options of REORG PLUS.

The Common Statistics component is installed when you install REORG PLUS.
This version of REORG PLUS requires a minimum of version 10.1.00 of the
Common Statistics component.
This feature enables the following enhancements to statistics processing in REORG PLUS:

— REORG PLUS now collects statistics on XML objects.

— REORG PLUS updates additional statistics in both the DASD MANAGER PLUS database statistics tables and the DB2 catalog.

— You can now request a statistics report without updating the statistics in the DASD MANAGER PLUS database. To support this feature, the BMCSTATS option has a new keyword, REPORT. For more information, see “REPORT” on page 260.

— You can now specify a table space sampling percentage to use for gathering statistics. A new command and installation option, TSSAMPLEPCT, supports this feature. For more information, see “TSSAMPLEPCT” on page 263 and page 733.

— You can optionally specify a separate DD statement in your JCL, ASUSRPRT DD, to contain the statistics report. For more information, see “ASUSRPRT data sets” on page 351.

This enhancement also encompasses the following changes:

— To update the DASD MANAGER PLUS database statistics, you must include the BMCPSWD and ASUBMAIN libraries in the STEPLIB of your reorganization job.

— You can no longer use a user exit supplied by the TERMEXIT option to override BMCSTATS NO or UPDATEDB2STATS NO to YES. REORG PLUS ignores the request from the exit.

— On restart, REORG PLUS does not update statistics if, in the original job, any participating table space partitions were completely loaded or any participating index partitions were completely built.

— REORG PLUS does not collect statistics for the following objects:
  - nonpartitioned indexes when running a partial reorganization
  - partitions in a partition-by-growth table space that are added during the reorganization

— REORG PLUS now ignores the CLUSTERRATIO command option. The cluster ratio value is calculated as if the value for CLUSTERRATIO were STANDARD.

— The Utility Products for DB2 Messages Manual now includes messages that the Common Statistics component produces.
For more information, see “Statistics options” on page 258.

The statistics report that is displayed in your SYSPRINT output has changed. This report is displayed only if you do not specify BMCSTATS YES, BMCSTATS REPORT, or UPDATEDB2STATS YES. The following messages have been removed:

— BMC50510 through BMC50514
— BMC50521 and BMC50522
— BMC50531 and BMC50532
— BMC50541 through BMC50547

The following messages have been added:

— BMC50515 through BMC50519
— BMC50525 through BMC50529

For more information about the new messages, see the Utility Products for DB2 Messages Manual.

Additional enhancements and changes

REORG PLUS now provides the option to offload eligible processing to an IBM System z® Integrated Information Processor (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 this version, the minimum version of XBM or SUF is 5.6.00 with PTF BPE0313.

NOTE

To enable DB2 Version 10 support, XBM and SUF also require PTF BPE0311.

The following installation and command options apply to this feature:

— The new ZIIP installation and command options enable this functionality. For more information, see “ZIIP” on page 252 or page 739.

— You can also use the existing XBMID installation or command option to specify an XBM subsystem to use to access this functionality. For more information, see “XBMID” on page 253 or page 738.

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

- REORG PLUS now supports reorganizing clone objects and base objects that participate (or have participated) in a clone relationship:

  — REORG PLUS natively reorganizes base objects when the base table space is instance number 1.

  — REORG PLUS invokes DSNUTILB to reorganize the following objects:
    - base objects when the base table space is instance number 2
    - clone objects

    A new command option, CLONE, enables you to tell REORG PLUS that you want to reorganize only the clone table in the specified table space, or the specified clone index. For more information, see “CLONE” on page 171.

    A new installation option, CLONE, enables support for these objects by invoking DSNUTILB. For more information, see page 678. For more information about DSNUTILB reorganization jobs, see “Reorganization jobs that invoke DSNUTILB” on page 72.

- You can now supply a DDLOUT DD statement in your reorganization job. The presence of this data set tells REORG PLUS to write to that data set all ALTER statements that it executes during partition rebalancing. For more information, see “DDLOUT data sets” on page 365.

- For SHRLEVEL CHANGE and SHRLEVEL REFERENCE reorganizations, you can now specify whether you want REORG PLUS to back out the reorganization when a failure occurs during processing of limit-key ALTER statements. This capability applies to ALTER statements that are processed either from your DDLIN data set or as a result of the REBALANCE option. A new installation option, ALTRFAIL, supports this change. For more information, see page 674.

- REORG PLUS now supports extended address volume (EAV) data sets.

- Enhancements to I/O handling routines have changed the way that REORG PLUS uses the values that you specify for the buffer installation options. For information about these changes, see the following options:

  — CBUFFS on page 678
  — UBUFFS on page 734
  — WBUFFS on page 737

  -NOTE-

  If, in an earlier version, you changed the values for any of these options from the values that were shipped with REORG PLUS, review the new information for these options carefully to ensure that you obtain the intended results for this version.
The default value for the LOB installation option has changed to YES. For more information about this option, see page 704.

REORG PLUS now allows you to determine how extents are allocated when extending to a new data set. A new installation option, MGEXTENT, enables this support. For more information, see page 712.

REORG PLUS now honors the MAXTAPE option when you are running a job that uses SMS-managed tapes.

If you currently use SMS-managed tapes, BMC strongly recommends that you review the MAXTAPE option to ensure that the current value is appropriate for your environment or job. Change the value if necessary to reduce the possibility that existing jobs will fail or will run at a decreased level of performance.

**NOTE**
You might have already performed this review. A technical bulletin dated March 26, 2010, addressed this issue for earlier versions of REORG PLUS. The associated PTFs are BPU2430 and BPU2431.

This version of REORG PLUS has the following changes to minimum requirements. For full requirement information, see “System setup” on page 65.

— z/OS Version 1.10
— DB2 Solution Common Code (SCC) version 10.1.00
— DB2 Utilities Common Code (D2U) version 10.1.00
— XBM or SUF version 5.6.00

**NOTE**
To enable DB2 Version 10 support, XBM and SUF also require PTF BPE0311.

The dynamic allocation information for specific data set types that was in Chapter 2 has been moved to the relevant data set sections in Chapter 4.
Summary of changes
Chapter 1 Introduction to REORG PLUS

This chapter presents the following topics:

Overview ................................................................. 43
  Disadvantages of disorganized data ................................ 43
  The REORG PLUS solution .......................................... 44
  REORG PLUS benefits ................................................ 44
BMC Software solution integration .................................. 45
  Product components ................................................ 45
  Technology components ............................................. 45
  Features ............................................................... 46
Tasks that REORG PLUS performs .................................. 46
Differences between REORG PLUS and the IBM DB2 REORG utility ................................................................ 48
How REORG PLUS works ............................................. 51
  REORG PLUS processing phases .................................. 51
  REORG PLUS data sets .............................................. 59
  Associated products and common components that REORG PLUS uses ......................................................... 61

Overview

The BMC Software utility products for DB2 address the needs of database administrators (DBAs) and system administrators by providing high-performance database administration and utility products. The REORG PLUS for DB2 product is a BMC Software product for reorganizing DB2 databases.

Disadvantages of disorganized data

Disorganized data decreases database efficiency in the following ways:

- In sequential processing, more I/Os are required to retrieve disorganized data than are needed to retrieve data items that are physically adjacent to one another.
The retrieval inefficiency caused by the disorganization slows DB2 application response time and decreases user productivity.

A disorganized database wastes DASD space.

For DB2 to operate at maximum cost-effectiveness, the physical structures of the database must be organized as efficiently as possible. Without good organization, system performance declines while costs, in both time and money, rise.

The REORG PLUS solution

BMC developed a product line to respond to the needs of DBAs, system administrators, and other DB2 users who require high-performance database administration and utility products. The REORG PLUS product efficiently reorganizes DB2 data. REORG PLUS replaces most of the functions of the IBM DB2 REORG utility and provides additional functions that are not available in other reorganization utilities.

REORG PLUS benefits

Advanced techniques and additional functions allow REORG PLUS to reorganize data faster than the IBM DB2 REORG utility does. In addition, REORG PLUS offers the following significant benefits over other reorganization utilities:

- **reduces costs of reorganizing DB2 data** because fewer CPU cycles and EXCPs are used
- **increases availability of DB2 data** because the time needed to reorganize the data is reduced
- **improves DB2 performance** by allowing more frequent reorganizations because of reduced reorganization costs and elapsed times

If you have the BMC Software EXTENDED BUFFER MANAGER (XBM) product or its SNAPSHOT UPGRADE FEATURE (SUF) installed, you can use REORG PLUS to perform online reorganizations. Online reorganizations allow full application access to your DB2 objects during most of the reorganization process, thus minimizing the amount of time that data is unavailable to you. You can request an online reorganization by specifying SHRLEVEL CHANGE on the REORG command.

Performing an online reorganization provides the following benefits:

- **allows full read/write (RW) access** to DB2 data during the reorganization
BMC Software solution integration

REORG PLUS is also a component of the Database Performance for DB2® solution. Database Performance helps database administrators (DBAs) determine the maintenance tasks that are required on their DB2 objects.

Database Performance integrates the features of several BMC products and technologies.

Product components

The Database Performance solution includes the following BMC products:

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

Technology components

The Database Performance solution includes the following BMC technologies:

- BMCSORT
- JCL Generation and Execution
- User Interface Middleware (UIM) Server
- BMC Mainframe DNA Host Services (DHS) and BMC Mainframe DNA
- DB2 Solution Common Code (SCC)
- DB2 Utilities Common Code (D2U)
- BMC Common Statistics (ATS)
Features

The following features are available only when you install the Database Performance for DB2 solution:

- You can 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. For more information, see the DASD MANAGER PLUS for DB2 User Guide.

- You can use the value BMC on the CONDEXEC installation or command option to instruct REORG PLUS to use the DASD MANAGER PLUS exceptions table to determine whether an object should be reorganized. Using the exceptions table gives REORG PLUS an expanded set of conditions to determine whether the reorganization is needed. For information about the CONDEXEC options, see page 679 or page 230.

- You can use the value BMCSTATS on the ANALYZE command option to enhance the performance of REORG PLUS. 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 page 199.

Tasks that REORG PLUS performs

REORG PLUS accomplishes the standard reorganization tasks and also offers the functional enhancements described in Table 1.

Table 1  REORG PLUS functional enhancements (part 1 of 2)

<table>
<thead>
<tr>
<th>Functional area</th>
<th>Enhanced tasks</th>
</tr>
</thead>
</table>
| resources       | - analyzes resources for the specific reorganization  
                     - provides data about resource requirements  
                     - optionally offloads eligible processing to an IBM System z\textsuperscript{®} Integrated Information Processor (zIIP)  
                     - optionally allocates sort work files, index work files, unload data files, archive files, and copy files dynamically |
| partitions      | - allows select and delete operations during a partial or full reorganization  
                     - rebalances partitions for all access levels, including SHRLEVEL CHANGE |
| conditional reorganizations | - uses the traditional values that are stored in the DB2 catalog  
                     - optionally uses an expanded set of exceptions and user controls from the DASD MANAGER PLUS exceptions table if you are using the Database Performance for DB2 solution |
| performance     | - provides data for fine-tuning reorganization performance  
                     - for an additional performance gain, offers an optional single processing phase  
                     - allows multitasking that is not limited by the number of CPUs |
Table 1  REORG PLUS functional enhancements (part 2 of 2)

<table>
<thead>
<tr>
<th>Functional area</th>
<th>Enhanced tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessibility</td>
<td>- optionally allows the objects that you are reorganizing to be available in read-only (RO) status during unload and reload processing</td>
</tr>
<tr>
<td></td>
<td>- offers a nondestructive reorganization process, meaning that the objects that you are reorganizing remain intact throughout the reorganization</td>
</tr>
<tr>
<td></td>
<td>- reclaims space that dropped tables and deleted rows occupied</td>
</tr>
<tr>
<td></td>
<td>- offers online reorganization by using SHRLEVEL CHANGE, allowing applications to have read-write (RW) access during most of the reorganization processing</td>
</tr>
<tr>
<td></td>
<td>- for a SHRLEVEL CHANGE reorganization, allows you to dynamically change the REORG command by using the XBM Utility Monitor or the MVS™ console</td>
</tr>
<tr>
<td>rebuilding</td>
<td>- removes row indirection</td>
</tr>
<tr>
<td>objects and data</td>
<td>- optionally purges unwanted or out-of-date information and can write this data to an archive data set (for all types of reorganizations)</td>
</tr>
<tr>
<td></td>
<td>- optionally updates column values to a specified constant value</td>
</tr>
<tr>
<td></td>
<td>- reorders out-of-cluster-order rows in single and multitable table spaces by table and clustering key or by clustering key only</td>
</tr>
<tr>
<td></td>
<td>- rebuilds associated indexes</td>
</tr>
<tr>
<td></td>
<td>- restores PCTFREE and FREEPAGE space in table spaces and indexes</td>
</tr>
<tr>
<td></td>
<td>- adds NULL or DEFAULT values to short rows after a table is altered to add new columns</td>
</tr>
<tr>
<td></td>
<td>- optionally reorganizes without reclustering rows</td>
</tr>
<tr>
<td></td>
<td>- optionally deletes and redefines user-defined data sets and data sets defined in DB2 storage groups as part of the reorganization</td>
</tr>
<tr>
<td>copies</td>
<td>- with a self-contained copy function, produces image copies or DSNICOPY-type copies concurrently with the reorganization</td>
</tr>
<tr>
<td></td>
<td>- optionally dynamically allocates the full and incremental image copy data sets</td>
</tr>
<tr>
<td></td>
<td>- optionally produces inline image copies as it reloads your tables</td>
</tr>
<tr>
<td>statistics and reports</td>
<td>- produces a statistics report</td>
</tr>
<tr>
<td></td>
<td>- optionally updates statistics in the DASD MANAGER PLUS statistics tables</td>
</tr>
<tr>
<td></td>
<td>- optionally updates statistics in DB2 catalog tables</td>
</tr>
<tr>
<td></td>
<td>- optionally maintains historical records of all REORG PLUS activities</td>
</tr>
<tr>
<td></td>
<td>- resets real-time statistics and timestamp values (in both DSNRTSDB and DB2 memory)</td>
</tr>
<tr>
<td></td>
<td>- optionally provides a report listing all ALTER statements that REORG PLUS executes during partition rebalancing</td>
</tr>
<tr>
<td>restart and</td>
<td>- provides restart capabilities</td>
</tr>
<tr>
<td>recovery</td>
<td>- optionally recovers from an abnormal termination</td>
</tr>
<tr>
<td></td>
<td>- lists the highest DB2 log relative byte address (RBA) or log record sequence number (LRSN) for the table space to facilitate recovery if the reorganization does not complete</td>
</tr>
<tr>
<td>user exits</td>
<td>provides user exit points that allow you to</td>
</tr>
<tr>
<td></td>
<td>- provide user-defined variables that you can use to construct data set name patterns for dynamically allocated files</td>
</tr>
<tr>
<td></td>
<td>- automatically resize DB2 VSAM objects</td>
</tr>
<tr>
<td></td>
<td>- reorder your storage group volumes or use a subset of them</td>
</tr>
<tr>
<td></td>
<td>- selectively specify REDEFINE NO for an object</td>
</tr>
<tr>
<td></td>
<td>- bypass the insert into BMCHIST and give you dynamic control over updates to BMCHIST, BMCSSTATS, real-time statistics, and UPDATEDB2STATS at termination</td>
</tr>
<tr>
<td></td>
<td>- create (and drop after successful DSNUTILB completion) mapping objects for use when invoking DSNUTILB</td>
</tr>
</tbody>
</table>
Differences between REORG PLUS and the IBM DB2 REORG utility

Table 2 summarizes the most important functional and operational differences between REORG PLUS and the IBM DB2 REORG utility. Table 3 on page 50 summarizes SHRLEVEL CHANGE differences.

**NOTE**

When REORG PLUS invokes DSNUTILB, you are using the IBM DB2 REORG utility. Therefore, many of these differences do not apply for a DSNUTILB reorganization.

---

**Table 2  Functional and operational differences between REORG PLUS and IBM DB2 REORG**

(part 1 of 2)

<table>
<thead>
<tr>
<th>Functional or operational area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authorization</td>
<td>REORG PLUS does not run as part of the DB2 subsystem. Therefore, to use REORG PLUS, you must have system authorization similar to that required by DB2 unless you are using the IBM Resource Access Control Facility (RACF) component of the z/OS Security Server and REORG PLUS is installed with OPNDB2ID=YES.</td>
</tr>
<tr>
<td>multiple reorganizations</td>
<td>REORG PLUS allows only one REORG command in the input data set (SYSIN). Therefore, you must run separate job steps to execute multiple reorganizations on different table spaces.</td>
</tr>
<tr>
<td>UNLOAD ONLY</td>
<td>REORG PLUS does not have an UNLOAD ONLY option. Therefore, you cannot use REORG PLUS to generate FORMAT UNLOAD data. You can use the UNLOAD PLUS for DB2 product from BMC Software to generate FORMAT UNLOAD data. UNLOAD PLUS provides advanced unload utility functions as well as basic unload tasks. <strong>Note</strong>: You can use the ARCHFORMAT DB2 option to tell REORG PLUS to write discards to your archive data set as FORMAT UNLOAD data.</td>
</tr>
<tr>
<td>reorganization phases</td>
<td>The SORT and BUILD phases of the DB2 REORG utility are combined into the RELOAD phase in REORG PLUS (two-phase reorganization), or into the REORG phase (single-phase reorganization).</td>
</tr>
<tr>
<td>indexes</td>
<td>To restore the clustering order of rows, REORG PLUS always sorts the unloaded rows (like the SORTDATA option of the DB2 REORG utility) and has no option to unload via the clustering index. This processing requires sufficient sort work space to sort not only the index keys, but all rows of the largest partition (or all rows of the table space if the table space is not partitioned).</td>
</tr>
</tbody>
</table>
**Differences between REORG PLUS and the IBM DB2 REORG utility**

**Table 2  Functional and operational differences between REORG PLUS and IBM DB2 REORG (part 2 of 2)**

<table>
<thead>
<tr>
<th>Functional or operational area</th>
<th>Description</th>
</tr>
</thead>
</table>
| start/stop status              | For SHRLEVEL NONE (the default), REORG PLUS stops the table space and index spaces that you are reorganizing at the beginning of unload processing. The space remains stopped throughout the job. For a partial reorganization, REORG PLUS stops and starts only those partitions that you specified with the PART option of the REORG command. REORG PLUS stops and starts nonpartitioned indexes in their entirety.  
For SHRLEVEL REFERENCE UNLOADONLY, REORG PLUS starts the table space and index spaces that you are reorganizing in RO status at the beginning of the UNLOAD phase. REORG PLUS later stops the spaces at the beginning of the RELOAD phase, and they remain stopped until the end of the job. For a partial reorganization, REORG PLUS stops and starts only those partitions that you specified with the PART option of the REORG command. REORG PLUS stops and starts nonpartitioned indexes in their entirety.  
For SHRLEVEL REFERENCE, REORG PLUS starts the table space and index spaces that you are reorganizing in RO status. The spaces remain in RO status until they are stopped at the beginning of the UTILTERM phase. For a partial reorganization, REORG PLUS stops and starts only those partitions that you specified with the PART option of the REORG command. REORG PLUS stops and starts nonpartitioned indexes in their entirety. |
| multiple volumes               | For multiple volume storage-group-defined table spaces and indexes, REORG PLUS attempts to reallocate the data set on the volume on which the data set currently resides if the volume is still defined in the storage group. After the current volume, the order of the volumes that REORG PLUS uses for allocating the DB2 VSAM data sets that you are reorganizing is unpredictable unless you use the DSRSEXIT user exit to specify a particular order. |
| EDITPROCs                      | REORG PLUS uses EDITPROCs to extract keys and update columns. |
| recoverability and restartability | REORG PLUS does not have a LOG YES option. Therefore, you must create and register a full image copy to ensure recoverability of the table space after reorganizing. You can create a full image copy by using the REORG PLUS COPY YES option. You can also use the BMC COPY PLUS for DB2® product or the IBM DB2 COPY utility. |
| partition rebalancing          | When you specify the REBALANCE command option, REORG PLUS computes limit key breaks based on rebalancing at the record level, which enables REORG PLUS to redistribute the number of rows across partitions. The DB2 REORG utility computes limit key breaks based on rebalancing at the page level.  
REORG PLUS enables you to rebalance partitions when running a partial reorganization.  
Alternatively, you can use a DDLIN data set to specify ALTER statements with new limit key values. REORG PLUS uses these new limit key values to rebalance partitions. |
### Differences between REORG PLUS and the IBM DB2 REORG utility

#### Table 3  SHRLEVEL CHANGE differences between REORG PLUS and IBM DB2 REORG (part 1 of 2)

<table>
<thead>
<tr>
<th>Functional or operational area</th>
<th>Description</th>
</tr>
</thead>
</table>
| performance                      | ■ Due to efficient processing in reorganization, log apply, and termination, REORG PLUS can successfully complete a SHRLEVEL CHANGE reorganization when the DB2 objects that you are reorganizing are under a heavier application load.  
■ REORG PLUS provides the ability to start the read-only phase of the reorganization at a specific time. |
| log apply process                | ■ You can specify the spill data set size. If insufficient memory exists to hold the RID translation maps or the log records, REORG PLUS allocates spill data sets on disk.  
■ REORG PLUS processing does not impact the DB2 buffer pools because REORG PLUS runs outside of DB2. |
| RID translation map              | ■ The RID translation map that REORG PLUS creates is not a DB2 object and therefore does not use DB2 resources.  
■ The REORG PLUS RID map processing does not require log record sequence numbers (LRSNs) or relative byte addresses (RBAs) to be kept for each RID map entry, resulting in less storage for the RID map.  
■ You can control the amount of virtual storage that is allocated for the RID map by using an installation option or an option on the REORG command.  
■ REORG PLUS does not require a mapping table to be predefined and does not require the mapping table name to be specified on the REORG command.  
■ REORG PLUS does not require any DB2 sorting services to order the RID map records. |
| image copy support               | REORG PLUS can  
■ update full image copy data sets on DASD  
■ create incremental image copies  
■ create full inline image copies  
When possible, REORG PLUS multitasks the creation of full and incremental image copies, reducing the amount of processing time. |
| altering limit keys              | REORG PLUS allows you to alter the limit keys to rebalance partitions during a SHRLEVEL CHANGE reorganization. You can perform the alter as part of the reorganization, so that the object is never put in REORG pending status (REORP). |
| defining data sets               | REORG PLUS provides additional syntax for more flexibility, including options that allow you to define memory size for the RID maps and log record buffers, and disk size for the spill data sets. You can also specify a data set name prefix to customize the spill data set names. |
How REORG PLUS works

This section describes the following information about how REORG PLUS works:

- REORG PLUS execution phases
- data sets that REORG PLUS uses
- common components that REORG PLUS uses

For a more detailed explanation of the execution phases for SHRLEVEL CHANGE, see “How SHRLEVEL CHANGE works” on page 618. The figures for the various types of SHRLEVEL CHANGE reorganizations are located in Chapter 7, beginning on page 623.

REORG PLUS processing phases

Table 4 describes the REORG PLUS processing phases. Figures 1 through 5 starting on page 54 illustrate the processing phases for the following types of reorganizations:

<table>
<thead>
<tr>
<th>Reorganization type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>two-phase table space</td>
<td>Figure 1 on page 54</td>
</tr>
<tr>
<td>two-phase index</td>
<td>Figure 2 on page 55</td>
</tr>
<tr>
<td>single-phase table space</td>
<td>Figure 3 on page 56</td>
</tr>
<tr>
<td>single-phase index space</td>
<td>Figure 4 on page 57</td>
</tr>
<tr>
<td>invoking DSNUTILB</td>
<td>Figure 5 on page 58</td>
</tr>
</tbody>
</table>

Table 3 SHRLEVEL CHANGE differences between REORG PLUS and IBM DB2 REORG (part 2 of 2)

<table>
<thead>
<tr>
<th>Functional or operational area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>altering execution</td>
<td>You can dynamically alter execution by using the Utility Monitor function of the XBM ISPF interface.</td>
</tr>
<tr>
<td>displaying status</td>
<td>You can display the status of the online reorganization by sending the Display command to REORG PLUS through the interface provided by the XBM Utility Monitor.</td>
</tr>
</tbody>
</table>
## REORG PLUS processing phases

### Table 4 REORG PLUS processing phases (part 1 of 2)

<table>
<thead>
<tr>
<th>Phase name</th>
<th>Description</th>
</tr>
</thead>
</table>
| UTILINIT   | During this phase, REORG PLUS completes the following processes:  
  - initializes the job  
  - performs DB2 catalog lookup  
  - reads, parses, and verifies the REORG command and the IDCAMS control statements in the SYSIDCIN data set |
| DSNUTILB   | REORG PLUS executes this phase only when the type of reorganization that you are running requires DSNUTILB. During this phase, REORG PLUS passes processing to DSNUTILB. This phase takes the place of the UNLOAD, RELOAD, and REORG phases, as well as the LOGAPPLY and LOGFINAL phases for a SHRLEVEL CHANGE reorganization. Additionally, this phase performs some of the tasks that the UTILTERM phase normally performs, such as updating statistics.  
  For more information about the types of jobs that require this phase, see “Reorganization jobs that invoke DSNUTILB” on page 72. |
| ANALYZE    | During this phase, REORG PLUS completes the following processes:  
  - analyzes the objects that you are reorganizing  
  - optionally produces statistics to help with data set allocation  
  REORG PLUS uses information from this phase to optimize the reorganization process. |
| UNLOAD     | During this phase, REORG PLUS completes the following processes:  
  - unloads the data from the table space or the index space  
  - sorts the data if ORDER YES is in effect  
  - for a table space reorganization, creates the unload data file (SYSREC)  
  - creates the index work files (SYSUT1)  
  - writes discarded rows to the archive data set (SYSARC)  
  - builds the compression dictionary and compresses the rows |
| RELOAD     | During this phase, REORG PLUS completes the following processes:  
  - redefines the VSAM data sets if the value of the REDEFINE option is YES  
  - reloads the data into the table space and index spaces  
  - sorts indexes as required before index build  
  - collects statistics  
  - produces the requested copies  
  - if required, adds partitions to a partition-by-growth table space (SHRLEVEL NONE and SHRLEVEL REFERENCE only) |
| REORG      | During this phase, REORG PLUS performs the functions of both the UNLOAD and RELOAD phase in a single processing phase. The SHRLEVEL specification affects the restartability of the job. |
| LOGAPPLY   | During this phase, REORG PLUS applies the stored log records to the reorganized staging data sets  
  This phase runs only if you specify SHRLEVEL CHANGE. For more information about the LOGAPPLY phase, see “How SHRLEVEL CHANGE works” on page 618. |
Table 4    REORG PLUS processing phases (part 2 of 2)

<table>
<thead>
<tr>
<th>Phase name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **LOGFINAL** | During this phase, REORG PLUS completes the following processes:  
  - prevents updates to the original data sets  
  - if required, adds partitions to a partition-by-growth table space  
  This phase runs only if you specify SHRLEVEL CHANGE. For more information about the LOGFINAL phase, see “How SHRLEVEL CHANGE works” on page 618. |
| **UTILTERM** | During this phase, REORG PLUS performs cleanup as follows:  
  - updates the BMCHIST table  
  - updates the DASD MANAGER PLUS statistics tables  
  - updates the DB2 catalog statistics  
  - updates the DB2 real-time statistics tables  
  - performs the rename or FASTSWITCH operations that are associated with the staging data sets when you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE  
  - cleans up the BMCSYNC and BMCUTIL tables  
  REORG PLUS executes only part of this phase when invoking DSNUTILB. |
| All phases | During all phases except the DSNUTILB phase, REORG PLUS updates the BMCUTIL and BMCSYNC DB2 tables. |
All phases use the SYSPRINT data set and update the BMCUTIL and BMCSYNC tables.

All sort processing uses the UTPRINT data set.
The UNLOAD phase updates the BMCDICT table.
The UTILTERM phase updates the BMCHIST table.

In a SHRLEVEL REFERENCE reorganization, during the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.
Figure 2 Two-phase index reorganization

In a SHRLEVEL REFERENCE reorganization, during the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

All phases use the SYSPRINT data set and update the BMCUTIL and BMCSYNC tables. All sort processing uses the UTPRINT data set. The UTILTERM phase updates the BMCHIST table.
REORG PLUS processing phases

Figure 3  Single-phase table space reorganization

All phases use the SYSPRINT data set and update the BMCUTIL and BMCSYNC tables.

All sort processing uses the UTPRINT data set.
The REORG phase updates the BMCDICT table.
The UTILTERM phase updates the BMCHIST table.
All phases use the SYSPRINT data set and update the BMCUTIL and BMCSYNC tables.

All sort processing uses the UTPRINT data set. The UTILTERM phase updates the BMCHIST table.

Figure 4  Single-phase index reorganization

The SYSUT1.nn data set is optional for single-phase index reorganization, but is required for restarting the job. Note: REORG PLUS does not use the SYSUT1 data set for a single-phase SHRLEVEL REFERENCE or SHRLEVEL CHANGE index reorganization.

In a SHRLEVEL REFERENCE reorganization, during the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.
Figure 5  Processing phases when invoking DSNUTILB

The UTILINIT and UTILTERM phases use the SYSPRINT data set.
REORG PLUS data sets

Table 5 lists the names (ddnames) of the data sets that REORG PLUS uses, and describes each data set. You can override most of the ddnames or ddname prefixes by using REORG command or installation options. For more information about each data set, see the pages listed in Table 5.

<table>
<thead>
<tr>
<th>Data set or ddname</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASUSRPRT</td>
<td>ASUSRPRT is an optional data set that contains the statistics report that is generated when you specify BMCSTATS YES, BMCSTATS REPORT, or UPDATEDB2STATS YES.</td>
<td>351</td>
</tr>
<tr>
<td>BMCFORCE</td>
<td>BMCFORCE is an optional data set that contains the thread cancelation report that is generated when you specify either FORCE REPORTONLY or FORCE_RPT YES.</td>
<td>352</td>
</tr>
<tr>
<td>DB2 data sets</td>
<td>REORG PLUS dynamically allocates the DB2 data sets (table spaces or index spaces) that you are reorganizing. Therefore, you do not need to specify them in your JCL.</td>
<td>not applicable</td>
</tr>
<tr>
<td>DDLIN</td>
<td>The DDLIN input data set contains the SQL ALTER INDEX and ALTER TABLE statements with the new limit key values to use to rebalance partitions. REORG PLUS uses this optional data set only during a table space reorganization.</td>
<td>360</td>
</tr>
<tr>
<td>DDLOUT</td>
<td>The DDLOUT output data set contains all SQL ALTER statements that REORG PLUS executes during partition rebalancing. REORG PLUS opens this optional data set only if executing an ALTER.</td>
<td>365</td>
</tr>
<tr>
<td>full copy data setsa</td>
<td>REORG PLUS creates one or more output copy data sets that contain the image copy or DSN1COPY-type copy of the reorganized table space. After completing the copy, REORG PLUS deallocates the data set in order to free the device unless you specify VOL=(,RETAIN) in the JCL. The copy data sets should not be temporary data sets.</td>
<td>352</td>
</tr>
<tr>
<td>incremental copy data setsa</td>
<td>During a SHRLEVEL CHANGE reorganization only, REORG PLUS might also create one or more output data sets that contain an incremental image copy. Depending on the option that you choose, REORG PLUS creates a standard image copy during the LOGFINAL phase. After completing the copy, REORG PLUS deallocates the data set in order to free the device unless you specify VOL=(,RETAIN) in the JCL. The incremental copy data sets should not be temporary data sets.</td>
<td>267, 653</td>
</tr>
<tr>
<td>SORTWKnn</td>
<td>SORTWKnn is the work data set that is used by BMCSORT. The data set is used in the UNLOAD and RELOAD phases for a two-phase reorganization, and in the REORG phase for a single-phase reorganization. You cannot allocate the sort work files as VIO data sets or tape data sets. You must allocate each individual work file on a single DASD unit.</td>
<td>366</td>
</tr>
</tbody>
</table>
### Table 5  REORG PLUS data sets (part 2 of 2)

<table>
<thead>
<tr>
<th>Data set or ddname</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSARC</td>
<td>The archive data set contains rows that REORG PLUS discards during a reorganization. REORG PLUS writes to this data set as the table space is unloaded. The SYSARC data set should not be a temporary data set. For DSNUTILB reorganizations, this data set is allocated as the discard data set that the DB2 REORG utility uses, and contains rows that are discarded during the reorganization.</td>
<td>370</td>
</tr>
<tr>
<td>SYSERR&lt;sup&gt;a&lt;/sup&gt;</td>
<td>This data set holds diagnostic messages that REORG PLUS might produce in error situations when running a SHRLEVEL CHANGE reorganization.</td>
<td>373</td>
</tr>
<tr>
<td>SYSEXEC</td>
<td>SYSEXEC specifies the library concatenation where REXX exits reside.</td>
<td>373</td>
</tr>
<tr>
<td>SYSIDCIN</td>
<td>The SYSIDCIN input data set contains the IDCAMS command statements that REORG PLUS uses to redefine user-defined (VCAT-defined) data sets, including the staging data sets that REORG PLUS uses when you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE. REORG PLUS reads, parses, and partially verifies the SYSIDCIN data set in the UTILINIT phase. The IDCAMS commands are issued before REORG PLUS reloads the data set. REORG PLUS uses the statements to redefine the VCAT-defined data sets only.</td>
<td>374</td>
</tr>
<tr>
<td>SYSIN</td>
<td>SYSIN is the input data set that contains the REORG command.</td>
<td>376</td>
</tr>
<tr>
<td>SYSPRIN2</td>
<td>The SYSPRIN2 output data set contains the same REORG PLUS messages that are output to SYSPRINT. In a worklist environment, SYSPRIN2 enables you to view REORG PLUS output in real time.</td>
<td>377</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>The SYSPRINT output data set contains REORG PLUS messages.</td>
<td>376</td>
</tr>
<tr>
<td>SYSPUNCH</td>
<td>The SYSPUNCH data set is used only for DSNUTILB reorganizations and contains LOAD statements that are generated when records are discarded during the reorganization.</td>
<td>378</td>
</tr>
<tr>
<td>SYSREC&lt;sub&gt;nn&lt;/sub&gt;&lt;sup&gt;a&lt;/sup&gt;</td>
<td>The SYSREC data set contains the table space’s unloaded rows. If you use multiple data sets, the &lt;i&gt;nn&lt;/i&gt; identifies a specific partition number. REORG PLUS does not use the SYSREC data set for a single-phase SHRLEVEL REFERENCE or SHRLEVEL CHANGE table space reorganization.</td>
<td>379</td>
</tr>
<tr>
<td>SYSTSPRT</td>
<td>REXX routes all output from the REXX ‘SAY’ statements to the SYSTSPRT data set.</td>
<td>382</td>
</tr>
<tr>
<td>SYSUT1&lt;sub&gt;nn&lt;/sub&gt;&lt;sup&gt;a&lt;/sup&gt;</td>
<td>The SYSUT1 data set contains the unloaded index keys. The &lt;i&gt;nn&lt;/i&gt; is required only if you specify multiple data sets. REORG PLUS does not use the SYSUT1 data set for a single-phase SHRLEVEL REFERENCE or SHRLEVEL CHANGE index reorganization.</td>
<td>382</td>
</tr>
<tr>
<td>UTPRINT&lt;sup&gt;a&lt;/sup&gt;</td>
<td>The UTPRINT data set indicates that sort messages should be reported. However, the actual messages for each sort appear in separate SYS&lt;sub&gt;nnnnn&lt;/sub&gt; data sets, where &lt;i&gt;nnnnn&lt;/i&gt; is a system-assigned sequential number.</td>
<td>386</td>
</tr>
</tbody>
</table>

<sup>a</sup> For reorganization jobs that invoke DSNUTILB, the description of how REORG PLUS uses this data set does not apply. For information about how this data set is used for these jobs, see the documentation for the REORG utility in the IBM DB2 Utility Guide and Reference.
Associated products and common components that REORG PLUS uses

In addition to its own processing components, REORG PLUS uses the common components described in Table 6.

Table 6  Products and components that REORG PLUS uses

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC Common Statistics</td>
<td>The BMC Common Statistics component of DASD MANAGER PLUS provides a common method for updating DB2 catalog statistics and DASD MANAGER PLUS statistics tables. This component also provides statistics reporting. If this component is installed during REORG PLUS installation, but is maintained separately from REORG PLUS.</td>
</tr>
<tr>
<td>BMCSORT</td>
<td>The BMCSORT technology is a common BMC technology. REORG PLUS uses BMCSORT to allocate sort work files and to perform sort processing. If this component is installed during REORG PLUS installation, but is maintained separately from REORG PLUS.</td>
</tr>
<tr>
<td>DB2 Solution Common Code (SCC)</td>
<td>BMC DB2 Solution Common Code (SCC) is a set of technologies that provide common processes for several BMC products for DB2. REORG PLUS uses SCC technologies for such processes as setting object statuses and compressing data. If this component is installed during REORG PLUS installation, but is maintained separately from REORG PLUS.</td>
</tr>
<tr>
<td>DB2 Utilities Common Code (D2U)</td>
<td>D2U is a set of technologies that provides common processes for the BMC Utility products for DB2 and the DASD MANAGER PLUS product. If this component is installed during REORG PLUS installation, but is maintained separately from REORG PLUS.</td>
</tr>
</tbody>
</table>
| EXTENDED BUFFER MANAGER (XBM) or SNAPSHOT UPGRADE FEATURE (SUF) | XBM or SUF provides the following capabilities:  
  - a snapshot image of data in a table space  
  - a snapshot copy of a nonpartitioned index  
  - zIIP processing  
  XBM and SUF are licensed, installed, and maintained separately from REORG PLUS. |
Operational considerations

This chapter presents the following topics:

System setup ................................................................. 65
   DB2 support .................................................................. 65
   System requirements ....................................................... 65
   Software requirements ..................................................... 66
   Required authorization ..................................................... 66
   MEMLIMIT system parameter ........................................... 71
Number of DB2 threads that REORG PLUS uses ....................... 71
Reorganization jobs that invoke DSNUTILB ............................... 72
   Enabling REORG PLUS to invoke DSNUTILB ..................... 75
   Installation options that enable DSNUTILB ....................... 76
   General restrictions for DSNUTILB reorganizations ............... 77
   Data set allocation for DSNUTILB reorganizations ............... 78
   Considerations for other REORG PLUS options when running DSNUTILB reorganizations ........................................... 79
Serialization and concurrency .............................................. 82
   Executing BMC utilities concurrently ................................ 82
   Object status for SHRLEVEL NONE ................................. 84
   Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE .................... 86
Dynamic data set allocation .................................................. 90
   Enabling dynamic allocation .......................................... 91
   Running with dynamic allocation in a worklist environment ........ 91
   Generating data set names .............................................. 92
   Specifying ddname prefixes ........................................... 93
   Deleting dynamically allocated data sets ............................ 94
   Dynamically allocating larger data sets with different properties than smaller data sets ......................................................... 94
   Using SMS ACS routines to influence dynamic allocation .......... 95
   Reaching the MAXTAPE limit during dynamic allocation .......... 95
   Changing dynamic allocation options on restart .................. 95
Work file validity and integrity checks .................................... 96
   Check for data set attributes ........................................... 96
   Check for data integrity .................................................. 98
System setup

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

DB2 support

This version of REORG PLUS supports IBM DB2 Versions 9, 10, and 11.

System requirements

This version of REORG PLUS has the following system requirements:

- IBM System z10® processor or a later 64-bit z/Architecture® processor
- IBM z/OS Version 1.10 or later
Software requirements

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

- REORG PLUS requires a minimum of the following versions of BMC common components:
  - version 2.4.01 of BMCSORT
  - version 11.1.00 of the DB2 Utilities Common Code (D2U)
  - version 11.1.00 with PTF BPJ0689 of the DB2 Solution Common Code (SCC)

- To update DB2 catalog statistics or the DASD MANAGER PLUS database statistics tables, REORG PLUS requires a minimum of version 11.1.00 of the BMC Common Statistics component.

- To use snapshot processing, REORG PLUS requires version 6.1.00 or later of the BMC EXTENDED BUFFER MANAGER (XBM) product or its SNAPSHOT UPGRADE FEATURE (SUF) technology.

  If you use the XBMID option to specify a particular XBM subsystem, that subsystem must be at this maintenance level. If you do not specify a particular XBM subsystem and snapshot processing is required, REORG PLUS searches for an XBM subsystem at this level.

- To offload eligible processing to a zIIP, REORG PLUS requires version 6.1.00 or later of XBM or SUF.

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

- To use any features that invoke DSNUTILB (see page 72), you must be licensed to use the IBM DB2 REORG utility.

Required authorization

Using the REORG PLUS product requires that you have the appropriate authorization within DB2 and through your system security package, such as RACF®. You need sufficient authorization to access resources and perform specified tasks during REORG PLUS processing.
Authorization verification mechanisms

If the DB2 DSNX@XAC authorization exit is available for your system, REORG PLUS uses this exit to verify authorization for external access. The exit is available from the following sources:

- IBM provides a sample exit with DB2 for the IBM Resource Access Control Facility (RACF®) component.
- CA Technologies provides the DSNX@XAC exit with CA-ACF2 Security for DB2 and CA-Top Secret Security for DB2.

BMC recommends this mechanism for implementing external security. The access control authorization exit must be available in the STEPLIB, JOBLIB, linklist, or in the SYS3.DSN exit.

If the DSNX@XAC exit is not available, REORG PLUS uses the standard DB2 method to check security.

DB2 authorization

To run all REORG PLUS jobs, you must have the following DB2 authorizations:

- sufficient DB2 authority to execute the REORG PLUS plan and all packages that the REORG PLUS plan uses
- authorization equivalent to the authorization that the comparable IBM DB2 REORG utility requires
- ALTER INDEX and ALTER TABLE privileges for the database containing the named table space or index (if not implicit in the authority that you have)

**NOTE**

REORG PLUS does not check for the DELETE privilege when you specify the SELECT or DELETE option. REORG PLUS does not check for the UPDATE privilege when you specify the UPDATE option.

Additional authorizations for SHRLEVEL CHANGE

To run a SHRLEVEL CHANGE reorganization, you must have the following privileges:

- TRACE privilege
- MONITOR2 privilege
- DISPLAY privilege (if not already granted to PUBLIC)
These privileges might be implicit in the authority that you have.

Additional authorizations for XML reorganizations

When reorganizing base table spaces that contain XML columns, you must have SELECT privileges on the following DB2 tables:

- SYSIBM.SYSSEQUENCES
- SYSIBM.SYSSEQUENCESDEP

When reorganizing user-defined XML indexes, you must have SELECT privileges on the SYSIBM.SYSXMLRELSES DB2 table.

These privileges might be implicit in the authority that you have.

Additional authorizations for using DSRSEXIT

To use the DSRSEXIT user exit with a value of YES (the default) for the BMC_ALTER_DB2_CATALOG variable (to have REORG PLUS update the DB2 catalog), the following additional requirements apply:

- For the ALTER TABLESPACE statement, you need one of the following authorizations:
  - ownership of the table space
  - DBADM authority for the database that contains the table
  - SYSADM or SYSCTRL authority
  - system DBADM (DB2 Version 10 or later)

- For the ALTER INDEX or ALTER TABLE statement, you need one of the following authorizations:
  - ownership of the index
  - ownership of the table on which the index is defined
  - DBADM authority for the database that contains the table
  - SYSADM or SYSCTRL authority
  - system DBADM (DB2 Version 10 or later)

Additional authorizations for using MAPTEXIT

To use the MAPTEXIT user exit, you must also have authority to create and drop objects on the DSNDB04 database.
**Additional authorizations for canceling threads**

To use the FORCE option to cancel DB2 threads that might prevent a successful drain during a reorganization job, you must have the following authorizations:

- DISPLAY privileges
- one of the following authorities:
  - SYSADM
  - SYSTOPR
  - SYSCTRL

These additional authorizations might be implicit in the authority that you have.

**Additional authorizations for using XBM or SUF**

To enhance performance, during portions of the reorganization process, REORG PLUS uses several features of the EXTENDED BUFFER MANAGER (XBM) product or its SNAPSHOT UPGRADE FEATURE (SUF). For information about security levels and authorizations for XBM, see the *BMC Products and Solutions for DB2 Configuration Guide*.

**Data set authorization**

REORG PLUS does not run as part of the DB2 subsystem. Therefore, you must have system authorization that is equivalent to the authorization that DB2 requires. You can obtain this authorization in one of the following ways:

- If you use RACF, specify OPNDB2ID=YES in your installation options.
- Establish authorization as described in “Establishing authorization when OPNDB2ID=NO” on page 70.

**Using RACF and OPNDB2ID=YES**

If you use RACF and OPNDB2ID=YES is set in REORG PLUS, the user who is running REORG PLUS is not required to have the authorizations that the following section describes. OPNDB2ID=YES tells REORG PLUS to use the DB2 RACF ID instead of the user’s RACF ID.

**NOTE**

Using OPNDB2ID=NO can improve performance, depending on the size of your data set profiles and the number of VSAM data sets that are involved in the reorganization.
Establishing authorization when OPNDB2ID=NO

Some sites use RACF or a similar system security package to protect underlying data sets and the Integrated Catalog Facility (ICF) catalog of a table or index space. In that case, you must have the following minimum levels of authorization:

- ALTER or CONTROL to access, update, and define DB2 data sets
- UPDATE or CONTROL to access and update the ICF catalog

If you establish authorizations at a node lower than the highest node, you must have the same privileges for the data sets that REORG PLUS uses during the renaming process for SHRLEVEL CHANGE and SHRLEVEL REFERENCE. These data sets vary depending on whether you are using the BMC naming convention (STAGEDSN=BMC) or the I/J naming convention (STAGEDSN=DSN), as follows:

- For STAGEDSN=BMC:
  - VCAT.BMCDBC.database.object.I0001
  - VCAT.BMCDBD.database.object.I0001
  - VCAT.OLDDBC.database.object.I0001
  - VCAT.OLDDBD.database.object.I0001
  - VCAT.BMCDBD.database.object.J0001
  - VCAT.BMCDBC.database.object.J0001
  - VCAT.OLDDBD.database.object.J0001
  - VCAT.OLDDBC.database.object.J0001

- For STAGEDSN=DSN (the default when you use the FASTSWITCH process):
  - VCAT.DSNDBC.database.object.I0001
  - VCAT.DSNDBD.database.object.I0001
  - VCAT.DSNDBC.database.object.J0001
  - VCAT.DSNDBD.database.object.J0001
  - VCAT.DSNDBC.database.object.S0001
  - VCAT.DSNDBD.database.object.S0001

For more information, see “Staging data sets and the rename process” on page 103 and “Staging data sets and the FASTSWITCH process” on page 105.
Using a security package other than RACF

The following procedure illustrates one method for granting these data set authorizations when your site uses a system security package other than RACF:

1. Associate users with a security group.
2. Grant EXECUTE privileges on the REORG PLUS product program (ARUUMAIN) to the security group.
3. Grant the minimum data set authority levels to ARUUMAIN, described in “Establishing authorization when OPNDB2ID=NO” on page 70.

MEMLIMIT system parameter

REORG PLUS requires above-the-bar memory and might abend if sufficient memory is not available. The default value for the System Management Facility (SMF) MEMLIMIT parameter is 2 GB.

This value is set in member SMFPRMxx in SYS1.PARMLIB. Use any of the following methods if you need to override the default value:

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

If you are unable to specify REGION=0M, BMC makes the following recommendations for the MEMLIMIT option:

- Specify NOLIMIT to allow unlimited above-the-bar memory.
- If you are unable to specify NOLIMIT, specify at least 4 GB. If you are reorganizing LOB or XML data, specify at least 32 GB.

Number of DB2 threads that REORG PLUS uses

The maximum number of batch DB2 threads that REORG PLUS uses concurrently is seven per job. REORG PLUS uses up to seven threads during the UTILINIT phase and two threads for the duration of the job.

For a job that invokes DSNUTILB, DSNUTILB uses additional threads. REORG PLUS does not use any additional threads for a DSNUTILB job.
Reorganization jobs that invoke DSNUTILB

REORG PLUS enables certain features by invoking DSNUTILB. This section describes considerations that apply when REORG PLUS invokes DSNUTILB.

Features that require DSNUTILB

For this version, REORG PLUS invokes DSNUTILB to enable the following features.

**NOTE**

REORG PLUS invokes DSNUTILB to enable new features quickly. REORG PLUS generally provides native support for these features (without invoking DSNUTILB) in a later version of the product or via PTF. You cannot use the DSNUTILB option (or any other option) to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively.

DB2 catalog objects

REORG PLUS invokes DSNUTILB when reorganizing the following DB2 catalog (DSNDB06 database) objects:

- the SYSCOPY table space
- catalog table spaces that contain LOB columns
- LOB catalog table spaces
- the SYSTSIPT table space if any of its indexes are versioned

REORG PLUS natively reorganizes all other table spaces and indexes in the catalog database.

LOB objects

REORG PLUS invokes DSNUTILB when reorganizing LOB objects in the following cases:

- reorganizing LOB table spaces when SHRLEVEL NONE or SHRLEVEL CHANGE is in effect
- reorganizing LOB table spaces and the base table space in the same job step:
  - You specify AUX YES.
— You do not specify the AUX option and any of the following conditions exist. (In this case, the DB2 REORG utility defaults to AUX YES.)

- You specify SHRLEVEL REFERENCE and REBALANCE.
- You are reorganizing more than one partition of a partition-by-growth table space.
- You specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE, and any partition is in REORP status.
- Reorganizing partition-by-growth table spaces that contain a LOB column.
- Reorganizing table spaces in which the length of an inline LOB column has been altered.
- Reorganizing indexes on an inline LOB column, or reorganizing table spaces that contain those indexes.

For more information about reorganizing LOB objects, see “LOB data” on page 116.

**Additional types of table spaces**

REORG PLUS invokes DSNUTILB to reorganize the following additional types of table spaces:

- Table spaces that contain tables that are defined as ORGANIZE BY HASH.
  REORG PLUS natively reorganizes indexes on tables that are defined as ORGANIZE BY HASH.

- Table spaces that contain pending DDL changes in those cases when the IBM DB2 REORG utility would materialize those changes.

  If the DB2 REORG utility would *not* materialize the pending changes, REORG PLUS reorganizes the table space natively but does not materialize the changes.

- Table spaces that contain columns that are defined with a CCSID specification.

- Table spaces that contain any of the following types of indexes:

  - Indexes that contain a key derived from certain expressions.

  REORG PLUS natively supports some indexes on expression. For more information, see “Indexes on expression” on page 148.
Reorganization jobs that invoke DSNUTILB

— indexes that contain keys with random ordering
— indexes that are defined with EXCLUDE NULL KEYS

Additional types of indexes

REORG PLUS invokes DSNUTILB to reorganize the following additional types of indexes:

- indexes that contain pending DDL changes in those cases when the IBM DB2 REORG utility would materialize those changes

  If the DB2 REORG utility would not materialize the pending changes, REORG PLUS reorganizes the index natively but does not materialize the changes.

- indexes that contain a key derived from certain expressions, including those on inline LOB columns and those on columns that are defined with a CCSID specification

  REORG PLUS natively supports some indexes on expression. For more information, see “Indexes on expression” on page 148.

- indexes that contain keys with random ordering

Additional types of reorganizations

REORG PLUS invokes DSNUTILB when any of the following conditions exists:

- specification of the RBALRSN_CONVERSION option when either of the following sets of conditions exists:

  — You specify SHRLEVEL CHANGE and request conversion to a format that is different from the current format of the object.

  — You specify RBALRSN_CONVERSION EXTENDED and all of the following conditions exist:

    ■ You are reorganizing a versioned XML table space.
    ■ The START_TS and END_TS column is currently defined as BINARY(8).
    ■ You are reorganizing all partitions.

- specification of the REBALANCE option when both of the following conditions exist:

  — You are running on DB2 Version 11.
  — You are reorganizing a table space that is defined with table-controlled partitioning.
specification of noncontiguous partitions when all of the following conditions exist:

— You are reorganizing a table space whose partition limit keys were altered before the reorganization.

— You are not rebalancing partitions during the reorganization.

— Your version of DB2 supports the ability to specify noncontiguous partitions.

### Enabling REORG PLUS to invoke DSNUTILB

To enable REORG PLUS to invoke DSNUTILB, perform the following tasks:

1. Ensure that you are licensed to use the IBM DB2 REORG utility.

2. For each feature that requires invoking DSNUTILB, ensure that the installation option that enables the feature is set to the appropriate enablement value. Table 7 on page 76 lists the option associated with each feature.

3. Ensure that the following installation or command options are set:

<table>
<thead>
<tr>
<th>Option</th>
<th>Required setting</th>
<th>More information</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUTILB</td>
<td>YES</td>
<td>page 237 or page 689</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>YES for the following DDTYPES:</td>
<td>page 296 or page 742</td>
</tr>
<tr>
<td></td>
<td>— all work file DDTYPES that the reorganization job requires</td>
<td></td>
</tr>
<tr>
<td></td>
<td>— (if you specify COPY YES) the LOCPFCPY DDTYPE and other copy DDTYPES if needed</td>
<td></td>
</tr>
<tr>
<td>SORTNUM</td>
<td>2 or greater</td>
<td>page 188 or page 725</td>
</tr>
</tbody>
</table>

4. For a SHRLEVEL CHANGE table space reorganization, ensure that a mapping table or database is specified and available:

   — For DB2 Version 9 or 10, complete one of the following actions:

     — Specify an existing mapping table with the MAPPINGTABLE command option (page 340).
— Provide a user exit that will create and drop the mapping objects for you; then, supply the user exit name in the MAPEXIT installation or command option.

For more information about the MAPEXIT user exit, see Appendix C, “REORG PLUS user exits.” For more information about the MAPEXIT options, see page 243 or page 709.

- (optional) For DB2 Version 11 or later, specify an existing mapping table or mapping database with the MAPPINGTABLE command option (page 340) or MAPPINGDATABASE command option (page 340).

5 If you anticipate that the IBM DB2 REORG utility will need data sets for discarded rows and for LOAD control cards for those discarded rows, ensure that dynamic allocation is enabled for the ARCHIVE and SYSPUNCH DDTYPES.

If the DB2 REORG utility needs these data sets and they are not dynamically allocated in your REORG PLUS job, the job terminates.

6 Use the information in the rest of this section to ensure that you create appropriate JCL for your job.

For existing jobs, you might need to make changes to your JCL. For example, Table 9 on page 80 describes which options will cause your job to fail when invoking DSNUTILB.

Installation options that enable DSNUTILB

In addition to specifying DSNUTILB=YES, you must specify the enablement value for the installation options shown in Table 7 to enable the features for which REORG PLUS invokes DSNUTILB.

**NOTE**

In a future release, REORG PLUS will no longer require or support the individual installation options for each feature that requires invoking DSNUTILB. You must still specify DSNUTILB=YES.

| Table 7 Installation options that enable DSNUTILB features (part 1 of 2) |
| --- | --- | --- | --- |
| Feature | Installation option | Enablement value | See page |
| catalog table spaces and indexes that are not supported natively | none | not applicable | not applicable |
| index columns that contain a CCSID specification | IXONEX | YES | 702 |
Chapter 2 Operational considerations 77

Table 7 Installation options that enable DSNUTILB features (part 2 of 2)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Installation option</th>
<th>Enablement value</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>indexes that contain a key derived from an expression when not supported natively</td>
<td>IXONEX</td>
<td>YES</td>
<td>702</td>
</tr>
<tr>
<td><strong>Note</strong>: This option includes indexes on inline LOB columns and indexes on columns that contain a CCSID specification. These indexes contain a key that is derived from an unsupported expression.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>indexes that contain keys with random ordering</td>
<td>IXRANDOM</td>
<td>YES</td>
<td>702</td>
</tr>
<tr>
<td>indexes that contain pending definition changes</td>
<td>PENDDDL</td>
<td>YES</td>
<td>715</td>
</tr>
<tr>
<td>LOB objects when not supported natively</td>
<td>LOB</td>
<td>YES</td>
<td>704</td>
</tr>
<tr>
<td>partitions that are not contiguous, when allowed under certain conditions</td>
<td>none</td>
<td>not applicable</td>
<td>not applicable</td>
</tr>
<tr>
<td>RBALRSN_CONVERSION option when not supported natively</td>
<td>none</td>
<td>not applicable</td>
<td>not applicable</td>
</tr>
<tr>
<td>REBALANCE option when reorganizing a table space defined with table-controlled partitioning on DB2 Version 11</td>
<td>none</td>
<td>not applicable</td>
<td>not applicable</td>
</tr>
<tr>
<td>table columns that contain a CCSID specification</td>
<td>UTILB_COLCCSID</td>
<td>UTILB</td>
<td>736</td>
</tr>
<tr>
<td><strong>Note</strong>: If a participating index contains columns with a CCSID specification, also specify IXONEX=YES.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>table spaces that contain indexes that are defined with EXCLUDE NULL KEYS</td>
<td>UTILB_NULLIX</td>
<td>UTILB</td>
<td>736</td>
</tr>
<tr>
<td>table spaces that contain pending definition changes</td>
<td>PENDDDL</td>
<td>YES</td>
<td>715</td>
</tr>
<tr>
<td><strong>Note</strong>: For pending changes that affect LOB objects or inline LOB columns, specify LOB=YES.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tables defined as ORGANIZE BY HASH</td>
<td>HASHAX</td>
<td>YES</td>
<td>696</td>
</tr>
</tbody>
</table>

General restrictions for DSNUTILB reorganizations

Note the following general restrictions when running a DSNUTILB reorganization:

- REORG PLUS invokes DSNUTILB after the UTILINIT phase. Therefore, most of the processing that REORG PLUS normally does to validate and copy your data is handled by the IBM DB2 REORG utility for these jobs. Functionality is limited to those features that are available in the DB2 REORG utility.

- The statement that REORG PLUS passes to DSNUTILB is limited to 32704 characters. This statement contains the utility command options and the TEMPLATE control statements for dynamically allocating your data sets.

- Do not specify FREE=CLOSE for your SYSPRINT data set.
- REORG PLUS ignores any SYSPRIN2 DD statement in your JCL.

- If any table in the table space that you are reorganizing uses a feature that REORG PLUS supports via DSNUTILB, REORG PLUS invokes DSNUTILB for that job.

- When you specify DELETEFILES YES, REORG PLUS deletes SYSREC and SYSUT1 data sets only when the reorganization completes successfully.

### Data set allocation for DSNUTILB reorganizations

When REORG PLUS invokes DSNUTILB, you must enable dynamic allocation for the following data sets by specifying ACTIVE YES for those DDTYPEs:

- all work data sets (WORK or UNLOAD DDTYPEs) that your job requires
- if you specify COPY YES, at least the primary local copy data set

Additionally, if the IBM DB2 REORG utility job requires data sets for discarded rows and for LOAD control statements for those discarded rows, you must enable dynamic allocation for the ARCHIVE and SYSPUNCH DDTYPEs.

If you specify any of these data sets in your JCL, REORG PLUS ignores them, regardless of your IFALLOC specification.

With the exceptions described in Table 8, REORG PLUS uses your dynamic allocation options to generate a TEMPLATE control statement for each data set. REORG PLUS then passes these TEMPLATE statements to DSNUTILB for data set allocation.

### Table 8 Dynamic allocation options considerations for a DSNUTILB reorganization (part 1 of 2)

<table>
<thead>
<tr>
<th>Command option</th>
<th>Considerations</th>
</tr>
</thead>
</table>
| ACTIVE         | requires a value of YES for all work files, and for at least the primary copy data set if you specify COPY YES  
                  - If ACTIVE is NO for these data sets, REORG PLUS issues message BMC50178E and terminates.  
                  - Additionally, if the DB2 REORG utility requires the discard and LOAD control card data sets, you must specify YES for the ARCHIVE and SYSPUNCH DDTYPEs. If these data sets are required but are not dynamically allocated, the reorganization terminates. |
| ALLOC          | ignores this option  
                  - DSNUTILB dynamically allocates SORTWK DD names. |
| AVGVOLSP       | ignores this option |
Considerations for other REORG PLUS options when running DSNUTILB reorganizations

Table 8 Dynamic allocation options considerations for a DSNUTILB reorganization (part 2 of 2)

<table>
<thead>
<tr>
<th>Command option</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNPAT</td>
<td>converts symbolic variables when an equivalent exists</td>
</tr>
<tr>
<td></td>
<td>If you specify a symbolic variable that does not have an equivalent, you will receive a DSNUTILB error. For information about which symbolic variables are converted, see Table 44 on page 316.</td>
</tr>
<tr>
<td></td>
<td>Also, be aware that user-defined variables are not valid for a DSNUTILB reorganization.</td>
</tr>
<tr>
<td>DSNTYPE</td>
<td>ignores this option</td>
</tr>
<tr>
<td>IFALLOC</td>
<td>ignores this option</td>
</tr>
<tr>
<td></td>
<td>If you specify DD statements in your JCL for any of the data sets required for a job, REORG PLUS ignores them. If your JCL contains a data set with the same name as the one that REORG PLUS generates for dynamic allocation, you might encounter a contention error.</td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td>translates this option to the MAXPRIME keyword of the TEMPLATE control statement</td>
</tr>
<tr>
<td>SIZEPCT</td>
<td>translates this option to the PCTPRIME keyword of the TEMPLATE control statement</td>
</tr>
<tr>
<td></td>
<td>If you specify a value greater than 100, REORG PLUS converts it to 100.</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>for copy data sets, translates this option to a LIMIT value in the TEMPLATE control statement</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS also builds the secondary TEMPLATE control statement to which the DB2 REORG utility will switch when the LIMIT value is exceeded.</td>
</tr>
<tr>
<td></td>
<td>for all other data sets, ignores this option</td>
</tr>
</tbody>
</table>

Considerations for other REORG PLUS options when running DSNUTILB reorganizations

For a DSNUTILB reorganization, REORG PLUS responds in one of the following ways when an option is not valid for the IBM DB2 REORG utility:

- ignores the option
- terminates with a message about the option
- translates the option to a valid DB2 REORG option and passes the translated option

In most other cases, REORG PLUS passes options that are valid for both REORG PLUS and the DB2 REORG utility.

Table 9 on page 80 lists the options that are not valid or that are translated, and describes how REORG PLUS responds if your job contains these options. For additional information about dynamic allocation options, see “Data set allocation for DSNUTILB reorganizations” on page 78.
### Table 9 Options that are incompatible or translated for a DSNUTILB reorganization (part 1 of 3)

<table>
<thead>
<tr>
<th>Command option</th>
<th>REORG PLUS response if you include the option</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE ONLY</td>
<td>issues message BMC50178E and terminates</td>
</tr>
<tr>
<td>ANALYZE (all others)</td>
<td>ignores this option</td>
</tr>
<tr>
<td>AMENDED</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ARCHFORMAT</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ARCROWS</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ASSOCIATE</td>
<td>ignores this option</td>
</tr>
<tr>
<td>AVAILPAGEPCT</td>
<td>ignores this option</td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>ignores this option</td>
</tr>
<tr>
<td>BMCHIST</td>
<td>ignores this option</td>
</tr>
<tr>
<td>BMCSTATS</td>
<td>ignores this option</td>
</tr>
<tr>
<td>COPY YES</td>
<td>invokes the copy function</td>
</tr>
<tr>
<td>COPYLVL PART</td>
<td>if COPY YES, converts to COPYLVL FULL</td>
</tr>
<tr>
<td>DDLDDN</td>
<td>ignores this option</td>
</tr>
<tr>
<td>DEADLINE</td>
<td>terminates if you include a time zone; otherwise, passes this option</td>
</tr>
<tr>
<td>DELETE</td>
<td>issues message BMC50178E and terminates</td>
</tr>
<tr>
<td>DSNUEXIT</td>
<td>ignores this option</td>
</tr>
<tr>
<td>DSPLOCKS</td>
<td>ignores this option</td>
</tr>
<tr>
<td>DSRSEXIT</td>
<td>ignores this option</td>
</tr>
<tr>
<td>FORCE</td>
<td>ignores this option</td>
</tr>
<tr>
<td>FSFALLBACK</td>
<td>ignores this option</td>
</tr>
<tr>
<td>FSTHRESHOLD</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ICDDN</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ICTYPE INCREMENTAL</td>
<td>if COPY YES, converts to ICTYPE UPDATE</td>
</tr>
<tr>
<td>IDCACHE</td>
<td>ignores this option</td>
</tr>
<tr>
<td>IDCDDN</td>
<td>ignores this option</td>
</tr>
<tr>
<td>IMPLICIT_TZ</td>
<td>ignores this option</td>
</tr>
<tr>
<td>INLINE NO</td>
<td>ignores this option</td>
</tr>
<tr>
<td>KEEPDICTIONARY</td>
<td>NO: ignores this option</td>
</tr>
<tr>
<td></td>
<td>YES:</td>
</tr>
<tr>
<td></td>
<td>■ if reorganizing a LOB table space using SHRLLEVEL CHANGE, ignores this option</td>
</tr>
<tr>
<td></td>
<td>■ otherwise, passes as KEEPDICTIONARY</td>
</tr>
<tr>
<td>LOGMEM</td>
<td>ignores this option</td>
</tr>
<tr>
<td>LOGSPILL</td>
<td>ignores this option</td>
</tr>
<tr>
<td>LOGTHRESHLD</td>
<td>ignores this option</td>
</tr>
<tr>
<td>LONGNAMETRUNC</td>
<td>ignores this option</td>
</tr>
</tbody>
</table>
Table 9  Options that are incompatible or translated for a DSNUTILB reorganization (part 2 of 3)

<table>
<thead>
<tr>
<th>Command option</th>
<th>REORG PLUS response if you include the option</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXNEWPARTS</td>
<td>ignores this option</td>
</tr>
<tr>
<td>MAXSORTMEMORY</td>
<td>ignores this option</td>
</tr>
<tr>
<td>MAXSORTS</td>
<td>ignores this option</td>
</tr>
<tr>
<td>MAXTAPE</td>
<td>ignores this option</td>
</tr>
<tr>
<td>MINSORTMEMORY</td>
<td>ignores this option</td>
</tr>
<tr>
<td>NLPCTFREE</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ON FAILURE</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ON MESSAGE</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ORDER YES ASSOCIATE BYTABLE</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ORIGINALDISP</td>
<td>ignores this option</td>
</tr>
<tr>
<td>PREFORMAT</td>
<td>NO: ignores this option</td>
</tr>
<tr>
<td></td>
<td>YES: passes as PREFORMAT</td>
</tr>
<tr>
<td>REBALANCE ON (n) COLUMNS</td>
<td>converts to REBALANCE</td>
</tr>
<tr>
<td>RECOVERYICDDN</td>
<td>ignores this option</td>
</tr>
<tr>
<td>REDEFINE YES</td>
<td>passes as REUSE</td>
</tr>
<tr>
<td>REGISTER</td>
<td>ignores this option</td>
</tr>
<tr>
<td>RIDMAPMEM</td>
<td>ignores this option</td>
</tr>
<tr>
<td>SELECT</td>
<td>issues message BMC50178E and terminates</td>
</tr>
<tr>
<td>SHORTMEMORY</td>
<td>ignores this option</td>
</tr>
<tr>
<td>SHRLEVEL REFERENCE</td>
<td>issues message BMC50178E and terminates</td>
</tr>
<tr>
<td>UNLOADONLY</td>
<td></td>
</tr>
<tr>
<td>SIXSNAP</td>
<td>ignores this option</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>if reorganizing a LOB table space, ignores this option</td>
</tr>
<tr>
<td>SPILLDSPAT</td>
<td>ignores this option</td>
</tr>
<tr>
<td>SPILLSTORCLAS</td>
<td>ignores this option</td>
</tr>
<tr>
<td>SPILLUNIT</td>
<td>ignores this option</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>ignores this option</td>
</tr>
<tr>
<td>SYNC</td>
<td>ignores this option</td>
</tr>
<tr>
<td>TIMEOUT TERM, (rc)</td>
<td>ignores this option</td>
</tr>
<tr>
<td>TERMEXIT</td>
<td>ignores this option</td>
</tr>
<tr>
<td>TOTALPAGEPCT</td>
<td>ignores this option</td>
</tr>
<tr>
<td>TSSAMPLEPCT</td>
<td>ignores this option</td>
</tr>
<tr>
<td>UPDATE</td>
<td>issues message BMC50178E and terminates</td>
</tr>
<tr>
<td>UPDATEDB2STATS YES</td>
<td>passes as STATISTICS TABLE (ALL) INDEX (ALL) REPORT YES UPDATE ALL</td>
</tr>
<tr>
<td>WTOMSG</td>
<td>ignores this option</td>
</tr>
</tbody>
</table>
Serialization and concurrency

This section discusses object status requirements and concurrency issues, which can vary with the command statement specifications. If an object is not in the allowed initial status, REORG PLUS issues a message and terminates the job.

Executing BMC utilities concurrently

All BMC Software utility products use the BMCUTIL table to control the use of utility IDs, which identify executions of BMC utilities. Each BMC utility product must have a unique ID for restart purposes. This unique ID is stored in the BMCUTIL table. For more information about this table, see “BMCUTIL table” on page 778.

BMC utility jobs register DB2 objects in the BMCSYNC table. The registering utility assigns a sharing level to each registered object. The sharing level controls access to that object from other BMC utilities. For partitioned DB2 spaces, registration is performed at the partition level.

The BMCSYNC table allows multiple BMC utilities (or multiple instances of a utility) to operate concurrently on different partitions of a DB2 space if no nonpartitioning indexes are involved. In addition, some BMC utilities can operate concurrently on the same object or partition. For information about which products can operate concurrently, see Table 10 on page 83. For additional serialization and concurrency issues for each utility, see that utility’s reference manual.

The “Access level” column in Table 10 refers to the value of the SHRLEVEL column in the BMCSYNC table. The level can be one of the following values:

- S indicates shared access. Any other utility that registers with shared access (S) can run against the object.
- X indicates exclusive access. No other utility can run against the object.
- A blank value indicates that no status is requested and any other utility can run against the object.

### Table 9 Options that are incompatible or translated for a DSNUTILB reorganization (part 3 of 3)

<table>
<thead>
<tr>
<th>Command option</th>
<th>REORG PLUS response if you include the option</th>
</tr>
</thead>
<tbody>
<tr>
<td>XBMD</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ZIIP</td>
<td>ignores this option</td>
</tr>
</tbody>
</table>
## Executing BMC utilities concurrently

<table>
<thead>
<tr>
<th>Product</th>
<th>Access level</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK PLUS</td>
<td>S</td>
<td>none</td>
</tr>
<tr>
<td>COPY PLUS</td>
<td>S or blank</td>
<td>If you specify COPY IMAGECOPY, COPY PLUS registers the object with no access status (blank). Otherwise, COPY PLUS registers the object with shared access (S).</td>
</tr>
<tr>
<td>DASD MANAGER PLUS (BMCSTATS)</td>
<td>S</td>
<td>none</td>
</tr>
<tr>
<td>LOADPLUS</td>
<td>X</td>
<td>If you specify PART, LOADPLUS registers only the specified partitions with exclusive access (X). If no nonpartitioned indexes exist on the table space, you can run other utilities on different partitions while running this job.</td>
</tr>
</tbody>
</table>
| RECOVER PLUS                  | X, S, or blank | RECOVER PLUS registers an object with shared access (S) under the following conditions:  
  - The table space for an index is registered with shared access if the index is being rebuilt and its table space is not recovered in the same job.  
  - A table space partition is registered with shared access if the keys for that partition are unloaded with a RECOVER UNLOADKEYS operation.  
  RECOVER PLUS registers an object with no access status (blank) if you specify the following commands or options:  
    - the ACCUM command  
    - OUTCOPY ONLY  
    - INDEP OUTSPACE  
  RECOVER PLUS registers the object with exclusive access (X) in all other cases. |
| RECOVERY MANAGER              | S            | none                   |
| REORG PLUS                    | X            | If you specify PART, REORG PLUS registers only the specified partitions with exclusive access (X). If no nonpartitioned indexes exist on the table space, you can run other utilities on different partitions while running this job. |
| UNLOAD PLUS                   | S            | none                   |

**WARNING**

Do not run an IBM utility, command, or SQL statement that attempts to manipulate the structure, data, or status of an object that a BMC utility is currently processing. For example, commands and SQL statements such as -STOP, -START, EXCHANGE, and ALTER will produce unpredictable results.
The setting of the LOCKROW installation option determines whether REORG PLUS uses MVS™ enqueues or SQL LOCK TABLE statements to serialize the BMCSYNC and BMCUTIL tables. For information about the LOCKROW option, see Appendix A, “REORG PLUS installation options.”

Object status for SHRLEVEL NONE

For SHRLEVEL NONE reorganizations (the default), this section describes the initial status requirements and how REORG PLUS changes the status of the objects during and after the reorganization.

**NOTE**
When REORG PLUS invokes DSNUTILB, status checking and changing is handled by DSNUTILB. For information about status handling for these jobs, see the documentation for the REORG utility in the IBM DB2 Utility Guide and Reference.

Initial status

The database that is associated with the objects that are participating in the reorganization must have one of the following initial statuses:

Table 11 Database initial status requirements for SHRLEVEL NONE

<table>
<thead>
<tr>
<th>Condition</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>you are using a DDLIN data set or the REBALANCE command option</td>
<td>read/write (RW)</td>
</tr>
</tbody>
</table>
| all others | one of the following statuses:  
  ■ RW  
  ■ read-only (RO) |

The objects that you are reorganizing can be in any status except those that Table 12 describes. An X in the table column indicates that the reorganization fails if the object is in the listed status for that type of reorganization.

Table 12 Restrictive statuses that are not permitted for SHRLEVEL NONE (part 1 of 2)

<table>
<thead>
<tr>
<th>Status that is not permitted</th>
<th>Reorganizations that fail (X)</th>
<th>Full table space</th>
<th>Partial table space</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREST (advisory restart pending)</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DBETE (DBET error)</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Table 12  Restrictive statuses that are not permitted for SHRLEVEL NONE
(part 2 of 2)

<table>
<thead>
<tr>
<th>Status that is not permitted</th>
<th>Reorganizations that fail (X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRECP (group RECOVER pending)</td>
<td>X(^a)</td>
</tr>
<tr>
<td>LPL (logical page list)</td>
<td>X</td>
</tr>
<tr>
<td>PRO (persistent read only)</td>
<td>X(^a)</td>
</tr>
<tr>
<td>PSRBD (page set REBUILD pending)</td>
<td>X</td>
</tr>
<tr>
<td>RBDP (REBUILD pending)</td>
<td>X</td>
</tr>
<tr>
<td>RBDP(^*) (logical part REBUILD pending)</td>
<td>X</td>
</tr>
<tr>
<td>RECP (RECOVER pending)</td>
<td>X(^a)</td>
</tr>
<tr>
<td>REFP (refresh pending)</td>
<td>X</td>
</tr>
<tr>
<td>RESTP (restart pending)</td>
<td>X</td>
</tr>
<tr>
<td>RREPL (read-or-replication-only)</td>
<td>X</td>
</tr>
<tr>
<td>UTRO (utility restrictive state, read-only access allowed)</td>
<td>X</td>
</tr>
<tr>
<td>UTRW (utility restrictive state, read/write access allowed)</td>
<td>X</td>
</tr>
<tr>
<td>UTUT (utility restrictive state, utility exclusive control)</td>
<td>X</td>
</tr>
<tr>
<td>WEPR (write page error range)</td>
<td>X</td>
</tr>
</tbody>
</table>

\(^a\) The table space cannot be in this status, but the associated indexes can be.

\(^b\) The reorganization terminates if one or more of the partitions being reorganized is in this status.

**Status changes for SHRLEVEL NONE**

Table 13 on page 86 lists status changes (for the database and the table space or index space) that occur during specific phases of a SHRLEVEL NONE reorganization. The following additional considerations apply:

- REORG PLUS does not set CHECK pending status for reorganizations that result in referential integrity violations.

- For a partial reorganization, REORG PLUS stops and starts only partitions that you specify with the PART option of the REORG command. REORG PLUS starts or stops nonpartitioned indexes in their entirety.
For SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE reorganizations, this section describes the initial status requirements and how REORG PLUS changes the status of the objects during and after the reorganization.

**NOTE**
When REORG PLUS invokes DSNUTILB, status checking and changing is handled by DSNUTILB. For information about status handling for these jobs, see the documentation for the REORG utility in the IBM DB2 Utility Guide and Reference.

### Initial status

The database that is associated with the objects that are participating in the reorganization must have one of the following initial statuses:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>you are using a DDLIN data set or the REBALANCE command option</td>
<td>read/write (RW)</td>
</tr>
</tbody>
</table>
Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE

The objects that you are reorganizing can be in any status except those that Table 15 describes. If the object is in a status that REORG PLUS does not permit, REORG PLUS terminates. An X in a column in Table 15 indicates the following information:

- For a full table space reorganization, the table space cannot be in that restrictive status (but any indexes, if applicable, can be unless otherwise noted).
- For a partial table space reorganization, the partitions that are participating in the reorganization and any nonpartitioned indexes (if applicable) cannot be in that restrictive status (but partitioned indexes, if applicable, can be).
- For an index reorganization, the index cannot be in that restrictive status.

Table 14  Database initial status requirements for SHRLEVEL NONE (part 2 of 2)

<table>
<thead>
<tr>
<th>Condition</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>all other SHRLEVEL CHANGE</td>
<td>RW</td>
</tr>
<tr>
<td>all other SHRLEVEL REFERENCE UNLOADONLY or SHRLEVEL REFERENCE</td>
<td>one of the following statuses: RW read-only (RO)</td>
</tr>
</tbody>
</table>

Table 15  Statuses that are not permitted for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE (part 1 of 2)

<table>
<thead>
<tr>
<th>Status that is not permitted</th>
<th>SHRLEVEL REFERENCE UNLOADONLY</th>
<th>SHRLEVEL REFERENCE</th>
<th>SHRLEVEL CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREST (advisory restart pending)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GRECP (group RECOVER pending)</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LPL (logical page list)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PRO (persistent read only)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PSRBD (page set REBUILD pending)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RBDP (REBUILD pending)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RBDP* (logical part REBUILD pending)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RECP (RECOVER pending)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>REFP (refresh pending)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RESTP (restart pending)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RO (read-only)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RREPL (read-or-replication-only)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>UTRO (utility restrictive state, read-only access allowed)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>UTRW (utility restrictive state, read/write access allowed)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE

Table 15  Statuses that are not permitted for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE (part 2 of 2)

<table>
<thead>
<tr>
<th>Status that is not permitted</th>
<th>SHRLEVEL REFERENCE UNLOADONLY</th>
<th>SHRLEVEL REFERENCE</th>
<th>SHRLEVEL CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTUT (utility restrictive state, utility exclusive control)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>WEPR (write page error range)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

\( a \) The reorganization terminates if one or more of the partitions being reorganized is in this status.

\( b \) When reorganizing a LOB table space, REORG PLUS also fails if the associated auxiliary index is in this status.

Status changes for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE

Table 16 through Table 18 on page 90 list status changes (for the database and the table space or index space) that occur during specific phases of a SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, or SHRLEVEL CHANGE reorganization. The following additional considerations apply:

- For SHRLEVEL REFERENCE UNLOADONLY and SHRLEVEL REFERENCE, for a partial reorganization, REORG PLUS starts or stops only partitions that you specify with the PART option of the REORG command. REORG PLUS starts or stops nonpartitioned indexes in their entirety.

- REORG PLUS does not set CHECK pending status for reorganizations that result in referential integrity violations.

Table 16  Status changes during a SHRLEVEL REFERENCE UNLOADONLY reorganization

<table>
<thead>
<tr>
<th>Phase</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILINIT and ANALYZE</td>
<td>The status of the spaces remains RW, RO, or UT (the initial status).</td>
</tr>
<tr>
<td>beginning of UNLOAD</td>
<td>REORG PLUS starts the table space and all associated indexes (for a table space reorganization) or the index space (for an index reorganization) in RO status.</td>
</tr>
</tbody>
</table>
| beginning of RELOAD | - REORG PLUS stops the table space and associated indexes (for a table space reorganization) or the index space (for an index reorganization).  
|                     | - Processing continues in the RELOAD and UTILTERM phases as with SHRLEVEL NONE. |
Table 17  Status changes during a SHRLEVEL REFERENCE reorganization

<table>
<thead>
<tr>
<th>Phase</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILINIT and ANALYZE</td>
<td>The status of the spaces remains RW, RO, or UT (the initial status).</td>
</tr>
<tr>
<td>beginning of UNLOAD (two-phase reorganization) or beginning of REORG (single-phase reorganization)</td>
<td>REORG PLUS starts the table space and all associated indexes (for a table space reorganization) or the index space (for an index reorganization) in RO status.</td>
</tr>
</tbody>
</table>
| beginning of UTILTERM                      | ■ REORG PLUS stops and sets the RECOVER pending status of the table space and all associated index spaces (for a table space reorganization) or the index space (for an index reorganization).  
  ■ After both of the following steps complete, REORG PLUS starts the table space and index spaces in the status that they were in when the reorganization began, and resets the RECOVER pending status:  
    — the rename or FASTSWITCH operations associated with the staging data sets complete  
    — the REORG or COPY row is registered in the SYSIBM.SYSCOPY catalog table  
  ■ If you specify COPY YES, REORG PLUS resets the COPY pending status on the table space. If you specify COPY NO, REORG PLUS sets the COPY pending status on the table space. In either case, REORG PLUS also sets any indexes defined with COPY YES to informational COPY pending (ICOPY) status. |
### Dynamic data set allocation

You activate dynamic data set allocation through command syntax or installation option defaults. When dynamic allocation is active, REORG PLUS calculates the optimal size and number of data sets and allocates them for you. Dynamic allocation reduces or eliminates the need to specify DD statements for these files in your JCL. REORG PLUS also optionally deletes the work files.

With dynamic allocation, you spend less time performing analysis to set up optimized JCL for REORG PLUS jobs. Also, you will not need to modify the JCL for the REORG PLUS job as DB2 objects change size or structure over time.

---

**Table 18  Status changes during a SHRLEVEL CHANGE reorganization**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILINIT and ANALYZE</td>
<td>The status of the spaces remains RW, RO, or UT (the initial status).</td>
</tr>
<tr>
<td>beginning of LOGFINAL</td>
<td>REORG PLUS prevents updates to the affected table and index spaces while applying the last of the log records.(^a)</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS either creates the incremental copy data sets (if incremental copies are being taken) or updates the full copies, depending on the ICTYPE option.(^a)</td>
</tr>
<tr>
<td>beginning of UTILTERM</td>
<td>REORG PLUS prevents any access to the objects.</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS sets the RECOVER pending status on the table space and all associated index spaces (for a table space reorganization) or the index space (for an index reorganization).</td>
</tr>
<tr>
<td></td>
<td>After completing the rename or FASTSWITCH operations that are associated with the staging data sets and registering the REORG or COPY row in the SYSIBM.SYSCOPY catalog table, REORG PLUS allows all access to the objects to resume. The table space and index spaces remain in the status that they had when the reorganization began. REORG PLUS then resets the RECOVER pending status.(^b)</td>
</tr>
</tbody>
</table>

\(^a\) For a partial reorganization, REORG PLUS prevents updates only to partitions that you specify (using the PART option of the REORG command) while it initializes XBM. In addition, REORG PLUS prevents updates to any nonpartitioned indexes.

\(^b\) If the value of the ORIGDISP installation option or the ORIGINALDISP command option is DELETE, REORG PLUS deletes the now-obsolete original data sets. If this option is set to RENAME and the FASTSWITCH process is not in effect, REORG PLUS renames the original data sets to the staging data set names. This renaming process preserves the space that the original data sets initially used. The process results in staging data sets that are ready for use in the next SHRLEVEL CHANGE reorganization.
Enabling dynamic allocation

REORG PLUS performs duplicate data set checking at data set allocation time. In a two-phase reorganization, dynamic allocation occurs at the beginning of the UNLOAD phase. In a single-phase reorganization, dynamic allocation occurs at the beginning of the REORG phase.

You can specify any of the following files to be dynamically allocated:

- unload data files (SYSREC)
- index work files (SYSUT1)
- sort work files (SORTWK)
- archive files, or discard files for DSNUTILB jobs (SYSARC)
- LOAD control statement data sets for DSNUTILB jobs (SYSPUNCH)
- full copy data sets (BCPY, BCPZ, BRCY, and BRCZ)
- incremental copy data sets (BICY, BICZ, BIRY, and BIRZ)

For information about allocating a particular data set type, see “REORG PLUS DD statements” on page 348.

Enabling dynamic allocation

To enable and use dynamic data set allocation quickly and simply, specify ACTIVE YES on your REORG PLUS command or in your installation options for each DDTYPE to dynamically allocate. For more options that you can use with dynamic allocation, see “Dynamic allocation options” on page 293.

Running with dynamic allocation in a worklist environment

When REORG PLUS runs in a worklist environment, REORG PLUS ignores the ACTIVE option in your installation options module. REORG PLUS dynamically allocates your data sets only if the invoking product (DASD MANAGER PLUS, CATALOG MANAGER, or CHANGE MANAGER) supplies the ACTIVE YES syntax.
Generating data set names

You can use the data set name pattern (DSNPAT) option to specify a pattern to generate a unique data set name. For some files, you can use a generation data group (GDG) name as the data set name.

Names created with DSNPAT

The DSNPAT installation or command option allows you to specify text and variable data for building data set names. If you cannot construct a data set name that meets your organization’s standards by using the text and the supplied variables, REORG PLUS provides an exit point that allows you to create your own variables for use with DSNPAT. Sample exits written in assembler, COBOL, C, and LE C are described in Appendix C, “REORG PLUS user exits,” and are provided in the HLQ.LLQSAMP library. (HLQ is the high-level qualifier specified during installation and LLQ is the low-level qualifier or prefix set during installation.) For more information about the DSNPAT option, see page 315.

The pattern that you specify in your DSNPAT option must allow REORG PLUS to generate unique data set names. For multiple SYSUT1 files, you must include the &DDNAME variable to generate unique names. For copy data sets, you might need to include additional variables, such as &VCAT, &DATEJ, or &TIME4, to generate unique names across multiple reorganizations. If REORG PLUS encounters non-unique data set names, it terminates the job.

GDG names

You can use generation data group (GDG) names for your dynamically allocated full and incremental copy data sets and for your SYSARC and SYSPUNCH files. Each DDTYPE must have a different GDG base.

GDG name format

The GDG format that you use to construct data set names is the same as the format that you use in JCL to allocate data sets through DD statements: you append the generation number in parentheses. The open parenthesis tells REORG PLUS that the pattern is a GDG name. The generation number must be an integer from 1 through 255.

An example of a GDG name is &TS.(+1). If you are using a substitution variable as the last variable before the open parenthesis, you must include a period before the open parenthesis.
**GDG base**

REORG PLUS has the following requirements for the number of GDG bases that you specify:

- Each DDTYPE must have a different GDG base.
- For copy data sets, each partition must have a different GDG base if you specify COPYLVL PART on the REORG command.

If the base does not exist, REORG PLUS creates it for you, using everything in the pattern up to the open parenthesis as the base name.

When defining the base, REORG PLUS uses the values of the following options:

- The GDGLIMIT installation or command option allows you to specify the number of generations to keep.
- If the GDGLIMIT value is exceeded, the GDGEMPTY option tells the system to uncatalog either all preexisting generations of this data set or only the oldest generation.
- The GDGSCRATCH installation option tells the system whether to delete the entry that was just uncataloged from the volume’s table of contents (VTOC). If the entry is deleted, the space on the volume becomes available to other users.

For more information, see the installation option descriptions in Appendix A, starting on page 752, and see “GDGLIMIT” on page 323.

**Specifying ddname prefixes**

If you specify more than one ddname prefix for dynamic allocation, the prefix for each ddname must be different enough for REORG PLUS to differentiate one prefix from another. To be different enough, if these prefixes are different only because one prefix has additional trailing bytes, then these trailing bytes must contain at least one nonnumeric byte. For example, the first set of prefixes that follow is sufficiently different, but the second set is not:

- acceptable set:
  - BMCRD
  - BMCRD11

- not acceptable set:
  - BMCRD
  - BMCRD11
Deleting dynamically allocated data sets

The prefixes that you specify must allow REORG PLUS to add the data set number (or partition number in the case of copy data sets) and still result in a valid ddname of eight characters. If the generated name would result in a ddname of less than eight characters, REORG PLUS pads the data set or partition number with leading zeros.

Deleting dynamically allocated data sets

To delete dynamically allocated data sets, specify DELETEFILES YES on your REORG command. After the job completes successfully, REORG PLUS automatically deletes the work files that it dynamically allocated and those allocated in your JCL. If you do not specify DELETEFILES YES, you must manually delete the dynamically allocated work files when your reorganization completes successfully. DELETEFILES YES does not apply to image copy data sets that REORG PLUS dynamically allocates.

You can also specify this preference with the DELFILES installation option. For more information about these options, see page 215 or page 684.

The SYSPRINT from your REORG PLUS job contains a report of the dynamically allocated work files. When you need to manually delete work files, you can use this report to determine which files to delete.

Dynamically allocating larger data sets with different properties than smaller data sets

You can use the THRESHLD option and associated dynamic allocation options to tell REORG PLUS to use different properties for larger data sets than for smaller ones. Data set allocations that exceed the threshold value will use the values for the second parameter of applicable dynamic allocation options.

For example, you could use the following options to tell REORG PLUS to send data sets greater than 720 MB to tape device TAPE1, and smaller data sets to DASD device SYSDA:

```
UNIT(SYSDA,TAPE1)
THRESHLD 720000
```

For more information about the THRESHLD option and the options that it affects, see “THRESHLD” on page 312.
Using SMS ACS routines to influence dynamic allocation

If your SMS automatic class selection (ACS) routines use the UNIT parameter to influence data set allocation, note the following information:

- When you specify SMSUNIT YES, REORG PLUS passes the UNIT option to SMS allocation in addition to passing the SMS class options and other normally passed options.

- When you specify SMSUNIT NO, REORG PLUS does not pass the UNIT option.

Reaching the MAXTAPE limit during dynamic allocation

When UNIT and THRESHLD specifications require that REORG PLUS dynamically allocate tape units, allocation occurs in the following priority order:

1. REORG PLUS attempts to allocate the greatest number of tape units required that will optimize multitasking.

2. If this number of tape units exceeds the MAXTAPE value, REORG PLUS decreases the multitasking level until the number of tape units required is less than or equal to the MAXTAPE value.

   This action might result in REORG PLUS dynamically allocating a single SYSUT1 data set, rather than one data set for each non-data-sorting index (thus decreasing multitasking).

3. If the minimum number of tape units required exceeds the MAXTAPE value, REORG PLUS issues a message and terminates.

   The value that you specify for the MAXTAPE option includes the units that are required for full and incremental copy data sets.

Changing dynamic allocation options on restart

Before restarting a job, you might need to change the options that affect dynamic data set allocation. For example, if specifying an invalid UNIT or overly restrictive MAXTAPE value causes the job to terminate, you need to change the relevant option before restarting the job.
The following restrictions apply to changes that you make to dynamic allocation options before restarting a job:

- You cannot change the value for the ACTIVE option on any restart.
- Changing any option on restart such that it results in different ddnames or a different number of DDs than the original option can produce an error. If you need to change the number of SYSREC and SYSUT1 work files, resubmit the job with a parameter of NEW.
- To change the value of other dynamic data set allocation options, specify RESTART(PHASE).

## Work file validity and integrity checks

REORG PLUS ensures that the work files it uses for reload processing are valid by performing the following types of file verification:

- check for data set attributes
  
The first check determines whether a work file has been allocated as a temporary data set or with DISP=MOD.

- check for data integrity
  
The second check ensures that the work file that REORG PLUS reads during reload processing is the same one that was created during unload processing.

### Check for data set attributes

This section describes how REORG PLUS responds when determining that your work files are temporary data sets. REORG PLUS defines a temporary data set as one whose normal or abnormal DISP is defined with one of the following values:

- DELETE
- NEW,PASS
- OLD,PASS and the original status is not OLD

### SYSREC, SYSUT1, and SYSARC data sets

If REORG PLUS determines that your SYSREC, SYSUT1, or SYSARC data sets are temporary, REORG PLUS responds based on the value for the FILECHK option.
FILECHK=FAIL

If FILECHK=FAIL and REORG PLUS finds that any of the SYSREC, SYSUT1, or SYSARC data sets are temporary, REORG PLUS terminates. On output, if the disposition is MOD, REORG PLUS resets it to empty.

If REORG PLUS is restarting in the RELOAD or REORG phase and the data set is temporary due to its normal disposition, then REORG PLUS overrides the normal disposition to KEEP to ensure against data loss after termination.

FILECHK=WARN

You can force REORG PLUS to process the SYSREC, SYSUT1, or SYSARC files in WARN mode by specifying FILECHK=WARN. In WARN mode, REORG PLUS issues a warning message for each work file that is allocated as a temporary data set, but continues processing.

**NOTE**

If you are running REORG PLUS in WARN mode, specifying DD DUMMY or DSN=NULLFILE for SYSREC or SYSUT1 data sets causes a failure in the RELOAD phase.

**Copy data sets**

**WARNING**

Use care when you specify DISP=MOD for an image copy data set because REORG PLUS does not reset the data set to empty but appends data to any data that is already present in the file.

If REORG PLUS finds that any of the full copy data sets are temporary data sets, REORG PLUS continues processing and issues a warning message, regardless of the value of FILECHK.

**NOTE**

For a two-phase reorganization, the warning message for the copy files occurs at open time, during the RELOAD phase. For a single-phase reorganization, the message occurs during the REORG phase. If you run REORG PLUS in two steps using the UNLOAD PAUSE option, you can code the DDs with DUMMY in the UNLOAD phase to avoid receiving message BMC50391E. This message indicates that REORG PLUS is unable to locate the copy data set.
For SHRLEVEL CHANGE, REORG PLUS terminates and issues an error message if it finds that any of the full or incremental copy files are temporary data sets, regardless of the value of FILECHK.

Check for data integrity

REORG PLUS performs a second check of the SYSREC and SYSUT1 work files to ensure data integrity. During unload processing, REORG PLUS creates the SYSREC and SYSUT1 work files with header information about the current utility execution. During reload processing, REORG PLUS checks this header information to ensure that the work file is the file that it is expecting to reload. If the header information does not match, REORG PLUS terminates and issues an error message.

SHRLEVEL considerations

The SHRLEVEL option specifies the level of access that DB2 has to the target spaces during REORG PLUS processing, as follows:

- SHRLEVEL NONE stops the objects that you are reorganizing and makes them unavailable.

- SHRLEVEL REFERENCE UNLOADONLY allows the objects to be available in read-only (RO) status during the UNLOAD phase.

- SHRLEVEL REFERENCE allows the objects to be available in RO status during unload and reload processing.

- SHRLEVEL CHANGE
  - allows the objects to be in read/write (RW) status during unload and reload processing and the LOGAPPLY phase
  - prevents updates to the objects during the LOGFINAL phase
  - prevents all access during the UTILTERM phase

This section provides operational considerations for the different SHRLEVEL values. For more information about the SHRLEVEL option, see “SHRLEVEL” on page 172.
**NOTE**

When you specify either SHRLEVEL REFERENCE UNLOADONLY or SHRLEVEL REFERENCE, you should be aware of the distinctions between them.

`SHRLEVEL REFERENCE UNLOADONLY` and `SHRLEVEL REFERENCE` are two distinct command option specifications. References in this document to the `SHRLEVEL REFERENCE UNLOADONLY` option are always explicit. Any discussion that mentions `SHRLEVEL REFERENCE` without the word UNLOADONLY applies only to the `SHRLEVEL REFERENCE` option.

---

**SHRLEVEL NONE**

SHRLEVEL NONE stops the objects that you are reorganizing and makes them unavailable. NONE is the default for the SHRLEVEL option.

REORG PLUS does not support SHRLEVEL NONE when reorganizing catalog objects.

---

**SHRLEVEL REFERENCE UNLOADONLY**

This option allows the table space and associated indexes (for a table space reorganization) or the index space (for an index reorganization) to remain in read-only status during the UNLOAD phase.

The objects that you are reorganizing have status requirements that are different from those for SHRLEVEL NONE. For more information, see “Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE” on page 86.

REORG PLUS does not support SHRLEVEL REFERENCE UNLOADONLY for the following reorganizations:

- single-phase reorganization
- DSNUTILB reorganization
- reorganization of catalog objects
SHRLEVEL REFERENCE

This option allows the table space and associated indexes (for a table space reorganization) or the index space (for an index reorganization) to remain in read-only status during the UNLOAD and RELOAD phases (for a two-phase reorganization) or the REORG phase (for a single-phase reorganization).

The objects that you are reorganizing have status requirements that are different from those for SHRLEVEL NONE. For more information, see “Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE” on page 86.

SHRLEVEL REFERENCE uses staging data sets as described in “Staging data sets” on page 101.

REORG PLUS does not support SHRLEVEL REFERENCE when reorganizing catalog objects.

SHRLEVEL CHANGE

This option allows the table space and associated indexes (for a table space reorganization) or the index space (for an index reorganization) to remain in RW status during all of the reorganization, except for a brief time during the UTILTERM phase when REORG PLUS replaces the original data sets with the staging data sets. For more information, see “Staging data sets” on page 101.

The objects that you are reorganizing have status requirements that are different from those for SHRLEVEL NONE. For more information, see “Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE” on page 86.

For more information about running a SHRLEVEL CHANGE reorganization, see Chapter 7, “Online reorganization.”
Staging data sets

When you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE, REORG PLUS uses staging data sets for shadow copies. REORG PLUS writes the reorganized objects to the staging data sets instead of to the original data sets. At the end of the job, REORG PLUS replaces the original data sets with the staging data sets in one of the following ways, depending on whether you use the rename process or the FASTSWITCH process:

- REORG PLUS uses the rename process by default. REORG PLUS “replaces” the original data sets with the staging data sets by renaming the VSAM data sets (discussed in “Staging data sets and the rename process” on page 103).

- In the FASTSWITCH process, REORG PLUS bypasses the VSAM rename process and changes the DB2 catalog entry to point to the staging data sets (discussed in “Staging data sets and the FASTSWITCH process” on page 105).

To use the FASTSWITCH process, you must take one of the following actions:

- Set the FASTSWITCH installation or command option to YES.
- Set the FASTSWITCH installation option to ZPARM, and ensure that the DB2 ZPARM value is YES.

You can also use special options to control whether REORG PLUS uses the rename or FASTSWITCH process under certain conditions:

- The FSFALLBACK option tells REORG PLUS to use rename processing if FASTSWITCH processing fails. For more information, see “FSFALLBACK” on page 220.

- The FSTHRESHOLD option lets you set a threshold number of staging data sets beyond which REORG PLUS performs FASTSWITCH processing. For data sets under this threshold, REORG PLUS performs rename processing. For more information, see “FSTHRESHOLD” on page 221.

Whether REORG PLUS renames the data sets or switches them, REORG PLUS leaves the original VSAM data sets intact throughout the reorganization process. This nondestructive process allows you to easily restart from a failure, or make the objects available without having to recover.

Allocating or defining staging data sets

Use Table 19 on page 102 to determine what action to take, regarding staging data sets, to enable REORG PLUS to perform your job correctly.
### Table 19  Staging data set actions

<table>
<thead>
<tr>
<th>REDEFINE command or installation option</th>
<th>Object you are reorganizing</th>
<th>Action</th>
</tr>
</thead>
</table>
| REDEFINE NO                             | VCAT-defined                | Preallocate the staging data sets and any data sets that you might need for expansion of a multi-data-set object before the RELOAD phase (for a two-phase reorganization) or the REORG phase (for a single-phase reorganization).
|                                         |                             | If you do not preallocate staging data sets and you specify REDEFINE NO, REORG PLUS fails because it cannot find the data sets. |
|                                         | storage-group-defined       | Preallocate the staging data sets and any data sets that you might need for expansion of a multi-data-set object before the RELOAD phase (for a two-phase reorganization) or the REORG phase (for a single-phase reorganization).
|                                         |                             | If you do not preallocate the staging or expansion data sets, REORG PLUS creates the data sets for you, using the same rules as if you had specified REDEFINE YES. |
| REDEFINE YES                            | VCAT-defined                | Provide the IDCAMS statements in the SYSIDCIN data set to delete and define the staging data sets.
|                                         |                             | If you do not provide any statements in SYSIDCIN, REORG PLUS treats the job as if you specified REDEFINE NO. |
|                                         | storage-group-defined       | Ensure that you have enough space available for REORG PLUS to allocate the staging data sets automatically.
|                                         |                             | REORG PLUS uses the list of volumes in the storage group for the original object unless you modify the volume list in the DSRSEXIT user exit. |
|                                         |                             | If you specified a value for the SMS DATACLAS in DSNZPARMs, REORG PLUS uses that value when it defines the VSAM data set. |
|                                         |                             | Using the DSRSEXIT user exit, you can change the size of VSAM data sets by changing the primary and secondary allocations for the new data sets within the exit. |
|                                         |                             | If no SMS classes exist in the DSNZPARMs or storage-group definition, you can add the classes by using the DSRSEXIT user exit. However, you cannot change existing specified SMS classes by using the DSRSEXIT user exit. |
|                                         |                             | You can change the value to REDEFINE NO at the object level by using the DSRSEXIT user exit. |
Staging data sets for nonpartitioned indexes during a partial reorganization

For partial reorganizations, if you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE, and you have defined nonpartitioned indexes, REORG PLUS copies the original nonpartitioned indexes to the staging data sets during the UNLOAD or REORG phase. If you specify UNLOAD PAUSE, REORG PLUS copies the indexes during the RELOAD phase after the RESTART.

Staging data sets and the rename process

If you use the rename process, you can use the STAGEDSN option to specify which naming convention you want REORG PLUS to use to allocate the staging data sets.

**NOTE**

If performing rename processing after falling back from a failed attempt to perform FASTSWITCH processing, REORG PLUS always uses STAGEDSN=DSN, regardless of the STAGEDSN value that you specified.

If the value of the STAGEDSN installation option is BMC, REORG PLUS names the staging data sets by replacing the DSNDBC node of the cluster and the DSNDDBD node of the data component with BMCDBC and BMCDBD. Table 20 illustrates this naming convention.

<table>
<thead>
<tr>
<th>Existing data set name</th>
<th>Staging data set name</th>
</tr>
</thead>
</table>

If your SMS, RACF, or other data-set-allocation rules require you to use a second-node qualifier of DSNDDBC (or DSNDDBD), you can set STAGEDSN=DSN. This setting instructs REORG PLUS to use the fifth-node qualifier to create your unique staging data set names. Table 21 illustrates this naming convention.

<table>
<thead>
<tr>
<th>Existing data set name</th>
<th>Staging data set name</th>
</tr>
</thead>
</table>
Staging data sets

Data set rename process

After writing the reorganized objects to the staging data sets, REORG PLUS renames the data sets by using the following process. Figure 6 on page 105 illustrates this process.

1. REORG PLUS prevents all access to the objects and renames the original DB2 VSAM data sets, using the naming convention that you selected with the STAGEDSN option.

   This part of the process makes the original data sets obsolete.

   A. If STAGEDSN=BMC, REORG PLUS replaces DSN in the DSNDDBC node of the cluster and in the DSNDBD node of the data component with OLD.

   **NOTE**  
   REORG PLUS terminates if the following data sets already exist:

   VCAT.OLDDBC.database.tablespace  
   VCAT.OLDDBD.database.tablespace

   B. If STAGEDSN=DSN, REORG PLUS renames the original data sets by replacing the I (or J) in the fifth node of the cluster and the fifth node of the data component with S to create a temporary name.

2. REORG PLUS changes the staging data set names back to the original data set names.

3. After successfully renaming the data sets, REORG PLUS starts the objects with their original statuses.

4. The next step depends on the value for the ORIGDISP or ORIGINALDISP option:

   ■ If the value for the ORIGDISP or ORIGINALDISP option is DELETE, REORG PLUS deletes the original data sets, which are now obsolete.

   ■ If the value for the ORIGDISP or ORIGINALDISP option is RENAME, REORG PLUS renames the original data sets to the staging data set names, depending on the value of STAGEDSN, as follows:

      — If STAGEDSN=BMC, REORG PLUS replaces OLD with BMC in the names.
      — If STAGEDSN=DSN, REORG PLUS replaces S with I or J, whichever letter corresponds to the VSAM node identifier for the staging data sets.

   RENAME preserves the space that was initially allocated for the original data sets. As a result, the data sets are ready for use as staging data sets in the next SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganization.
Tasks to perform if your reorganization job terminates

If you terminate the reorganization job, or the job abnormally terminates and you do not intend to restart it, you can delete the staging data sets. However, if the job terminates during the rename process, BMC recommends that you restart the job rather than attempt to rename the staging data sets manually.

Staging data sets and the FASTSWITCH process

If you are using FASTSWITCH processing and you want to preallocate or define the staging data sets, you must use the original DB2 VSAM data set names as a starting point, but replace the I qualifier with J, or the J qualifier with I. Table 22 on page 106 illustrates this naming convention (which is equivalent to specifying STAGEDSN=DSN).
After writing the reorganized objects to the staging data sets, REORG PLUS replaces the original data sets by using the following process. Figure 7 on page 107 illustrates this process.

1. REORG PLUS prevents all access to the objects and updates the DB2 catalog (SYSIBM.SYSTABLEPART and SYSIBM.SYSINDEXPART) to point to the staging data sets.

   This part of the process makes the original data sets obsolete.

2. After the switch is successful and the REORG or COPY row is registered in the SYSIBM.SYSCOPY catalog table, REORG PLUS starts the objects with their original statuses.

3. The next step depends on the value for the ORIGDISP or ORIGINALDISP option:

   - If the value for the ORIGDISP or ORIGINALDISP option is DELETE, REORG PLUS deletes the now-obsolete original data sets.
   - If the value for the ORIGDISP or ORIGINALDISP option is RENAME, REORG PLUS retains the staging data sets but does not rename them.

   The data sets still exist with their original names, preserving the space. You can use them as staging data sets in a future SHRLEVEL REFERENCE or CHANGE reorganization.
Staging data sets

Chapter 2 Operational considerations

Figure 7 Data set name FASTSWITCH process

Tasks to perform if your reorganization job terminates

If you terminate the reorganization job, or the job abnormally terminates and you do not intend to restart it, you can delete the staging data sets. However, if the job terminates during the FASTSWITCH process, BMC strongly recommends that you restart the job. If you cannot restart the job, see “Not completing the reorganization in the UTILTERM phase” on page 400.
Large number of partitions

REORG PLUS can reorganize objects that contain up to 4096 partitions. However, reorganizing objects with a large number of partitions increases the potential for encountering performance problems and memory restrictions. Consider the information in this section to help you avoid these issues.

Recommended option values for reorganizing a large number of partitions

Changing the following command or installation option values can improve the performance of your job and help avoid memory or other restrictions:

- If you need to make copies, consider one of the following options to avoid encountering data set allocation restrictions of the operating system or REORG PLUS memory restrictions:
  - Limit the number of copies per partition.
  - If your site’s recovery strategy allows for full copies, use one of the following options:
    - If you are reorganizing all partitions, create a single copy by specifying COPYLVL FULL.
    - If you are reorganizing a subset of partitions, create a single copy by specifying a single physically contiguous range of partitions, COPYSUBSET=YES in the installation options module, and COPYLVL FULL on your REORG command.

For more information, see the command option “COPYLVL” on page 269 and the COPYSUBSET installation option on page 682.

- If you dynamically allocate partition-level copies, REORG PLUS appends the partition number to the default ddname prefix. If you dynamically allocate partition-level copy data sets for more than 99 partitions, ensure that you change the value of the COPYDDN installation or command option to specify a ddname prefix; the prefix plus the highest partition number must not exceed eight characters. If you are also dynamically allocating remote copies, change the value of the RECOVERYDDN command option (or RCVYDDN installation option) to specify a ddname prefix for your remote copy data sets; the prefix plus the highest partition number must not exceed eight characters. For more information, see the command options “COPYDDN” on page 271 and “RECOVERYDDN” on page 273 or the COPYDDN and RCVYDDN installation options on page 680 and page 716.
Avoiding constrained resources when reorganizing a large number of partitions

When you are reorganizing a large number of partitions, use the following information to avoid or work around constrained resources issues:

- Ensure that you have specified a region size that allows the system to allocate as much virtual storage as possible to the REORG PLUS job. BMC recommends that you specify REGION=0M in the JOB or EXEC statement of your execution JCL.

- If you have specified REGION=0M and your job fails with a constrained resources error, consider reorganizing fewer partitions in a single job.

- When you are reorganizing a large number of partitions and dynamic allocation is active, BMC recommends that you specify DYNAMNBR=1600 on the EXEC statement of your execution JCL. Including this parameter enables the system to acquire adequate resources immediately before their use and to release them immediately after use.

- Reorganizing a large number of compressed partitions might result in constrained resources. If you encounter this problem, consider specifying fewer partitions in a single job.
Additional recommendations when reorganizing a large number of partitions

If you are reorganizing a large number of partitions, you might need to increase the size of the following common utility table spaces from the standard size that was allocated during installation:

- **BMCSYNC**

  Estimate the allocation for this table space based on all of the following factors:
  - the number of utilities that you are running concurrently
  - the number of partitions that you are processing concurrently
  - the number of files that you are dynamically allocating

- **BMCDICT**

  If you are reorganizing compressed data, estimate the allocation for this table space by multiplying by 64 KB the number of compressed partitions that you are operating on concurrently.

**DEFINE NO objects**

You can reorganize an object that was defined with the DEFINE NO attribute. However, the reorganization fails in the following situations:

- You are reorganizing an object that was defined with the DEFINE NO attribute, but DB2 has not yet materialized the table space or index.

- *(SHRLEVEL CHANGE)* While you are reorganizing a table space that contains at least one index that was defined with the DEFINE NO attribute, a row is inserted into one of the tables. The inserted row causes DB2 to define the VSAM data set for the index.
Multi-data-set DB2 objects

The following requirements and considerations apply to multi-data-set objects:

- For multi-data-set table spaces and index spaces that are VCAT-defined, you must perform one of the following tasks to prevent REORG PLUS from terminating:
  - Predefine all required data sets.
  - Specify REDEFINE YES and include a SYSIDCIN data set that defines the required data sets.

- For storage-group-defined multi-data-set objects, REORG PLUS defines additional data sets if needed for expansion, even if REDEFINE NO is in effect.

- If you are using Instant Snapshot technology to copy nonpartitioned indexes and the copy of one component of the multi-data-set object fails, the copy for the entire index fails.
  - If the value of the SIXSNAP command or installation option is YES, REORG PLUS terminates.
  - If the value of the SIXSNAP command is AUTO, REORG PLUS changes SIXSNAP to NO and recopies all components in the multi-data-set index, using the software-based copy method.

For possible timeout considerations when using the SIXSNAP function, see “Considerations for SIXSNAP use” on page 124.

- You can use the DSRSEXIT user exit to redefine a multi-data-set object. For more information, see “Sample DSRSEXIT REXX user exit” on page 844.

Catalog objects

REORG PLUS lets you reorganize DB2 catalog (database DSNDB06) table spaces and indexes when running under DB2 Version 10 new-function mode or later.

REORG PLUS invokes DSNUTILB when reorganizing the following catalog objects:

- the SYSCOPY table space
- LOB catalog table spaces
- catalog table spaces that contain LOB columns
- the SYSTSIPT table space if any of its indexes are versioned

In all other cases, REORG PLUS reorganizes the catalog object natively.
Requirements when reorganizing catalog objects

REORG PLUS has the following requirements for reorganizing catalog objects. If your job does not meet these requirements, REORG PLUS terminates.

- You must be running DB2 Version 10 new-function mode or later.
- You must specify SHRLEVEL CHANGE.
- A full image copy of the table space that you are reorganizing must be registered in SYSIBM.SYSCOPY before the reorganization runs.

For those reorganizations for which REORG PLUS invokes DSNUTILB, ensure that you meet the requirements and set the options appropriately for a DSNUTILB job. For more information, see “Reorganization jobs that invoke DSNUTILB” on page 72.

REORG PLUS options that are incompatible with reorganizing catalog objects

Some REORG PLUS command and installation options are incompatible with reorganizing a catalog object natively. Table 23 describes how REORG PLUS responds if these options are in effect.

Table 23  Options that are incompatible when reorganizing catalog objects

<table>
<thead>
<tr>
<th>Option</th>
<th>REORG PLUS response</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPYRFAIL=COPYPEND</td>
<td>changes this option to CPYRFAIL=TERM</td>
</tr>
<tr>
<td>DELETE</td>
<td>issues message BMC50056E and terminates</td>
</tr>
<tr>
<td>FASTSWITCH YES</td>
<td>changes this option to FASTSWITCH NO</td>
</tr>
<tr>
<td>MAXNEWPARTS</td>
<td>ignores this option</td>
</tr>
<tr>
<td>PREFORMAT YES</td>
<td>changes this option to PREFORMAT NO</td>
</tr>
<tr>
<td>SELECT</td>
<td>issues message BMC50056E and terminates</td>
</tr>
<tr>
<td>SHRLEVEL NONE</td>
<td>issues message BMC50119E and terminates</td>
</tr>
<tr>
<td>SHRLEVEL REFERENCE</td>
<td></td>
</tr>
</tbody>
</table>

Additional considerations for reorganizing catalog objects

The following considerations apply to reorganizing catalog objects:

- You cannot reorganize any object in the directory database (DSNDB01).
REORG PLUS allows you to update catalog or DASD MANAGER PLUS statistics only when reorganizing a catalog object natively.

If the reorganization fails during the rename process, REORG PLUS automatically backs out the reorganization and leaves the spaces in their original statuses.

**XML data**

REORG PLUS lets you reorganize XML data.

**Considerations when reorganizing the base table space for XML data**

The following considerations apply when you are reorganizing a table space that contains an XML column.

**Partition rebalancing**

If one of the following conditions exists, REORG PLUS terminates:

- You specify REBALANCE
- A DDLIN data set in your JCL includes an ALTER statement that would alter the last partition, and the table space is one of the following types:
  - a table space that uses table-controlled partitioning
  - a table space defined with the LARGE or DSSIZE attribute

**Additional incompatible REORG PLUS command options**

If you specify the SELECT or DELETE option, REORG PLUS issues message BMC51222E and terminates.

If you specify one of the following options, REORG PLUS issues BMC50125E and terminates:

- an UPDATE option that attempts to update an XML or document ID column
- a WHERE clause that references an XML column
### Document ID column

When you add the first XML column to a table via an ALTER statement, the document ID column (DOCID) is not added at the same time. If REORG PLUS finds that this column is missing, it adds the column and populates it with an assigned value. REORG PLUS assigns this value based on the value in the MAXASSIGNEDVAL column of the SYSIBM.SYSEQUENCES table and the cache of document ID values that REORG PLUS reserves on this column.

### Reserving the cache

For each unload and log apply task that requires it, REORG PLUS reserves at least one cache of document ID values on the MAXASSIGNEDVAL column of the SYSIBM.SYSEQUENCES table. When reserving the cache, REORG PLUS updates the MAXASSIGNEDVAL field with the last value in the cache that it is reserving.

### Controlling the cache

You can use the IDCACHE installation or command option to control the number of values in the cache that REORG PLUS reserves. BMC recommends that you use the default value of 10000. Specifying a cache that is too large or too small might cause REORG PLUS to retrieve values that it will not use. Specifying a smaller cache size can also impact performance because REORG PLUS must access the DB2 catalog more frequently. For more information about these options, see page 255 or page 699.

### Authorization

For authorization requirements that apply when reorganizing tables that contain an XML column, see “Additional authorizations for XML reorganizations” on page 68.

### Considerations when reorganizing the XML table space

The following considerations apply when reorganizing the XML table space. When reorganizing the XML table space, REORG PLUS updates the associated node ID index and any user-defined XML indexes.

**NOTE**

You can also reorganize the indexes on an XML table space with the REORG INDEX command option. For information about authorization requirements when reorganizing a user-defined XML index, see “Additional authorizations for XML reorganizations” on page 68.
XML columns that support XML versions

REORG PLUS terminates if you attempt to reorganize a versioned XML table space with SHRLEVEL CHANGE and an update occurs on the XML table space.

ANALYZE processing

If you specify ANALYZE SCAN for an XML table space, REORG PLUS changes the value to ANALYZE SAMPLE, issues message BMC51328I, and continues processing. For ANALYZE SAMPLE on an XML table space, REORG PLUS obtains key counts from the DB2 real-time statistics tables.

REORG PLUS sizes the SYSUT1 data sets for an XML table space reorganization based on the number of keys in the XML and node ID indexes.

Additional incompatible command options and data sets

With the noted exception, REORG PLUS terminates when any of the following conditions exists:

- You include a DDLIN data set in your JCL.
- You specify any of the following options:
  - REBALANCE
  - SELECT
  - DELETE
  - UPDATE

**NOTE**

For partition-by-growth table spaces, REORG PLUS does not terminate if you include a DDLIN data set in your JCL or specify REBALANCE. In this case, REORG PLUS issues a warning that this function is not valid for partition-by-growth table spaces and continues processing.

Partition-by-growth table spaces

If your XML table space is a partition-by-growth table space, see “Partition-by-growth table spaces” on page 125 for additional considerations.
LOB data

This section provides instructions and considerations for reorganizing LOB data.

Reorganizing the base table space and its indexes

In general, REORG PLUS reorganizes the base table space (and its indexes) for LOB data natively. REORG PLUS invokes DSNUTILB to reorganize a base table space that contains a LOB column when any of the following conditions exists:

- The table space contains an index on an inline LOB column.
- The table space contains an inline LOB column whose length definition was altered before the reorganization.
- The table space is a partition-by-growth table space.
- You are reorganizing both the base table space and the LOB table space in the same job step. This occurs when any of the following conditions exists:
  - You specify AUX YES.
  - You do not specify the AUX option, but you are reorganizing a table space that contains a LOB column and any of the following conditions exists. (In this case, the DB2 REORG utility defaults to AUX YES.)
    - You specify SHRLEVEL REFERENCE and REBALANCE.
    - You are reorganizing more than one partition of a partition-by-growth table space
    - You specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE, and any partition is in REORP status.

To reorganize base table spaces and their indexes when REORG PLUS invokes DSNUTILB

1 Ensure that you meet the requirements for a DSNUTILB job, as described in “Reorganization jobs that invoke DSNUTILB” on page 72.

If you do not supply the correct options or environment for invoking DSNUTILB, REORG PLUS issues BMC50178E and terminates.
2 Ensure that the installation and command options are set appropriately for a DSNUTILB reorganization job, as described in “Reorganization jobs that invoke DSNUTILB” on page 72.

3 If an index is defined on an inline LOB column, ensure that the value of the IXONEX installation option is YES.

For more information about this option, see page 702.

4 If an inline LOB column was altered before the reorganization, complete the following steps:

A Ensure that the value of the LOB installation option is YES.

For more information about this option, see page 704.

B Specify AUX YES on the REORG command.

For more information about this option, see page 256.

5 If you want to reorganize the base table space and LOB table space in the same job, complete the following steps:

A Ensure that the value of the LOB installation option is YES.

For more information about this option, see page 704.

B Specify AUX YES on the REORG command.

For more information about this option, see page 256.

To reorganize base table spaces and their indexes in all other cases

Create your job as usual, making any necessary adjustments based on the information in “Considerations when reorganizing the base table space for LOB data” on page 118.
Considerations when reorganizing the base table space for LOB data

REORG PLUS terminates when any of the following conditions exists:

- You specify one of the following options:
  - SELECT
  - DELETE
  - an UPDATE option that attempts to update a LOB column
  - a WHERE clause that references a LOB column
  - AUX YES and UNLOAD PAUSE.

- You are rebalancing partitions during the reorganization when running natively.

Reorganizing LOB table spaces

How REORG PLUS handles LOB table spaces depends on the type of job that you are running:

- For SHRLEVEL REFERENCE, REORG PLUS natively reorganizes the LOB table space and updates the auxiliary index.

- For SHRLEVEL NONE and SHRLEVEL CHANGE, REORG PLUS invokes DSNUTILB to reorganize the LOB table space.

- When reorganizing both the LOB table space and base table space in the same job step, REORG PLUS invokes DSNUTILB.

**NOTE**

For any SHRLEVEL option, REORG PLUS natively reorganizes the auxiliary index on a LOB table space when you specify REORG INDEX.

To reorganize LOB table spaces when invoking DSNUTILB

1. Ensure that you meet the requirements for a DSNUTILB job, as described in “Reorganization jobs that invoke DSNUTILB” on page 72.

   If you do not supply the correct options or environment for invoking DSNUTILB, REORG PLUS issues BMC50178E and terminates.
2 Ensure that the installation and command options are set appropriately for a DSNUTILB reorganization job, as described in “Reorganization jobs that invoke DSNUTILB” on page 72.

3 Ensure that the value of the LOB installation option is YES.

   For more information about this option, see page 704.

4 If you want to reorganize the LOB table space and base table space in the same job, also complete the following steps:

   A Specify the base table space as the table space that you are reorganizing.

   B Specify AUX YES on the REORG command.

   For more information about the AUX option, see page 256.

To reorganize LOB table spaces when SHRLEVEL REFERENCE is in effect

Create your job as usual, making any adjustments as needed based on the information in “Considerations when reorganizing LOB table spaces (SHRLEVEL REFERENCE).”

Considerations when reorganizing LOB table spaces (SHRLEVEL REFERENCE)

When SHRLEVEL REFERENCE is in effect, REORG PLUS uses single-phase processing to reorganize your LOB table space and update the auxiliary index. The following considerations apply in this case.

Object status requirements

In addition to the status restrictions for any other SHRLEVEL REFERENCE reorganization job, the auxiliary index on the LOB table space must not be in page set rebuild pending (PSRBD) or rebuild pending (RBDP) status.

Data set requirements

REORG PLUS does not use any sort work or SYSUT1 data sets that you allocate when reorganizing LOB table spaces with SHRLEVEL REFERENCE.
Incompatible REORG PLUS command options

Some REORG PLUS command options are not available when you are reorganizing a LOB table space with SHRLEVEL REFERENCE. Table 24 describes the options that are not available and how REORG PLUS responds if your job contains these options.

Table 24 Command options that are incompatible when reorganizing LOB table spaces

<table>
<thead>
<tr>
<th>Command option</th>
<th>REORG PLUS response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE BMCSTATS</td>
<td>ignores this option</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>does not sample or scan, but adds high-used relative byte addresses (HURBAs) from all data sets to estimate copy data set size</td>
</tr>
<tr>
<td>ANALYZE SAMPLE</td>
<td></td>
</tr>
<tr>
<td>ANALYZE SCAN</td>
<td></td>
</tr>
<tr>
<td>CONDEXEC BMC</td>
<td>issues message BMC50056E and terminates</td>
</tr>
<tr>
<td>CONDEXEC YES</td>
<td>issues message and terminates</td>
</tr>
<tr>
<td>DELETE</td>
<td>issues message BMC50056E and terminates if specifying this option would cause REORG PLUS to check criteria for conditional reorganization</td>
</tr>
<tr>
<td>INDREFLIMIT</td>
<td>issues message BMC50056E and terminates if specifying this option would cause REORG PLUS to check criteria for conditional reorganization</td>
</tr>
<tr>
<td>OFFPOSLIMIT</td>
<td>issues message BMC50056E and terminates if specifying this option would cause REORG PLUS to check criteria for conditional reorganization</td>
</tr>
<tr>
<td>ORDER YES</td>
<td>changes this option to ORDER NO</td>
</tr>
<tr>
<td>REBALANCE</td>
<td>ignores this option</td>
</tr>
<tr>
<td>SELECT</td>
<td>issues message and terminates</td>
</tr>
<tr>
<td>SHRLEVEL CHANGE</td>
<td>issues message BMC50115E and terminates</td>
</tr>
<tr>
<td>UNLOAD CONTINUE</td>
<td>changes this option to UNLOAD RELOAD</td>
</tr>
<tr>
<td>UNLOAD PAUSE</td>
<td>issues message BMC50115E and terminates</td>
</tr>
<tr>
<td>UPDATE</td>
<td>issues message and terminates</td>
</tr>
</tbody>
</table>

Floating-point data

You will receive inexact results if an ALTER changes a table space column from an integer or decimal type to FLOAT before you reorganize the table space. The value in the floating-point column might not be identical to the value in the original integer or decimal column. (By definition, floating-point calculations are inexact. Therefore, conversions involving floating-point data produce inexact results.)
Partial reorganization

This section discusses some of the differences between a partial reorganization (a reorganization in which you specify the PART option) and a full reorganization.

General considerations for a partial reorganization

Consider the following items when you are performing a partial reorganization:

- Full table space reorganizations rebuild and reorganize all indexes. However, when you perform a partial table space reorganization, REORG PLUS does not rebuild the nonpartitioned indexes. Instead, it updates the RID entries only for keys that are associated with the rows of the partitions that you are reorganizing.

- Because REORG PLUS does not rebuild indexes during a partial reorganization, the following restrictions apply:
  - REORG PLUS cannot update a row change timestamp column when a row is modified if the column that is implicitly updated is part of a nonpartitioned index key.
  - REORG PLUS cannot convert the RBA or LRSN format of a nonpartitioned index.
  - REORG PLUS does not collect statistics for nonpartitioned indexes.

- You cannot run two or more partial table space reorganization jobs concurrently if nonpartitioned indexes exist.

SHRLEVEL NONE considerations for a partial reorganization

In addition to the general considerations, the following considerations apply to a partial reorganization when SHRLEVEL NONE is in effect:

- REORG PLUS stops and starts only those partitions specified with the PART option of the REORG command. REORG PLUS stops or starts nonpartitioned indexes in their entirety.
If a failure occurs after REORG PLUS starts updating one of the following indexes, you can restart the reorganization, but the index will be left in RBDP status after the restarted job completes:

- a compressed, non-unique, nonpartitioned index
- a document ID index for which REORG PLUS has generated document ID values

This situation might occur if the job is the first reorganization after adding the first XML column to the table.

SHRLEVEL REFERENCE or SHRLEVEL CHANGE considerations for a partial reorganization

In addition to the general considerations, the following considerations apply when performing a partial reorganization with SHRLEVEL REFERENCE or SHRLEVEL CHANGE specified:

- If nonpartitioned indexes are defined, REORG PLUS copies the original nonpartitioned indexes to the staging data sets during the UNLOAD or REORG phase.

- If you specify UNLOAD PAUSE with SHRLEVEL REFERENCE, REORG PLUS copies the indexes during the RELOAD phase, after the restart.

Instant Snapshot with nonpartitioned indexes

You can substantially improve the performance of copying storage-group-defined nonpartitioned indexes by using the XBM product’s Instant Snapshot technology. Instant Snapshots are hardware-based copies that do not require the I/O that is needed to make a software-based copy.

**To use Instant Snapshot technology to copy nonpartitioned indexes**

1. Ensure that you have a currently supported version of XBM or SUF as described in “Software requirements” on page 66.

2. Ensure that you have the supported intelligent storage devices (as documented in EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide).
3 Specify AUTO or YES on the SIXSNAP command or installation option.

For information about the SIXSNAP command option, see “SIXSNAP” on page 251.

4 (optional) Use the XBMID installation or command option to specify the XBM subsystem that you want REORG PLUS to use for snapshot processing.

For information about the XBMID command option, see “XBMID” on page 253.

5 Review the information in the rest of this section for any additional actions to take.

**Using the SIXSNAP option**

The SIXSNAP option determines whether REORG PLUS uses the Instant Snapshot technology of XBM and SUF to create a copy of storage-group-defined nonpartitioned indexes. To use Instant Snapshot, you must have the supported intelligent storage devices and you must specify YES or AUTO for the SIXSNAP command or installation option. For a list of supported devices, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

Table 25 shows how the value of the SIXSNAP command or installation option determines the action that REORG PLUS takes if a copy failure occurs.

### Table 25  SIXSNAP and Instant Snapshot

<table>
<thead>
<tr>
<th>SIXSNAP value</th>
<th>Instant Snapshot fails for</th>
<th>REORG PLUS action</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>one or more nonpartitioned indexes</td>
<td>makes a software-based copy of each index that failed</td>
</tr>
<tr>
<td></td>
<td>one or more components of a multi-data-set object</td>
<td>makes a software-based copy of every component of that multi-data-set index</td>
</tr>
<tr>
<td>YES</td>
<td>any index</td>
<td>terminates</td>
</tr>
</tbody>
</table>

BMC recommends that you specify AUTO instead of YES, especially if your hardware vendor is STK. For more information, see “Hardware vendor considerations” on page 125.

Table 26 on page 124 shows how the REDEFINE command or installation option can also impact the type of copies that REORG PLUS makes.
Instant Snapshot with nonpartitioned indexes

Table 26  REDEFINE and SIXSNAP

<table>
<thead>
<tr>
<th>REDEFINE value</th>
<th>SIXSNAP value</th>
<th>REORG PLUS action</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>AUTO</td>
<td>changes the value of SIXSNAP to NO and makes a software-based copy</td>
</tr>
<tr>
<td>YES</td>
<td></td>
<td>terminates the job</td>
</tr>
<tr>
<td>YES</td>
<td>AUTO or YES</td>
<td>performs the SIXSNAP function</td>
</tr>
</tbody>
</table>

For more information about invoking Instant Snapshot, see “SIXSNAP” on page 251. For information about the XBM and SUF requirements, see “XBM and SUF considerations” on page 151.

Considerations for SIXSNAP use

When using SIXSNAP, certain situations can cause timeouts to occur. Also, the allocation on DASD varies, depending on the hardware vendor. This section discusses these considerations.

Timeout considerations

Using Instant Snapshot can improve the performance of the reorganization, but can also cause applications to time out. For example, a timeout might occur when REORG PLUS copies a large number of nonpartitioned indexes or a multi-data-set index that contains a large number of pieces.

The size of the data sets to be copied does not matter. The number of data sets affects timeouts because the Instant Snapshot requests (one per index or piece) are serial. If these requests take longer than the standard SQL timeout value (IRLMRWT) that is defined in DSNZPARMs for your site, the application times out with SQL code –911.

To calculate the time required to perform the Instant Snapshot, allow approximately 8 to 10 seconds (depending on the hardware and version of the microcode) per index or piece. If this value is less than the IRLMRWT value, the application should not time out.

A timeout can also occur when REORG PLUS begins Instant Snapshot processing but discovers that the required hardware is not available.
Hardware vendor considerations

The results of using SIXSNAP can vary depending on the hardware vendor that you are using for the DASD devices.

- SMS controlled

  If the target data set (snapped copy) is SMS-controlled and uses fewer volumes than the source data set, the additional volumes remain candidate volumes with a volume name of * (asterisk).

- EMC

  Regardless of the order in which you pass the storage group volumes, EMC sorts the volumes from the one with the greatest amount of available space to the one with the least amount of available space, and allocates them in that order.

- STK

  STK allocates the volumes in the order that you pass them. However, if the first volume in the list does not contain enough room for the primary allocation, the allocation fails. If you specify SIXSNAP YES, the reorganization fails if this volume allocation fails. If you specify SIXSNAP AUTO, REORG PLUS changes it to SIXSNAP NO and performs the software-based copy. Therefore, BMC recommends that you specify SIXSNAP AUTO instead of YES.

Partition-by-growth table spaces

You can use REORG PLUS to perform both full and partial reorganizations of partition-by-growth table spaces. The strategy that REORG PLUS uses to place rows within the table space depends on whether the table contains a LOB column, as follows:

- If the table contains a LOB column, REORG PLUS does not change any row’s partition. For an ORDER YES reorganization, REORG PLUS sorts the rows within each partition by the clustering index.

- If the table does not contain a LOB column, REORG PLUS can move rows within partition ranges. Within each range, REORG PLUS places rows from lowest RID to highest. For an ORDER YES reorganization, REORG PLUS sorts each range by the clustering index.

When processing partition-by-growth table spaces, REORG PLUS starts at most one unload, reload, and reorg task per range of contiguous partitions. The task that handles the last partition handles the partitions that might be added during the reorganization process.
Partition-by-growth table spaces have special considerations. In addition to the information in the rest of this section, see the following sections:

- “Building a dictionary” on page 146
- “Copy data sets” on page 352
- “SYSREC data sets” on page 379
- “Partition-by-growth table spaces” on page 393

Extension of partition-by-growth table spaces by REORG PLUS

Either REORG PLUS or DB2 might add partitions to a partition-by-growth table space during the reorganization process. If REORG PLUS runs out of space in the existing set of partitions, it can add partitions to the table space if all of the following conditions exist:

- The table has no LOB columns.
- You include the last partition in the reorganization, either by running a full table space reorganization, or by specifying it using the PART option (page 175).
- The value of the DB2 MAXPARTITIONS parameter is greater than the number of partitions in the table space.
- The value of the MAXNEWPARTS option (page 177) is greater than 0.

Note the following considerations when REORG PLUS adds partitions to a partition-by-growth table space:

- You can specify multiple ranges of partitions (that is, contiguous sets of reorganized partitions). When you do so, if it cannot reload all of the rows unloaded from a range back into that range, REORG PLUS terminates. REORG PLUS considers added partitions to be in the range that includes the last partition.
- If you are making copies, REORG PLUS copies all partitions that it adds to the table space. For information about copy data set requirements, see “Partition-by-growth table spaces” on page 358.
Extension of partition-by-growth table spaces by DB2

DB2 might add partitions during a SHRLEVEL CHANGE reorganization of a partition-by-growth table space or an index defined on its table. REORG PLUS can support added partitions up to the limit specified by the MAXNEWPARTS option (page 177). If the number of partitions that DB2 adds exceeds MAXNEWPARTS, REORG PLUS terminates and issues message BMC53025E.

REORG PLUS copies all partitions that DB2 adds for a full reorganization. For a partial reorganization, REORG PLUS copies partitions that DB2 adds when both of the following conditions exist:

- The last partition is included in the reorganization.
- The table does not include a LOB column.

For information about copy data set requirements, see “Partition-by-growth table spaces” on page 358.

Partition rebalancing

REORG PLUS supports partition rebalancing in the following ways:

- REORG PLUS reorganizes partitions that have had the values of their partition keys previously altered and are now in REORG pending (REORP) status.

- REORG PLUS reorganizes logically contiguous partitions for any type of SHRLEVEL table space reorganization where you alter the partition’s limit key during the reorganization by using one of the following methods:
  
  — providing a DDLIN data set (defined by the DDLIN DD statement)

  For more information, see “DDLIN data set considerations” on page 131.

  — specifying the REBALANCE command option

  When you specify the REBALANCE command option, REORG PLUS automatically rebalances partitions during the reorganization. By specifying REBALANCE, you can have REORG PLUS determine the limit keys for you. For more information about this option, see “REBALANCE” on page 179.
To rebalance partitions

1. Either include a DDLIN data set in your JCL or specify REBALANCE on the REORG command.

2. Include a SYSARC data set, if needed.

   If the partitioned table space is LARGE (either by definition or by default) or defined with DSSIZE, and the last partition is in REORP status, REORG PLUS requires a SYSARC data set. Any rows that REORG PLUS discards because their primary key is greater than the newly designated limit key for the highest-numbered partition are written to the SYSARC data set. You can allocate the SYSARC data set in JCL or have REORG PLUS dynamically allocate it for you.

3. (optional) Include a DDLOUT DD statement in your JCL if you want to keep a record of the ALTER statements that REORG PLUS executes.

4. Specify ORDER YES or omit the ORDER keyword on the REORG command.

   This type of reorganization does not support ORDER NO. If you specify ORDER NO, REORG PLUS terminates and issues message BMC51209E.

5. Specify COPY YES on the REORG command.

6. Include all contiguous partitions in the same reorganization that are in REORP status or that you want to rebalance.

   For example, if the following set of conditions exists, you must reorganize partitions 2 through 6 in the same reorganization job:

   - You are reorganizing a table space with seven partitions.
   - Only partitions 2 and 3 are in REORP status.
   - You want to alter the limit keys on partitions 4 and 5, but not the remaining partitions.

7. Make any necessary changes to your job based on the information in the rest of this section.

General rebalancing considerations

The following general restrictions and considerations apply to partition rebalancing:

- When you include both a DDLIN data set and the REBALANCE command option in a reorganization job, DDLIN data set ALTER statements override the REBALANCE option for the partitions specified in the DDLIN ALTER statements.
■ REORG PLUS processes all related partitions in the same unload task. Consequently, the task distribution might not be as even as in a normal reorganization.

■ You can rebalance up to 255 ranges of logically contiguous partitions in a single execution of REORG PLUS.

■ With SHRLEVEL CHANGE, you can use the DDLOUT DD statement to obtain only a report (without completing the reorganization) of the ALTER statements that REORG PLUS would use to rebalance partitions. For more information, see “DDLOUT data sets” on page 365.

■ With SHRLEVEL CHANGE and SHRLEVEL REFERENCE, you can use the ALTRFAIL installation option (page 674) to tell REORG PLUS how to handle failures that occur when processing limit-key ALTER statements. This capability applies to ALTER statements that are processed either from your DDLIN data set or as a result of the REBALANCE option.

Limit key considerations for rebalancing

The following information applies to the limit keys used during rebalancing.

Limit key considerations

The following considerations apply to limit keys used during rebalancing:

■ You can use the ON n COLUMNS clause of the REBALANCE option to specify the number of columns to use in the limit key.

■ REORG PLUS supports the following data types in limit keys when you rebalance partitions:
  — CHAR
  — DATE
  — DECIMAL
  — INTEGER
  — SMALLINT
  — BIGINT
  — TIME
  — TIMESTAMP
  — TIMESTAMP WITH TIME ZONE
  — VARCHAR
Limit key restrictions

REORG PLUS does not support the following when you rebalance partitions:

- The following types of constants as limit key values:
  - floating point
  - LOB
  - row ID
  - graphic types
  - binary string types
  - decimal floating point

- Limit key columns that use a FIELDPROC in SQL ALTER INDEX or ALTER TABLE statements.

Table space considerations for rebalancing

The following rebalance considerations apply to the type of table space that you are reorganizing:

- You cannot use either partition rebalancing method to rebalance the partitions of the following types of table spaces:
  - partition-by-growth table spaces
  - table spaces for clone objects or base objects that participate (or have participated) in a clone relationship
  - XML table spaces
  - LOB table spaces
  - table spaces that contain LOB columns, when reorganizing natively
  - in certain cases, table spaces that contain XML columns (see “Partition rebalancing” on page 113)

For partition-by-growth table spaces and LOB table spaces, REORG PLUS ignores the rebalance request and completes normally. For all other restricted partitioned table spaces, REORG PLUS terminates.
For a table space that is defined with the LARGE or DSSIZE attribute and for table-controlled objects, REORG PLUS honors the limit key of the last partition. If the limit key changes, REORG PLUS discards into the SYSARC data set any keys that are greater than the limit key. You can use the BMC LOADPLUS for DB2 product or the IBM DB2 LOAD utility to reload the data or to load it to a different table space.

For a non-LARGE or non-DSSIZE table space that uses index-controlled partitioning, the limit key of the last partition is informational only. REORG PLUS does not discard any rows and does not use the SYSARC data set.

When rebalancing partitions of a table space with compression, REORG PLUS builds a new dictionary even if you specified KEEPDICTIONARY=YES in your installation options module or KEEPDICTIONARY on the REORG command.

REORP status considerations for rebalancing

REORG PLUS treats each partition that is participating in the reorganization as if it were in REORP status. You must reorganize the entire group of logically contiguous partitions that are either in REORP status or treated as if they were. Not doing so causes REORG PLUS to terminate the job.

Reorganization of objects that are in REORP status (or treated as if they were in REORP status) requires registered image copies:

- If your job specifies the COPY NO command option, REORG PLUS changes the option to COPY YES and issues message BMC51218I.

- If the value of the CPYRFAIL installation option is COPYPEND, REORG PLUS changes the value to TERM for this job, issues message BMC50138I, and continues terminating the job. The value change ensures that REORG PLUS can return the table space to its original state.

DDLIN data set considerations

The DDLIN data set should contain only an SQL ALTER INDEX statement or, for table-controlled partitioning, an ALTER TABLE statement for each limit key that you want to change. REORG PLUS ignores ALTER statements that do not apply to the current reorganization. Thus, you can use a single DDLIN data set for multiple jobs. For more information about this data set, see “DDLIN data set” on page 360.
REORG PLUS does not explicitly execute the SQL statements. Instead, REORG PLUS extracts the partition number and the associated limit key values from the statements. REORG PLUS reorganizes the data in the partitions based on the new key values and alters the limit keys during the UTILTERM phase, thus rebalancing the data in the partitions.

If the limit key values are altered, DB2 invalidates any plans and packages that are associated with the altered objects. If the value of the DB2 ZPARM AUTO BIND is YES or COEXIST, DB2 rebinds the plans and packages automatically the next time that they are executed. If the value of AUTO BIND is NO, you must rebind the plans and packages before your applications can access the altered objects.

When processing a DDLIN data set, REORG PLUS echoes all of the statements in the data set to SYSPRINT and issues a message for each statement that it will process. For sample SYSPRINT output, see “Sample ALTER INDEX statements and resulting messages” on page 363 and “Sample ALTER TABLE statements and resulting messages” on page 364.

Additional considerations
In addition to the considerations described in the previous sections, the following additional considerations apply to use of the DDLIN data set:

- REORG PLUS terminates when either of the following conditions exists:
  - Definition changes are pending on the table space or an index on the table space.
  - You are running on DB2 Version 11 and are reorganizing a table space that is defined with table-controlled partitioning.

- If the DDLIN data set contains an alter limit key value that is equal to the existing limit key value, REORG PLUS performs the reorganization, but no rebalancing is required and no ALTER is performed for that partition.

- When you use a DDLIN data set to rebalance partitions, and also use dynamic allocation of SYSREC data sets during partition rebalancing, REORG PLUS cannot predict how much data movement will occur between rebalanced partitions. Therefore, REORG PLUS might inadequately size the SYSREC for a partition that has a large amount of data rebalanced into it. Also, when a table space has a non-unique partitioning key, REORG PLUS might not evenly distribute rows across the partitions. In these situations, BMC recommends that you code the required SYSRECnn DD statements for these types of partitions in the JCL. Continue to allow REORG PLUS to dynamically allocate all other SYSRECnn data sets.

- You can use a previously created DDLOUT data set as DDLIN input. For more information about this data set, see “DDLOUT data sets” on page 365.
REBALANCE option considerations

In addition to the considerations described in the previous sections, the following additional considerations apply to use of the REBALANCE option.

**DSNUTILB conditions**

REORG PLUS invokes DSNUTILB when you specify REBALANCE and either of the following conditions exists:

- You are reorganizing a table space that is defined with table-controlled partitioning and you are running on DB2 version 11.
- You are reorganizing a table space that contains a LOB column and the following conditions exist:
  - You specify SHRLEVEL REFERENCE.
  - You do not specify the AUX option on the REORG command.
    
    In this case, the DB2 REORG utility defaults to AUX YES.

For more information about running a DSNUTILB reorganization, see “Reorganization jobs that invoke DSNUTILB” on page 72.

**Specifying partitions**

Use care when specifying partitions. REORG PLUS rebalances partitions by logical partition number, rather than physical partition number. If contiguous physical partitions are not contiguous by logical partition number, REORG PLUS cannot rebalance them. The following example illustrates this consideration.

A table space has four partitions. The logical and physical partition numbers match.

<table>
<thead>
<tr>
<th>Logical</th>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
You then alter the table space to rotate the first partition to last, resulting in the following mapping of logical and physical partitions:

<table>
<thead>
<tr>
<th>Partition number</th>
<th>Logical</th>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Next, you add a partition to the table space, resulting in the following mapping of logical and physical partitions:

<table>
<thead>
<tr>
<th>Partition number</th>
<th>Logical</th>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Finally, you rotate the first partition to last a second time, resulting in the following mapping of logical and physical partitions:

<table>
<thead>
<tr>
<th>Partition number</th>
<th>Logical</th>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

If you then try to reorganize the table space and rebalance partitions 1:3, the job fails because these partitions are not logically contiguous. Their logical partition numbers are 3, 5, and 1 respectively.
Rebalancing partitions when the clustering index does not match the partitioning key

If a partitioned table has a clustering index that is not the partitioning index and you run a reorganization that specifies REBALANCE on the table space, the rows will no longer be in clustering order and all reorganized partitions will be placed in AREO* status. To put the rows back in clustering order and reset the AREO* status, you must run another reorganization without specifying REBALANCE. The following example illustrates this consideration.

The following SQL creates a table space, a table, and an index with different columns for partitioning and clustering.

```sql
CREATE TABLESPACE tableSpaceName IN databaseName
USING STOGROUP storageGroup
NUMPARTS 4 BUFFERPOOL BP0;
CREATE TABLE tableSpaceName (C01 CHAR(5) NOT NULL,
C02 CHAR(5) NOT NULL,
C03 CHAR(5) NOT NULL)
IN databaseName.tableSpaceName
PARTITION BY (C01)
(PART 1 VALUES ('00001'),
PART 2 VALUES ('00002'),
PART 3 VALUES ('00003'),
PART 4 VALUES ('00004'));
CREATE INDEX indexName ON tableSpace(C02) CLUSTER;
```

For the first table space reorganization, you would use the following syntax to rebalance the data across the four partitions:

```sql
REORG TABLESPACE databaseName.tableSpaceName REBALANCE
```

The partitions are placed in AREO* status after the reorganization completes. This status indicates that you should run another reorganization to order the rows in clustering order. You would use the following syntax for the second reorganization:

```sql
REORG TABLESPACE databaseName.tableSpaceName
```
Conditional reorganization

A conditional reorganization means that REORG PLUS performs the reorganization only if it is needed. The CONDEXEC command and installation options tell REORG PLUS whether to check the criteria for a conditional reorganization. The CONDEXEC options also tell REORG PLUS whether to use information from the DB2 catalog (CONDEXEC YES) or the DASD MANAGER PLUS exceptions table (CONDEXEC BMC). Using the exceptions table allows you to set more conditions that might cause the reorganization to be performed. However, the BMC value is available only if you are using REORG PLUS as part of the Database Performance for DB2 solution. For more information about the BMC value, see “Conditional reorganizations using the DASD MANAGER PLUS exceptions table” on page 138.

**NOTE**

Conditional execution is not supported when you are reorganizing LOB table spaces. REORG PLUS terminates if you are reorganizing a LOB table space, and command or installation options would cause REORG PLUS to check the criteria for a conditional reorganization.

Conditional reorganizations using the DB2 catalog

After determining whether to check to see if a conditional reorganization should be performed by using the DB2 catalog, REORG PLUS uses the limit command and installation options and the CONDEXEC option to determine what action to take. The limit command options are OFFPOSLIMIT, INDREFLIMIT, and LEAFDISTLIMIT; the limit installation options are OFFPOSLM, INDREFLM, and LEAFDLSLM.

**CONDEXEC options**

If the value of the CONDEXEC command or installation option is YES, REORG PLUS uses the limit command options or installation options. REORG PLUS compares the value that you specify on the limit options to the value that it calculates using the data that it retrieves from the DB2 catalog. After issuing a report, REORG PLUS reorganizes the object if the calculated value exceeds the limit that you set. To have REORG PLUS issue the report with the recommendations but without performing any reorganizations, specify REPORTONLY in the command.

Table 27 on page 137 shows the actions that REORG PLUS takes based on the value of the CONDEXEC command and installation options. The command option overrides the installation option.
Limit options

REORG PLUS examines the limit command and installation options when CONDEXEC YES is in effect. It also examines the limit command options if you do not specify CONDEXEC on the command, regardless of the CONDEXEC installation option value.

Table 28 shows how the limit command options interact with the limit installation options and the resulting REORG PLUS action.

<table>
<thead>
<tr>
<th>CONDEXEC value</th>
<th>REORG PLUS action</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>command option is NO</td>
<td>ignores all limit command and installation option values and performs the reorganization</td>
<td>Not applicable</td>
</tr>
<tr>
<td>command option is YES</td>
<td>uses the limit command or installation option values to determine whether to perform the reorganization</td>
<td>Table 28</td>
</tr>
<tr>
<td>installation option is NO</td>
<td>ignores the limit installation options and performs the reorganization unless a limit command option is specified</td>
<td></td>
</tr>
<tr>
<td>installation option is YES</td>
<td>uses the limit installation options to determine whether to perform the reorganization, unless those options are overridden by command options</td>
<td></td>
</tr>
<tr>
<td>command or installation option is BMC</td>
<td>ignores all limit command and installation options and uses the DASD MANAGER PLUS exceptions table to determine whether to perform the reorganization</td>
<td>“Conditional reorganizations using the DASD MANAGER PLUS exceptions table” on page 138</td>
</tr>
</tbody>
</table>

The command syntax for CONDEXEC and the limit options begins on page 230. For a description of the installation options, see Appendix A, “REORG PLUS installation options.”
Conditional reorganizations using the DASD MANAGER PLUS exceptions table

When you specify a threshold for an exception in the BMCTRG function of DASD MANAGER PLUS and then specify one or more objects, BMCTRG evaluates the objects against the exception criteria. If you tell it to save exceptions, BMCTRG puts a row into the DASD MANAGER PLUS exceptions table for every exception that meets or exceeds the threshold.

If the value of the CONDEXEC command or installation option is BMC, REORG PLUS completes the following actions:

1. Examines the DASD MANAGER PLUS exceptions table to determine whether the reorganization is needed.

2. If REORG PLUS finds an exception for the object to be reorganized and the exception is active, REORG PLUS reorganizes the object. (If REORG PLUS does not find an exception for the object, it does not perform the reorganization.)

3. REORG PLUS updates the exceptions table to indicate that the exception is not active.

Using the DASD MANAGER PLUS exceptions table allows REORG PLUS to use an expanded set of conditions to trigger a conditional reorganization. If you have existing REORG PLUS jobs that run on a regular schedule and you use CONDEXEC BMC, REORG PLUS will reorganize only objects that need reorganization. To have REORG PLUS issue an exceptions report but not perform any reorganizations, specify REPORTONLY on the command. For a description of the CONDEXEC command option, see page 230. For a description of REPORTONLY, see page 236.

**NOTE**
The value BMC for the CONDEXEC option is available only if you are using the Database Performance for DB2 solution. Otherwise, REORG PLUS ignores the option and continues with the reorganization.

Considerations when using DASD MANAGER PLUS exceptions

Table 29 on page 139 describes the general actions that REORG PLUS takes based on the type of exception found in the DASD MANAGER PLUS exceptions table.
The following additional considerations apply to the actions that REORG PLUS takes when using the DASD MANAGER PLUS exceptions table:

- If the exceptions table contains multiple rows for the table space that you are reorganizing, REORG PLUS updates the exceptions table as follows:
  
  — For a full table space reorganization, REORG PLUS sets all rows to inactive status.
  
  — For a partial table space reorganization, REORG PLUS sets to inactive status all rows that apply to the participating partitions.

- For any index built during a table space reorganization, REORG PLUS sets the exceptions for the index to inactive. If you request a conditional index reorganization on this index after the table space reorganization, REORG PLUS does not run the index reorganization.

- Table exceptions do not roll up to the table space level. Therefore, table exceptions can trigger either a table space or index reorganization. For information about which exceptions trigger which type of reorganization, see Table 30 on page 140.
Exceptions that trigger a reorganization

Table 30 lists exceptions that cause REORG PLUS to reorganize the object. The table lists the following information:

- type of object that will be reorganized
- value that REORG PLUS checks in the exceptions table
- brief description of the exception
- corresponding field name that you set on the DASD MANAGER PLUS panels for the BMCTRIG utility

For a detailed description of the exceptions, see the *DASD MANAGER PLUS for DB2 Reference Manual*.

Table 30  Exceptions that initiate a reorganization (part 1 of 3)

<table>
<thead>
<tr>
<th>Reorganization type</th>
<th>Value in the exceptions table</th>
<th>DASD MANAGER PLUS field name</th>
<th>Brief description of the exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>either table space or index</td>
<td>CARD</td>
<td>Card</td>
<td>percent increase in the cardinality of a table or index&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>DSEXENT</td>
<td>Dsexents</td>
<td>number of extents at the data set level</td>
</tr>
<tr>
<td></td>
<td>EXTENTS</td>
<td>Extents</td>
<td>table spaces or indexes that have reached a certain number of extents&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>REORMDEL</td>
<td>Mass del reorg</td>
<td>mass deletion&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>REORMMODS</td>
<td>Mods since reorg</td>
<td>initiates a reorganization after a large number of rows have been modified&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>REORSPAC</td>
<td>Reorgspace</td>
<td>percentage of space used that is over or under the minimum space required for the object&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>Space</td>
<td>percentage of change in tracks that an object uses&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>BMCRUSER</td>
<td>BMCRUSER</td>
<td>user-defined exception in DASD MANAGER PLUS</td>
</tr>
</tbody>
</table>

<sup>a</sup> Note: For information about setting up a user-defined exception and naming it BMCRUSER, see the *DASD MANAGER PLUS for DB2 User Guide*.  

---

*Conditional reorganizations using the DASD MANAGER PLUS exceptions table*
### Table 30  Exceptions that initiate a reorganization (part 2 of 3)

<table>
<thead>
<tr>
<th>Reorganization type</th>
<th>Value in the exceptions table</th>
<th>DASD MANAGER PLUS field name</th>
<th>Brief description of the exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>table space</td>
<td>FARIND</td>
<td>Farind</td>
<td>percentage of rows in a table space that are more than 16 pages from their original page</td>
</tr>
</tbody>
</table>
|                     | FAROFF                         | Faroff                        | percentage of rows that are more than 16 pages from the optimal position\(^b\)  
|                     |                                |                               | This exception is evaluated only for clustering indexes. A high Faroff percentage indicates that clustering might be degrading. |
|                     | PACTHI                         | PctActivHi                   | table spaces in which the percentage of active pages is greater than or equal to the specified value |
|                     | PACTLO                         | PctActivLo                   | table spaces in which the percentage of active pages is less than or equal to the specified value |
|                     | PCTCLUS                        | PctClus                      | cluster ratio of the object\(^b\) |
|                     | PCTDROP                        | Pct Dropped Rows             | percentage of space that is occupied by dropped rows |
|                     | REORPEND                       | Reorg Pend                   | indication that the partition or object is in REORG pending status |
|                     | TOTALIND                       | Totalind                     | percentage of rows in a table that are not in their optimal positions |
|                     | TOTALOFF                       | Totaloff                     | percentage of table space rows that are not in optimal position by the index key\(^b\) |
|                     | UNCLUST                        | Unclust inserts              | number of unclustered inserts |
Table 30  Exceptions that initiate a reorganization (part 3 of 3)

<table>
<thead>
<tr>
<th>Reorganization type</th>
<th>Value in the exceptions table</th>
<th>DASD MANAGER PLUS field name</th>
<th>Brief description of the exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>APPNDINS</td>
<td>Append inserts</td>
<td>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></td>
<td>AREOPENED</td>
<td>Advisory pending</td>
<td>whether the index is in advisory REORG pending status</td>
</tr>
<tr>
<td></td>
<td>LEAFDIST</td>
<td>Leafdist</td>
<td>the distance in page IDs between successive leaf pages during a sequential access of the index</td>
</tr>
<tr>
<td></td>
<td>LEAFFOFF</td>
<td>LeafFarOff</td>
<td>percentage of leaf pages that are physically located far from the previous leaf page</td>
</tr>
<tr>
<td></td>
<td>LEAFOFF</td>
<td>LeafTotOff</td>
<td>percentage of leaf pages that are not in optimal position</td>
</tr>
<tr>
<td></td>
<td>LEVELINC</td>
<td>LevelInc</td>
<td>increase in the number of index levels</td>
</tr>
<tr>
<td></td>
<td>LEVELMIN</td>
<td>LevelMin</td>
<td>number of index levels that are greater than the minimum number required</td>
</tr>
<tr>
<td></td>
<td>LEVELS</td>
<td>Levels</td>
<td>number of index levels</td>
</tr>
<tr>
<td></td>
<td>NUNIFORM</td>
<td>NonUniform</td>
<td>nonuniformity of an index based on the values that appear in the SYSCOLDISTSTATS catalog table. Up to 10 distinct values can appear in the catalog table.</td>
</tr>
<tr>
<td></td>
<td>PSEUDODL</td>
<td>Pseudo Deleted Key</td>
<td>percentage of keys that have been pseudo-deleted within an index partitiona</td>
</tr>
<tr>
<td></td>
<td>ROWS/KEY</td>
<td>Rows/Key</td>
<td>average number of rows per key value</td>
</tr>
</tbody>
</table>

a  This exception is evaluated at the partition level.

b  This exception is an index-based trigger.

**DB2 real-time statistics**

REORG PLUS resets the real-time statistics and updates timestamp values (in both the DSNRTSDB and DB2 memory) during the UTILTERM phase. Table 31 on page 143 lists the columns in tables SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS that REORG PLUS resets for each partition and the updated value.
**NOTE**

When REORG PLUS invokes DSNUTILB, updating real-time statistics tables is handled by DSNUTILB. Therefore, the information in this section about how the tables are updated does not apply to this type of reorganization job.

Table 31  Real-time statistics updates (part 1 of 2)

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Updated value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SYSIBM.SYSTABLESPACESTATS</code></td>
<td><code>UPDATESTATSTIME</code></td>
<td>timestamp of the update</td>
</tr>
<tr>
<td></td>
<td><code>NACTIVE</code></td>
<td>actual value</td>
</tr>
<tr>
<td></td>
<td><code>NPAGES</code></td>
<td>actual value</td>
</tr>
<tr>
<td></td>
<td><code>EXTENTS</code></td>
<td>actual value</td>
</tr>
<tr>
<td></td>
<td><code>REORGLASTTIME</code></td>
<td>timestamp of the reset</td>
</tr>
<tr>
<td></td>
<td><code>REORGINSERTS</code></td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td><code>REORGDELETES</code></td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td><code>REORGUPDATES</code></td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td><code>REORGUNCLUSTINS</code></td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td><code>REORGDISORGLOB</code></td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td><code>REORGMASSDELETE</code></td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td><code>REORGNEARINDREF</code></td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td><code>REORGFARINDREF</code></td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td><code>STATSLASTTIME</code></td>
<td><code>(UPDATEDB2STATS YES only)</code></td>
</tr>
<tr>
<td></td>
<td><code>COPYLASTTIME</code></td>
<td>timestamp of the update when REORG PLUS took the copy</td>
</tr>
<tr>
<td></td>
<td><code>COPYUPDATEDPAGES</code></td>
<td>zero when a copy is taken</td>
</tr>
<tr>
<td></td>
<td><code>COPYCHANGES</code></td>
<td>zero when a copy is taken</td>
</tr>
<tr>
<td></td>
<td><code>COPYUPDATELRSN</code></td>
<td>null when a copy is taken</td>
</tr>
<tr>
<td></td>
<td><code>COPYUPDATETIME</code></td>
<td>null when a copy is taken</td>
</tr>
<tr>
<td></td>
<td><code>SPACE</code></td>
<td>actual value</td>
</tr>
<tr>
<td></td>
<td><code>TOTALROWS</code></td>
<td>actual value</td>
</tr>
<tr>
<td></td>
<td><code>DATASIZE</code></td>
<td>actual value</td>
</tr>
</tbody>
</table>
For a performance gain, you can combine the UNLOAD and RELOAD phases into a single processing phase, named REORG. You do so by setting the value of the UNLOAD command or installation option to RELOAD.

The following considerations apply to single-phase reorganizations:

- When running a single-phase reorganization with SHRLEVEL NONE (the default), you can omit the following data sets:
  - for a table space reorganization, the SYSREC data set
  - for an index reorganization, the SYSUT1 data set

  Omitting the data set provides a performance boost. However, if you omit the data set, your job might not be restartable.

---

**Table 31  Real-time statistics updates (part 2 of 2)**

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Updated value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIBM.SYSINDEXSPACESTATS&lt;sup&gt;a&lt;/sup&gt;</td>
<td>UPDATESTATSTIME</td>
<td>timestamp of the reset</td>
</tr>
<tr>
<td></td>
<td>NLEVELS</td>
<td>actual value</td>
</tr>
<tr>
<td></td>
<td>NLEAF</td>
<td>actual value</td>
</tr>
<tr>
<td></td>
<td>NACTIVE</td>
<td>actual value</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>actual value</td>
</tr>
<tr>
<td></td>
<td>EXTENTS</td>
<td>actual value</td>
</tr>
<tr>
<td></td>
<td>REORGLASTTIME</td>
<td>timestamp of the reset</td>
</tr>
<tr>
<td></td>
<td>REORGINSETS</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>REORGDELETES</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>REORGAPPENDINSERT</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>REORGPSUEDODELETES</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>REORGMASSDELETE</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>REORGLEANFAR</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>REORGLEAFFAR</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>REORGNUMLEVELS</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>STATSLASTTIME</td>
<td>(UPDATEDB2STATS YES only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>timestamp of the update</td>
</tr>
<tr>
<td></td>
<td>COPYLASTTIME</td>
<td>timestamp of the update when REORG PLUS took the copy</td>
</tr>
<tr>
<td></td>
<td>TOTALENTRIES</td>
<td>actual value</td>
</tr>
</tbody>
</table>

<sup>a</sup> For nonpartitioned indexes in partial reorganizations, REORG PLUS updates only the SPACE and EXTENTS columns of this table.
When running a single-phase reorganization with SHRLEVEL REFERENCE or SHRLEVEL CHANGE, REORG PLUS does not use the following data sets, even if specified:

— for a table space reorganization, the SYSREC data set
— for an index reorganization, the SYSUT1 data set

For SHRLEVEL REFERENCE, the job is restartable due to the nondestructive nature of this type of reorganization.

For SHRLEVEL CHANGE, REORG PLUS is not restartable until the UTILTERM phase begins. There is no need to restart the job before the UTILTERM phase begins because the original data sets are not modified.

For detailed usage requirements for the SYSREC and SYSUT1 data sets, see the following references:

- “SYSREC data sets” on page 379
- “SYSUT1 data sets” on page 382

For information about restarting the job during a single-phase reorganization, see “Restarting REORG PLUS” on page 391.

Table space compression

This section describes how REORG PLUS handles table space compression.

The KEEPDICTIONARY installation option (page 703) determines how REORG PLUS handles compression at a global level. You can override this value by specifying the KEEPDICTIONARY command option at either the table space or partition level. If you specify the KEEPDICTIONARY command option without a value, REORG PLUS uses KEEPDICTIONARY YES, regardless of the value of the installation option.
Considerations when using table space compression

Note the following considerations when using table space compression with REORG PLUS:

- When both of the following conditions exist, specify a member subsystem ID (SSID) for your reorganization job instead of a group attachment name. Ensure that the DB2 version of the SSID that you specify corresponds to the version of the DB2 load library that you specify in your STEPLIB.
  - You are running REORG PLUS in a data sharing environment.
  - The subsystems in your data sharing group are not all at the same DB2 version level.

- Reorganizing a large number of compressed partitions might result in constrained resources. If you encounter this problem, consider specifying fewer partitions in a single job. (Whether the number of compressed partitions that you have is considered to be a large number depends on your environment.)

- REORG PLUS reports the percentage of compressed rows in messages BMC50512I and BMC50522I. This percentage does not include rows that are compressed by an EDITPROC.

- When REORG PLUS invokes DSNUTILB, compression is handled by DSNUTILB. For information about how compression is handled for these jobs, see the documentation for the REORG utility in the IBM DB2 Utility Guide and Reference.

Building a dictionary

REORG PLUS builds a new dictionary when any of the following conditions exists:

- The value of KEEPDICTIONARY is NO.

- The value of KEEPDICTIONARY is YES, and one of the following conditions is in effect:
  - A dictionary does not exist.
  - REORG PLUS is converting your data from BRF to RRF and the value of the DB2 subsystem parameter HONOR_KEEPDICTIONARY is NO.

- You are rebalancing partitions of a table space with compression. (For more information about partition rebalancing, see page 127.)
After REORG PLUS builds the dictionary, REORG PLUS uses it to compress the data rows. REORG PLUS stores the dictionary in the BMCDICT table until the reorganization completes.

**NOTE**

REORG PLUS stores a dictionary in BMCDICT for each partition that you are reorganizing. If you are reorganizing a table space with a large number of partitions, you might need to increase the size of the BMCDICT table space from the standard size that was allocated during installation. For more information about the BMCDICT table, see page 769.

Note the following considerations about building a dictionary:

- REORG PLUS reserves enough pages for the largest dictionary size.

- If REORG PLUS does not unload enough rows to build a complete dictionary, REORG PLUS
  - does not compress any rows
  - builds a non-optimal dictionary for future use

- For partition-by-growth table spaces, whether REORG PLUS builds a new dictionary depends on the value of the ORDER option, as follows:

  — For ORDER NO, REORG PLUS ignores KEEPDICTIONARY YES and builds a new dictionary for each added partition that has enough rows to build a dictionary.

  — For ORDER YES, REORG PLUS copies the dictionary for each added partition from the previous partition, regardless of the KEEPDICTIONARY value.

**Keeping a dictionary**

If the value of KEEPDICTIONARY is YES and a dictionary already exists, REORG PLUS uses the dictionary for compression. When using the existing dictionary, REORG PLUS compresses all of the data rows and does not have to spend processing time identifying patterns and building the dictionary.
Index compression

The following considerations apply to index compression:

- When the following conditions exist, you must specify REDEFINE YES:
  - Before the reorganization, an ALTER occurred that changed the compression attribute of a participating index.
  - The DSVCI system parameter is set to YES, indicating that the control interval (CI) size is variable.

- Compressed indexes have additional restart considerations. For more information, see “Compressed indexes” on page 393.

Indexes on expression

REORG PLUS natively supports reorganizing indexes that contain a key derived from certain expressions, and reorganizing table spaces that contain those indexes. With the exceptions described in “Restrictions on indexes on expression” on page 149, REORG PLUS currently supports expressions that contain the following SQL functions:

- UPPER
- LOWER
- SUBSTR
- DATE
- DAY
- MONTH
- YEAR

For all other functions that IBM supports for indexes on expression, REORG PLUS invokes DSNUTILB.
Restrictions on indexes on expression

When any of the following conditions exists, REORG PLUS does not provide native support and invokes DSNUTILB instead:

- A SUBSTR function has any of the following characteristics:
  - The function is on a numeric column.
  - The function uses columns in the table space to supply start and end values.
  - The function is on a LOB column (for example, an index on an inline LOB column).

- An expression contains nested functions.

- An UPPER or LOWER function has any of the following characteristics:
  - The function includes a length parameter.
  - The locale name has a value other than UNI or blank.
  - The column contains mixed or DBCS data, and the locale name is blank.
  - For ASCII and EBCDIC objects, the column contains mixed data.

- A DATE, DAY, MONTH, or YEAR function has any of the following characteristics:
  - The function references more than one column.
  - The column data type is not DATE, TIMESTAMP, or TIMESTAMP WITH TIME ZONE.

For more information, see “Reorganization jobs that invoke DSNUTILB” on page 72.

Reordered row format

REORG PLUS supports reordered row format (RRF) when you are running on DB2 Version 9 (new-function mode) or later, as follows:

- REORG PLUS reorganizes data that is in RRF.
- REORG PLUS converts data from basic row format (BRF) to RRF.

Considerations
The following considerations apply to RRF support in REORG PLUS:

- When converting objects from BRF to RRF, REORG PLUS ignores FSFALLBACK YES.
REORG PLUS does not support mixed-format partitions in a single reorganization if you have disabled RRF. In this case, you must run separate partial reorganizations: one to reorganize the BRF partitions, and another to reorganize the RRF partitions.

If a table space is compressed and a REORG PLUS job would convert the row format, REORG PLUS builds a new dictionary except when both of the following options are in effect:

— The value of the REORG PLUS KEEPDICTIONARY option is YES.
— The value of the DB2 subsystem parameter HONORKEEPDICTIONARY is YES.

REORG PLUS does not convert objects from BRF to RRF when any of the following conditions exists:

— Any table in the table space has an EDITPROC or VALIDPROC.
— You are reorganizing a catalog object.
— You are reorganizing a clone object or a base table space for which a clone relationship exists.
— REORG PLUS is running on a subsystem that has been migrated directly from DB2 Version 8 to DB2 Version 10 and has not yet been migrated to new-function mode.

**Recoverability of the reorganized table space**

If you specify COPY NO (or do not specify a COPY option), you must take some action to ensure that DB2 can recover the table space, if necessary, after the reorganization. For example, you can

■ Create an image copy by running COPY PLUS or the IBM DB2 COPY utility.
■ Execute DSN1COPY or any other suitable substitute that your system provides.

If you use the table space as a read-only table space, specify COPY NO or specify COPY YES REGISTER NONE and DD DUMMY for the copy data sets.
Referential integrity

REORG PLUS performs no checking for referential constraint violations, nor does it set the CHKP status for reorganizations that result in violations. To maintain referential integrity, consider the following information:

- If you use the UPDATE option and update a column that is part of a foreign key, ensure that the new column value does not violate any referential constraints.

- If you use the SELECT or DELETE option to delete rows from a primary table, you must ensure that the dependent rows are deleted as well.

DB2 user exits

REORG PLUS invokes EDITPROCs during processing. If the value of the UXSTATE installation option is SUP, REORG PLUS invokes the exit in supervisor state (and PSW key=7). If you can ensure that all EDITPROCs called by REORG PLUS are able to run in problem state, you can specify UXSTATE=PROB in the REORG PLUS installation options module.

XBM and SUF considerations

To enhance performance, REORG PLUS uses several features of XBM or SUF during portions of the reorganization process. If you have installed the required version of XBM or SUF, REORG PLUS can use the following XBM or SUF functions:

- Software snapshot functions for a SHRLEVEL CHANGE reorganization

- Instant Snapshot technology for the nonpartitioned indexes during a SHRLEVEL REFERENCE or SHRLEVEL CHANGE partial table space reorganization

- zIIP processing

XBM and SUF are licensed, installed, and maintained separately from REORG PLUS. You can use either XBM or SUF, depending on the license that you have obtained:

- A license for the full version of the XBM product authorizes you to use all XBM features.

- A license for SUF authorizes you to use only the snapshot and zIIP-processing features of XBM.
Using XBM or SUF with SHRLEVEL CHANGE

NOTE
If you are licensed only for a BMC solution that contains REORG PLUS, your license authorizes you to use SUF, not the full version of XBM.

For more information, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

Using XBM or SUF with SHRLEVEL CHANGE

By using the software snapshot functions of XBM or SUF when performing a SHRLEVEL CHANGE reorganization, REORG PLUS can reorganize and apply the log records to a consistent image of the data while the data is available for RW operations. Using XBM or SUF also allows you to monitor and alter the execution of the reorganization by using the MVS console or the XBM Utility Monitor.

For REORG PLUS processing, software snapshots are most efficient and require the least amount of time to register. For proper REORG PLUS operation, ensure that XBM or SUF is configured with sufficient cache for the number of concurrent snapshots and the level of DB2 update activity at your site. For specific cache configuration information, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

REORG PLUS does not use the traditional hardware snapshot functions of XBM or SUF, even if XBM or SUF is configured to take hardware snapshots.

For the steps to take to run a SHRLEVEL CHANGE reorganization, see “Running a SHRLEVEL CHANGE reorganization” on page 631.

Using XBM or SUF with nonpartitioned indexes

For a SHRLEVEL REFERENCE or SHRLEVEL CHANGE partial table space reorganization, REORG PLUS can use Instant Snapshot technology to copy each nonpartitioned index to a staging data set. Instant Snapshot copies the entire nonpartitioned index in one operation, thus eliminating the multiple reads and writes of the traditional copy method (for SHRLEVEL REFERENCE) and the software snapshots (for SHRLEVEL CHANGE).

For the steps to take to use Instant Snapshot technology in this case, see “To use Instant Snapshot technology to copy nonpartitioned indexes” on page 122.
Using XBM or SUF to enable zIIP processing

You can use a component of XBM or SUF to enable zIIP processing during your reorganization.

To enable zIIP processing

1. Ensure that you have a version of XBM or SUF that supports zIIP processing, as described in “Software requirements” on page 66.

2. Ensure that an XBM subsystem is started.

3. (optional) Use the XBMID installation or command option to specify the XBM subsystem that you want REORG PLUS to access.

   For information about the XBMID command option, see “XBMID” on page 253.

4. Ensure that the value of the ZIIP installation or command option is YES.

   For information about the ZIIP command option, see “ZIIP” on page 252.

DB2 features that REORG PLUS does not support

This version of REORG PLUS does not support, or provides limited support, for the following features of DB2.

- This version of REORG PLUS does not support the following objects or features:
  - logging previously existing compression dictionaries
  - indexes in which updated entries remain in their original position on the page
  - IBM FlashCopy® image copies
  - row- and column-level security
  - use of striped data sets for DB2 VSAM objects
  - user-defined column default values that are greater than 255 bytes
  - LISTDEF and TEMPLATE control statements

   However, REORG PLUS provides the DSNPAT installation or command option, which allows you to specify a data set name pattern for your dynamically allocated data sets.
REORG PLUS does not support the following types of columns or constants on the WHERE clause of SELECT, DELETE, or UPDATE statements:

- XML columns
- LOB columns
- floating-point columns
- decimal floating point columns
- row ID columns
- binary string constants or columns
- columns that are defined with a FIELDPROC

Support for reorganizing Unicode data or reorganizing data into Unicode objects has the following limitations:

- REORG PLUS processes objects with Unicode names only if all characters in the name can be translated to EBCDIC.

- REORG PLUS does not support SYSIN input in Unicode format.
Syntax of the REORG command

This chapter presents the following topics:

- Command syntax rules for REORG PLUS .................................................. 155
- Alphabetical listing of REORG PLUS command options ............................ 156
- Command syntax diagrams for REORG PLUS ........................................... 160
- Descriptions of REORG PLUS command options .................................... 170
  - Basic processing options ................................................................. 170
  - Statistics options ................................................................. 258
  - Copy options for REORG TABLESPACE ........................................... 264
  - Selective unload and update options for REORG TABLESPACE .......... 279
  - Dynamic allocation options ......................................................... 293
  - SHRLEVEL CHANGE options ....................................................... 326

Command syntax rules for REORG PLUS

The following general rules apply to the REORG command syntax:

- In a REORG command, REORG PLUS considers any line beginning with an asterisk (*) in column 1 to be a comment and ignores it.

- When encountering two consecutive hyphens in a line (except within a delimited token), REORG PLUS considers everything on that line after the hyphens to be a comment and ignores the remainder of the line.

- If you specify the same command option more than once, REORG PLUS uses only the last option that you specify. For example, if you specify the following options, REORG PLUS accepts UNLOAD CONTINUE as the processing option:

  UNLOAD RELOAD UNLOAD CONTINUE

- When you use a signed token, do not place a space between the sign and the value.
You can split a token (such as a keyword, identifier, or constant) across a line. However, REORG PLUS ignores anything in columns 73 through 80.

A blank, or a delimiter if the token is delimited, indicates the end of the token. If an undelimited token ends in column 72, column 1 on the next line must be blank.

**NOTE**

If you use applications that automate JCL submission and resolve symbolic variables within your JCL, the resulting control cards might not appear as they do in the JCL that you created. These applications might produce an invalid command statement.

In the syntax diagrams in the following pages, underlined options indicate default options. For more information about how to read the diagrams, see “Syntax diagrams” on page 23.

When specifying object names, you can specify objects that contain Unicode data. However, you cannot specify delimited object names that do not have a character representation in EBCDIC.

**Alphabetical listing of REORG PLUS command options**

As a quick reference, Table 32 lists the REORG PLUS command options alphabetically and indicates where to find each option description. The third column indicates whether the option is valid for an index-only reorganization.

**Table 32 REORG PLUS command options (part 1 of 5)**

<table>
<thead>
<tr>
<th>Command option</th>
<th>See page</th>
<th>Valid for REORG INDEX?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>296</td>
<td>yes</td>
</tr>
<tr>
<td>ALLOC</td>
<td>299</td>
<td>yes</td>
</tr>
<tr>
<td>AMENDED</td>
<td>214</td>
<td>no</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>199</td>
<td>yes</td>
</tr>
<tr>
<td>ARCHDDN</td>
<td>185</td>
<td>no</td>
</tr>
<tr>
<td>ARCHFORMAT</td>
<td>185</td>
<td>no</td>
</tr>
<tr>
<td>ARROWS</td>
<td>229</td>
<td>no</td>
</tr>
<tr>
<td>ASSOCIATE</td>
<td>191</td>
<td>no</td>
</tr>
<tr>
<td>AUTOESTSPACE</td>
<td>256</td>
<td>no</td>
</tr>
<tr>
<td>AUX</td>
<td>256</td>
<td>no</td>
</tr>
<tr>
<td>AVAILPAGEPCT</td>
<td>193</td>
<td>yes</td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>306</td>
<td>yes</td>
</tr>
<tr>
<td>Command option</td>
<td>See page</td>
<td>Valid for REORG INDEX?</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>BMCHIST</td>
<td>208</td>
<td>yes</td>
</tr>
<tr>
<td>BMCSTATS</td>
<td>259</td>
<td>yes</td>
</tr>
<tr>
<td>CLONE</td>
<td>171</td>
<td>yes</td>
</tr>
<tr>
<td>CONDEXEC</td>
<td>230</td>
<td>yes</td>
</tr>
<tr>
<td>COPY</td>
<td>264</td>
<td>no</td>
</tr>
<tr>
<td>COPYDDN</td>
<td>271</td>
<td>no</td>
</tr>
<tr>
<td>COPYLVL</td>
<td>269</td>
<td>no</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>325</td>
<td>yes</td>
</tr>
<tr>
<td>DDLDDN</td>
<td>187</td>
<td>no</td>
</tr>
<tr>
<td>DDTYPE</td>
<td>294</td>
<td>yes</td>
</tr>
<tr>
<td>DEADLINE</td>
<td>327</td>
<td>yes</td>
</tr>
<tr>
<td>DELAY</td>
<td>330</td>
<td>yes</td>
</tr>
<tr>
<td>DELETE</td>
<td>284</td>
<td>no</td>
</tr>
<tr>
<td>DELETEFILES</td>
<td>215</td>
<td>yes</td>
</tr>
<tr>
<td>DRAIN</td>
<td>340</td>
<td>yes</td>
</tr>
<tr>
<td>DRAIN_WAIT</td>
<td>245</td>
<td>yes</td>
</tr>
<tr>
<td>DSNPAT</td>
<td>315</td>
<td>yes</td>
</tr>
<tr>
<td>DSNTYPE</td>
<td>309</td>
<td>yes</td>
</tr>
<tr>
<td>DSNUEXIT</td>
<td>240</td>
<td>yes</td>
</tr>
<tr>
<td>DSNUUTILB</td>
<td>237</td>
<td>yes</td>
</tr>
<tr>
<td>DSPLOCKS</td>
<td>247</td>
<td>yes</td>
</tr>
<tr>
<td>DSRSEXIT</td>
<td>241</td>
<td>yes</td>
</tr>
<tr>
<td>EXPDT</td>
<td>322</td>
<td>no</td>
</tr>
<tr>
<td>FASTSWITCH</td>
<td>218</td>
<td>yes</td>
</tr>
<tr>
<td>FORCE</td>
<td>248</td>
<td>yes</td>
</tr>
<tr>
<td>FORCE_AT</td>
<td>248</td>
<td>yes</td>
</tr>
<tr>
<td>FORCE_DELAY</td>
<td>249</td>
<td>yes</td>
</tr>
<tr>
<td>FORCE_RPT</td>
<td>249</td>
<td>yes</td>
</tr>
<tr>
<td>FSFALLBACK</td>
<td>220</td>
<td>yes</td>
</tr>
<tr>
<td>FSTHRESHOLD</td>
<td>221</td>
<td>yes</td>
</tr>
<tr>
<td>GDGLIMIT</td>
<td>323</td>
<td>no</td>
</tr>
<tr>
<td>ICDNN</td>
<td>275</td>
<td>no</td>
</tr>
<tr>
<td>ICTYPE</td>
<td>267</td>
<td>no</td>
</tr>
<tr>
<td>IDCACHE</td>
<td>255</td>
<td>no</td>
</tr>
<tr>
<td>IDCDDN</td>
<td>214</td>
<td>yes</td>
</tr>
<tr>
<td>JFALLOC</td>
<td>297</td>
<td>yes</td>
</tr>
<tr>
<td>IMPLICIT_TZ</td>
<td>281</td>
<td>no</td>
</tr>
<tr>
<td>INDFRFLIMIT</td>
<td>233</td>
<td>no</td>
</tr>
</tbody>
</table>
## Table 32  REORG PLUS command options (part 3 of 5)

<table>
<thead>
<tr>
<th>Command option</th>
<th>See page</th>
<th>Valid for REORG INDEX?</th>
</tr>
</thead>
<tbody>
<tr>
<td>INLINE</td>
<td>265</td>
<td>no</td>
</tr>
<tr>
<td>KEEPDICTIONARY</td>
<td>206</td>
<td>no</td>
</tr>
<tr>
<td>LEAFDISTLIMIT</td>
<td>234</td>
<td>yes</td>
</tr>
<tr>
<td>LOG</td>
<td>209</td>
<td>no</td>
</tr>
<tr>
<td>LOGFINAL</td>
<td>338</td>
<td>yes</td>
</tr>
<tr>
<td>LOGMEM</td>
<td>337</td>
<td>yes</td>
</tr>
<tr>
<td>LOGSPILL</td>
<td>338</td>
<td>yes</td>
</tr>
<tr>
<td>LOGTHRESHLD</td>
<td>326</td>
<td>yes</td>
</tr>
<tr>
<td>LONGLOG</td>
<td>329</td>
<td>yes</td>
</tr>
<tr>
<td>LONGNAMETRUNC</td>
<td>178</td>
<td>yes</td>
</tr>
<tr>
<td>MAPPINGDATABASE</td>
<td>340</td>
<td>no</td>
</tr>
<tr>
<td>MAPPINGTABLE</td>
<td>340</td>
<td>no</td>
</tr>
<tr>
<td>MAPTEXT</td>
<td>243</td>
<td>no</td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td>307</td>
<td>yes</td>
</tr>
<tr>
<td>MAXNEWPARTS</td>
<td>177</td>
<td>no</td>
</tr>
<tr>
<td>MAXRO</td>
<td>326</td>
<td>yes</td>
</tr>
<tr>
<td>MAXSORTMEMORY</td>
<td>197</td>
<td>yes</td>
</tr>
<tr>
<td>MAXSORTS</td>
<td>189</td>
<td>yes</td>
</tr>
<tr>
<td>MAXTAPE</td>
<td>293</td>
<td>yes</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>325</td>
<td>yes</td>
</tr>
<tr>
<td>MINSORTMEMORY</td>
<td>197</td>
<td>yes</td>
</tr>
<tr>
<td>NLPCTFREE</td>
<td>210</td>
<td>yes</td>
</tr>
<tr>
<td>NOSYSREC</td>
<td>205</td>
<td>no</td>
</tr>
<tr>
<td>OFFPOSLIMIT</td>
<td>231</td>
<td>no</td>
</tr>
<tr>
<td>ON FAILURE</td>
<td>224</td>
<td>yes</td>
</tr>
<tr>
<td>ON MESSAGE</td>
<td>222</td>
<td>no</td>
</tr>
<tr>
<td>ORDER</td>
<td>190</td>
<td>yes</td>
</tr>
<tr>
<td>ORIGINALDISP</td>
<td>217</td>
<td>yes</td>
</tr>
<tr>
<td>PART</td>
<td>175</td>
<td>yes</td>
</tr>
<tr>
<td>PREFORMAT</td>
<td>227</td>
<td>yes</td>
</tr>
<tr>
<td>RBALRSN_CONVERSION</td>
<td>228</td>
<td>yes</td>
</tr>
<tr>
<td>REBALANCE</td>
<td>179</td>
<td>no</td>
</tr>
<tr>
<td>RECOVERYDDN</td>
<td>273</td>
<td>no</td>
</tr>
<tr>
<td>RECOVERYICDDN</td>
<td>277</td>
<td>no</td>
</tr>
<tr>
<td>REDEFINE</td>
<td>210</td>
<td>yes</td>
</tr>
<tr>
<td>REGISTER</td>
<td>267</td>
<td>no</td>
</tr>
<tr>
<td>REORG INDEX</td>
<td>171</td>
<td>yes</td>
</tr>
<tr>
<td>REORG TABLESPACE</td>
<td>170</td>
<td>no</td>
</tr>
</tbody>
</table>
### Table 32  REORG PLUS command options (part 4 of 5)

<table>
<thead>
<tr>
<th>Command option</th>
<th>See page</th>
<th>Valid for REORG INDEX?</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORTONLY</td>
<td>236</td>
<td>yes</td>
</tr>
<tr>
<td>RETPD</td>
<td>322</td>
<td>no</td>
</tr>
<tr>
<td>RETRY</td>
<td>246</td>
<td>yes</td>
</tr>
<tr>
<td>RETRY_DELAY</td>
<td>246</td>
<td>yes</td>
</tr>
<tr>
<td>REUSE</td>
<td>213</td>
<td>yes</td>
</tr>
<tr>
<td>RIDMAPMEM</td>
<td>336</td>
<td>yes</td>
</tr>
<tr>
<td>SELECT</td>
<td>282</td>
<td>no</td>
</tr>
<tr>
<td>SHORTMEMORY</td>
<td>195</td>
<td>yes</td>
</tr>
<tr>
<td>SHRLEVEL</td>
<td>172</td>
<td>yes</td>
</tr>
<tr>
<td>SIXSNAP</td>
<td>251</td>
<td>no</td>
</tr>
<tr>
<td>SIZEPCT</td>
<td>311</td>
<td>yes</td>
</tr>
<tr>
<td>SMS</td>
<td>300</td>
<td>yes</td>
</tr>
<tr>
<td>SMSUNIT</td>
<td>301</td>
<td>yes</td>
</tr>
<tr>
<td>SORTDATA</td>
<td>190</td>
<td>no</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>187</td>
<td>yes</td>
</tr>
<tr>
<td>SORTKEYS</td>
<td>174</td>
<td>no</td>
</tr>
<tr>
<td>SORTNUM</td>
<td>188</td>
<td>yes</td>
</tr>
<tr>
<td>SPACE</td>
<td>312</td>
<td>yes</td>
</tr>
<tr>
<td>SPILLDSNPAT</td>
<td>331</td>
<td>yes</td>
</tr>
<tr>
<td>SPILLSTORCLAS</td>
<td>331</td>
<td>yes</td>
</tr>
<tr>
<td>SPILLUNIT</td>
<td>330</td>
<td>yes</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>325</td>
<td>yes</td>
</tr>
<tr>
<td>SYNC</td>
<td>208</td>
<td>yes</td>
</tr>
<tr>
<td>TERMEXIT</td>
<td>242</td>
<td>yes</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>312</td>
<td>yes</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>238</td>
<td>yes</td>
</tr>
<tr>
<td>TOTALPAGEPCT</td>
<td>194</td>
<td>yes</td>
</tr>
<tr>
<td>TSSAMPLEPCT</td>
<td>263</td>
<td>no</td>
</tr>
<tr>
<td>UNIT</td>
<td>302</td>
<td>yes</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>302</td>
<td>yes</td>
</tr>
<tr>
<td>UNLDDN</td>
<td>182</td>
<td>no</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>204</td>
<td>yes</td>
</tr>
<tr>
<td>UPDATE</td>
<td>286</td>
<td>no</td>
</tr>
<tr>
<td>UPDATEDB2STATS</td>
<td>261</td>
<td>yes</td>
</tr>
<tr>
<td>VOLCNT</td>
<td>304</td>
<td>yes</td>
</tr>
<tr>
<td>WHERE condition</td>
<td>288</td>
<td>no</td>
</tr>
<tr>
<td>WORKDDN</td>
<td>183</td>
<td>yes</td>
</tr>
<tr>
<td>WТОMSG</td>
<td>254</td>
<td>yes</td>
</tr>
</tbody>
</table>
Command syntax diagrams for REORG PLUS

Figure 8 shows the possible options allowed on the REORG command. For information about how to read the diagrams, see “Syntax diagrams” on page 23.

Table 32  REORG PLUS command options (part 5 of 5)

<table>
<thead>
<tr>
<th>Command option</th>
<th>See page</th>
<th>Valid for REORG INDEX?</th>
</tr>
</thead>
<tbody>
<tr>
<td>XBIMID</td>
<td>253</td>
<td>yes</td>
</tr>
<tr>
<td>ZIIP</td>
<td>252</td>
<td>yes</td>
</tr>
</tbody>
</table>

---

Figure 8  REORG PLUS command syntax diagram (part 1 of 6)
Figure 8  REORG PLUS command syntax diagram (part 2 of 6)

- **SORTDEV** `deviceType`\(^a\)
- **SORTNUM** \(^a\) \(^32\) integer
- **MAXSORTS** \(^a\) \(^0\) integer
- **SORTDATA** \(^b\)
- **ORDER** YES \(^b\) | NO
- **ASSOCIATE** \(^b\) \(^NO\) | \(^BYTABLE\) | \(^BYCLUSTERKEY\)
- **SHORTMEMORY** \(^a\) \(^0\) | \(^CONTINUE\) | \(^FAIL\)
- **AVAILPAGEPCT** \(^a\) \(^b\) \(^100\) integer
- **TOTALPAGEPCT** \(^a\) \(^b\) \(^0\) integer
- **CONTINUE**
- **FAIL**
- **MINSORTMEMORY** integer\(^a\)
- **MAXSORTMEMORY** integer\(^a\)
- **ANALYZE**
- **PAUSE** ONLY
- **SAMPLE** SCAN
- **HURBA**
- **BMCSTATS**
- **PAGEPCTFREE** STANDARD
- **BMC** \(^integer\)
- **REDEFINE** \(^a\) \(^YES\) | \(^NO\)
- **REUSE**
- **RELOAD** CONTINUE | PAUSE
- **RENAME**
- **DELETEFILES** \(^a\) \(^YES\) | \(^NO\)
- **ORIGINALDISP** \(^a\) \(^DELETE\) | \(^RENAME\)

\(^a\) Option has a corresponding installation option.
\(^b\) This option is applicable for a table space reorganization only.
Figure 8  REORG PLUS command syntax diagram (part 3 of 6)

- **FASTSWITCH**
  - YES
  - NO

- **FSFALLBACK**
  - NO
  - YES

- **FSTHRESHOLD**
  - 0
  - integer

- **ON MESSAGE**
  - 50253
  - 50254

- **STOP UTILITY**
- **CONTINUE UTILITY**
- **RETCODE**
  - integer

- **ON FAILURE**
  - **UTILINIT**
  - **TERMINATE UTILITY**
  - **RETCODE**
  - integer

- **PREFORMAT**
  - YES
  - NO

- **RBALRSN_CONVERSION**
  - NONE
  - BASIC
  - EXTENDED

- **ARCROWS**
  - integer

- **CONDEXC**
  - YES
  - NO
  - BMC

- **OFFPOSLIMIT**
  - integer

- **INDREFLIMIT**
  - integer

- **LEAFDISTLIMIT**
  - integer

- **REPORTONLY**
- **DSNUTILB**
  - YES
  - NO

- **TIMEOUT**
  - ABEND
  - rc

---

*a Option has a corresponding installation option.
b This option is applicable for a table space reorganization only.
Figure 8  REORG PLUS command syntax diagram (part 4 of 6)

- Option has a corresponding installation option.
- This option is applicable for a table space reorganization only.
- This option is applicable for an index space reorganization only.
Statistics options

- **BMCSTATS**: NO, YES
- **UPDATER2STATS**: NO, YES

Copy options

- **COPY**: NO, YES
- **INLINE**: YES, NO
- **REGISTER**: ALL, NONE
- **ICTYPE**: AUTO, UPDATE, INCREMENTAL
- **ddname**

- **COPYLVL**: PART, FULL
- **COPYDDN**: BCPY, BCPZ, (ddname1, ddname2)

- **RECOVERYDDN**: BRCY, BRCZ, (ddname1, ddname2)

- **ICDDN**: BICY, BICZ, (ddname1, ddname2)

- **RECOVERYICDDN**: BIRY, BIRZ, (ddname1, ddname2)

---

*a Option has a corresponding installation option.
*b This option is applicable for a table space reorganization only.
Figure 8  REORG PLUS command syntax diagram (part 6 of 6)

Selective unload and update options  page 279

![Syntax diagram](image)

Dynamic data set allocation options  page 293

![Syntax diagram](image)

Figure 9 shows syntax diagram details. The number on each detail diagram corresponds to a preceding section of the REORG PLUS command syntax or to another diagram in this section.

Figure 9  Detail syntax diagrams (part 1 of 5)

![Syntax diagram](image)
This option is applicable for a table space reorganization only.

3 update block detailb page 286

```
UPDATE tableName
   cref1, cref2
   SET columnName = const,
      WHERE condition block
```

4 condition block detaib page 288

```
AND OR
   NOT (condition block)
```

5 predicate block detailb page 290

```
columnName < constant
   <= NULL
   <> CURRENT DATE
   = CURRENT TIMESTAMP
   !=
   >= WITH TIME ZONE
   >
   IN (constant)
```

b This option is applicable for a table space reorganization only.
Figure 9  Detail syntax diagrams (part 3 of 5)

DD type block detail

Option has a corresponding installation option.

See SMS class block detail on page 169.
Figure 9  Detail syntax diagrams (part 4 of 5)

DD type block detail continued

- Volcnt\(^{a}\) — (25 integer\(^{1}\) AUTO, 25 integer\(^{2}\) AUTO)

- Avgvolsp\(^{a}\) — (- (3000 integer\(^{1}\) TRK CYL), - (3000 integer\(^{2}\) TRK CYL))

- Maxextsz\(^{a}\) — (- (- 0 integer\(^{1}\) TRK CYL), - (- 0 integer\(^{2}\) TRK CYL))

- Dsntype\(^{a}\) — (NONE, LARGE, BASIC, EXTREQ, EXTREQ, EXTREQ, EXTREQ)

- Sizepct\(^{a}\) — (100 primary, 100 secondary)

- Space (primary,secondary)\(^{c}\) — CYL TRK

- Threshld\(^{a}\) — 0 integer

- Dsnpat\(^{a}\) — 'pattern'

- Expdt date\(^{a,b}\)

- Retpd integer\(^{a,b}\)

- Gdglimit\(^{a,b}\) — S integer

\(^a\) Option has a corresponding installation option.
\(^b\) This option is applicable for a table space reorganization only.
\(^c\) This option is valid for a DSNUTILB reorganization only.
Figure 9  Detail syntax diagrams (part 5 of 5)

7  SMS class block detail  page 324

\[\text{DATACLAS}^a \quad \text{MGMTCLAS}^a \quad \text{STORCLAS}^a \quad ( \quad \text{NONE}^a \quad \text{class1} \quad ), \quad \text{NONE}^a \quad \text{class2} \quad )\]

8  change block detail

\[\text{LOGTHRESHLD} \text{ integer}^a \quad \text{MAXRO}^a \text{ integer}^a \quad \text{DEFER}\]

\[\text{DEADLINE}^a \quad \text{NONE}^a \quad \text{timeStamp}^a \quad \text{LONGLOG}^a \quad \text{CONTINUE}^a \quad \text{TERM}^a \quad \text{DRAIN}^a \quad \text{DELAY} \text{ integer}^a\]

\[\text{SPILLUNIT}^a \quad \text{unitName}^a \quad \text{NONE} \quad \text{SPILLSTORCLAS}^a \quad \text{class}^a \quad \text{NONE}\]

\[\text{SPILLDSNPAT} \text{' pattern'}^a \quad \text{RIDMAPMEM} \text{ integer}^a \quad \text{LOGMEM} \text{ integer}^a\]

\[\text{LOGSPILL}^a \quad \text{' -- primary' } \quad \text{LOGFINAL}^a \text{' -- secondary'} \quad \text{NONE}^a \quad \text{timeStamp}\]

\[\text{DRAIN}^a \quad \text{ALL} \quad \text{WRITERS} \quad \text{MAPPINGTABLE} \text{ tableName} \quad \text{MAPPINGDATABASE} \text{ databaseName}\]

^a Option has a corresponding installation option.
Descriptions of REORG PLUS command options

This section describes options in the order in which they appear in the preceding syntax diagrams, which group the options according to these functions:

- basic processing options
- copy options
- statistics options
- selective unload options
- dynamic data set allocation options
- SHRLEVEL CHANGE options

Options that are not applicable to an index-only reorganization are noted as such.

Basic processing options

The basic processing options control most aspects of REORG PLUS execution.

**REORG**

REORG is the keyword for the REORG PLUS utility command.

**TABLESPACE**

*This option applies to a table space reorganization only.*

The TABLESPACE option names the table space to be reorganized. All indexes that are associated with the table space are also reorganized.

**databaseName**

If you do not specify a database name, BMC supplies the value DSNDB04. The database name specifies the database to be reorganized and cannot be DSNDB01 or DSNDB07.
**tableSpaceName**

The table space name specifies the name of the table space to be reorganized. The table space and associated index spaces must be started in the appropriate status before beginning the REORG PLUS job. For status information, see “Serialization and concurrency” on page 82.

**INDEX**

*This option does not apply to a table space reorganization.*

The INDEX option names the index (and thus the corresponding index space) to be reorganized. The index name is the qualified name of the index. If you do not specify the authorization ID qualifier, REORG PLUS uses the DB2® primary authorization ID of the user who is running the REORG PLUS job.

If you specify the INDEX option, REORG PLUS reorganizes only the index. REORG PLUS does not reorganize data in the associated table space.

**CLONE**

The CLONE option indicates that you want to reorganize only the clone table in the specified table space, or the specified index on a clone table. If you specify this option but no clone table exists in the specified table space, or the specified index is not on a clone table, REORG PLUS terminates.

**Restrictions**

The following restrictions apply to reorganizing clone objects:

- You cannot use FASTSWITCH processing when reorganizing clone objects. REORG PLUS changes FASTSWITCH YES to FASTSWITCH NO in this case.

- You cannot convert RBA or LRSN format when reorganizing a clone object or a base table space for which a clone relationship exists. REORG PLUS ignores the RBALRSN_CONVERSION option in this case.

- You cannot rebalance partitions on a table space for which a clone relationship exists. REORG PLUS takes the following actions, based on the type of table space involved:
  - For range-partitioned table spaces, REORG PLUS terminates.
  - For partition-by-growth table spaces, REORG PLUS ignores the rebalance request and completes normally.

- If you also specify BMCSTATS YES or UPDATEDB2STATS YES, REORG PLUS collects statistics only on the base objects, not on the clone table.
SHRLEVEL

The SHRLEVEL option specifies the level of access that DB2 has to the target spaces during REORG PLUS processing. For important information about using this option, see “SHRLEVEL considerations” on page 98.

NONE

SHRLEVEL NONE is the default. It specifies that the objects that you are reorganizing are stopped and unavailable during the entire reorganization.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

Restriction
SHRLEVEL NONE is not valid when reorganizing catalog objects. REORG PLUS issues a message and terminates.

REFERENCE UNLOADONLY

This option does not apply to a single-phase reorganization.

This option specifies that the objects that you are reorganizing are available in read-only status during the UNLOAD phase. They are then stopped at the beginning of the RELOAD phase and remain unavailable throughout the remainder of the reorganization.

Restrictions
REORG PLUS issues a message and terminates when either of the following conditions exists:

- REORG PLUS is invoking DSNUTILB.
- You are reorganizing catalog objects.
REFERENCE

SHRLEVEL REFERENCE specifies that the objects that you are reorganizing are available in read-only status during unload and reload processing. REORG PLUS writes the reorganized data to staging data sets and leaves the original VSAM data sets intact throughout the reorganization process. This nondestructive type of reorganization allows you to easily restart from a failure or make the objects available without having to recover.

When invoking DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility for processing.

Considerations
The following considerations apply to SHRLEVEL REFERENCE:

- SHRLEVEL REFERENCE is not valid when reorganizing catalog objects. REORG PLUS issues message BMC50119E and terminates.
- When reorganizing a table space that contains a LOB column, REORG PLUS invokes DSNUTILB when the following conditions exist:
  - You do not specify the AUX option on the REORG command.
  - Either of the following conditions exists:
    - You specify REBALANCE.
    - Any table space partition is in REORP status.

In this case, the DB2 REORG utility defaults to AUX YES.

For more information, see “SHRLEVEL REFERENCE” on page 100.

CHANGE

SHRLEVEL CHANGE allows the objects that you are reorganizing to remain in RW status during unload and reload processing. REORG PLUS writes the reorganized data to staging data sets and leaves the original VSAM data sets intact throughout the reorganization process. This nondestructive type of reorganization makes the DB2 data sets available in RW status during most of the reorganization.

To use the SHRLEVEL CHANGE option, you must have installed the BMC Software EXTENDED BUFFER MANAGER (XBM) product or SNAPSHOT UPGRADE FEATURE (SUF) component of XBM.

Considerations
The following considerations apply to SHRLEVEL CHANGE:

- When invoking DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility for processing.
When reorganizing a table space that contains a LOB column, REORG PLUS invokes DSNUTILB when the following conditions exist:

- You do not specify the AUX option on the REORG command.
- Any table space partition is in REORP status.

In this case, the DB2 REORG utility defaults to AUX YES.

For additional information, see the following sections:

- For information about the syntax for the additional options to specify with SHRLEVEL CHANGE, see “SHRLEVEL CHANGE options” on page 326.

- For information about how the function works, see Chapter 7, “Online reorganization.”

**SORTKEYS**

SORTKEYS is available only for compatibility with the IBM DB2 REORG utility command, and REORG PLUS treats this option as a comment.
PART

The PART option specifies the partition or partitions of the partitioned table space or index to be reorganized. If you do not specify PART, REORG PLUS reorganizes all partitions of the table space or index.

For more information about partial reorganizations, including restrictions, see “Partial reorganization” on page 121.

Specifying partition numbers
Note the following information about specifying partition numbers with the PART option:

- You can specify partitions by number using integer values from 1 through 4096.
- Individual partitions in a list can be in any order (but partitions within a range must be in ascending order).
- You can specify a mixture of individual partitions and ranges of partitions.
- If you specify a partition number more than once, REORG PLUS ignores any occurrence after the first.
- If you want to have REORG PLUS create a single image copy for a subset of partitions, specify only one set of contiguous partitions, either individually or as a range.
- If you are rebalancing partitions during the reorganization, you must specify a contiguous range of logical partitions. For more information, see “Partition rebalancing” on page 127.
- REORG PLUS invokes DSNUUTILB when you specify noncontiguous partitions for a table space whose partition limit keys were altered before the reorganization.
The following examples illustrate valid PART specifications using partition numbers:

<table>
<thead>
<tr>
<th>Specification</th>
<th>REORG PLUS reorganizes these partitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PART 2:4</td>
<td>2, 3, and 4</td>
</tr>
<tr>
<td>PART 3:5,7,9:11,13,2</td>
<td>3, 4, 5, 7, 9, 10, 11, 13, and 2</td>
</tr>
<tr>
<td>PART 100:104,4096</td>
<td>100, 101, 102, 103, 104, and 4096</td>
</tr>
</tbody>
</table>

Reorganizing a large number of partitions might result in constrained resources. If you encounter this problem, consider specifying smaller groups of partitions in multiple steps. Whether the number of partitions that you have is considered to be a large number depends on your environment. For more information, see “Large number of partitions” on page 108.

**DSNUTILB reorganization jobs**

For a DSNUTILB reorganization, you must specify your partitions as follows:

- For a table space reorganization, your specification must be one that the version of the IBM DB2 REORG utility that you are running supports.
- For an index reorganization, you can specify only PART partitionNumber.

If you specify other values, REORG PLUS issues message BMC50178E and terminates.

**LAST**

When you are reorganizing a partition-by-growth table space, you can specify the last partition either by partition number or by specifying the LAST option. Specifying LAST is useful when you are reorganizing partition-by-growth table spaces because it frees you from having to track which partition number is the last partition before you begin the reorganization. LAST is valid only for partition-by-growth table spaces. If you specify LAST for any other type of table space, REORG PLUS terminates and issues message BMC50173E. For more information about reorganizing partition-by-growth table spaces, see page 125.

The following examples illustrate valid PART specifications using the LAST option:

<table>
<thead>
<tr>
<th>Specification</th>
<th>REORG PLUS reorganizes these partitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PART 100-LAST</td>
<td>100 through the last partition</td>
</tr>
<tr>
<td>PART LAST</td>
<td>last partition</td>
</tr>
<tr>
<td>PART 100,LAST</td>
<td>100 and last</td>
</tr>
</tbody>
</table>

**Restriction**

For a DSNUTILB reorganization, the LAST option is not valid.
MAXNEWPARTS

This option applies to partition-by-growth table spaces only.

The MAXNEWPARTS option specifies the following limits:

- the maximum number of partitions that REORG PLUS can add during a reorganization

  Partition extension is also constrained by the DB2 MAXPARTITIONS value with which the table space was created.

- the maximum number of partitions added by DB2 that REORG PLUS can support (only applies to a SHRLEVEL CHANGE reorganization of a table space or an index defined on the table)

  If DB2 adds more partitions than the value specified for MAXNEWPARTS, REORG PLUS terminates and issues message BMC53025E.

You can specify any integer from 0 through 4096 for MAXNEWPARTS. However, BMC recommends that you specify only the number of new partitions that you expect the table space to require. A smaller MAXNEWPARTS value reduces memory requirements and the number of additional partition-level data sets that you must allocate. For special requirements for copy data sets and SYSREC data sets, see page 358 and page 380.

Restrictions
REORG PLUS ignores this option when either of the following conditions exists:

- REORG PLUS invokes DSNUTILB.
- You are reorganizing a catalog object.

Specifying the default
You can specify the default for the MAXNEWPARTS command option in your installation options module by using the MAXNEWPARTS installation option (page 709). REORG PLUS was shipped with a default value of 2 for this option. The command option overrides the default that is in the installation options module.
LONGNAMETRUNC

LONGNAMETRUNC tells REORG PLUS where to truncate names that are longer than the area that is available in REORG PLUS report-style messages.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the LONGNAMETRUNC command option in your installation options module by using the LONGNAMETRUNC installation option. REORG PLUS was shipped with a default value of MIDDLE for this option. The command option overrides the default value that is in the installation options module.

MIDDLE

MIDDLE tells REORG PLUS to truncate outward from the middle of the name. You can shorten this value to M. REORG PLUS indicates the truncated section with the symbols << as shown in the following example:

BEGINNING

BEGINNING, or B, tells REORG PLUS to truncate from the beginning of the name. REORG PLUS indicates the truncated section with the symbols << as shown in the following example:

END

END, or E, tells REORG PLUS to truncate from the end of the name. REORG PLUS indicates the truncated section with the symbols >> as shown in the following example:
REBALANCE

This option does not apply to index-only or nonpartitioned table space reorganizations.

For index-controlled and table-controlled partitioned table spaces, you can use the REBALANCE command option to rebalance partitions. In a single execution of REORG PLUS, you can rebalance up to 255 ranges of logically contiguous partitions.

REBALANCE tells REORG PLUS to define new partition boundaries and evenly redistribute rows across the reorganized partitions. When you specify a list of partitions, REORG PLUS rebalances all logically contiguous partitions. REORG PLUS computes limit key breaks based on rebalancing at the record level, which enables REORG PLUS to redistribute the number of rows across partitions.

You can also use REBALANCE with all SHRLEVEL options, including SHRLEVEL CHANGE. However, you must have sufficient unique limit key values to allow REORG PLUS to establish new balanced limit keys. If the number of duplicate keys results in an empty partition, the reorganization fails because REORG PLUS cannot determine the correct limit key.

--- WARNING ---
When you use SHRLEVEL NONE, insufficient unique limit key values might cause an error that can only be repaired with a recovery.

Restrictions
The following restrictions apply when you use the REBALANCE option. For more information, including requirements and considerations, see “Partition rebalancing” on page 127.

- When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing. DB2 REORG utility restrictions apply, as described in the IBM documentation.
You cannot rebalance the partitions of the following table space types:

- partition-by-growth table spaces
- table spaces for clone objects or base objects that participate (or have participated) in a clone relationship
- XML table spaces
- LOB table spaces
- table spaces that contain a LOB column, when reorganizing natively
- table spaces that contain an XML column

**NOTE**

In many cases, you can rebalance table spaces that contain an XML column by including a DDLIN data set in your JCL. For restrictions, see “DDLIN data set” on page 360.

When you restart a reorganization, you cannot change the REBALANCE specification.

REORG PLUS invokes DSNUTILB when either of the following conditions exist:

- You are reorganizing a table space that is defined with table-controlled partitioning and you are running on DB2 Version 11.
- You are reorganizing a table space that contains a LOB column and the following conditions exist:
  - You specify SHRLEVEL REFERENCE.
  - You do not specify the AUX option on the REORG command.

In this case, the DB2 REORG utility defaults to AUX YES.

You cannot specify the following combination of options when all partitioning columns are ascending and any column except the last column is nullable:

- SHRLEVEL NONE
- UNLOAD RELOAD (single-phase processing)
- REBALANCE
Specifying partitions
Use care when specifying partitions. REORG PLUS rebalances partitions by logical partition number, rather than physical partition number. If contiguous physical partitions are not contiguous by logical partition number, REORG PLUS cannot rebalance them. For more information, see “Specifying partitions” on page 133.

ON n COLUMNS

ON n COLUMNS tells REORG PLUS how many columns to use for the limit key. Specify any value from 1 through the number of columns in the limit key definition.

When invoking DSNUTILB, REORG PLUS changes REBALANCE ON n COLUMNS to REBALANCE.
**UNLDDN**

*This option applies to a table space reorganization only.*

UNLDDN allows you to override the default ddname (SYSREC) or ddname prefix for the output data set that contains the unloaded rows to be reorganized. For information about specifying and using the SYSREC data set, see “SYSREC data sets” on page 379.

If you use this command option to override the default name in the installation options module, you must also change the ddname in your JCL.

**DSNUTILB reorganization jobs**

When invoking DSNUTILB, REORG PLUS uses this option only to pass a value for the &DDNAME variable for the data set name pattern. REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

**Multiple data sets**

If you use multiple unload data sets, the ddname that you specify in this option becomes a prefix. The ddname that you specify in the JCL must have the partition number \(nn\) appended to this prefix. The length of \(nn\) can be from one through seven characters, depending on the length of the ddname prefix. The maximum length of the ddname with the prefix must be eight bytes or less. For example, if you have 103 unload data sets, you can specify the ddname here as SYSRC, and specify SYSRC103 in your JCL.

**Dynamic data set allocation**

If dynamic data set allocation is active and you specify more than one ddname prefix for dynamic allocation, the prefix for each ddname must be different enough for REORG PLUS to differentiate one prefix from another. To be different enough, if these prefixes are different only because one prefix has additional trailing bytes, then these trailing bytes must contain at least one nonnumeric byte. For example, the first set of prefixes that follow is sufficiently different, but the second set is not:

- acceptable set:
  - BMCRD
  - BMCRDWK
Basic processing options

- not acceptable set:

<table>
<thead>
<tr>
<th>BMCRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCRD11</td>
</tr>
</tbody>
</table>

When you dynamically allocate SYSREC data sets, REORG PLUS appends the data set number to the ddname prefix that you specify. To dynamically allocate more than 99 SYSREC data sets, use this option to specify a ddname prefix. The prefix plus the highest data set number must not exceed eight characters. For more information, see “Specifying ddname prefixes” on page 93.

**Specifying the default**
You can specify the default for the UNLDDN command option in your installation options module by using the UNLDDN installation option (page 734). REORG PLUS was shipped with a default value of SYSREC for this option. The command option overrides the default that is in the installation options module.

**WORKDDN**

WORKDDN allows you to override the default ddname or ddname prefix of the work data set. SORTOUT is not used by REORG PLUS but is kept for compatibility with the IBM DB2 REORG utility command syntax. For specification guidelines and detailed information about the use of the SYSUT1 work data set, see “SYSUT1 data sets” on page 382.

If you use this command option to override the default name in the installation options module, you must also change the ddname in your JCL.

**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS uses this option only to pass a value for the &DDNAME variable for the data set name pattern. REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

**Multiple data sets**
If you use multiple work data sets, the ddname that you specify in this option becomes a prefix. The ddname that you specify in the JCL must have the partition number \( nn \) appended to this prefix. The length of \( nn \) can be from one through seven characters, depending on the length of the ddname prefix. The maximum length of the ddname with the prefix must be eight bytes or less. If you are using more than nine work data sets (\( nn \) is 10 or greater), the prefix that is specified in this option can have a maximum of six characters.
**Dynamic data set allocation**

If dynamic data set allocation is active and you specify more than one ddname prefix for dynamic allocation, the prefix for each ddname must be different enough for REORG PLUS to differentiate one prefix from another. To be different enough, if these prefixes are different only because one prefix has additional trailing bytes, then these trailing bytes must contain at least one nonnumeric byte. For example, the first set of prefixes that follow is sufficiently different, but the second set is not:

- **acceptable set:**
  - BMCRD
  - BMCRDWK

- **not acceptable set:**
  - BMCRD
  - BMCRD11

When you dynamically allocate SYSUT1 data sets, REORG PLUS appends the data set number to the ddname prefix that you specify. To dynamically allocate more than 99 SYSUT1 data sets, use this option to specify a ddname prefix. The prefix plus the highest data set number must not exceed eight characters. For more information, see “Specifying ddname prefixes” on page 93.

**Specifying the default**

You can specify the default for the WORKDDN command option in your installation options module by using the WORKDDN installation option (page 737). REORG PLUS was shipped with a default value of SYSUT1 for this option. The command option overrides the default that is in the installation options module.
ARCHDDN

This option applies to a table space reorganization only.

ARCHDDN allows you to override the default ddname of the archive data set (SYSARC). REORG PLUS uses the archive data set to contain rows that it discards during a reorganization. For more information about the SYSARC data set, see “SYSARC data set” on page 370.

If you use this command option to override the default name in the installation options module, you must also change the ddname in your JCL.

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS uses this option only to pass a value for the &DDNAME variable for the data set name pattern. REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

Specifying the default
You can specify the default for the ARCHDDN command option in your installation options module by using the ARCHDDN installation option (page 675). REORG PLUS was shipped with a default value of SYSARC for this option. The command option overrides the default that is in the installation options module.

ARCHFORMAT

This option applies to a table space reorganization only.

ARCHFORMAT allows you to specify the format of your archive data set (SYSARC). For more information about the SYSARC data set, see “SYSARC data set” on page 370.

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the ARCHFORMAT command option in your installation options module by using the ARCHFORMAT installation option (page 675). REORG PLUS was shipped with a default value of BMC for this option. The command option overrides the default that is in the installation options module.
Basic processing options

**BMC**

ARCHFORMAT BMC tells REORG PLUS to produce the archive data set in BMC internal format. You can reload the rows from the archive data set in this format by using the FORMAT BMC option of LOADPLUS.

An archive data set produced with ARCHFORMAT BMC uses less space than one produced with ARCHFORMAT DB2.

**DB2**

ARCHFORMAT DB2 tells REORG PLUS to produce the archive data set in the same format that the IBM DB2 REORG utility produces when you specify UNLOAD ONLY. You can reload the rows from the archive data set in this format by using the FORMAT UNLOAD option of either LOADPLUS or the IBM DB2 LOAD utility.

---

**NOTE**

This format is not the same format that the DB2 REORG utility uses for discard data sets.
Basic processing options

**DDLDDN**

*This option applies to a table space reorganization only.*

DDLDDN allows you to override the default ddname of the DDLIN data set. For information about using a DDLIN data set in your REORG PLUS jobs, see page 360.

If you use this command option to override the default name in the installation options module, you must also change the ddname in your JCL.

**Restriction**

When invoking DSNUTILB, REORG PLUS ignores this option.

**Specifying the default**

You can specify the default for the DDLDDN command option in your installation options module by using the DDLDDN installation option (page 683). REORG PLUS was shipped with a default value of DDLIN for this option. The command option overrides the default that is in the installation options module.

**SORTDEVT**

The SORTDEVT option specifies the device type for the sort work files that are allocated dynamically.

For non-DSNUTILB jobs, if the first parameter of the SORTDEVT installation option is null, this command option overrides the first parameter of the BMCSORT DYNALOC installation option. If the value of the third parameter in the BMCSORT DYNALOC installation option is OFF, specifying a value for the SORTDEVT option turns BMCSORT dynamic allocation on.

**DSNUTILB reorganizations**

If you are reorganizing a LOB table space, REORG PLUS ignores this option for a DSNUTILB reorganization. In all other cases, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

**Specifying the default**

You can specify the default for the SORTDEVT command option in your installation options module by using the SORTDEVT installation option (page 724). REORG PLUS was shipped with a default value of (.SYSALLDA) for this option (where the first parameter affects non-DSNUTILB jobs and the second parameter affects DSNUTILB jobs). The command option overrides the default that is in the installation options module for both parameters.
SORTNUM

The SORTNUM option affects the allocation of sort work files in the following cases. You can specify any integer value from 0 through 255.

**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility as the number of sort work files to allocate dynamically. For this type of reorganization, the value must be 2 or greater.

**All other reorganization jobs**
This value is in effect when BMCSORT is allocating your sort work files dynamically. Table 33 describes the action that BMCSORT takes for each value that you can specify for the SORTNUM option. The table also provides any additional considerations for these values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Additional considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>BMCSORT honors the value of the third parameter of the BMCSORT DYNALOC installation option. (This parameter tells BMCSORT whether to dynamically allocate sort work files.)</td>
<td>For more information about how this parameter affects dynamic allocation, see “Dynamically allocating SORTWK data sets” on page 367. For more information about the parameter itself, see “DYNALOC installation option” on page 761.</td>
</tr>
<tr>
<td>1–32</td>
<td>BMCSORT dynamically allocates the number of sort work files that it needs for each sort task, up to 32 minus any preallocated sort work files. This number is per sort task.</td>
<td>Preallocated sort work files include sort work files that are allocated in your JCL and any sort work files that REORG PLUS dynamically allocates.</td>
</tr>
<tr>
<td>33–255</td>
<td>BMCSORT dynamically allocates the number of sort work files that it needs for each sort task, up to the number that you specify minus any preallocated sort work files. This number is per sort task.</td>
<td></td>
</tr>
</tbody>
</table>

If the value of the third parameter in the BMCSORT DYNALOC installation option is OFF, specifying a value greater than 0 for the SORTNUM option turns BMCSORT dynamic allocation on and BMCSORT allocates sort work files as needed. For information about when BMCSORT allocates your sort work files dynamically, see the “SORTWK data sets” on page 366.
Specifying the default
You can specify the default for the SORTNUM command option in your installation options module by using the SORTNUM installation option (page 725). REORG PLUS was shipped with a default value of 32 for this option. The command option overrides the default that is in the installation options module.

MAXSORTS

The MAXSORTS option allows you to specify the maximum number of sort tasks that REORG PLUS can run concurrently.

Table 34 describes the effects that MAXSORTS and its relationship with the SMAX installation option have on REORG PLUS processing:

Table 34  Effects of MAXSORTS and SMAX options on REORG PLUS processing

<table>
<thead>
<tr>
<th>If MAXSORTS value is</th>
<th>And SMAX value is</th>
<th>Then REORG PLUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>any</td>
<td>uses the SMAX value</td>
</tr>
<tr>
<td>greater than 0</td>
<td>any</td>
<td>uses MAXSORTS and starts only one task per CPU $^a$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>multitasks based on the values of the BILDMAX, RORGMAX, and UNLDMAX options</td>
</tr>
</tbody>
</table>

$^a$ If you want to improve performance by starting more than one task per CPU, specify 0 for MAXSORTS and SMAX and use the multitasking options.

For more information about the multitasking options, see the following references:

- Table 90 on page 586
- “Sort processing options” on page 580
- SMAX installation option on page 723
- “Multitasking processes that invoke BMCSORT” on page 587

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.
SORTDATA

This option applies to a table space reorganization only.

SORTDATA is available only for compatibility with the IBM DB2 REORG utility command, and REORG PLUS treats this option as a comment. REORG PLUS never uses the clustering index to unload the data.

ORDER

The ORDER option tells REORG PLUS how and whether to order the table rows. For performance information about this option, see “Memory requirements of the ORDER command option” on page 597.

YES

When ORDER YES (the default) is in effect, REORG PLUS orders the rows in the tables of the table space by their clustering key or, if there is no clustering key, by partitioning key. REORG PLUS orders the data in segmented table spaces, even if no clustering index exists for the table space.

NOTE
To simplify terminology, this book refers to an index that is used to sort data as a data-sorting index. For traditional table spaces, this is a clustering index. For table-controlled partitioned table spaces, this is either a clustering index or partitioning index.

Restrictions
REORG PLUS ignores ORDER YES or changes it to ORDER NO when any of the following conditions exists:

- You are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect.
- All of the following conditions exist:
  - You are reorganizing a table space that has only one table.
  - The table space is not partitioned or is partition-by-growth.
  - No clustering index exists.
  - You are performing a two-phase reorganization (UNLOAD CONTINUE).
ASSOCIATE

This option does not apply to an index-only reorganization.

Use the ASSOCIATE option to specify how you want REORG PLUS to order rows for multi-table table spaces.

BYTABLE. BYTABLE, which is the default, tells REORG PLUS to sort the rows by table as well as by each table’s clustering key. If no clustering key exists, REORG PLUS uses X’00’ s. REORG PLUS always sorts segmented table spaces by table.

Restrictions
Note the following restrictions for ASSOCIATE BYTABLE:

- For a two-phase reorganization, this option is meaningless for a single-table table space with no clustering index. In this case, REORG PLUS functions as if you specified ORDER NO.

- When invoking DSNUTILB, REORG PLUS ignores this option.

BYCLUSTERKEY. BYCLUSTERKEY sorts the rows by each table’s clustering key only. This option is useful for multi-table, simple table spaces to group rows of different tables together by a common clustering key.

When invoking DSNUTILB, REORG PLUS passes this option as SORTDATA YES to the IBM DB2 REORG utility for processing.

Restriction
REORG PLUS fails if you specify BYCLUSTERKEY for a multi-table, segmented table space.

NO

For table space reorganizations, if you specify ORDER NO, REORG PLUS performs no ordering, and the data rows retain the order of the table before reorganization. REORG PLUS processes clustering indexes together with nonclustering indexes.

For index reorganizations, ORDER NO performs no ordering of the key/RID pairs.

When invoking DSNUTILB, REORG PLUS passes this option as SORTDATA NO to the IBM DB2 REORG utility for processing.
Restrictions
ORDER NO has the following restrictions:

- REORG PLUS changes ORDER NO to ORDER YES when you are rebalancing partitions.
- REORG PLUS terminates when you specify ORDER NO under any of the following conditions:
  - You are reorganizing a multi-table, segmented table space.
  - For a single-phase reorganization, you are performing a SHRLEVEL NONE reorganization.
  - For an index-only reorganization, you are performing a SHRLEVEL CHANGE reorganization.
  - All of the following conditions apply:
    - the table uses table-controlled partitioning
    - the table has no clustering index
    - you are using a DDLIN data set to alter limit keys or using the REBALANCE command option to rebalance partitions
REORG PLUS uses the AVAILPAGEPCT option to control virtual storage above the 16-MB line that REORG PLUS allocates to BMCSORT for concurrent sort processing. AVAILPAGEPCT specifies the maximum percentage of available 4-KB pages, as obtained from the system, that REORG PLUS can allocate. You can specify any integer from 0 through 100.

A value of 0 tells REORG PLUS to ignore the number of available pages when allocating sort memory. A value of 1 through 100 tells REORG PLUS to use up to the specified percentage of available pages when allocating sort memory. For example, AVAILPAGEPCT 50 tells REORG PLUS to use no more than 50 percent of the available pages.

Additional considerations
The following additional information applies to the AVAILPAGEPCT option:

- Because available pages are rarely subject to system paging, changing this value will have a minimal effect, if any, on system performance.

- When you specify values greater than 0 for both AVAILPAGEPCT and TOTALPAGEPCT, REORG PLUS uses the lesser of the two calculated results as the maximum amount of memory for sort processing.

- If REORG PLUS cannot perform an optimal sort due to an insufficient number of available or total pages that it is enabled to allocate, the SHORTMEMORY option (page 721 and page 195) controls the action that REORG PLUS takes.

- When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the AVAILPAGEPCT command option in your installation options module by using the AVAILPAGEPCT installation option (page 676). REORG PLUS was shipped with a default value of 100 for this option. The command option overrides the default that is in the installation options module.

NOTE
REORG PLUS defines available pages as pages that have not been used. Total pages (which you can control with the TOTALPAGEPCT option), are pages that are underutilized and are available for use.
TOTALPAGEPCT

REORG PLUS uses the TOTALPAGEPCT option to control virtual storage allocated to BMCSORT for concurrent sort processing. TOTALPAGEPCT specifies the maximum percentage of total 4-KB pages, as obtained from the system, that REORG PLUS can allocate. You can specify any integer from 0 through 100.

**NOTE**

REORG PLUS defines total pages as pages that are underutilized and are available for use. Available pages (which you can control with the AVAILPAGEPCT option), are pages that have not been used.

A value of 0 tells REORG PLUS to ignore the number of total pages when allocating sort memory.

A value of 1 through 100 tells REORG PLUS to use up to the specified percentage of total pages when allocating sort memory. For example, TOTALPAGEPCT 50 tells REORG PLUS to use no more than 50 percent of the total pages.

Additional considerations

The following additional information applies to the TOTALPAGEPCT option:

- When you specify values greater than 0 for both TOTALPAGEPCT and AVAILPAGEPCT, REORG PLUS uses the lesser of the two calculated results as the maximum amount of memory for sort processing.

- If REORG PLUS is unable to start any tasks because of restraints on sort memory caused by a low number of total or available pages, the SHORTMEMORY installation (page 195) or command option controls the action that REORG PLUS takes.

- When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the TOTALPAGEPCT command option in your installation options module by using the TOTALPAGEPCT installation option (page 732). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.
The SHORTMEMORY option controls the action that REORG PLUS takes when one of the following memory shortages exists during sort processing:

- The system contains insufficient available pages of memory for REORG PLUS to perform an optimal sort.

- The region contains insufficient memory for REORG PLUS to perform one task with the amount of memory required to perform an optimal sort. However, the region does contain at least 1024 KB of memory or the minimum amount of memory specified by the MINSORTMEMORY option. If the region contains at least 1024 KB of memory and the amount of memory specified by the MINSORTMEMORY option is available, REORG PLUS uses the greater amount of memory.

Table 35 on page 196 shows the relationship between SHORTMEMORY values and the following conditions:

- memory data obtained from the system
- memory in the region
- value specified for the SMCORE installation option (page 724)
- value specified for the MINSORTMEMORY installation or command option (page 712 and page 197)
Basic processing options

Table 35  Action REORG PLUS takes when memory resources are constrained during sort processing

<table>
<thead>
<tr>
<th>Location of memory</th>
<th>Amount of memory is</th>
<th>SHORTMEMORY value</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory in the system</td>
<td>insufficient to run one optimal sort task based on the amount of data to be sorted</td>
<td>CONTINUE: REORG PLUS runs one task with 1024 KB of memory or the amount of memory that you specified with MINSORTMEMORY, whichever is greater. FAIL: REORG PLUS fails.</td>
</tr>
<tr>
<td></td>
<td>insufficient as specified by MINSORTMEMORY</td>
<td></td>
</tr>
<tr>
<td>virtual memory in the region</td>
<td>insufficient to run one optimal sort task based on the amount of data to be sorted</td>
<td>CONTINUE: REORG PLUS runs one task with the available memory. FAIL: REORG PLUS fails.</td>
</tr>
<tr>
<td></td>
<td>but</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sufficient as specified by MINSORTMEMORY or at least 1024 KB, whichever is greater</td>
<td></td>
</tr>
<tr>
<td></td>
<td>insufficient as specified by MINSORTMEMORY or less than 1024 KB, whichever is greater</td>
<td></td>
</tr>
<tr>
<td></td>
<td>insufficient as specified by the first parameter of SMCORE</td>
<td></td>
</tr>
</tbody>
</table>

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the SHORTMEMORY command option in your installation options module by using the SHORTMEMORY installation option (page 721). REORG PLUS was shipped with a default value of CONTINUE for this option. The command option overrides the default that is in the installation options module.

CONTINUE
CONTINUE indicates that, when a memory shortage exists, REORG PLUS should issue message BMC50364I and continue sort processing, as described in Table 35.

FAIL
FAIL indicates that, when a memory shortage exists, REORG PLUS should fail. Be aware when specifying FAIL that sufficient memory might exist to sort during the UNLOAD phase of a two-phase reorganization or the REORG phase of a single-phase reorganization. However, because of other system conditions, insufficient available pages might exist during the index build process, which occurs during the RELOAD process (two-phase reorganization) or following the REORG process (single-phase reorganization).
MINSORTMEMORY

The MINSORTMEMORY option specifies the minimum amount of memory, in kilobytes, that REORG PLUS should allocate to each sort task.

A value of 0 tells REORG PLUS to automatically compute the minimum amount of memory that is needed to optimally perform each sort task. In addition to 0, you can specify any number of kilobytes between 1024 and the value that you specify for the MAXSORTMEMORY installation or command option (page 197 and page 711).

**NOTE**

BMC recommends that you use a value of 0.

For information about how this option interacts with the SMCORE installation option, see “Sort processing options” on page 580.

**Restriction**

When invoking DSNUTILB, REORG PLUS ignores this option.

**Specifying the default**

You can specify the default for the MINSORTMEMORY command option in your installation options module by using the MINSORTMEMORY installation option (page 712). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.

MAXSORTMEMORY

The MAXSORTMEMORY option specifies the maximum amount of memory, in kilobytes, that REORG PLUS can allocate to each sort task.

A value of 0 tells REORG PLUS to automatically compute the maximum amount of memory that is needed to optimally perform each sort task. In addition to 0, you can specify any number of kilobytes between the value that you specify for the MINSORTMEMORY installation (page 712) or command option and 2097152.
For information about how this option interacts with the SMCORE installation option, see “Sort processing options” on page 580.

**Restriction**
When invoking DSNUTILB, REORG PLUS ignores this option.

**Specifying the default**
You can specify the default for the MAXSORTMEMORY command option in your installation options module by using the MAXSORTMEMORY installation option (page 711). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.

**NOTE**
BMC recommends that you use a value of 0.
ANALYZE gathers information about the objects that you are reorganizing and provides estimated sizes for the following data sets:

- unload (SYSREC)
- work (SYSUT1)
- sort (SORTWK)
- full image copy (BCPY, BCPZ, BRCY, and BRCZ)
- incremental image copy (BICY, BICZ, BIRY, and BIRZ)

REORG PLUS provides all estimates in both kilobytes and cylinders by device type. REORG PLUS calculates the data set size based on cardinality (the number of rows) and average row length (table space only). REORG PLUS obtains these values by using one of the following methods:

- sampling the object (SAMPLE)
- scanning the index (SCAN)
- retrieving the data from the DB2 catalog (HURBA)
- retrieving the data from the DASD MANAGER PLUS statistics tables (BMCSTATS)

REORG PLUS does not stop the table space and associated index spaces that participate in the reorganization during the ANALYZE phase.

Table 36 on page 200 shows the combinations of ANALYZE options and the effects that they have on how REORG PLUS determines cardinality and average row length.
Basic processing options

If you do not specify ANALYZE on the command or you specify ANALYZE with no values (the default), REORG PLUS determines whether to use sampling or scanning. REORG PLUS performs a full analysis and continues processing, using information from the analysis to dynamically allocate data sets.

Additional consideration
When reorganizing a LOB table space, REORG PLUS does not sample or scan to estimate copy data set size, but adds HURBAs from all data sets.

### Table 36  Effects of ANALYZE options on cardinality and average row length estimation

<table>
<thead>
<tr>
<th>ANALYZE keywords</th>
<th>Cardinality</th>
<th>Average row length</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ANALYZE not specified) ANALYZE ANALYZE PAUSE ANALYZE ONLY</td>
<td>REORG PLUS decides whether to sample the table space (for a table space reorganization) or the index (for an index reorganization), or to scan the index leaf pages (for either type of reorganization).</td>
<td>REORG PLUS samples the table space.</td>
</tr>
<tr>
<td>ANALYZE SAMPLE ANALYZE PAUSE SAMPLE ANALYZE ONLY SAMPLE</td>
<td>REORG PLUS samples the table space or index space. For an XML table space reorganization, REORG PLUS obtains index cardinality from the DB2 real-time statistics tables.</td>
<td></td>
</tr>
<tr>
<td>ANALYZE SCAN ANALYZE PAUSE SCAN ANALYZE ONLY SCAN</td>
<td>For a table space reorganization, REORG PLUS scans one index for each table that has an index. If a table does not have an index, REORG PLUS uses sampling for that table. For an index reorganization, REORG PLUS scans the index.</td>
<td>REORG PLUS estimates the length based on half the length of any VARCHAR columns in the table and the length of any fixed columns. REORG PLUS does not consider compression in the calculations.</td>
</tr>
<tr>
<td>ANALYZE HURBA</td>
<td>REORG PLUS makes an estimate based on the HURBA of the object that you are reorganizing, average row length, page size, and free space. If you specify SHRLEVEL CHANGE or activate dynamic allocation, REORG PLUS ignores the HURBA keyword.</td>
<td>REORG PLUS estimates the length based on half the length of any VARCHAR columns in the table and the length of any fixed columns. REORG PLUS does not consider compression in the calculations.</td>
</tr>
<tr>
<td>ANALYZE PAUSE HURBA ANALYZE ONLY HURBA</td>
<td>REORG PLUS changes the keywords to ANALYZE PAUSE or ANALYZE ONLY.</td>
<td></td>
</tr>
<tr>
<td>ANALYZE BMCSTATS ANALYZE PAUSE BMCSTATS ANALYZE ONLY BMCSTATS</td>
<td>REORG PLUS uses cardinality from the DASD MANAGER PLUS statistics table.</td>
<td>REORG PLUS uses average row length from the DASD MANAGER PLUS statistics table.</td>
</tr>
</tbody>
</table>

**no value**

If you do not specify ANALYZE on the command or you specify ANALYZE with no values (the default), REORG-plus determines whether to use sampling or scanning. REORG PLUS performs a full analysis and continues processing, using information from the analysis to dynamically allocate data sets.

Additional consideration
When reorganizing a LOB table space, REORG PLUS does not sample or scan to estimate copy data set size, but adds HURBAs from all data sets.
PAUSE

If you specify ANALYZE PAUSE, REORG PLUS ends the processing after the ANALYZE phase is complete and displays a report. You can use the output of the ANALYZE phase to specify the number and allocations of the work and copy data sets yourself, or you can have REORG PLUS use dynamic allocation for the data sets when you restart REORG PLUS at the next phase.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

ONLY

ANALYZE ONLY tells REORG PLUS to generate a report that provides estimates of the total space that REORG PLUS will need for the work and copy data sets, and the recommended number of each type of work data set. You can use the output of the ANALYZE phase to specify the number and allocations of the data sets on the REORG job that actually performs the reorganization. If you specify ANALYZE ONLY, REORG PLUS terminates after the ANALYZE phase and cannot be restarted.

Restriction
This option is not valid when REORG PLUS invokes DSNUTILB. REORG PLUS issues message BMC50178E and terminates.

SAMPLE

SAMPLE tells REORG PLUS to read the minimum number of pages needed to determine a reasonable estimate for cardinality.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Additional considerations
The following considerations apply to ANALYZE SAMPLE:

- When reorganizing a LOB table space, REORG PLUS does not sample or scan to estimate copy data set size, but adds HURBAs from all data sets.

- When dynamically allocating the SYSUT1 data sets for an XML table space reorganization, REORG PLUS sizes the data sets based on the number of keys in the XML and node ID indexes. During ANALYZE SAMPLE, REORG PLUS obtains these key counts from the DB2 real-time statistics tables.

SCAN

SCAN tells REORG PLUS to read every leaf page in one or more indexes to determine the exact cardinality.
Restrictions
The following restrictions apply to ANALYZE SCAN:

- When invoking DSNUTILB, REORG PLUS ignores this option.
- REORG PLUS does not support SCAN for an XML table space reorganization. If you specify ANALYZE SCAN for an XML table space, REORG PLUS changes the value to ANALYZE SAMPLE and continues processing.

Additional consideration
When reorganizing a LOB table space, REORG PLUS does not sample or scan to estimate copy data set size, but adds HURBAs from all data sets.

HURBA

HURBA tells REORG PLUS to use information obtained from the DB2 catalogs and the HURBA to determine a gross estimate for cardinality and the average row length. You can save processing time by specifying HURBA, but you might get less accurate numbers than if you specify another value.

Restrictions
The following restrictions apply to ANALYZE HURBA:

- For the following functions, REORG PLUS changes ANALYZE HURBA to ANALYZE to obtain more accurate numbers:
  - dynamic allocation
  - SHRLEVEL CHANGE
  - ANALYZE PAUSE HURBA or ANALYZE ONLY HURBA
- When invoking DSNUTILB, REORG PLUS ignores this option.
- When you also specify the REBALANCE option, REORG PLUS ignores it.

BMCSTATS

BMCSTATS tells REORG PLUS to use information contained in the DASD MANAGER PLUS statistics tables to determine cardinality and average row length.

If the information in the DASD MANAGER PLUS statistics tables is current, you can save processing time by specifying ANALYZE BMCSTATS. The results will be as accurate as if you had specified ANALYZE SCAN.

If the information in the DASD MANAGER PLUS statistics tables for the object that you are reorganizing is missing or incomplete, REORG PLUS changes BMCSTATS to SAMPLE and continues with the reorganization. To populate the statistics tables, either run the BMCSTATS component of DASD MANAGER PLUS or run a reorganization with BMCSTATS YES.
**NOTE**
The ANALYZE BMCSTATS option is available only if you are using REORG PLUS as a component of the Database Performance for DB2 solution. If you are not, REORG PLUS changes BMCSTATS to SAMPLE and continues with the reorganization.

**Restrictions**
REORG PLUS ignores this option in the following cases:

- when invoking DSNUTILB
- when reorganizing LOB table spaces
UNLOAD

The UNLOAD option instructs REORG PLUS to continue processing after the UNLOAD phase, to suspend execution after the UNLOAD phase, or to combine the UNLOAD and RELOAD phases into a single processing phase.

RELOAD

UNLOAD RELOAD tells REORG PLUS to combine the UNLOAD and RELOAD phases into a single processing phase named REORG. In this single phase, REORG PLUS provides significant CPU and elapsed time savings over a two-phase reorganization. For performance benefits and considerations when using this single phase, see “REORG phase for a single-phase reorganization” on page 609.

Additional considerations

The following considerations apply to UNLOAD RELOAD:

- If you specify UNLOAD RELOAD and want any VCAT-defined data sets to be deleted and redefined as part of the reorganization, the value of the REDEFINE command or installation option must be YES. You must also provide the necessary IDCAMS commands in the SYSIDCIN data set. For more information, see the REDEFINE option on page 210 and “SYSIDCIN data set” on page 374.

- If you specify UNLOAD RELOAD with ORDER NO, see page 190 for more information about ORDER NO and the various types of reorganizations.

- When invoking DSNUTILB, REORG PLUS passes the value of this option to the IBM DB2 REORG utility as NOSYSREC.

CONTINUE

The CONTINUE option specifies that REORG PLUS continue with a two-phase reorganization process after the data has been unloaded.

Restriction

When you are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect, REORG PLUS changes UNLOAD CONTINUE to UNLOAD RELOAD.
PAUSE

This option does not apply to a SHRLEVEL CHANGE reorganization.

UNLOAD PAUSE instructs REORG PLUS to stop the processing after the data has been unloaded. You can then restart the job in the RELOAD phase. The PAUSE option is useful if you need to redefine data sets during reorganization.

Messages that are displayed at the end of the UNLOAD phase provide the estimated amount of space required to rebuild an index or reload a table space. Key compression of nonleaf pages is not considered when making these estimates.

NOTE

If you are performing a partial reorganization (not reorganizing all partitions of a partitioned table space), do not redefine the nonpartitioned index data sets. REORG PLUS does not rebuild these data sets during a partial reorganization, but only updates them.

If you specify UNLOAD PAUSE with SHRLEVEL REFERENCE for a partial reorganization with nonpartitioned indexes, copying the nonpartitioned indexes is deferred until the RELOAD phase following the restart.

Restriction

REORG PLUS terminates when either of the following conditions exists:

- You are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect.
- You specify AUX YES to reorganize the base and LOB table space in a single job step.

NOSYSREC

Specifying NOSYSREC is the same as specifying UNLOAD RELOAD.
This option applies to a table space reorganization only.

The KEEPDICITIONARY option tells REORG PLUS whether to keep the existing compression dictionary. If you specify KEEPDICITIONARY without a value, REORG PLUS assumes KEEPDICITIONARY YES. For more information about using compression, see “Table space compression” on page 145.

Restrictions
The following restrictions apply to the KEEPDICITIONARY option:

- The KEEPDICITIONARY option is valid only if the table space or partition that you are reorganizing has the COMPRESS YES attribute.

- If a table space is compressed and a REORG PLUS job would convert the row format from BRF to RRF, REORG PLUS builds a new dictionary except when both of the following options are in effect:
  
  — The value of the REORG PLUS KEEPDICITIONARY option is YES.
  — The value of the DB2 subsystem parameter HONOR_KEEPDICITIONARY is YES.

Specifying the default
You can specify the default for the KEEPDICITIONARY command option in your installation options module by using the KEEPDICITIONARY installation option (page 703). REORG PLUS was shipped with a default value of NO for this option. The command option overrides the default that is in the installation options module.

YES

If you specify KEEPDICITIONARY YES, REORG PLUS keeps the existing compression dictionary. If a dictionary does not exist, REORG PLUS builds the dictionary and compresses the data.
DSNUTILB reorganizations
REORG PLUS ignores this option for a DSNUTILB reorganization when both of the following conditions exist:

- You are reorganizing a LOB table space.
- You specify SHRLEVEL CHANGE.

In all other cases, REORG PLUS passes the value of this option to the IBM DB2 REORG utility as KEEPDICTIONARY.

Restrictions
REORG PLUS ignores a value of YES and treats the option as if you specified KEEPDICTIONARY NO when either of the following conditions exists:

- You are performing partition rebalancing.
- You are reorganizing a partition-by-growth table space.

NO

If you specify KEEPDICTIONARY NO, REORG PLUS builds a new compression dictionary and compresses the data.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.
**BMCHIST**

This option enables you to choose whether to insert a utility history row into the BMC BMCHIST table when the reorganization successfully completes. This insert occurs in the UTILTERM phase.

You can also use the TERMEXIT option and the TERMEXIT user exit to dynamically control processing of BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS at execution time. For more information, see “TERMEXIT” on page 242 and “Using TERMEXIT to control BMCHIST and statistics updates” on page 849.

**Restriction**
When invoking DSNUTILB, REORG PLUS ignores this option.

**Specifying the default**
You can specify the default for the BMCHIST command option in your installation options module by using the BMCHIST installation option (page 678). REORG PLUS was shipped with no default value for this option. The command option overrides the default that is in the installation options module.

**SYNC**

REORG PLUS writes records to the BMCSYNC table that show the number of 1-KB rows that REORG PLUS processed during the UNLOAD and RELOAD phases for a two-phase reorganization, or during the REORG phase for a single-phase reorganization. You can use this information to determine how far the REORG job has progressed.

By default, REORG PLUS writes records only after it has read from or loaded the last row or key in a table space, index, or partition. If you want records written more often, specify an integer with the SYNC option to identify the number of 1-KB rows that you want REORG PLUS to process between writing to the BMCSYNC table.

**Additional considerations**
Note the following additional information about the SYNC option:

- REORG PLUS records monitoring and restart sync points in the BMCSYNC table as the job progresses. You can issue an SQL statement to query this table to determine how far the reorganization has progressed and the status of objects that you are reorganizing.
The SYNC option does not control the sync points that are used to restart REORG PLUS. The restart sync points are established only after the last row or key is loaded in a table space, index, or partition.

When invoking DSNUTILB, REORG PLUS ignores this option.

**LOG NO**

LOG NO is not used by REORG PLUS but is provided for compatibility with the IBM DB2 REORG utility command syntax.

When invoking DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility for processing.
**NLPCTFREE**

NLPCTFREE specifies the percentage of each nonleaf index page to reserve as free space when REORG PLUS rebuilds the indexes.

*Restriction*
When invoking DSNUTILB, REORG PLUS ignores this option.

**STANDARD**

The STANDARD option (the default) tells REORG PLUS to use the value that you specified in the DB2 PCTFREE option when you created the index. Like DB2, REORG PLUS leaves up to 10 percent of a nonleaf page free. If you specified a value greater than 10, only 10 percent is left free.

**BMC**

The BMC option tells REORG PLUS to honor the DB2 PCTFREE value in the DB2 catalog, even if the value is greater than 10.

**integer**

Specifying an integer identifies the percentage of each nonleaf index page to reserve as free space. You can specify any integer from 0 through 99.

**REDEFINE**

This option controls whether REORG PLUS deletes and redefines the VSAM data sets for the table space or index space as part of the reorganization. REORG PLUS can redefine both user-defined (VCAT-defined) data sets and data sets that are defined in DB2 storage groups (STOGROUP-defined).
Additional considerations
Note the following additional information about the REDEFINE option:

- When the following conditions exist, you must specify REDEFINE YES:
  - Before the reorganization, an ALTER changed the compression attribute of a participating index.
  - The DSVCNI system parameter is set to YES, indicating that the control interval (CI) size is variable.

- If you are reorganizing a large number of partitions, consider specifying REDEFINE NO. This value minimizes the time that REORG PLUS requires to delete and redefine the existing VSAM data sets for the table space or indexes.

- If you do not specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE, REORG PLUS does not redefine nonpartitioned indexes when performing a partial reorganization. For more information, see “Staging data sets” on page 101.

- When you specify a REXX exit name on the DSRSEXIT command or installation option, REORG PLUS calls the exit before performing DELETE/DEFINE processing for each DB2 VSAM component that it is going to define.

In addition to the applications described in this section, you can use the DSRSEXIT user exit to change REDEFINE YES to REDEFINE NO for a specific object. For more information, see “DSRSEXIT” on page 241 and “Using DSRSEXIT to manage VSAM data set redefinition” on page 834.

Specifying the default
You can specify the default for the REDEFINE command option in your installation options module by using the REDEFINE installation option (page 717). REORG PLUS was shipped with a default value of YES for this option. The command option overrides the default that is in the installation options module.

YES

For SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY, REORG PLUS deletes and redefines the VSAM data sets for the space before reloading it. For SHRLEVEL REFERENCE or SHRLEVEL CHANGE, REORG PLUS deletes and defines the staging data sets.

VCAT considerations
For VCAT-defined objects, the following considerations apply:

- When you specify UNLOAD PAUSE, you still have the option of deleting and redefining VCAT-defined data sets.
Basic processing options

When you specify UNLOAD CONTINUE or UNLOAD RELOAD, the only way to delete and redefine VCAT-defined data sets as part of the reorganization is to use REDEFINE YES and provide the SYSIDCIN data set containing the necessary IDCAMS control statements. Otherwise, REORG PLUS treats the job as if you had specified REDEFINE NO.

For specification guidelines and detailed information about the use of the SYSIDCIN data set, see “SYSIDCIN data set” on page 374.

Storage group considerations
For storage-group-defined objects, the following considerations apply:

- When you specify this option for storage-group-defined data sets, do not provide a SYSIDCIN data set.

- For a storage-group-defined table space or index, REORG PLUS uses the value in DSNZPARMs for the SMS DATAACLAS if a value exists, as follows:
  - For a table space, REORG PLUS uses the value in SMSDCFL.
  - For an index, REORG PLUS uses the value in SMSDCIX.

- For a storage-group-defined table space or index, REORG PLUS attempts to reallocate the data set on the volume on which it currently resides if the volume is still defined in the storage group.

The order in which REORG PLUS retrieves subsequent volumes from the storage group for the purposes of allocating VSAM data sets is not predictable. You can use the DSRSEXIT user exit to sort the volumes into a different sequence. For information about this exit, see DSRSEXIT on page 241 and “Using DSRSEXIT to manage VSAM data set redefinition” on page 834.

- The redefined table space data sets will have a CI size that corresponds with the page size of the assigned buffer pool if the value of your DSVCI system parameter is set to YES.

Reorganize a table space

The REDEFINE NO option tells REORG PLUS not to delete and redefine the existing VSAM data sets for the table space or indexes. REORG PLUS instead issues message BMC50391I, reuses the existing data sets, and resets the high-used RBA.

VCAT considerations
For VCAT-defined objects, REORG PLUS

- extends to another data set, if needed, as long as that data set is already defined
- does not define any additional data sets
If you specify REDEFINE NO with SHRLEVEL REFERENCE or SHRLEVEL CHANGE, you must preallocate the staging data sets before the reorganization for VCAT-defined objects.

**Storage group considerations**
For storage-group-defined objects, the following considerations apply:

- For SHRLEVEL REFERENCE or SHRLEVEL CHANGE, REORG PLUS creates any staging data sets that you do not preallocate, but reuses any staging data sets that you do preallocate.

- For multi-data-set objects, REORG PLUS extends to another data set if needed, and creates it if the data set does not exist. When the reorganization completes, REORG PLUS deletes any data set that it did not use.

- The CI size for any additional data sets that REORG PLUS creates is based on the value of your DSVCI DB2 system parameter and the page size that is defined in the table space that you are reorganizing.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility as REUSE.

**REUSE**

This option is used for compatibility with the IBM DB2 REORG utility command syntax. If you specify REUSE and REDEFINE YES, REORG PLUS uses the last keyword that it finds in the command string.

If you specify REUSE, REORG PLUS functions differently depending on the type of reorganization job that you are running:

- When invoking DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility for processing.

- For all other jobs, REORG PLUS functions as if you specified REDEFINE NO.
**IDCDDN**

The IDCDDN option allows you to override the default ddname for the input data set containing the IDCAMS command statements that REORG PLUS uses to redefine VSAM data sets. The default is SYSIDCIN. For specification guidelines and detailed information about the use of the SYSIDCIN data set, see “SYSIDCIN data set” on page 374.

**AMENDED**

*This option applies to a table space reorganization only.*

The AMENDED option specifies that one or more of the tables in the table space were changed by using the BMC Software DATA PACKER for DB2 AMEND function. The option also specifies whether you want the rows of each table to be re-encoded (using the table’s defined EDITPROC) during the reorganization.

**NOTE**

If a table has an index and an EDITPROC, the EDITPROC is normally invoked to decode the row, regardless of the value of this option.

**NO**

Specifying AMENDED NO, the default, prevents REORG PLUS from re-encoding the rows in the table space.

**YES**

AMENDED YES causes REORG PLUS to invoke any table’s EDITPROC to both decode and then re-encode the rows.
DELETEFILES

DELETEFILES allows you to specify whether you want REORG PLUS to delete all physical sequential data sets whose ddnames match the SYSREC, SYSUT1, and SORTWK ddnames or ddname prefixes after a reorganization.

**NOTE**
To restart your job during DELETEFILES processing, use RESTART, not RESTART(PHASE).

If you are running REORG PLUS in a worklist environment, REORG PLUS ignores the value that you specified in the DELFILES installation option and processes the job as if you had specified DELFILES=NO. If you want to delete your data sets, you must specify DELETEFILES YES on the command.

**Specifying the default**
You can specify the default for the DELETEFILES command option in your installation options module by using the DELFILES installation option (page 684). REORG PLUS was shipped with a default value of YES for this option. The command overrides the default that is in the installation options module.

**YES**

DELETEFILES YES tells REORG PLUS to perform DELETEFILES processing, which depends on whether the job completed successfully.

**Successful job completion**
For non-DSNUTILB reorganizations, REORG PLUS deletes all physical sequential data sets whose ddnames match the SYSREC, SYSUT1, and SORTWK ddnames or ddname prefixes.

For DSNUTILB reorganizations, REORG PLUS deletes the following data sets:

- the SYSREC and SYSUT1 data sets that were used during the reorganization
- all SYSREC, SYSUT1, and SORTWK data sets allocated in your JCL
Unsuccessful job completion
For non-DSNUTILB reorganizations, if the job does not complete successfully, REORG PLUS performs DELETEFILES processing as part of TERMINATE processing in any of the following circumstances:

- You specify TIMEOUT TERM on the command or installation option and a drain timeout occurs.
- You specify ON FAILURE phase TERMINATE UTILITY in the command for one of the following phases, and REORG PLUS terminates with an error in that phase:
  - UNLOAD
  - RELOAD
  - LOGAPPLY (includes the LOGFINAL phase)
  - UTILTERM

  If an error occurs in the UTILTERM phase after REORG PLUS has started renaming the data sets, REORG PLUS stops the job instead of terminating it. In this case, REORG PLUS does not delete any image copy data sets or staging VSAM data sets.

- The reorganization is a SHRLEVEL CHANGE reorganization.

**NOTE**
For a SHRLEVEL CHANGE reorganization, the default is ON FAILURE phase TERMINATE UTILITY for most cases before the UTILTERM phase. You must specify ON FAILURE phase STOP UTILITY if you do not want REORG PLUS to perform TERMINATE processing. For more information, see Table 37 on page 225.

During DELETEFILES processing for an unsuccessful job, REORG PLUS deletes the following data sets:

- any unregistered full and incremental copy data sets (local and remote) whose ddname matches a copy data set prefix
- all physical sequential data sets whose ddnames match the SYSREC, SYSUT1, and SORTWK ddnames or ddname prefixes
- staging VSAM data sets if ORIGINALDISP=DELETE

For DSNUTILB reorganizations, REORG PLUS does not perform DELETEFILES processing for an unsuccessful job.

**NO**
DELETEFILES NO tells REORG PLUS not to delete any data sets after either a successful or unsuccessful reorganization.
ORIGINAlDISP

This option applies to SHRLEVEL REFERENCE and SHRLEVEL CHANGE only.

ORIGINAlDISP allows you to specify whether you want REORG PLUS to delete or rename the original data sets after it has renamed the staging data sets and completed the reorganization successfully.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the ORIGINAlDISP command option in your installation options module by using the ORIGDISP installation option (page 714). REORG PLUS was shipped with a default value of DELETE for this option. The command overrides the default that is in the installation options module.

DELETE

If you specify this option, REORG PLUS deletes the original data sets.

RENAME

If you specify this option, REORG PLUS renames the original data sets to the staging data set names. ORIGINAlDISP RENAME enables you to preserve the space initially allocated for the original data sets by renaming them to the staging data set names. They are then ready to be used as the staging data sets in a subsequent reorganization.

REORG PLUS ignores the ORIGINAlDISP RENAME option when FASTSWITCH YES is in effect. In this case, no rename is needed. The data sets keep their original names, and the space is preserved for use in a subsequent reorganization. For information about the naming conventions for staging data sets, see “Staging data sets” on page 101.
FASTSWITCH

This option applies to SHRLEVEL REFERENCE or SHRLEVEL CHANGE.

The FASTSWITCH option determines the action that REORG PLUS takes in the UTILTERM phase regarding the staging data sets. For more information, see “Staging data sets and the FASTSWITCH process” on page 105.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

Additional considerations
Consider the following information about the FASTSWITCH process:

- You can use the FSFALLBACK (page 220) and FSTHRESHOLD (page 221) options to control whether REORG PLUS performs FASTSWITCH or rename processing under particular conditions.

- BMC recommends specifying FASTSWITCH YES when you are reorganizing objects with more than 200 data sets, such as a segmented table space that contains many tables that each have an index.

- The FASTSWITCH process requires updates to the DB2 catalog and directory. In some environments, REORG PLUS might have difficulty acquiring the necessary locks, which results in contention or deadlocks. Minimizing use of the FASTSWITCH process can reduce this contention and allow the process to be more effective when it is needed. Frequently reorganizing the DB2 catalog and directory can also help prevent contention.

- BMC recommends that you do not specify FASTSWITCH YES when your staging data sets are named according to the STAGEDSN=BMC method.

Specifying the default
You can specify the default for the FASTSWITCH command option in your installation options module by using the FASTSWITCH installation option (page 691). REORG PLUS was shipped with a default value of NO for this option. The command overrides the default that is in the installation options module.

YES

YES tells REORG PLUS to bypass the VSAM rename process and directly updates the DB2 catalog to use the staging data set names.
Restrictions
REORG PLUS changes FASTSWITCH YES to FASTSWITCH NO when reorganizing either of the following types of objects:

- clone objects
- catalog objects

NO

NO tells REORG PLUS to rename the staging data sets to the names of the original data sets. This value can help to prevent potential contention issues between the DB2 catalog and directory.
FSFALLBACK

This option applies to SHRLEVEL REFERENCE or SHRLEVEL CHANGE.

The FSFALLBACK option tells REORG PLUS whether to perform rename processing of your staging data sets when FASTSWITCH processing fails.

Specifying the default
You can specify the default for the FSFALLBACK command option in your installation options module by using the FSFALLBACK installation option (page 695). REORG PLUS was shipped with a default value of NO for this option. The command overrides the default that is in the installation options module.

NO

NO tells REORG PLUS to terminate instead of falling back to rename processing if FASTSWITCH processing fails.

YES

YES tells REORG PLUS to fall back to rename processing if FASTSWITCH processing fails.

Restrictions
The following restrictions apply to FSFALLBACK YES:

- For the following types of reorganizations, REORG PLUS falls back to rename processing only after successfully backing out the FASTSWITCH changes:
  - an index reorganization
  - a partial table space reorganization with a participating nonpartitioned index

- REORG PLUS ignores this option when any of the following conditions exists:
  - FASTSWITCH NO is in effect.
  - REORG PLUS is invoking DSNUTILB.
  - REORG PLUS is converting objects from basic row format to reordered row format.
FSTHRESHOLD

This option applies to SHRLEVEL REFERENCE or SHRLEVEL CHANGE.

The FSTHRESHOLD option provides a threshold above which REORG PLUS performs FASTSWITCH processing. If the number of staging data sets used in a single reorganization is greater than or equal to this value, REORG PLUS performs FASTSWITCH processing. Otherwise, REORG PLUS performs rename processing.

You can specify a value from 0 through 32767. A value of 0 tells REORG PLUS to always perform FASTSWITCH processing.

Restriction
REORG PLUS ignores this option when FASTSWITCH NO is in effect.

Specifying the default
You can specify the default for the FSTHRESHOLD command option in your installation options module by using the FSTHRESHOLD installation option (page 695). REORG PLUS was shipped with a default value of NO for this option. The command overrides the default that is in the installation options module.
ON MESSAGE

This option applies to a table space reorganization only.

ON MESSAGE allows you to specify, for two different error situations, whether REORG PLUS should stop or continue processing when it encounters the error and issues its corresponding error message. If you do not specify ON MESSAGE, the utility stops when it encounters the error. You can repeat this option.

The following message numbers are valid for this option:

- BMC50253 taskNumber: DBID dbid and PSID psid DO NOT MATCH THOSE FOUND IN DATASET dataSetName
- BMC50254 taskNumber: UNEXPECTED EOF (PAGE=n) IN DATASET dataSetName

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

STOP UTILITY

STOP UTILITY (the default) tells REORG PLUS to stop the utility when the specified message is issued. The utility ID is left in a restartable state.

CONTINUE UTILITY

CONTINUE UTILITY tells REORG PLUS to continue processing after issuing the corresponding message. Under some fatal conditions, REORG PLUS will not be able to continue. However, if you specify CONTINUE, and REORG PLUS can continue, REORG PLUS issues the message number with a suffix of I (indicating that it is an informational message).

NOTE

BMC recommends that you specify CONTINUE only after your utility job has stopped due to an error and you have either corrected the problem and restarted the job or determined that you want to restart the job without making any changes.
**RETCODE**
Use this option to designate the return code that REORG PLUS issues when it encounters the message that you specify with CONTINUE UTILITY. Note the following results for certain values:

- If you do *not* specify a value for the RETCODE option, REORG PLUS issues the original error return code.

- If you specify 31 or less, REORG PLUS issues the return code.

- If you specify 32 or greater, REORG PLUS issues a user abend code that is equal to the specified integer. Do not specify a value greater than 4095.

**NOTE**
For a SHRLEVEL CHANGE reorganization, REORG PLUS operates as if you specified ON MESSAGE 50254 CONTINUE UTILITY RETCODE 0 in the ANALYZE phase, even if you do not specify ON MESSAGE.
ON FAILURE

ON FAILURE allows you to recover REORG PLUS from an abnormal termination (when REORG PLUS abends or terminates with a return code that is greater than or equal to eight). You can specify for each phase whether you want REORG PLUS to terminate with a specified return code or stop. If you do not specify ON FAILURE, an abnormal termination stops REORG PLUS.

Additional considerations
The following considerations apply to the ON FAILURE option:

- For a failure in the RELOAD phase of a SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY reorganization, you can only stop the job.

- For a SHRLEVEL CHANGE reorganization, REORG PLUS always functions as if you specified TERMINATE UTILITY.

- When invoking DSNUTILB, REORG PLUS ignores this option.

TERMINATE UTILITY

If you specify ON FAILURE TERMINATE UTILITY and an abend occurs, REORG PLUS cannot honor the TERMINATE option if all of the following conditions exist:

- You are running any type of reorganization except SHRLEVEL CHANGE.
- REORG PLUS is unable to start the table space during the UTILTERM phase.
- The abend code is any cancel type (‘x22’).

The ON FAILURE option is valid also for a single-phase reorganization (that is, when UNLOAD RELOAD is in effect). In this case, REORG PLUS combines the UNLOAD and RELOAD phases into one processing phase named REORG. If a failure occurs during unload processing of a single-phase reorganization, REORG PLUS considers the failure to be in the UNLOAD phase. In contrast, if the failure occurs during reload processing, REORG PLUS considers the failure to be in the RELOAD phase.
Table 37 describes processing in each phase when REORG PLUS abnormally terminates and ON FAILURE TERMINATE UTILITY is in effect. For information about the return code that REORG PLUS issues, see page 226.

### Table 37  Description of the TERMINATE UTILITY option

<table>
<thead>
<tr>
<th>Abend in phase</th>
<th>If you specify TERMINATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILINIT ANALYZE</td>
<td>REORG PLUS deletes the row containing the utility ID from the BMCUTIL, BMCSYNC, and BMCDICT tables.</td>
</tr>
</tbody>
</table>
| UNLOAD | REORG PLUS deletes the row containing the utility ID from the BMCUTIL, BMCSYNC, and BMCDICT tables. REORG PLUS starts all table and index spaces that are participating in the reorganization.  
If DELETEFILES YES is in effect, REORG PLUS deletes all of the data sets, any unregistered copy data sets, and (if ORIGINALDISP is DELETE) the staging VSAM data sets. |
| RELOAD | REORG PLUS deletes the row containing the utility ID from the BMCUTIL, BMCSYNC, and BMCDICT tables.  
For SHRLEVEL REFERENCE, REORG PLUS starts all table and index spaces taking part in the reorganization.  
For SHRLEVEL CHANGE, REORG PLUS leaves all objects in their original status.  
(For SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY, you can only stop REORG PLUS.)  
If DELETEFILES YES is in effect, REORG PLUS deletes all of the data sets, any unregistered copy data sets, and the staging VSAM data sets (if ORIGINALDISP is DELETE). |
| LOGAPPLY | REORG PLUS deletes the row containing the utility ID from the BMCUTIL, BMCSYNC, and BMCDICT tables.  
REORG PLUS leaves all objects in their original status.  
If DELETEFILES YES is in effect, REORG PLUS deletes all of the data sets, any unregistered copy data sets, and the staging VSAM data sets (if ORIGINALDISP is DELETE).  
**Note:** The LOGAPPLY phase name applies to both the LOGAPPLY and LOGFINAL phases of a SHRLEVEL CHANGE reorganization. |
| UTILTERM | Depending on normal UTILTERM processing and the use of the COPY YES option, REORG PLUS might not start the table and index spaces.  
If DELETEFILES YES is in effect, REORG PLUS deletes all of the data sets, any unregistered copy data sets, and the staging VSAM data sets (if ORIGINALDISP is DELETE).  
If you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE and the failure occurs during the rename or FASTSWITCH process associated with the staging data sets, REORG PLUS stops the utility instead of terminating it. REORG PLUS leaves the utility ID in a restartable state. |
RETCODE
Use this option to designate the return code that REORG PLUS issues when it encounters the message that you specify with TERMINATE UTILITY. Note the following results for certain values:

- If you do not specify a value for the RETCODE option, REORG PLUS issues the original error return code.

- If you specify 31 or less, REORG PLUS issues the return code.

- If you specify 32 or greater, REORG PLUS issues a user abend code that is equal to the specified integer. Do not specify a value greater than 4095.

If you specify the TIMEOUT command or installation option with TERM, rc and a drain timeout failure occurs, the return code that you specified for TERM overrides the return code from ON FAILURE.

STOP UTILITY

In any phase, when REORG PLUS abnormally terminates and you specify STOP UTILITY for that phase, REORG PLUS stops and leaves the utility ID in a restartable state.

You can also use STOP UTILITY to stop processing and avoid data set renaming when one of the following conditions exists:

- REORG PLUS encounters a key or RID mismatch during the index update process.
- REORG PLUS detects duplicates during the index build process.

**NOTE**
If you do not use STOP UTILITY and REORG PLUS encounters a key or RID mismatch during index update or detects duplicates during index build, REORG PLUS continues the reorganization and

- completes the LOGAPPLY and LOGFINAL phases
- renames the data sets
- ends with return code 8
- places the corrupted index in RBDP (REBUILD pending) status
The PREFORMAT option tells REORG PLUS whether to preformat the unused pages of the data set. If you specify PREFORMAT without a value, REORG PLUS assumes PREFORMAT YES.

For information about PREFORMAT and the SYSIDCIN data set, see “SYSIDCIN data set” on page 374.

**Specifying the default**

You can specify the default for the PREFORMAT command option in your installation options module by using the PREFORMAT installation option (page 715). REORG PLUS was shipped with a default value of NO for this option. The command overrides the default that is in the installation options module.

**YES**

If you specify PREFORMAT YES, REORG PLUS preformats the unused portion of the data set, writing full pages that have been initialized with zeros up to the high-allocated RBA of the table space and index spaces. Preformatting occurs after REORG PLUS reorganizes the data and indexes.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility as PREFORMAT.

**Restriction**

When reorganizing a catalog object, REORG PLUS changes this option to PREFORMAT NO.

**NO**

If you specify PREFORMAT NO, REORG PLUS does not preformat the unused pages.

**Restriction**

When invoking DSNUTILB, REORG PLUS ignores this option.
**RBALRSN_CONVERSION**

*(DB2 Version 11 or later)* The RBALRSN_CONVERSION option tells REORG PLUS which RBA or LRSN format to use for the reorganized object after the reorganization completes. If you do not specify a value for this option, REORG PLUS uses the value of the UTILITY_OBJECT_CONVERSION DB2 subsystem parameter.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

**Restrictions**
The following restrictions apply to this option:

- When both of the following conditions exist, REORG PLUS invokes DSNUTILB:
  - You specify SHRLEVEL CHANGE.
  - You request conversion to a format that is different from the current format of the object.

- If you are running a partial reorganization, nonpartitioned indexes remain in their original format.

- If a clone relationship exists for the object being reorganized, REORG PLUS ignores this option.

**NONE**

NONE indicates that you do not want REORG PLUS to convert the RBA or LRSN format.

**BASIC**

BASIC tells REORG PLUS to convert to basic (6-byte) format any object whose RBA or LRSN is in extended (10-byte) format.

**EXTENDED**

EXTENDED tells REORG PLUS to convert to extended (10-byte) format any object whose RBA or LRSN is in basic (6-byte) format.

**Restrictions**

When reorganizing a versioned XML table space where the START_TS and END_TS column is defined as BINARY(8), REORG PLUS operates as follows:

- For a partial reorganization, REORG PLUS performs no conversion.
- For a full reorganization, REORG PLUS invokes DSNUTILB.
For requirements and restrictions when REORG PLUS invokes DSNUTILB, see “Reorganization jobs that invoke DSNUTILB” on page 72.

**ARCROWS**

This option applies to a table space reorganization only.

ARCROWS is an estimate of the number of rows that REORG PLUS will discard to a dynamically allocated archive (SYSARC) data set.

If you dynamically allocate the archive data set, REORG PLUS uses the value that you specify for ARCROWS to determine the size of the archive data set. If you do not specify a value for ARCROWS, REORG PLUS allocates enough space to hold all of the rows in the table space.

When you specify a value for both the ARCROWS command option and the SIZEPCT installation or command option, REORG PLUS ignores SIZEPCT for the archive data set and uses only ARCROWS to determine the archive data set size.

BMC recommends that you overestimate the value instead of underestimating it. Overestimating causes REORG PLUS to allocate files that are larger than it needs, but underestimating can cause REORG PLUS to terminate.

**Restriction**

When invoking DSNUTILB, REORG PLUS ignores this option.
The CONDEXEC option specifies whether REORG PLUS should consider performing a conditional reorganization. For more information about conditional reorganizations, see “Conditional reorganization” on page 136.

**Specifying the default**
You can specify the default for the CONDEXEC command option in your installation options module by using the CONDEXEC installation option (page 679). REORG PLUS was shipped with a default value of NO for this option. The command overrides the default that is in the installation options module.

**YES**
YES tells REORG PLUS to examine the values of the limit command options (if specified) and the limit installation options in conjunction with the values in the DB2 catalog to determine whether to perform the reorganization. For a description of the limit command options (OFFPOSLIMIT, INDREFLIMIT, and LEAFDISTLIMIT), see page 231, page 233, and page 234. For a description of the limit installation options (OFFPOSLM, INDREFLM, and LEAFDSLM), see page 713, page 699, and page 704.

**Restriction**
REORG PLUS terminates if you specify this option when reorganizing a LOB table space.

**NO**
NO tells REORG PLUS not to perform a conditional reorganization. Instead, REORG PLUS proceeds with the reorganization, regardless of the existence or values of the limit command and installation options (OFFPOSLM, INDREFLM, and LEAFDSLM). If you set values for the limit options in your installation options module, you can specify CONDEXEC NO on the REORG command to ignore those values for this job.

**BMC**
BMC tells REORG PLUS to use information from the DASD MANAGER PLUS exceptions table to determine whether the reorganization is needed. The BMCTRIG feature of DASD MANAGER PLUS puts exception rows into the exceptions table based on customer rules and BMCTRIG’s own analysis, thus providing an expanded set of criteria for determining whether a reorganization is needed. To populate the exceptions tables, run the BMCTRIG component of DASD MANAGER PLUS.
REORG PLUS issues one of the following return codes after examining the DASD MANAGER PLUS exceptions table:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No calculated value exceeded an exception value. REORG PLUS does not perform a reorganization.</td>
</tr>
<tr>
<td>2</td>
<td>A calculated value exceeded an exception value. REORG PLUS performs a reorganization. If REORG PLUS issues any return code greater than 2 during processing, that return code supersedes return code 2.</td>
</tr>
</tbody>
</table>

For more information, see the following references:

- “Conditional reorganizations using the DASD MANAGER PLUS exceptions table” on page 138
- “Conditional reorganization” on page 136.

**Restrictions:**
The following restrictions apply to CONDEXEC BMC:

- This option is available only if you are using REORG PLUS as a component of the Database Performance solution. If you do not have a license for the solution, REORG PLUS ignores the option and continues with the reorganization.

- REORG PLUS terminates if you specify this option when reorganizing a LOB table space.

- When invoking DSNUTILB, REORG PLUS ignores this option.

- If the information in the DASD MANAGER PLUS tables for the object that you are reorganizing is missing or incomplete, REORG PLUS changes BMC to NO and continues with the reorganization.

**OFFPOSLIMIT**

This option applies to a table space reorganization only.

The OFFPOSLIMIT option allows you to set conditions under which REORG PLUS reorganizes a table space. Valid values are 0 through 100. REORG PLUS also issues a report listing each object it examined, the DB2 catalog values retrieved for each object, and whether each object should be reorganized.
For every table in the table space named in the REORG command, REORG PLUS performs the following calculation. The calculation applies to the specified partitions in SYSIBM.SYSINDEXPART for the table’s explicit clustering index.

\[(\text{NEAROFFPOS} + \text{FAROFFPOS}) \times 100 / \text{CARDF} = \text{integer}\]

If any calculated integer value exceeds the OFFPOSLIMIT value, REORG PLUS reorganizes the object.

If you specify the OFFPOSLIMIT command option without an integer value, REORG PLUS uses the integer value in the corresponding OFFPOSLM installation option. If you specify CONDEXEC NO or BMC on the REORG command, REORG PLUS ignores the OFFPOSLIMIT command option. For a detailed description of the effects of the various command and installation options on a conditional reorganization, see “Conditional reorganization” on page 136.

**NOTE**

You can obtain the report that recommends objects for reorganization without performing any reorganizations. To do so, also specify REPORTONLY in the REORG command.

When you specify CONDEXEC YES for either the installation or command option and you specify an OFFPOSLIMIT value, REORG PLUS issues one of the following return codes:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No calculated value exceeded the OFFPOSLIMIT value. REORG PLUS does not perform a reorganization.</td>
</tr>
<tr>
<td>2</td>
<td>A calculated value exceeded the OFFPOSLIMIT value. REORG PLUS performs a reorganization. If REORG PLUS issues any return code greater than 2 during processing, that return code supersedes return code 2.</td>
</tr>
</tbody>
</table>

**Restriction**

If you are reorganizing a LOB table space and specifying this option would cause REORG PLUS to check criteria for conditional execution, REORG PLUS terminates.

**Specifying the default**

You can specify the default for the OFFPOSLIMIT command option in your installation options module by using the OFFPOSLM installation option (page 713). REORG PLUS was shipped with a default value of 10 for this option. The command overrides the default that is in the installation options module.
INDREFLIMIT

This option applies to a table space reorganization only.

The INDREFLIMIT option allows you to set conditions under which REORG PLUS reorganizes a table space. Valid values are 0 through 100. REORG PLUS also issues a report listing each object it examined, the DB2 catalog values retrieved for each object, and whether each object should be reorganized.

For the table space named in the REORG command, REORG PLUS performs the following calculation. The calculation applies to the specified partitions in SYSIBM.SYSTABLEPART for the table space.

\[(\text{NEARINDREF} + \text{FARINDREF}) \times 100 / \text{CARDF=integer}\]

If any calculated integer value exceeds the INDREFLIMIT value, REORG PLUS reorganizes the object.

If you specify the INDREFLIMIT command option without an integer value, REORG PLUS uses the integer value in the corresponding INDREFLIM installation option. If you specify CONDEXEC NO or BMC on the REORG command, REORG PLUS ignores the INDREFLIMIT command option. For more information, see “Conditional reorganization” on page 136.

**NOTE**

You can obtain the report that recommends objects for reorganization without performing any reorganizations. To do so, also specify REPORTONLY in the command.
When you specify CONDEXEC YES for either the installation or command option and you specify an INDREFLIMIT value, REORG PLUS issues one of the following return codes:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No calculated value exceeded the INDREFLIMIT value. REORG PLUS does not perform a reorganization.</td>
</tr>
<tr>
<td>2</td>
<td>A calculated value exceeded the INDREFLIMIT value. REORG PLUS performs a reorganization. If REORG PLUS issues any return code greater than 2 during processing, that return code supersedes return code 2.</td>
</tr>
</tbody>
</table>

**Restriction**

If you are reorganizing a LOB table space and specifying this option would cause REORG PLUS to check criteria for conditional execution, REORG PLUS terminates.

**Specifying the default**

You can specify the default for the INDREFLIMIT command option in your installation options module by using the INDREFLM installation option (page 699). REORG PLUS was shipped with a default value of 10 for this option. The command overrides the default that is in the installation options module.

**LEAFDISTLIMIT**

*This option applies only to an index-only reorganization.*

The LEAFDISTLIMIT option allows you to set conditions under which REORG PLUS reorganizes an index. Specify any positive integer (0 or greater). REORG PLUS issues a report that lists each object it examined, the DB2 catalog values retrieved for each object, and whether each object should be reorganized.

For every partition named in the REORG command, REORG PLUS compares the LEAFDISTLIMIT value to the LEAFDIST value in SYSIBM.SYSINDEXPART for the specified index. If any LEAFDIST value exceeds the LEAFDISTLIMIT value, REORG PLUS reorganizes the object.

If you specify the LEAFDISTLIMIT command option without a value, REORG PLUS uses the value in the corresponding LEAFDSLM installation option. If you specify CONDEXEC NO or BMC on the REORG command, REORG PLUS ignores the LEAFDISTLIMIT command option. For more information, see “Conditional reorganization” on page 136.

**NOTE**

You can obtain the report that recommends objects for reorganization without performing any reorganizations. To do so, also specify REPORTONLY on the command.
When you specify CONDEXEC YES for either the installation or command option and specify a LEAFDISTLIMIT value, REORG PLUS issues one of the following return codes:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No calculated value exceeded the LEAFDISTLIMIT value. REORG PLUS does not perform a reorganization.</td>
</tr>
<tr>
<td>2</td>
<td>A calculated value exceeded the LEAFDISTLIMIT value. REORG PLUS performs a reorganization. If REORG PLUS issues any return code greater than 2 during processing, that return code supersedes return code 2.</td>
</tr>
</tbody>
</table>

**Specifying the default**

You can specify the default for the LEAFDISTLIMIT command option in your installation options module by using the LEAFDSLM installation option (page 704). REORG PLUS was shipped with a default value of 200 for this option. The command overrides the default that is in the installation options module.
REPORTONLY

When you specify the REPORTONLY option, REORG PLUS produces a report that lists the tests and results for a conditional reorganization but does not perform the reorganization.

If you specify REPORTONLY along with CONDEXEC YES on the REORG command or in the installation options module, the report lists

- all of the objects that REORG PLUS considered
- the limits that REORG PLUS retrieved for the objects from the DB2 catalog
- whether the objects should be reorganized based on the values of the limit command options (OFFPOSLIMIT, INDREFLIMIT, and LEAFDISTLIMIT), or their corresponding values in the installation options module

If you specify REPORTONLY with CONDEXEC BMC on the REORG command or in the installation options module, the report lists the exceptions that REORG PLUS found in the DASD MANAGER PLUS exceptions table. However, if you specify REPORTONLY with CONDEXEC BMC and you are not using REORG PLUS as part of the Database Performance for DB2 solution (that is, no valid solution password is found), REORG PLUS terminates.

If the value of the CONDEXEC command or installation option is NO, REORG PLUS ignores the REPORTONLY option.

As a result of the REPORTONLY option, REORG PLUS issues one of the following return codes:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No value exceeded the value that you specified in a limit option or no exceptions were found in the DASD MANAGER PLUS exceptions table.</td>
</tr>
<tr>
<td>2</td>
<td>A value exceeded the value that you specified in a limit option or an exception was found in the DASD MANAGER PLUS exceptions table. One or more objects should be reorganized. If REORG PLUS issues any return code greater than 2 during processing, that return code supersedes return code 2.</td>
</tr>
</tbody>
</table>

For a description of when REORG PLUS uses the limit command values and when it uses the installation values to do the calculations, see “Conditional reorganization” on page 136.
Basic processing options

DSNUTILB

The DSNUTILB option tells REORG PLUS whether to invoke DSNUTILB to pass processing to the IBM DB2 REORG utility. REORG PLUS uses this option to enable support for certain features. For the list of features, see “Reorganization jobs that invoke DSNUTILB” on page 72.

Specifying the default
You can specify the default for the DSNUTILB command option in your installation options module by using the DSNUTILB installation option (page 689). REORG PLUS was shipped with a default value of YES for this option. The command option overrides the default that is in the installation options module.

YES

DSNUTILB YES tells REORG PLUS to invoke DSNUTILB when DSNUTILB is required to support the object type that is involved in the reorganization. If any table in or index on the table space that you are reorganizing uses a feature that REORG PLUS supports via DSNUTILB, REORG PLUS invokes DSNUTILB for that job.

NOTE
REORG PLUS invokes DSNUTILB to enable new features quickly. REORG PLUS generally provides native support for these features (without invoking DSNUTILB) in a later version of the product or via PTF. You cannot use the DSNUTILB option (or any other option) to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively.

For requirements and restrictions when REORG PLUS invokes DSNUTILB, see “Reorganization jobs that invoke DSNUTILB” on page 72.

NO

DSNUTILB NO tells REORG PLUS not to invoke DSNUTILB when DSNUTILB is required to support the object type that is involved in the reorganization. If DSNUTILB NO is in effect when it is required, REORG PLUS terminates.
TIMEOUT

The TIMEOUT option specifies the action that REORG PLUS should take after it has exhausted all attempts (including retries) to obtain a drain.

The value of the return code depends on what you specify for the TIMEOUT installation and command option and the ON FAILURE command option. Table 38 describes the return code information.

Table 38  Return code hierarchy for the TIMEOUT option

<table>
<thead>
<tr>
<th>TIMEOUT option</th>
<th>ON FAILURE with RETCODE specified</th>
<th>Return code is returned from</th>
</tr>
</thead>
<tbody>
<tr>
<td>TERM</td>
<td>yes</td>
<td>ON FAILURE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The return code is the value that you specified with RETCODE.</td>
</tr>
<tr>
<td>TERM</td>
<td>no</td>
<td>TIMEOUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The return code is 8.</td>
</tr>
<tr>
<td>TERM, rc</td>
<td>does not matter</td>
<td>TIMEOUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The return code is the value that you specified with TERM.</td>
</tr>
<tr>
<td>ABEND</td>
<td>does not matter</td>
<td>TIMEOUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The return code is 12.</td>
</tr>
<tr>
<td>not specified</td>
<td>no</td>
<td>REORG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The return code is 8.</td>
</tr>
</tbody>
</table>

Specifying the default

You can specify the default for the TIMEOUT command option in your installation options module by using the TIMEOUT installation option (page 731). REORG PLUS was shipped with a default value of TERM for this option. The command overrides the default that is in the installation options module.
TERM

If you specify TERM and a timeout condition occurs, REORG PLUS

- issues messages BMC50020I and BMC50285E
- leaves the objects in their original states
- terminates the job
- deletes the data sets and unregistered copy data sets if DELETEFILES YES is in effect

rc
TERM, rc allows you to specify the return code that you want REORG PLUS to issue when terminating due to a timeout condition. Note the following results for certain values:

- When you specify 31 or less, REORG PLUS issues that integer as the return code.
- When you specify 32 or greater, REORG PLUS issues a user abend that is equal to the specified integer.

The return code that you specify for this option overrides any return code that you specify for ON FAILURE TERMINATE UTILITY.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

ABEND

If you specify ABEND and a timeout condition occurs, REORG PLUS

- abends with user code 3200
- leaves the objects in their original states
- leaves an entry in the BMCUTIL table

If you are running a SHRLEVEL CHANGE reorganization, you must resubmit the job with TERM instead of NEW on the EXEC statement.
Basic processing options

The DSNUEXIT option allows you to specify the name of a user-written exit that creates user-defined variables. You can use these variables with the DSNPAT or SPIILDSNPAT option to create patterns for names for dynamically allocated data sets. For details about DSNPAT, see page 315. For details about SPIILDSNPAT, see page 331. For more information about creating a DSNUEXIT user exit and user-defined variables, see Appendix C, “REORG PLUS user exits.”

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the DSNUEXIT command option in your installation options module by using the DSNUEXIT installation option (page 688). REORG PLUS was shipped with a default value of (NONE,ASM) for this option. The command overrides the default that is in the installation options module.

exitName
The exitName is the name of the user-written exit.

NONE
Specify NONE in place of an exit name to override a default exit name in your installation options module. This tells REORG PLUS that you do not want to call an exit to create user-defined variables.

programLanguage
You can specify one of the following program languages to indicate the language of the specified exit:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Specifies an exit written in</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td>assembler (the default)</td>
</tr>
<tr>
<td>COBOL2</td>
<td>COBOL II</td>
</tr>
</tbody>
</table>
The DSRSEXIT option allows you to specify the name of a user-written REXX exit that REORG PLUS calls once for each object immediately before redefining the object. You can use this exit to

- resize the DB2 VSAM data sets by changing the value of the primary and secondary allocation quantities, and update the DB2 catalog with the changed quantity values for use with subsequent allocations

REORG PLUS issues an SQL ALTER TABLESPACE or ALTER INDEX statement to update the DB2 catalog with the changed quantity values for use with subsequent allocations. You can modify a variable in your exit to prevent REORG PLUS from issuing this command.

- override REDEFINE YES with REDEFINE NO selectively for an object that meets your criteria

- order the volumes in the current storage group for each object

- restrict the volume list that you return to REORG PLUS to a subset of the original storage group volume list

- add SMS classes

If no SMS classes exist in the DSNZPARMs or storage-group definition, you can add the classes by using the DSRSEXIT user exit. However, you cannot change existing specified SMS classes by using the DSRSEXIT user exit.

You can use the DSRSEXIT user exit only for storage-group-defined objects. If you want to alter the size of VCAT-defined objects, use the SYSIDCIN data set. For more information, see “SYSIDCIN data set” on page 374.

For details about this user exit, see Appendix C, “REORG PLUS user exits.”

---

**Table 39  Program language keywords (part 2 of 2)**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Specifies an exit written in</th>
</tr>
</thead>
<tbody>
<tr>
<td>LE_COBOL</td>
<td>Language Environment COBOL</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>LE_C</td>
<td>Language Environment C</td>
</tr>
</tbody>
</table>

---

**WARNING**

The program language, whether specified here or in your installation options module, must match the source language of the specified user exit. If the language does not match, your user exit might not operate correctly.
Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the DSRSEXIT command option in your installation options module by using the DSRSEXIT installation option (page 690). REORG PLUS was shipped with a default value of (NONE,REXX) for this option. The command overrides the default that is in the installation options module.

exitName

The exitName is the name of the user-written exit.

NONE

Specify NONE in place of an exit name to override a default exit name in your installation options module. This tells REORG PLUS that you do not want to call a DSRSEXIT user exit.

programLanguage

The exit that you specify for DSRSEXIT must be written in REXX (the default). If you specify any other language, your user exit might not operate correctly.

TERMEXIT

The TERMEXIT option allows you to specify the name of a user-written exit that gives you dynamic control over several options during termination processing. Use the TERMEXIT user exit to dynamically control processing of BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS. For details about this user exit, see Appendix C, “REORG PLUS user exits.”

Restrictions
The following restrictions apply to the TERMEXIT option and to using a TERMEXIT user exit:

- When invoking DSNUTILB, REORG PLUS ignores this option.

- You cannot use a user exit supplied by the TERMEXIT option to override BMCSTATS NO or UPDATEDB2STATS NO to YES.

Specifying the default
You can specify the default for the TERMEXIT command option in your installation options module by using the TERMEXIT installation option (page 730). REORG PLUS was shipped with a default value of (NONE,REXX) for this option. The command overrides the default that is in the installation options module.
exitName

The `exitName` is the name of the user-written exit.

NONE

Specify NONE in place of an exit name to override a default exit name in your installation options module. This tells REORG PLUS that you do not want to call a TERMEXIT user exit.

programLanguage

The exit that you specify for TERMEXIT must be written in REXX (the default). If you specify any other language, your user exit might not operate correctly.

MAPTEXIT

(For DB2 Versions 9 and 10) The MAPTEXIT option allows you to specify the name of a user-written exit that creates mapping objects for use with a DSNUTILB reorganization. This exit also drops the mapping objects after DSNUTILB processing completes successfully.

For details about this user exit, see Appendix C, “REORG PLUS user exits.”

Restrictions

The following restrictions apply to the MAPTEXIT option and the MAPTEXIT user exit:

- REORG PLUS uses this exit only when invoking DSNUTILB for a SHRLEVEL CHANGE reorganization.
- REORG PLUS ignores this option when any of the following conditions exists:
  - You are running on DB2 Version 11 or later.
  - You also specify the MAPPINGTABLE option.
  - You specify REORG INDEX.

Specifying the default

You can specify the default for the MAPTEXIT command option in your installation options module by using the MAPTEXIT installation option (page 709). REORG PLUS was shipped with a default value of (NONE,REXX) for this option. The command overrides the default that is in the installation options module.

exitName

The `exitName` is the name of the user-written exit.
NONE

Specify NONE in place of an exit name to override a default exit name in your installation options module. NONE tells REORG PLUS that you do not want to call a user exit to create and drop mapping table objects.

programLanguage

The exit that you specify for MAPEXIT must be written in REXX (the default). If you specify any other language, your user exit might not operate correctly.
DRAIN_WAIT

The DRAIN_WAIT option specifies the drain timeout value to use. If it cannot drain all of the objects within the time period specified by DRAIN_WAIT, REORG PLUS completes the following process:

1. releases the drains that it has obtained so far
2. waits the length of time that you specify in the RETRY_DELAY command option (or DRNDELAY installation option)
3. tries again to drain the objects for the number of times that you specify in the RETRY command option (or DRNRETRY installation option)

NOTE

The FORCE option tells REORG PLUS whether to cancel DB2 threads that might prevent a successful drain and, if so, when to cancel them. Depending on the FORCE option that you specify, the drain process might differ from the process described here. For more information, see “FORCE” on page 248.

When invoking DSNUTILB, REORG PLUS converts the DRAIN_WAIT option based on the DSNZPARM settings, and passes this option to the IBM DB2 REORG utility as DRAIN_WAIT integer.

Specifying the default

You can specify the default for the DRAIN_WAIT command option in your installation options module by using the DRNWAIT installation option (page 687). REORG PLUS was shipped with a default value of NONE for this option. The command overrides the default that is in the installation options module.

NONE

NONE means that the drain request issued by REORG PLUS times out immediately if the drain cannot acquire the lock. NONE prevents any application transactions from being queued during the drain process. BMC recommends that you specify NONE in high-transaction environments, such as SAP.

When invoking DSNUTILB, REORG PLUS converts this option to DRAIN_WAIT 1.
Basic processing options

**UTIL**

UTIL tells REORG PLUS to use the standard DB2 utility timeout value defined in DSNZPARMs for your site (IRLMRWT multiplied by UTIMOUT). The wait time applies to each object involved in the reorganization.

When invoking DSNUTILB, REORG PLUS converts this option to DRAIN_WAIT \( \text{integer} \), where \( \text{integer} \) is the product of SPRTMOUT multiplied by SPRMUTO.

**SQL**

If you specify SQL, REORG PLUS uses the standard SQL timeout value (IRLMRWT) as the drain timeout value. The wait time applies to each object involved in the reorganization.

When invoking DSNUTILB, REORG PLUS converts DRAIN_WAIT SQL to DRAIN_WAIT \( \text{integer} \), where \( \text{integer} \) equals SPRMOUT.

**integer**

Specify any integer value from 0 through 1800, as follows:

- 0 is equivalent to the value UTIL.
- 1 through 1800 specifies the number of seconds to wait to obtain the drain for each drain retry before timing out.

**RETRY**

The RETRY option specifies the maximum number of times that you want REORG PLUS to attempt to obtain a drain before it terminates. The number of attempts can range from 0 through 255.

*Specifying the default*

You can specify the default for the RETRY command option in your installation options module by using the DRNRETRY installation option (page 686). REORG PLUS was shipped with a default value of 255 for this option. The command overrides the default that is in the installation options module.

**RETRY_DELAY**

After a drain times out, the RETRY_DELAY option specifies the minimum number of seconds that you want REORG PLUS to wait before it tries again to obtain the drain. The number of seconds can range from 1 through 1800.
Specifying the default
You can specify the default for the RETRY_DELAY command option in your installation options module by using the DRNDelay installation option (page 686). REORG PLUS was shipped with a default value of 1 for this option. The command overrides the default that is in the installation options module.

**DSPLOCKS**

The DSPLOCKS option tells REORG PLUS what action to take regarding displaying claims and locks if a drain attempt times out.

**Restriction**
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the DSPLOCKS command option in your installation options module by using the DSPLOCKS installation option (page 689). REORG PLUS was shipped with a default value of DRNFAIL for this option. The command overrides the default that is in the installation options module.

**DRNFAIL**

DRNFAIL tells REORG PLUS to display the claims and locks once, after the final attempt to obtain the drain times out.

**NONE**

NONE tells REORG PLUS not to display any claims or locks.

**RETRY**

RETRY tells REORG PLUS to display claims and locks after each drain timeout.
FORCE

The FORCE option specifies whether to cancel DB2 threads that might prevent a drain process from completing.

When invoking DSNUTILB, REORG PLUS ignores this option.

*Specifying the default*
You can specify a default for the FORCE command option in your installation options module by using the FORCE installation option (page 693). REORG PLUS was shipped with a default value of NONE for this option. The FORCE command option overrides the default that is in the installation options module.

NONE

FORCE NONE tells REORG PLUS not to cancel DB2 threads that might prevent the drain process from completing.

READERS

FORCE READERS tells REORG PLUS to cancel read claimers at the point specified by the FORCE_AT option.

FORCE_AT

The FORCE_AT option tells REORG PLUS at which point during the drain process to cancel DB2 threads that might prevent the drain process from completing.

*Specifying the default*
You can specify a default for the FORCE_AT command option in your installation options module by using the first parameter of the FORCE_AT installation option (page 693). REORG PLUS was shipped with a default value of START for this option. The FORCE_AT command option overrides the default that is in the installation options module.
Additional considerations
When determining which option to choose, consider the following information:

- To obtain the same behavior as the FORCE option of the IBM DB2 REORG TABLESPACE utility, specify FORCE_AT LASTRETRY.

- The cancelation process includes a small amount of time to issue the cancel commands, plus any time that DB2 needs to roll back the canceled transactions. If the drain wait time (controlled by the DRAIN_WAIT option) is shorter than the total cancelation process time, the drain fails. FORCE_AT START and FORCE_AT RETRY allow for additional drain attempts after the start of the cancelation process; therefore, these options increase the likelihood that the drain will succeed when a short drain wait time is in effect. (Alternatively, you can increase the drain wait time.)

  **START.** FORCE_AT START tells REORG PLUS to start canceling threads when the drain request begins.

  **RETRY.** FORCE_AT RETRY tells REORG PLUS to start canceling threads the first time the drain process times out and REORG PLUS attempts to retry the drain.

  **LASTRETRY.** FORCE_AT LASTRETRY tells REORG PLUS to start canceling threads at the beginning of the last retry attempt following a drain process timeout. You can control the number of drain retry attempts with the RETRY command option (page 246) or DRNRETRY installation option (page 686).

**FORCE_DELAY**
The FORCE_DELAY option tells REORG PLUS how long to wait, in hundredths of a second, before starting the thread cancelation process. For example, specify 7 to tell REORG PLUS to wait .07 seconds.

Specify an integer value of 0 or greater. A value of 0 tells REORG PLUS to start the cancelation process immediately upon reaching the point specified by the FORCE_AT option.

**Specifying the default**
You can specify a default for the FORCE_DELAY command option in your installation options module by using the second parameter of the FORCE_AT installation option (page 693). REORG PLUS was shipped with a default value of 3 for this parameter. The FORCE_DELAY command option overrides the default that is in the installation options module.

**FORCE_RPT**
The FORCE_RPT option tells REORG PLUS whether to display a report of the canceled threads.

  **NO.** NO tells REORG PLUS not to display the report.
**YES**. YES tells REORG PLUS to display the report. If your JCL includes a BMCFORCE DD statement, REORG PLUS sends the canceled threads report to that data set. Otherwise, the report is displayed in the REORG PLUS SYSPRINT.

**Specifying the default**
You can specify a default for the FORCE_RPT command option in your installation options module by using the FORCE_RPT installation option (page 694). REORG PLUS was shipped with a default value of NONE for this option. The FORCE_RPT command option overrides the default that is in the installation options module.

**ALL**
FORCE ALL tells REORG PLUS to cancel both read and write claimers at the point specified by the FORCE_AT option.

**FORCE_AT**
For information about this option, see “FORCE_AT” on page 248.

**FORCE_DELAY**
For information about this option, see “FORCE_DELAY” on page 249.

**FORCE_RPT**
For information about this option, see “FORCE_RPT” on page 249.

**REPORTONLY**
FORCE REPORTONLY tells REORG PLUS to display a report of the threads that it would have canceled if you had specified FORCE ALL. When reporting which threads it would have canceled, REORG PLUS takes into account the current values of FORCE_AT and FORCE_DELAY.

If your JCL includes a BMCFORCE DD statement, REORG PLUS sends the canceled threads report to that data set. Otherwise, the report is displayed in the REORG PLUS SYSPRINT.

**FORCE_AT**
For information about this option, see “FORCE_AT” on page 248.

**FORCE_DELAY**
For information about this option, see “FORCE_DELAY” on page 249.

**FORCE_RPT**
REORG PLUS ignores FORCE_RPT when you specify FORCE REPORTONLY.
SIXSNAP

This option applies only to a SHRLEVEL REFERENCE or SHRLEVEL CHANGE partial table space reorganization.

The SIXSNAP option tells REORG PLUS whether to use the Instant Snapshot technology of XBM and SUF to create a copy of storage-group-defined nonpartitioned indexes. Using Instant Snapshot can improve performance because Instant Snapshot uses intelligent storage devices to copy the index data sets in one operation, rather than performing multiple I/O operations. To use Instant Snapshot, the supported intelligent storage devices must be available.

For detailed information about the SIXSNAP function and considerations, see “Instant Snapshot with nonpartitioned indexes” on page 122. For a list of the supported devices, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the SIXSNAP command option in your installation options module by using the SIXSNAP installation option (page 723). REORG PLUS was shipped with a default value of NO for this option. The command overrides the default that is in the installation options module.

AUTO

AUTO tells REORG PLUS to attempt to use Instant Snapshot. If the Instant Snapshot copy fails, REORG PLUS changes SIXSNAP to NO and uses the software-based copy method to recopy the index that could not be copied with Instant Snapshot.

YES

YES tells REORG PLUS to use only Instant Snapshot technology. If the Instant Snapshot copy fails, REORG PLUS terminates.

NO

NO tells REORG PLUS to use the software-based copy method to copy each nonpartitioned index.
ZIIP

The ZIIP option tells REORG PLUS whether to attempt to use IBM® System z® Integrated Information Processors (zIIPs). REORG PLUS can use enclave service request blocks (SRBs) to enable zIIP processing automatically while running jobs. Using zIIP processing can reduce the overall CPU time for REORG PLUS jobs.

**Restriction**

When invoking DSNUTILB, REORG PLUS ignores this option.

**Specifying the default**

You can specify the default for the ZIIP command option in your installation options module by using the ZIIP installation option (page 739). REORG PLUS was shipped with a default value of ENABLED for this option. The ZIIP command option overrides the default that is in the installation options module.

**ENABLED**

ZIIP ENABLED tells REORG PLUS to attempt to offload eligible processing to an available zIIP. If the zIIP is busy or not available, normal processing continues on a general-purpose processor.

To enable and use zIIP processing with REORG PLUS, you must

- have an installed authorized version of XBM or SUF
- start and maintain an XBM subsystem in your environment

**NOTE**

You can specify a particular XBM subsystem to use by specifying a value for the XBMID installation or command option. For more information, see “XBMID” on page 253 or page 738.

- have a zIIP available in your environment

For more information, see the following references:

- For XBM or SUF requirements and considerations, see “XBM and SUF considerations” on page 151.
Basic processing options

For the steps to enable zIIP processing, see “Using XBM or SUF to enable zIIP processing” on page 153.

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

**DISABLED**

ZIIP DISABLED tells REORG PLUS to not attempt to use zIIP processing.

**XBMID**

Specify XBMID to identify a specific active XBM subsystem to use for snapshot processing and zIIP processing in REORG PLUS.

The variable ssid (subsystem ID) is the unique identifier that you specified when you installed XBM or SUF. If you are using XBM or SUF in a DB2 data sharing environment, you can use the value of the XBMGROUP parameter in place of the ssid. The XBMGROUP is the name of the cross-system coupling facility (XCF) group that is defined to the XBM subsystem, and its default value is XBMGROUP.

If you specify an XBM subsystem, it must be:

- available
- at a supported maintenance level
- enabled for the required function

If you do not specify an XBM subsystem (either here or with the XBMID installation option), REORG PLUS automatically searches for an XBM subsystem that meets the same criteria. If you have multiple subsystems that meet these criteria, REORG PLUS can use any one of these subsystems.

For more information about using XBM or SUF with REORG PLUS, see “XBM and SUF considerations” on page 151. For more information about XBM and SUF, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

**Restriction**

When invoking DSNUTILB, REORG PLUS ignores this option.

**Specifying the default**

You can specify the default for the XBMID command option in your installation options module by using the XBMID installation option (page 738). The command option overrides the default that is in the installation options module.
WTOMSG

*This option applies to SHRLEVEL CHANGE only.*

Specify WTOMSG SUFSTART to write message BMC50008I to the MVS™ system log to indicate that the REORG PLUS snapshot initialization has successfully completed. You can use the text of this message to trigger the submission of jobs that you want to run concurrently with the REORG PLUS job.

Specify a text string of up to 50 characters enclosed in single quotes to indicate the message to print in the MVS system log. REORG PLUS truncates strings greater than 50 characters. Quotes cannot appear within the text string. The message has the following format:

```
BMC50008I 'text'
```

*Restriction*

When invoking DSNUTILB, REORG PLUS ignores this option.
IDCACHE

The IDCACHE option allows you to specify the size of the cache that REORG PLUS uses when populating document ID columns. Specify the size as the number of values to reserve. REORG PLUS reserves at least one cache for each unload and log apply task that requires it. The valid values for this option are 1 through 2,147,483,647.

REORG PLUS reserves this cache of numbers in SYSIBM.SYSSEQUENCES. BMC recommends that you use the default value of 10000. Specifying a cache that is too large or too small might cause REORG PLUS to retrieve values that it will not use. Specifying a smaller cache size can also impact performance because REORG PLUS must access and update the DB2 catalog more frequently.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the IDCACHE command option in your installation options module by using the IDCACHE installation option (page 699). REORG PLUS was shipped with a default value of 10000 for this option. The command overrides the default that is in the installation options module.
AUTOESTSPACE

This option applies to a DSNUTILB table space reorganization only.

The AUTOESTSPACE option specifies whether to calculate the optimal hash space size for the table space. REORG PLUS ignores this option if the table space does not contain a table defined as ORGANIZE BY HASH.

Currently, this option applies only when REORG PLUS invokes DSNUTILB. When not invoking DSNUTILB, REORG PLUS ignores this option.

YES

YES (the default) indicates that real-time statistics values are to be used to calculate the hash space.

NO

NO indicates that the HASH SPACE value that is defined on the table is to be used.

AUX

This option applies to a DSNUTILB table space reorganization only.

The AUX option specifies whether to reorganize associated LOB table spaces while reorganizing the base table space. Currently, this option applies only when REORG PLUS invokes DSNUTILB. When not invoking DSNUTILB, REORG PLUS ignores this option.

If you do not specify this option, the default is AUX NO except in certain cases. When you do not specify this option, you are reorganizing a table space that contains a LOB column, and any of the following conditions exists, REORG PLUS invokes DSNUTILB. In this case, the DB2 REORG utility defaults to AUX YES:

- You specify SHRLEVEL REFERENCE and REBALANCE.
- You are reorganizing more than one partition of a partition-by-growth table space
- You specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE, and any partition is in REORP status.
YES

YES indicates that associated LOB table spaces are to be reorganized when you specify REORG TABLESPACE on the base table space. This option applies only to the LOB table spaces associated with the partitions that are being reorganized.

The LOB and DSNUTILB installation options must be set to YES.

Restrictions
AUX YES has the following restrictions:

- If you are reorganizing a nonpartitioned base table space, REORG PLUS changes AUX YES to AUX NO.
- REORG PLUS terminates if AUX YES is in effect and you specify UNLOAD PAUSE.

NO

NO indicates that only the base table space is to be reorganized when you specify REORG TABLESPACE on the base table space. Associated LOB table spaces are not reorganized.

Restrictions
REORG PLUS terminates if AUX NO is in effect for a table space that contains a LOB column and any of the following conditions exists:

- Participating partitions are in REORP status.
- You specify REBALANCE.
- The table space contains an inline LOB column whose length definition was altered before the reorganization.

NOTE

In REORG PLUS version 10.2 and earlier, the AUXREORG installation option enabled you to specify a default for the AUX command option. REORG PLUS no longer requires or uses this option.
Statistics options

The statistics options allow you to update statistical information in the DASD MANAGER PLUS database statistics tables and in the DB2 catalog. REORG PLUS uses the BMC Common Statistics component to update these statistics.

**NOTE**

REORG PLUS automatically updates the DB2 real-time statistics tables. For more information, see “DB2 real-time statistics” on page 142.

Reporting options

The Common Statistics component generates a report of the statistics that it updates. You can direct the output of this report in any of the following ways:

- **Send the output to a data set that is separate from the REORG PLUS SYSPRINT.**
  
  To use this method, specify an ASUSRPRT DD statement in your JCL.

- **Send the output to the REORG PLUS SYSPRINT.**
  
  To use this method, do not specify an ASUSRPRT DD statement in your JCL. REORG PLUS automatically sends the report to your REORG PLUS SYSPRINT.

  **NOTE**
  
  The Common Statistics component does not send this output to a REORG PLUS SYSPRINT data set.

- **Suppress the output.**
  
  To suppress the output, specify //ASUSRPRT DD DUMMY in your JCL.

For descriptions of the fields that appear in this report, see the DASD MANAGER PLUS for DB2 Reference Manual.
BMCSTATS

The BMCSTATS option tells REORG PLUS whether to use the BMC statistics that it gathers to update the DASD MANAGER PLUS database statistics tables.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

NO

BMCSTATS NO, which is the default, tells REORG PLUS not to update the DASD MANAGER PLUS database statistics tables.

YES

BMCSTATS YES tells REORG PLUS to update the DASD MANAGER PLUS database statistics tables. This option requires that you have the DASD MANAGER PLUS product installed.

Restrictions
The following restrictions apply to BMCSTATS YES:

- REORG PLUS changes BMCSTATS YES to BMCSTATS NO when any of the following conditions exists:
  
  — Statistics collection fails in any one of the reload or build tasks
  — On restart, a participating table space partition was completely loaded or index partition was completely built in the original job

- REORG PLUS ignores this option when you are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect.
REORG PLUS does not collect statistics for the following objects and activities:

- clone tables
- nonpartitioned indexes when running a partial reorganization
- activities that occur during the LOGAPPLY phase

**Additional considerations**

The following additional considerations apply to BMCSTATS YES:

- You must include the BMCPSWD data set in the STEPLIB of your reorganization job.

- For any columns that have a FIELDPROC defined, REORG PLUS gathers statistics for the encoded values from the FIELDPROC. REORG PLUS does not gather statistics for column values that are stored in SYSIBM.SYSFIELDS.

- For multi-column keys, REORG PLUS uses only the first column to update column statistics in the DASD MANAGER PLUS tables.

- The TERMEXIT user exit provides REORG PLUS with user-defined variables that allow you to dynamically control processing of updates to the BMCHIST table and statistics tables during termination processing. You cannot use a TERMEXIT user exit to change BMCSTATS NO to YES. For information about the TERMEXIT user exit, see “Using TERMEXIT to control BMCHIST and statistics updates” on page 849.

- You can control the amount of table space sampling that REORG PLUS does for these statistics. For more information, see “TSSAMPLEPCT” on page 263.

- Due to differences in execution timing, options, and sampling, the statistics that REORG PLUS generates might differ slightly from the statistics produced by the BMCSTATS utility of DASD MANAGER PLUS.

**REPORT**

BMCSTATS REPORT tells REORG PLUS to gather statistics and produce the statistics report without updating the DASD MANAGER PLUS statistics tables.

**NOTE**

This option uses the same amount of processing overhead as specifying BMCSTATS YES.

**Restrictions**

The same restrictions that apply to BMCSTATS YES also apply to BMCSTATS REPORT. For information, see “YES” on page 259.
Additional considerations
Note the following additional information for BMCSTATS REPORT:

- For any columns that have a FIELDPROC defined, REORG PLUS gathers statistics for the encoded values from the FIELDPROC. REORG PLUS does not gather statistics for column values that are stored in SYSIBM.SYSFIELDS.

- For multi-column keys, REORG PLUS uses only the first column to update column statistics in the DASD MANAGER PLUS tables.

- You can control the amount of table space sampling that REORG PLUS does for these statistics. For more information, see “TSSAMPLEPCT” on page 263.

- Due to differences in execution timing, options, and sampling, the statistics that REORG PLUS generates might differ slightly from the statistics produced by the BMCSTATS utility of DASD MANAGER PLUS.

UPDATEDB2STATS

UPDATEDB2STATS tells REORG PLUS whether to update statistics in the DB2 catalog. DB2 uses these statistics to determine the access paths that the DB2 optimizer selects.

NO

UPDATEDB2STATS NO, which is the default, tells REORG PLUS not to update statistics in the DB2 catalog.

YES

UPDATEDB2STATS YES tells REORG PLUS to update statistics in the DB2 catalog. REORG PLUS uses the BMC Common Statistics component to update access path statistics.

NOTE
REORG PLUS passes the following options to the Common Statistics component. REORG PLUS does not enable you to change these options.

- UPDATEDB2 ALL
- HISTORY ALL
- FORCEROLLUP N
- KEYCARD Y

For information about how these options affect which statistics are updated in the DB2 catalog, see the DASD MANAGER PLUS documentation.
Statistics options

Restrictions
The following restrictions apply to UPDATEDB2STATS YES:

- REORG PLUS changes UPDATEDB2STATS YES to UPDATEDB2STATS NO when any of the following conditions exists:
  - Statistics collection fails in any reload or build task.
  - On restart, a participating table space partition was completely loaded, or any index partition was completely built in the original job.

- REORG PLUS ignores this option when you are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect.

- REORG PLUS does not collect statistics for the following objects and activities:
  - clone tables
  - nonpartitioned indexes when running a partial reorganization
  - activities that occur during the LOGAPPLY phase

Additional considerations
The following additional considerations apply to UPDATEDB2STATS YES:

- The TERMEXIT user exit provides REORG PLUS with user-defined variables that allow you to dynamically control processing of updates to the BMCHIST table and statistics tables during termination processing. You cannot use a TERMEXIT user exit to change UPDATEDB2STATS NO to YES.

  For information about the TERMEXIT user exit, see “Using TERMEXIT to control BMCHIST and statistics updates” on page 849.

- When invoking DSNUTILB, REORG PLUS passes this option to the STATISTICS option of the IBM DB2 REORG utility as STATISTICS TABLE (ALL) INDEX (ALL) REPORT YES UPDATE ALL.

- For multi-column keys, REORG PLUS uses only the first column to update statistics in the SYSIBM.SYSCOLUMNS table. If the cardinality for the first key column changes dramatically, BMC recommends that you run the IBM RUNSTATS utility to ensure that the DB2 optimizer selects the appropriate path.

- You can control the amount of table space sampling that REORG PLUS does for these statistics. For more information, see “TSSAMPLEPCT” on page 263.
TSSAMPLEPCT

This option applies to a table space reorganization only.

The TSSAMPLEPCT option enables you to specify a percentage of table space pages that you want REORG PLUS to sample when gathering statistics. The following values are valid:

- 1 through 50 tells REORG PLUS to sample the specified percentage of the table space pages.
- 100 tells REORG PLUS to read all table space pages instead of sampling.

NOTE

Values 51 through 99 are not valid.

Restrictions

REORG PLUS ignores the TSSAMPLEPCT option for either of the following types of reorganizations:

- a DSNUTILB reorganization
- an index reorganization

Specifying the default

You can specify the default for the TSSAMPLEPCT command option in your installation options module by using the TSSAMPLEPCT installation option (page 733). REORG PLUS was shipped with a default value of 100 for this option. The command option overrides the default that is in the installation options module.
Copy options for REORG TABLESPACE

The copy options indicate whether REORG PLUS is to create a copy of the table space and, if so, what type of copy. You can have REORG PLUS create an inline image copy as it reloads the table space, or a standard image copy or DSN1COPY-type copy after it reloads the table space. Additional options control other aspects of the copy process, including the ability to override the default ddnames of the data sets.

COPY

The COPY option tells REORG PLUS whether or not to produce a copy of the table space or selected partitions. Additional options tell REORG PLUS what type of copy to produce.

For a SHRLEVEL CHANGE reorganization, REORG PLUS creates a DB2 image copy of a nonpartitioned table space, all partitions of a partitioned table space, or selected partitions of a partitioned table space after the reorganization. For this type of reorganization, COPY YES is required and will be set even if you do not specify the COPY option or you specify COPY NO.

NO

COPY NO, which is the default (except when you use SHRLEVEL CHANGE), tells REORG PLUS not to make a copy of the table space or selected partitions. After the reorganization, REORG PLUS sets the COPY pending status and completes with return code 4.

Reorganization of an object in REORP status requires registered image copies. If you specify COPY NO, REORG PLUS changes the option to COPY YES and issues message BMC51218I.
YES

If you specify COPY YES, REORG PLUS creates a DB2 image copy or DSN1COPY-type copy of a nonpartitioned table space, all partitions of a partitioned table space, or selected partitions of a partitioned table space. For partition-by-growth table spaces, REORG PLUS also copies partitions that it or DB2 adds during the reorganization, as described in “Partition-by-growth table spaces” on page 125.

REORG PLUS creates all copies except inline image copies with system pages at the beginning of the data set (in the same way that the IBM DB2 COPY utility creates image copies when you specify SYSTEMPAGES YES).

DSNUTILB reorganization jobs
ACTIVE YES must be specified for at least the primary copy data set (DDTYPE LOCPFCPY). Otherwise, REORG PLUS terminates. All copy data sets for a DSNUTILB reorganization are dynamically allocated. REORG PLUS ignores any copy data set allocations in your JCL.

Additional considerations
The following considerations apply to COPY YES:

- For information about allocating data sets for your copies, including copy data set requirements, see “Copy data sets” on page 352.

- If you are running a SHRLEVEL CHANGE reorganization, you might need to make an incremental copy for each corresponding full copy specified, depending on the ICTYPE specification (see “ICTYPE” on page 267). For information about SHRLEVEL CHANGE and the incremental copy data sets, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 653.

- If the table space was in COPY pending status before the reorganization, REORG PLUS resets the COPY pending status unless you specify REGISTER NONE.

- The difference between an image copy and a DSN1COPY-type copy is one of registration: an image copy is registered, but a DSN1COPY is not.

INLINE
The INLINE option tells REORG PLUS whether to make an inline image copy as it reloads the table space. When invoking DSNUTILB, REORG PLUS passes INLINE to the IBM DB2 REORG utility, which creates inline, single data set copies.

Specifying the default
You can specify the default for the INLINE command option in your installation options module by using the INLINECP installation option (page 700). REORG PLUS was shipped with a default value of YES for this option. The command option, when used with COPY YES, overrides the default that is in the installation options module.
YES. INLINE YES tells REORG PLUS to create an inline image copy as it reloads the table space rather than after. Therefore, specifying INLINE YES can reduce the elapsed time of your reorganization job.

The inline image copies that REORG PLUS creates have the same characteristics as inline copies that the DB2 REORG utility creates.

NOTE
If you specify INLINE YES for a SHRLEVEL CHANGE reorganization, REORG PLUS uses more memory than it would with INLINE NO because it keeps all of the image copy data sets open until the LOGFINAL phase completes processing.

Other than in the exception situation noted, if any of the following conditions exists with your reorganization job, REORG PLUS behaves as if you specified INLINE NO, regardless of the value you specified:

- The copy data sets are on a stacked tape.
- The size of the table space page is greater than 4 KB, you are reorganizing multiple partitions, and you have a single image copy data set.
- An exception to this condition exists for partition-by-growth table spaces. REORG PLUS can create inline image copies of partition-by-growth table spaces.
- You specify SHRLEVEL CHANGE and ICTYPE INCREMENTAL.
- For a partitioned table space, you restart a SHRLEVEL NONE or REFERENCE reorganization, you have a single image copy data set, and at least one (but not all) of the partitions was reloaded before the failure.

NO. INLINE NO tells REORG PLUS not to create an inline image copy as it reloads the table space, but to create a DB2 image copy or DSN1COPY-type copy after it reloads the table space.

Restrictions
REORG PLUS terminates if you also specify ICTYPE UPDATE and you attempt to create a single image copy for multiple partitions, as in the following scenarios:

- You allocate a single copy data set in your JCL but specify multiple partitions.
- You specify COPYLVL FULL and COPYSUBSET=YES.
REGISTER
COPY YES REGISTER tells REORG PLUS to register some, none, or all of the full and incremental copies with DB2 in the SYSIBM.SYSCOPY table.

Restrictions
Note the following restrictions on registering copies:

- When invoking DSNUTILB, REORG PLUS ignores this option.
- REORG PLUS does not register any copies if any one is rejected because of a duplicate entry in the SYSIBM.SYSCOPY table.
- For a SHRLEVEL CHANGE reorganization, REORG PLUS forces REGISTER ALL (even if you specify NONE or a list of ddnames) and issues message BMC53008I.

  ALL. REGISTER ALL registers all copies requested in the JCL by ddname.

  NONE. REGISTER NONE does not register any of the copies with DB2 and does not leave the object in COPY pending status.

  ddname. This option enables you to specify by ddname the copy data sets to register. REORG PLUS registers only the copies in the data sets specified by this option. The ddnames that you specify in this option must be present in your JCL.

ICTYPE
This option applies to SHRLEVEL CHANGE only.

Based on the value of this option, REORG PLUS updates the full image copy data sets or creates incremental image copy data sets. You will achieve better performance and reduce the amount of time during which application updates are prevented if REORG PLUS does not have to create incremental copies during the LOGFINAL phase. BMC recommends that you use ICTYPE AUTO to allow REORG PLUS to determine which type of copy is best.

REORG PLUS does not support updating SMS-managed striped data sets for the copy data sets unless you specify ICTYPE AUTO or UPDATE and INLINE YES (or INLINECP=YES).

If your installation allows SMS data sets to go to tape for the SMS classes that you specified for dynamically allocated copy data sets, you must specify ICTYPE INCREMENTAL if all of the following statements are true:

- You specify SHRLEVEL CHANGE.
- You specify SMS YES.
- The value of the SMSUNIT command or installation option is NO.
- The value of the INLINE command option or INLINECP installation option is NO.
If you specify SHRLEVEL CHANGE and ICTYPE INCREMENTAL, REORG PLUS behaves as if you specified INLINE NO, regardless of the value that you specified for INLINE.

**Specifying the default**

You can specify the default for the ICTYPE command option in your installation options module by using the ICTYPE installation option (page 697). REORG PLUS was shipped with a default value of AUTO for this option. The command option, when used with COPY YES, overrides the default that is in the installation options module.

**AUTO.** This value tells REORG PLUS to determine which type of copy is best.

REORG PLUS updates the full copy data sets if *either* of the following statements is true:

- The value of the INLINE command option or INLINECP installation option is YES.
- All of the full copy data sets are on DASD, and one full copy data set exists for each partition that you are reorganizing.

If neither condition is met, REORG PLUS creates incremental image copies. You must define or dynamically allocate these incremental copy data sets.

When invoking DSNUTILB, REORG PLUS ignores this option and DSNUTILB uses ICTYPE UPDATE.

**UPDATE.** If you specify UPDATE, REORG PLUS always updates the full copy data sets. If you also specify INLINE YES on the command or INLINECP=YES in the installation option, REORG PLUS appends the updated pages to the full copy data sets. When the value of INLINE or INLINECP is YES, the copy data sets can be on tape.

If you also specify INLINE NO or INLINECP=NO, REORG PLUS updates the full copy data sets in place.

**INCREMENTAL.** If you specify INCREMENTAL, REORG PLUS creates the incremental copy data sets. You must define or dynamically allocate these data sets. If the data sets do not exist, REORG PLUS terminates.

When invoking DSNUTILB, REORG PLUS changes ICTYPE INCREMENTAL to ICTYPE UPDATE.
COPYLVL

REORG PLUS uses the COPYLVL option only when dynamic allocation is active for copy data sets. COPYLVL tells REORG PLUS how to assign the dynamically allocated image copy data sets in either of the following cases:

- You are reorganizing all partitions of a partitioned table space.
- You are reorganizing a subset of contiguous partitions by using the PART command option and you specify YES for the COPYSUBSET installation option.

Specifying COPYLVL PART can improve performance when reorganizing partitioned objects. However, if you are reorganizing a large number of partitions, consider specifying COPYLVL FULL to avoid encountering memory or data set allocation restrictions. Whether the number of partitions that you have is considered to be a large number of partitions depends on your environment. For more information regarding full copy data sets, see “Copy data sets” on page 352. For more information regarding incremental copy data sets, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 653.

Specifying the default
You can specify the default for the COPYLVL command option in your installation options module by using the COPYLVL installation option (page 680). REORG PLUS was shipped with a default value of PART for this option. The command option overrides the default that is in the installation options module.

PART

COPYLVL PART tells REORG PLUS to allocate individual full copy data sets and incremental copy data sets (if created) for each partition that you are reorganizing.

Restriction
REORG PLUS changes COPYLVL PART to COPYLVL FULL when invoking DSNUTILB.

Additional considerations
The following additional considerations apply to COPYLVL PART:

- If you specify a tape device for the UNIT option, REORG PLUS allocates a tape unit for each partition.
If you are using a GDG name, each partition must have a different GDG base.

If you are reorganizing a table space that contains more than 99 partitions, use the COPYDDN option (page 271) to specify a ddname prefix for the copy data sets that results in eight characters or less after REORG PLUS appends the highest partition number. If you are also making remote copies, use the RECOVERYDDN option (page 273) in the same way.

FULL

COPYLVL FULL tells REORG PLUS to allocate a single full copy data set to contain all of the partitions that you are reorganizing. If incremental copy data sets are created, REORG PLUS also allocates a single incremental copy data set to contain all of the partitions that you are reorganizing.

Restrictions
REORG PLUS changes COPYLVL FULL to COPYLVL PART when either of the following conditions exists:

- You specify a subset of partitions, but the value of the COPYSUBSET installation option is NO.
- The value of the COPYSUBSET installation option is YES and you specify a subset of partitions, but they are not contiguous.
COPYDDN

COPYDDN allows you to override the default ddnames or ddname prefixes of the local copy data sets that were specified in your installation options. The ddnames correspond to the data sets that receive a full image copy or DSN1COPY-type copy of the table space or partitions that you are reorganizing.

If you specify $ddname2$ for this option, you must either allocate this data set in your JCL, or dynamic allocation must be active for both the primary and backup copy data sets. If you specify $ddname2$ and dynamic allocation is active for only the primary copy data set, REORG PLUS terminates because it expects a second copy data set.

If you are registering the copies, $ddname1$ will be the DB2 local primary and $ddname2$ will be the local backup. For information about specifying and using these data sets, see “Copy data sets” on page 352.

If you use this command option to override the default name in the installation options module, you must also change the ddnames in your JCL.

When REORG PLUS invokes DSNUTILB (and you specify COPY YES), REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

Partition-level copies
If you are making partition-level copies, specify only the ddname prefix (no $nn$) in this option. If you are dynamically allocating copy data sets (and you specify COPYLVL PART), REORG PLUS appends the partition number to the ddname prefix. Specify a prefix that results in eight characters or less after REORG PLUS appends the highest partition number. For more information, see “Specifying ddname prefixes” on page 93.

If you are not dynamically allocating copy data sets, the ddname that you specify in the JCL must have the partition number $nn$ appended to this prefix. The length of $nn$ can be from one to seven characters, depending on the length of the ddname prefix. The maximum length of the ddname with the prefix must be eight bytes or less. For example, if you are reorganizing partition 157, you could specify $ddname1$ as BCOPY, and specify BCOPY157 in your JCL.
Dynamic allocation
If dynamic allocation is active and you specify more than one ddname prefix for
dynamic allocation, the prefix for each ddname must be different enough for REORG
PLUS to differentiate one prefix from another. To be different enough, if these
prefixes are different only because one prefix has additional trailing bytes, then these
trailing bytes must contain at least one nonnumeric byte. For example, the first set of
prefixes that follow is sufficiently different, but the second set is not:

- acceptable set:
  - BMCRD
  - BMCRDWK

- not acceptable set:
  - BMCRD
  - BMCRD11

Specifying the default
You can specify the default for the COPYDDN command option in your installation
options module by using the COPYDDN installation option (page 680). REORG
PLUS was shipped with a default value of (BCPY, BCPZ) for this option. The
command option overrides the default that is in the installation options module.
RECOVERYDDN allows you to override the default ddnames or ddname prefixes that were specified in your installation options for remote copy data sets. The ddnames correspond to the data sets that receive a full image copy or DSN1COPY-type copy of the table space or partitions that you are reorganizing.

If you specify `ddname2` for this option, you must either allocate this data set in your JCL, or dynamic allocation must be active for both the primary and backup remote copy data sets. If you specify `ddname2` and dynamic allocation is active for only the primary copy data set, REORG PLUS terminates because it expects a second copy data set.

If you are registering the copies, `ddname1` will be the DB2 remote primary copy, and `ddname2` will be the remote backup copy. For information about specifying and using these data sets, see “Copy data sets” on page 352.

If you use this command option to override the default name in the installation options module, you must also change the ddnames in your JCL.

When REORG PLUS invokes DSNUTILB (and you specify COPY YES), REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

**Partition-level copies**

If you are making partition-level copies, specify only the ddname prefix (no `nn`) in this option. If you are dynamically allocating copy data sets (and you specify COPYLVL PART), REORG PLUS appends the partition number to the ddname prefix. Specify a prefix that results in eight characters or less after REORG PLUS appends the highest partition number. For more information, see “Specifying ddname prefixes” on page 93.

If you are not dynamically allocating copy data sets, the ddname that you specify in the JCL must have the partition number `nn` appended to this prefix. The length of `nn` can be from one to seven characters, depending on the length of the ddname prefix. The maximum length of the ddname with the prefix must be eight bytes or less. For example, if you are reorganizing partition 203, you could specify `ddname1` as `BMCRY`, and specify `BMCRY203` in your JCL.
Dynamic allocation
If dynamic allocation is active and you specify more than one ddname prefix for
dynamic allocation, the prefix for each ddname must be different enough for REORG
PLUS to differentiate one prefix from another. That is, if the prefixes differ only
because one prefix has additional trailing bytes, the trailing bytes must contain at
least one nonnumeric byte. For example, the first set of prefixes that follow is
sufficiently different, but the second set is not:

- acceptable set:
  
  ```
  BMCRD
  BMCRDWK
  ```

- not acceptable set:
  
  ```
  BMCRD
  BMCRD11
  ```

Specifying the default
You can specify the default for the RECOVERYDDN command option in your
installation options module by using the RCVYDDN installation option (page 716).
REORG PLUS was shipped with a default value of (BRCY, BRCZ) for this option. The
command option overrides the default that is in the installation options module.
ICDDN

This option applies to SHRLEVEL CHANGE only.

ICDDN allows you to override the default ddnames or ddname prefixes of the local incremental copy data sets that were specified in your installation options. The ddnames correspond to the data sets that receive an incremental image copy of the table space or partitions that you are reorganizing.

When you register the copies, \textsl{ddname1} will be the DB2 local primary and \textsl{ddname2} will be the local backup. For specification guidelines and detailed information about the use of the copy data sets, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 653.

If you use this command option to override the default name in the installation options module, you must also change the ddnames in your JCL.

\textit{Restriction}
When invoking DSNUTILB, REORG PLUS ignores this option.

\textit{Partition-level copies}
If you are making partition-level copies, specify only the ddname prefix (no \textit{nn}) in this option. If you are dynamically allocating copy data sets (and you specify COPYLVL PART), REORG PLUS appends the partition number to the ddname prefix. Specify a prefix that results in eight characters or less after REORG PLUS appends the highest partition number. For more information, see “Specifying ddname prefixes” on page 93.

If you are not dynamically allocating copy data sets, the ddname that you specify in the JCL must have the partition number \textit{nn} appended to this prefix. The length of \textit{nn} can be from one to seven characters, depending on the length of the ddname prefix. The maximum length of the ddname with the prefix must be eight bytes or less.
**Dynamic allocation**
If dynamic allocation is active and you specify more than one ddname prefix for dynamic allocation, the prefix for each ddname must be different enough for REORG PLUS to differentiate one prefix from another. That is, if the prefixes differ only because one prefix has additional trailing bytes, the trailing bytes must contain at least one nonnumeric byte. For example, the first set of prefixes that follow is sufficiently different, but the second set is not:

- acceptable set:

  | BMCRD |
  | BMCRDWK |

- not acceptable set:

  | BMCRD |
  | BMCRD11 |

**Specifying the default**
You can specify the default for the ICDDN command option in your installation options module by using the ICDDN installation option (page 696). REORG PLUS was shipped with a default value of (BICY, BICZ) for this option. The command option overrides the default that is in the installation options module.
RECOVERYICDDN

This option applies to SHRLEVEL CHANGE only.

RECOVERYICDDN allows you to override the installation default ddnames corresponding to the data sets that receive a remote incremental image copy or DSN1COPY-type copy of the table space or partitions that you are reorganizing.

When you register the copies, \textit{ddname1} will be the DB2 remote primary and \textit{ddname2} will be the remote backup. For specification guidelines and detailed information about the use of the copy data sets, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 653.

If you use this command option to override the default name in the installation options module, you must also change the ddnames in your JCL.

\textbf{Restriction}

When invoking DSNUTILB, REORG PLUS ignores this option.

\textbf{Partition-level copies}

If you are making partition-level copies, specify only the ddname prefix (no \textit{nn}) in this option. If you are dynamically allocating copy data sets (and you specify COPYLVL PART), REORG PLUS appends the partition number to the ddname prefix. Specify a prefix that results in eight characters or less after REORG PLUS appends the highest partition number. For more information, see “Specifying ddname prefixes” on page 93.

If you are not dynamically allocating copy data sets, the ddname that you specify in the JCL must have the partition number \textit{nn} appended to this prefix. The length of \textit{nn} can be from one to seven characters, depending on the length of the ddname prefix. The maximum length of the ddname with the prefix must be eight bytes or less.
**Dynamic allocation**

If dynamic data set allocation is active and you specify more than one ddname prefix for dynamic allocation, the prefix for each ddname must be different enough for REORG PLUS to differentiate one prefix from another. That is, if the prefixes differ only because one prefix has additional trailing bytes, the trailing bytes must contain at least one nonnumeric byte. For example, the first set of prefixes that follow is sufficiently different, but the second set is not:

- acceptable set:
  - BMCRD
  - BMCRDWK

- not acceptable set:
  - BMCRD
  - BMCRD11

**Specifying the default**

You can specify the default for the RECOVERYICDDN command option in your installation options module by using the RCVICDDN installation option (page 716). REORG PLUS was shipped with a default value of (BIRY, BIRZ) for this option. The command option overrides the default that is in the installation options module.
Selective unload and update options for REORG TABLESPACE

The SELECT and DELETE options allow you to specify conditions that must be met in order for a row to be unloaded and reloaded during the reorganization. The UPDATE option allows you to reset column values to a constant.

Restrictions
These options do not apply to the following jobs:

- index-only reorganizations
- DSNUTILB jobs
- XML table space reorganizations
- LOB table space reorganizations
- table space reorganizations of table spaces that contain a LOB column

NOTE
You can specify UPDATE unless either of the following conditions exists:

- The UPDATE statement attempts to update a LOB column.
- An associated WHERE clause references a LOB column.

- catalog table space reorganizations
- reorganizations of table spaces that include archive-enabled tables
- reorganizations of table spaces that contain system period temporal tables that are enabled for versioning
DBCS support

REORG PLUS supports the use of the double-byte character set (DBCS). You can use DBCS characters in DB2 identifiers, such as table or column names, or in the constants specified in your SELECT, DELETE, or UPDATE options. You can also include DBCS characters in DB2 object names, such as database or table space names.
Data translation of command constants

REORG PLUS translates character constants from EBCDIC (using the DB2 installation default EBCDIC SBCS CCSID) to the encoding scheme of the table. REORG PLUS translates the following character constants in order to compare constants and row data:

- predicate block constants
- IN constants

REORG PLUS translates character constants that are placed in the table data from EBCDIC (using the DB2 installation default EBCDIC SBCS CCSID) to the output table encoding scheme. REORG PLUS translates SET values on the UPDATE option for output.

**IMPLICIT_TZ**

This option does not apply to an index-only reorganization.

The IMPLICIT_TZ option enables you to specify the time zone that you want REORG PLUS to use when any of the following conditions exists on a WHERE or SET clause:

- You specify CURRENT TIMESTAMP WITH TIME ZONE.
- You specify a constant for a TIMESTAMP WITH TIME ZONE column, but the constant that you specify does not include a time zone.

REORG PLUS adds the time zone that you specify to the constant.

Specify a time zone, within quotation marks, as the difference between local time and Coordinated Universal Time (UTC). You can specify a value from -12:59 through +14:00. If you do not specify a value for the IMPLICIT_TZ option, REORG PLUS uses the DSNHDECP IMPLICIT_TIMEZONE value.

Any time zone that you specify in a SELECT, DELETE, or UPDATE statement overrides the value of this option for that statement.

**Restrictions**

When invoking DSNUTILB, REORG PLUS ignores this option.
This option does not apply to an index-only reorganization.

The SELECT * FROM option tells REORG PLUS to select rows that match the specified WHERE clause conditions from the named table. REORG PLUS unloads the selected rows and reloads them to the table space. The table name specifies the table to which the select criterion applies. If you do not specify the creator name, REORG PLUS uses the authorization ID of the user running the reorganization.

Restrictions
REORG PLUS terminates when you specify this option for the following jobs:

- index-only reorganizations
- DSNUTILB jobs
- XML table space reorganizations
- LOB table space reorganizations
- reorganizations of table spaces that contain a LOB column
- catalog table space reorganizations
- reorganizations of table spaces that include archive-enabled tables
- reorganizations of table spaces that contain system period temporal tables that are enabled for versioning

For restrictions on the WHERE clause of your SELECT statement, see “WHERE” on page 288.

Additional considerations
Note the following additional information about the SELECT * FROM option:

- You can specify only one SELECT or one DELETE clause per table.

- REORG PLUS deletes rows that are not selected and optionally writes them to the SYSARC archive data set.

For specification guidelines and detailed information about the use of the SYSARC data set, see “SYSARC data set” on page 370.
If you are running a SHRLEVEL CHANGE reorganization, see “SHRLEVEL CHANGE considerations for using SELECT or DELETE” on page 649.

**WARNING**

If the rows that you did not select are in a table with a primary key, REORG PLUS does not guarantee or maintain any referential integrity of the primary keys after those rows are deleted and does not set the CHECK pending status. You might need to run either the BMC CHECK PLUS for DB2 product or the IBM DB2 CHECK DATA utility on dependent tables to restore referential integrity, or run REORG PLUS on dependent tables to delete dependent rows.

**WHERE**

The WHERE clause enables you to specify which rows are to be selected. For a detailed description of the WHERE condition, see page 288.

**NOTE**

When you use WHERE on a partial reorganization, REORG PLUS processes only the specified partitions. If the WHERE condition matches rows in any of the partitions that are not being reorganized, those rows are unaffected. You do not need to specify predicates to limit the WHERE condition to only partitions that you are reorganizing.
DELETE FROM

This option does not apply to an index-only reorganization.

The DELETE FROM option tells REORG PLUS to delete the rows that match the specified WHERE clause conditions from the named table. Rows that REORG PLUS does not delete are unloaded and used to reload the table space. The table name specifies the table to which the delete criterion applies. If you do not specify the creator name, REORG PLUS uses the authorization ID of the user running the reorganization.

Restrictions
REORG PLUS terminates when you specify this option for the following jobs:

- index-only reorganizations
- DSNUTILB jobs
- XML table space reorganizations
- LOB table space reorganizations
- reorganizations of table spaces that contain a LOB column
- catalog table space reorganizations
- reorganizations of table spaces that include archive-enabled tables
- reorganizations of table spaces that contain system period temporal tables that are enabled for versioning

For restrictions on the WHERE clause of your DELETE statement, see “WHERE” on page 288.

Additional considerations
Note the following additional information about the DELETE FROM option:

- You can specify only one SELECT or one DELETE clause per table.
- REORG PLUS optionally writes the deleted rows to the SYSARC archive data set.

For specification guidelines and detailed information about the use of the SYSARC data set, see “SYSARC data set” on page 370.

- If you are running a SHRLEVEL CHANGE reorganization, see “SHRLEVEL CHANGE considerations for using SELECT or DELETE” on page 649.
**WARNING**

If you are deleting rows from a table with a primary key, REORG PLUS does not guarantee or maintain any referential integrity of the primary keys and does not set the CHECK pending status. You might need to run either the BMC CHECK PLUS product or the IBM DB2 CHECK DATA utility on dependent tables to restore referential integrity, or run REORG PLUS on dependent tables to delete dependent rows.

**WHERE**

The WHERE clause enables you to specify which rows are to be deleted. For a detailed description of the WHERE condition, see page 288.

**NOTE**

When you use WHERE on a partial reorganization, REORG PLUS processes only the specified partitions. If the WHERE condition matches rows in any of the partitions that are not being reorganized, those rows are unaffected. You do not need to specify predicates to limit the WHERE condition to only partitions that you are reorganizing.
UPDATE tableName SET

You cannot use this option for an index-only or SHRLEVEL CHANGE reorganization.

UPDATE tableName SET enables you to reset column values to a constant. The table name specifies the table to which the update criterion applies. If you do not specify the creator name, REORG PLUS uses the authorization ID of the user running the reorganization. SET specifies that values in the named columns that match the specified WHERE clause will be set to one of the values listed in Table 40.

Table 40 Values for SET

<table>
<thead>
<tr>
<th>SET values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td>updates the column with the specified value</td>
</tr>
<tr>
<td></td>
<td>Observe the rules for constants in Table 42 on page 291. The associated</td>
</tr>
<tr>
<td></td>
<td>constant must match the data type of the column (that is, numeric to numeric,</td>
</tr>
<tr>
<td></td>
<td>string to string, and date/time to date/time).</td>
</tr>
<tr>
<td>NULL</td>
<td>resets the column values to null</td>
</tr>
<tr>
<td>CURRENT DATE</td>
<td>resets the column values to the current date</td>
</tr>
<tr>
<td>CURRENT TIME</td>
<td>resets the column values to the current time</td>
</tr>
<tr>
<td>CURRENT TIMESTAMP</td>
<td>resets the column values to the current timestamp</td>
</tr>
<tr>
<td>CURRENT TIMESTAMP WITH TIME ZONE</td>
<td>resets the column values to the current timestamp, including the time zone from the IMPLICIT_TZ option (see page 281)</td>
</tr>
</tbody>
</table>

Restrictions
The following restrictions apply to the UPDATE statement:

- REORG PLUS terminates when you specify UPDATE for the following jobs:
  - reorganizations that invoke DSNUTILB
  - SHRLEVEL CHANGE jobs
Selective unload and update options for REORG TABLESPACE

— index-only reorganizations
— XML table space reorganizations
— LOB table space reorganizations
— catalog table space reorganizations
— reorganizations of table spaces that include archive-enabled tables
— reorganizations of table spaces that contain system period temporal tables that are enabled for versioning

■ REORG PLUS terminates when you attempt to use the UPDATE statement to modify the following keys or columns:

— partitioning keys
— unique index keys
— any key if you are performing a partial reorganization
— row ID columns
— LOB or XML columns
— document ID columns
— floating-point columns
— decimal floating-point columns
— identity columns
— binary string type columns
— columns that are defined with a FIELDPROC
— row change timestamp columns

For restrictions on the WHERE clause of your UPDATE statement, see “WHERE” on page 288.

Additional considerations
Note the following additional information about the UPDATE statement:

■ You can specify only one UPDATE per table.
■ During processing, REORG PLUS updates any row change timestamp column when it updates rows.

WHERE

The WHERE clause enables you to specify which rows are to be updated. For a detailed description of the WHERE condition, see page 288.

NOTE

If you are updating a foreign key column, REORG PLUS does not detect referential violations and does not set the CHECK pending status. You might need to add rows to the primary tables to restore referential integrity.
The WHERE clause enables you to specify the condition that must be true for a row to be selected, deleted, or updated. In the WHERE condition block, a condition is a combination of predicates using AND, OR, NOT, and ( ) operators. Table 41 shows the result of the WHERE clause when you use the AND, OR, or NOT operator. Predicates are described on page 290.

Table 41  Truth table for AND, OR, NOT

<table>
<thead>
<tr>
<th>Result of predicate</th>
<th>Result of WHERE condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>q</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

Predicates are evaluated left to right, with the following operator precedence:

- ( )
- NOT
- AND
- OR

After the resulting value of the condition is determined, evaluation of any more predicates stops. If the result is true, the row is selected, deleted, or updated. If the result is false, the row is not selected, deleted, or updated.
Restrictions
REORG PLUS terminates when you specify any of the following types of columns or constants on your WHERE clause:

- XML columns
- LOB columns
- floating-point columns
- decimal floating-point columns
- row ID columns
- binary string constants or columns
- columns that are defined with a FIELDPROC
A predicate is a comparison of a single column value to a constant or list of constants. The column name identifies a column of the named table that is used in the comparison. You can specify the following comparison operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>less than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>not equal</td>
</tr>
<tr>
<td>=</td>
<td>equal</td>
</tr>
<tr>
<td>!=</td>
<td>not equal</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
</tr>
<tr>
<td>IN</td>
<td>equal to any</td>
</tr>
<tr>
<td>NOT IN</td>
<td>not equal to all</td>
</tr>
</tbody>
</table>

The constant specifies a value to be compared to the column value. REORG PLUS then evaluates the result based on the preceding relative operator. Observe the rules for constants in Table 42 on page 291. For the comparison, the associated constant must match the data type of the column (that is, numeric to numeric, string to string, and date/time to date/time).

You should avoid nonstandard comparisons. For more information, see rules for constants in the IBM DB2 SQL Reference.
If you specify a list of constants, you can improve performance by specifying the constants in ascending sequential order. When you specify the list in this order, you save processing time because REORG PLUS does not sort the list.

**Table 42 Rules for constants (part 1 of 2)**

<table>
<thead>
<tr>
<th>Type of constant</th>
<th>Additional details</th>
</tr>
</thead>
</table>
| integer          | identical to DB2 SQL integer constants  
|                  | The constant must be within the range of the column’s data type. |
| decimal          | identical to DB2 SQL decimal constants  
|                  | The constant must be within the range of the column’s data type.  
|                  | **Note:** To determine whether your site has set the decimal point to a comma, check the DB2 subsystem DSNHDECP setting. To enable REORG PLUS to distinguish comma decimal points from comma separators, include a space after any comma that you use as a separator.  
|                  | Example of using the comma as a decimal point:  
|                  | ```sql  
|                  | SELECT * FROM owner.table  
|                  | WHERE SALARY > 100.00  
|                  | ```  
|                  | Example of using the comma as both a decimal point and a separator:  
|                  | ```sql  
|                  | SELECT * FROM owner.table  
|                  | WHERE SALARY IN (100,00, 200,00, 300,00)  
|                  | ```  
| character string | identical to DB2 SQL strings  
|                  | Use ‘’ to denote an empty string.  
|                  | **Note:** REORG PLUS does not validate character data against CCSIDs. |
| hexadecimal strings | identical to DB2 SQL hexadecimal strings, except that string length is limited to 256 bytes  
|                  | **Notes:**  
|                  | - REORG PLUS does not validate hexadecimal string data against CCSIDs.  
|                  | - REORG PLUS does not support the hexadecimal string constants BX’xxxx’, UX’xxxx’, or GX’xxxx’. |
| graphic string   | limited to 256 bytes |
| date/time string | with the following exceptions, identical to DB2 SQL date/time strings  
|                  | **Exceptions:**  
|                  | - Use only ISO, USA, EUR, and JIS formats.  
|                  | - Precision on a timestamp constant must be less than or equal to the precision defined in the column. |
|                  | **Note:** If you specify a time zone on a timestamp constant, this value overrides any value that you specify for the IMPLICIT_TZ option. |
Table 42 Rules for constants (part 2 of 2)

<table>
<thead>
<tr>
<th>Type of constant</th>
<th>Additional details</th>
</tr>
</thead>
<tbody>
<tr>
<td>floating-point</td>
<td>not supported</td>
</tr>
<tr>
<td>decimal floating point</td>
<td></td>
</tr>
<tr>
<td>binary string</td>
<td></td>
</tr>
<tr>
<td>LOB</td>
<td></td>
</tr>
<tr>
<td>row ID</td>
<td></td>
</tr>
</tbody>
</table>

**NULL**

You can specify the keyword NULL as a constant for columns that can be set to null. NULL is generally used with the =, <>, ¬=, IN, and NOT IN operators, but you can specify it with any of the allowed operators.

The null value is always higher than all other values. To bypass columns that allow nulls, specify AND ¬ = NULL.

**CURRENT DATE – labeledDuration**

The CURRENT DATE option describes the current date or, optionally, describes the current date minus a duration in days, months, or years. The rules for date/time arithmetic using labeled durations are identical to the DB2 SQL rules stated in the IBM DB2 SQL Reference manual. The CURRENT DATE is taken from the local time-of-day clock during the UTILINIT phase.

**CURRENT TIMESTAMP WITH TIME ZONE – labeledDuration**

The CURRENT TIMESTAMP option describes the current timestamp. Optionally, you can include one of the following specifications with CURRENT TIMESTAMP:

- WITH TIME ZONE to include the time zone (from the IMPLICIT_TZ option, described on page 281) in the timestamp

- a duration (preceded by a minus sign) in days, months, or years, which results in the current timestamp minus that duration. The rules for date/time arithmetic using labeled durations are identical to the DB2 SQL rules stated in the IBM DB2 SQL Reference manual.

The CURRENT TIMESTAMP is taken from the local time-of-day clock during the UTILINIT phase.
Dynamic allocation options

Dynamic allocation options allow you to have REORG PLUS optimally allocate the size and number of work files and copy data sets needed to process your reorganization. Dynamic allocation provides the following benefits:

- eliminates the need for you to include DD statements for the specified file types in your JCL
- eliminates the need for you to calculate data set allocation sizes
- automatically provides an optimal allocation for the highest possible multitasking level
- allows you to use symbolic variables and user-defined variables to assist in creating data set names

The following sections provide the syntax required for enabling and using dynamic data set allocation. For more information, see “Dynamic data set allocation” on page 90.

```
MAXTAPE integer
```

MAXTAPE

The MAXTAPE option allows you to specify a maximum number of tape devices that REORG PLUS will use when dynamically allocating data sets. For details regarding how REORG PLUS handles dynamic data set allocation when it reaches your MAXTAPE limit, see “Reaching the MAXTAPE limit during dynamic allocation” on page 95.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the MAXTAPE command option in your installation options module by using the MAXTAPE installation option (page 711). REORG PLUS was shipped with a default value of 3 for this option. The command option overrides the default that is in the installation options module.
This option tells REORG PLUS to dynamically allocate one or more types of data sets, or to override an active dynamic data set allocation value in the installation options module. The DD type block provides additional options that you can use to direct REORG PLUS in various aspects of the dynamic data set allocation process. You can repeat this DD type block for more than one data set type. Table 43 lists the keywords that you can specify for the data set types that you want to allocate dynamically.

### Table 43  DDTYPE keywords (part 1 of 2)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Applies specifications for dynamic allocation to</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLOAD</td>
<td>unload files (SYSREC)</td>
</tr>
<tr>
<td>WORK</td>
<td>index data sets (SYSUT1)</td>
</tr>
<tr>
<td>SORTWORK</td>
<td>sort work files (SORTWK)</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>archive file, or discard file for DSNUTILB jobs (SYSARC)</td>
</tr>
<tr>
<td>SYSPUNCH</td>
<td>LOAD control statement data set for DSNUTILB jobs (SYSPUNCH)</td>
</tr>
</tbody>
</table>

**Note:** REORG PLUS ignores this DDTYPE option when invoking DSNUTILB to reorganize a LOB table space.
Specifying the default
You can also specify the DD type in your installation options module by using the
DDTYPE installation option (page 741). Any dynamic data set allocation options that
you specify with the DDTYPE command option override the defaults established at
installation for the same DDTYPE.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Applies specifications for dynamic allocation to</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCPFCPY</td>
<td>local primary full copy data set (BCPY)</td>
</tr>
<tr>
<td>LOCPICPY</td>
<td>local primary incremental copy data set (BICY)</td>
</tr>
<tr>
<td>LOCBFCPY</td>
<td>local backup full copy data set (BCPZ)</td>
</tr>
<tr>
<td>LOCBICPY</td>
<td>local backup incremental copy data set (BICZ)</td>
</tr>
<tr>
<td>REMPFCPY</td>
<td>remote primary full copy data set (BRCY)</td>
</tr>
<tr>
<td>REMPICPY</td>
<td>remote primary incremental copy data set (BIRY)</td>
</tr>
<tr>
<td>REMBFCPY</td>
<td>remote backup full copy data set (BRCZ)</td>
</tr>
<tr>
<td>REMBICPY</td>
<td>remote backup incremental copy data set (BIRZ)</td>
</tr>
</tbody>
</table>
The ACTIVE option allows you to specify whether you want dynamic allocation active for the specified data set type. If you specify the ACTIVE keyword without a value, REORG PLUS assumes ACTIVE YES.

**DSNUTILB reorganization jobs**
For a DSNUTILB reorganization, you must specify ACTIVE YES for the following DDTYPEs:

- all work file DDTYPEs that the reorganization job requires
- if you specify COPY YES, the LOCPFCPY DDTYPE (and other copy DDTYPEs if you need them)

Additionally, if the IBM DB2 REORG utility job requires data sets for discarded rows and for LOAD control statements for those discarded rows, you must enable dynamic allocation for the ARCHIVE and SYSPUNCH DDTYPEs. If the DB2 REORG utility needs these data sets and they are not dynamically allocated in your REORG PLUS job, the job terminates.

If you specify any of these data sets in your JCL, REORG PLUS ignores them, regardless of your IFALLOC specification.

**NOTE**
All copy data sets for a DSNUTILB reorganization are dynamically allocated even if you specify ACTIVE YES for only the primary local copy data set.

**Specifying the default**
You can specify the default for the ACTIVE command option in your installation options module by using the ACTIVE installation option (page 742). REORG PLUS was shipped with a default value of NO or YES for this option, depending on the DDTYPE value. The command option overrides the default that is in the installation options module.

When running in a worklist environment, REORG PLUS ignores the ACTIVE option in your installation options module. REORG PLUS dynamically allocates your data sets only if the invoking product (DASD MANAGER PLUS for DB2, CATALOG MANAGER for DB2, or CHANGE MANAGER for DB2) supplies the ACTIVE YES syntax.
YES

YES tells REORG PLUS to activate dynamic allocation for the specified data set type. ACTIVE YES must be in effect for REORG PLUS to use the remaining dynamic allocation options.

NOTE

When determining whether to dynamically allocate a data set, REORG PLUS takes into account the values of both the ACTIVE and IFALLOC options for that data set type.

NO

NO tells REORG PLUS to deactivate dynamic allocation for the specified data set type. Use this option if you set dynamic data set allocation active in your installation options but you do not want REORG PLUS to allocate the specified data sets dynamically for a particular processing.

IFALLOC

The IFALLOC option tells REORG PLUS to either free the data sets specified in your JCL or use them and, if necessary, allocate additional data sets.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the IFALLOC command option in your installation options module by using the IFALLOC installation option (page 752). REORG PLUS was shipped with a default value of USE for this option. The command option overrides the default that is in the installation options module.

USE

Specifying USE tells REORG PLUS to use the data sets that you allocated in the JCL. If the number of SYSREC, SYSUT1, SORTWK, full copy, or incremental copy files specified in your JCL is insufficient for processing, REORG PLUS (or BMCSORT in the case of sort work files) dynamically allocates the additional files that your job needs. REORG PLUS uses both the dynamically allocated data sets and those that you specify in your JCL. REORG PLUS allocates only additional files, not additional space for a single file.
Dynamic allocation options

When you specify IFALLOC USE, the ddnames that you specify in the JCL must be eight bytes long and must match the corresponding ddnames displayed in one of the following reports in the SYSPRINT data set:

- **REORG PLUS DASD REQUIREMENT ESTIMATES** (messages BMC51260I through BMC51263I)
  
  REORG PLUS issues this report when you specify ANALYZE PAUSE or ANALYZE ONLY.

- **DYNAMIC FILE ALLOCATION REPORT** (messages BMC50445I through BMC50448I)
  
  REORG PLUS issues this report when dynamic data set allocation is active.

The following example shows a valid ddname:

```
SYSU0001
```

**FREE**

Specifying FREE tells REORG PLUS to free the data sets allocated in your JCL and use only dynamically allocated data sets.

**WARNING**

If you specify this option for a reorganization job in a worklist environment, REORG PLUS frees the data sets for the remainder of the worklist execution. This can result in an error if a subsequent REORG PLUS job in the worklist requires these data sets.
ALLOCATION OPTIONS

This option applies only to sort work data sets.

The ALLOC option allows you to specify the method that you want REORG PLUS to use when dynamically allocating your sort work files.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the ALLOC command option in your installation options module by using the ALLOC installation option (page 744). REORG PLUS was shipped with a default value of ANY for this option. The command option overrides the default that is in the installation options module.

OPTIMIZED

This option tells REORG PLUS to allocate as much DASD as necessary for the best performance results. If REORG PLUS cannot obtain an optimal allocation, processing terminates. For details about how REORG PLUS allocates sort work files for optimized and nonoptimized performance, see page 369.

MINIMAL

This option tells REORG PLUS to allocate the smallest amount of DASD necessary to process your reorganization. Using this option can decrease performance.

ANY

This option tells REORG PLUS to attempt an OPTIMIZED allocation. If this optimized allocation fails, REORG PLUS attempts a MINIMAL allocation. If REORG PLUS cannot obtain a minimal allocation, it terminates.
The SMS option tells REORG PLUS whether to pass the SMS classes and extended data set types (EXTREQ and EXTPREF) to SMS during dynamic allocation. Whether REORG PLUS actually performs an SMS allocation depends on your site.

**Specifying the default**
You can specify the default for the SMS command option in your installation options module by using the SMS installation option (page 755). REORG PLUS was shipped with a default value of NO for this option. The command option overrides the default that is in the installation options module.

**YES**
YES tells REORG PLUS to pass SMS classes and extended data set types during dynamic allocation. The following considerations apply to SMS YES:

- When you specify SMS YES, REORG PLUS ignores the value that you specify for MAXEXTSZ.
- To pass the UNIT value to SMS during dynamic allocation, also specify the SMSUNIT option.
- When invoking DSNUTILB, REORG PLUS includes the SMS classes in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

**NO**
NO tells REORG PLUS not to pass SMS classes and extended data set types during dynamic allocation. When you specify SMS NO, REORG PLUS always passes the UNIT value during dynamic allocation.
SMSUNIT

The SMSUNIT option tells REORG PLUS whether to pass the UNIT value in the SMS allocation parameter list to SMS during dynamic allocation. If you do not specify SMS YES, REORG PLUS ignores the SMSUNIT option. REORG PLUS does not modify any other parameters based on this option.

Specifying the default
You can specify the default for the SMSUNIT command option in your installation options module by using the SMSUNIT installation option (page 756). REORG PLUS was shipped with a default value of NO for this option. The command option overrides the default that is in the installation options module.

NO

NO tells REORG PLUS not to pass the value of the UNIT option.

YES

YES tells REORG PLUS to pass the value of the UNIT option.

When invoking DSNUTILB, REORG PLUS includes the UNIT option values in the TEMPLATE control statements that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.
UNIT

For non-SMS-managed data sets, the UNIT option allows you to specify the primary and secondary unit names that REORG PLUS uses for dynamic data set allocation. The unit names cannot exceed eight characters. Enclose the two unit names within parentheses and separate them with a comma.

For SMS-managed data sets when you specify SMS YES and SMSUNIT YES, this option supplies the unit names that REORG PLUS passes in the SMS allocation parameter list. DSNUTILB reorganization jobs function differently, as described in the DSNUTILB reorganization jobs section.

The THRESHLD installation or command option (page 312) controls which of the specified unit names REORG PLUS selects.

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second UNIT parameter in that template. For information about when REORG PLUS builds a secondary template, see “THRESHLD” on page 312.

Specifying the default
You can specify the default for the UNIT command option in your installation options module by using the UNIT installation option (page 758). REORG PLUS was shipped with a default value of (SYSALLDA,SYSALLDA) for this option. The command option overrides the default that is in the installation options module.

UNITCNT

This option enables you to specify the number of devices to allocate when dynamically allocating data sets. Valid values are 0 through 59. A value of 0 tells REORG PLUS to use the system default.
If you specify a second value (*integer2*), REORG PLUS uses this number when the value for THRESHLD (page 312) is exceeded. This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second UNITCNT parameter in that template. For information about when REORG PLUS builds a secondary template, see “THRESHLD” on page 312.

**Additional considerations**
The following additional considerations apply to the UNITCNT option:

- REORG PLUS ignores this option for sort work files.
- To avoid performance problems, specify only the number of devices that you need.

**Specifying the default**
You can specify the default for the UNITCNT command option in your installation options module by using the UNITCNT installation option (page 759). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.
VOLCNT

This option enables you to specify the maximum number of volumes to use during dynamic allocation. Valid values are:

- 0, which tells REORG PLUS to not specify a volume count for dynamic allocation.
- Integer values 1 through 255 to specify the number of volumes.
- AUTO, which tells REORG PLUS to compute the volume count based on the amount of data, adjusted for the estimated space required (based on, for example, the value of the SIZEPCT installation or command option).

REORG PLUS computes the volume count by dividing the size estimate by the value specified for the AVGVOLSP installation or command option (see page 744 or page 306).

If you specify a second value (integer2), REORG PLUS uses that value when the value for the THRESHLD installation or command option is exceeded (see page 757 or page 312). This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second VOLCNT parameter in that template. For information about when REORG PLUS builds a secondary template, see “THRESHLD” on page 312.

Additional considerations
Note the following additional information about the VOLCNT option:

- This option is not available for DDTYPE SORTWORK. If you specify a value for this option for sort work files, REORG PLUS changes the value to 1.
Dynamic allocation options

- To avoid performance problems, specify only the number of volumes that you need.

- For an SMS-managed data set, REORG PLUS recommends that you specify a value of 0 if your ACS routines are set up to provide a volume count.

**Specifying the default**
You can specify the default for the VOLCNT command option in your installation options module by using the VOLCNT installation option (page 760). REORG PLUS was shipped with a default value of 25 for this option. The command option overrides the default that is in the installation options module.
The AVGVOLSP option enables you to specify the average amount of space that is available on each device. Use this option to reflect the average space available on volumes that are eligible to contain the dynamically allocated data set.

Specify one or two integer values, and include one of the following units of measure:

- K for kilobytes
- TRK for tracks (the default)
- CYL for cylinders

REORG PLUS uses AVGVOLSP only when you specify AUTO for the corresponding first or second parameter of the VOLCNT installation or command option.

If you specify a second value (integer2), REORG PLUS uses that value when the value for the THRESHLD option is exceeded. This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

**Restrictions**

The following restrictions apply to the AVGVOLSP option:

- The AVGVOLSP option is not valid for DDTYPE SORTWORK.
- When invoking DSNUTILB, REORG PLUS ignores this option.

**Additional considerations**

Note the following additional information about the AVGVOLSP option:

- You should not use AVGVOLSP to specify the maximum space on all devices or volumes unless the volumes to be used are empty.

- If you specify a value for AVGVOLSP that is too small, REORG PLUS computes a value for VOLCNT that is too large. However, if you specify a value for this option that is too large, REORG PLUS computes a value for VOLCNT that is too small.

**Specifying the default**

You can specify the default for the AVGVOLSP command option in your installation options module by using the AVGVOLSP installation option (page 744). REORG PLUS was shipped with a default value of ((30000,TRK),(30000,TRK)) for this option. The command option overrides the default that is in the installation options module.
MAXEXTSZ

For any extent that REORG PLUS allocates for a dynamically allocated data set, this option enables you to specify the maximum allowable value for the primary space allocation. Because the secondary quantity cannot exceed the primary quantity, MAXEXTSZ also controls the maximum secondary quantity.

Specify MAXEXTSZ by using one of the following values:

- 0 if you do not want to set a limit for space allocation
- an integer for the units specified with the UNIT installation or command option (see page 758 or page 302), with the unit of measure as follows:
  — K for kilobytes (the default)
  — TRK for tracks
  — CYL for cylinders

You can specify values for MAXEXTSZ by using one of the following options:

- If you specify a single value (integer1), that value applies to both units (to ensure backward compatibility with previous REORG PLUS versions).

- If you specify a single value and a comma as follows, REORG PLUS takes the missing value from the MAXEXTSZ installation option:

  \[
  \text{MAXEXTSZ} \left( \text{integer1,} \ K \right) \]
  
  or

  \[
  \text{MAXEXTSZ} \left( , \text{integer2,} \ K \right)
  \]

- If you specify a second value (integer2), REORG PLUS uses that value when the value specified for the THRESHLD installation or command option is exceeded (see page 757 or page 312). This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

Specifying two MAXEXTSZ values might be useful if characteristics of the unit that you specify for allocations below the threshold are different from those of the unit that you specify for allocations above the threshold.
Regardless of the amount of space that REORG PLUS determines that it needs, REORG PLUS will not allocate more than your specified MAXEXTSZ limit for either the primary or the secondary quantity.

- If the amount of required space that REORG PLUS calculates is greater than the MAXEXTSZ limit for the primary quantity, REORG PLUS uses the secondary extents to hold the remainder of the required primary space.

- If the amount of required space that REORG PLUS calculates cannot be accommodated because of MAXEXTSZ restrictions, the job might terminate with an out-of-space condition on the data set.

**DSNUTILB reorganization jobs**

When invoking DSNUTILB, REORG PLUS includes the first parameter of this option as the MAXPRIME value in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For DSNUTILB, the unit of measure is always cylinders.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second parameter of this option as the MAXPRIME value in that template. For information about when REORG PLUS builds a secondary template, see “THRESHLD” on page 312.

**Additional considerations**

Note the following additional information about the MAXEXTSZ option:

- The MAXEXTSZ option is not valid for DDTYPE SORTWORK.
- REORG PLUS ignores MAXEXTSZ when you specify SMS YES.
- REORG PLUS checks the value of MAXEXTSZ after applying SIZEPCT to the allocation amount.

**Specifying the default**

You can specify the default for the MAXEXTSZ command option in your installation options module by using the MAXEXTSZ installation option (page 753). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.
**DSNTYPE**

The DSNTYPE option enables you to specify the type of data set that you want REORG PLUS to allocate. If you specify a second value, REORG PLUS uses that value when the value that is in effect for the THRESHLD option is exceeded (see page 757 or page 312).

**Restrictions**
The following restrictions apply to the DSNTYPE option:

- The DSNTYPE option is not valid for tape data sets.
- When invoking DSNUTILB, REORG PLUS ignores this option.

**Specifying the default**
You can specify the default for the DSNTYPE command option in your installation options module by using the DSNTYPE installation option (page 750). REORG PLUS was shipped with a default value of (NONE,NONE) for this option. The command option overrides the default that is in the installation options module.

**NONE**

NONE tells REORG PLUS to not use any extended attributes for this data set allocation.

**LARGE**

LARGE tells REORG PLUS to allocate this data set as a large format sequential data set. This option enables data sets larger than 65,535 tracks.

**BASIC**

BASIC tells REORG PLUS to allocate this data set as a basic sequential data set. This data set will be limited to 65,535 tracks.
**EXTREQ**

EXTREQ tells REORG PLUS to allocate this data set as an extended format data set.

This option is ignored if either of the following conditions exists:

- The SMS option is NO.
- The DDTYPE is SORTWORK.

**EXTPREF**

EXTPREF indicates that you prefer that REORG PLUS allocate this data set as an extended format data set. If this allocation is not possible, the data set is allocated as a basic format data set.

This option is ignored if either of the following conditions exists:

- The SMS option is NO.
- The DDTYPE is SORTWORK.
SIZEPCT

The SIZEPCT option allows you to adjust, by percentages, the allocated data set sizes that REORG PLUS calculates. The values that you specify must be greater than 0.

- primary indicates the percentage of the primary data set size calculated by REORG PLUS that you actually want allocated.

- secondary indicates the percentage of the secondary data set size calculated by REORG PLUS that you actually want allocated.

If you specify a secondary size that is greater than the primary size, REORG PLUS changes the secondary value to equal the primary value.

If you are performing a SHRLEVEL CHANGE reorganization that is creating incremental copy data sets, you can use the SIZEPCT option to reduce the size of those data sets. Because REORG PLUS uses the high-used RBA of the table space to determine the primary quantity, the incremental copy data sets can be larger than needed. If you use your estimate of the percentage of pages in the table space that will be updated during the reorganization as the primary SIZEPCT value, less space will be allocated for the data sets.

When you specify a value for both the SIZEPCT installation or command option and the ARCROWS command option, REORG PLUS ignores SIZEPCT for the archive data set and uses only ARCROWS to determine the archive data set size.

**DSNUTILB reorganization jobs**

When invoking DSNUTILB, REORG PLUS includes the first parameter of this option as the PCTPRIME value in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. If you specify a value greater than 100, REORG PLUS converts it to 100.

**Specifying the default**

You can specify the default for the SIZEPCT command option in your installation options module by using the SIZEPCT installation option (page 755). REORG PLUS was shipped with a default value of (100,100) for this option for most DDTYPEs and (5,100) for the incremental copy DDTYPEs (LOCPICPY, LOCBICPY, REMPICPY, and REMBICPY). The command option overrides the default that is in the installation options module.
Dynamic allocation options

SPACE

This option is only valid for a DSNUTILB reorganization.

The SPACE option enables you to override the DSNUTILB space calculation for the dynamically allocated data set. This option also overrides the value for the SIZEPCT option.

When you use this option, you must specify integer values for both the primary and secondary data sets. The numbers that you specify must be greater than 0. Include the unit of measure as follows:

- CYL for cylinders
- TRK for tracks

When not invoking DSNUTILB, REORG PLUS ignores this option.

THRESHLD

The THRESHLD option allows you to specify a threshold value, in kilobytes, above which REORG PLUS applies secondary values to allocated data sets. REORG PLUS tests this threshold for each data set to be allocated. If the size for a particular data set is greater than the threshold, REORG PLUS performs the following tasks:

- When SMS is NO, REORG PLUS uses the secondary values of the following options for the allocated data sets:
  - UNIT (unit name)
  - UNITCNT (unit count)
  - VOLCNT (volume count)
  - AVGVOLSP (average volume space)
  - MAXEXTSZ (extent size)
  - DSNTYPE (data set type), unless DSNTYPE is EXTREQ or EXTPREF
When SMS is YES, REORG PLUS uses the secondary values of the following options (or passes them to SMS if applicable):

- STORCLAS, MGMTCLAS, or DATACLAS (class name for the SMS classes)
- VOLCNT (volume count)
- AVGVOLSP (average volume space)
- DSNTYPE (data set type)
- if SMSUNIT is YES
  - UNIT (unit name)
  - UNITCNT (unit count)

If you specify 0, or if the threshold is not exceeded, REORG PLUS uses the primary values for these options.

**TIP**
If you use THRESHLD to send larger data sets to tape, consider setting the MAXTAPE option to limit the number of tapes that you use.

**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS handles this option differently, depending on the type of data set that you are allocating:

- For copy data sets, REORG PLUS translates this option to a LIMIT value in the TEMPLATE control statement that REORG PLUS builds for the IBM DB2 REORG utility. REORG PLUS also builds the secondary TEMPLATE control statement to which the DB2 REORG utility will switch when the LIMIT is exceeded.

  REORG PLUS assumes that the value that you supply for THRESHLD is in kilobytes. REORG PLUS translates that value to the appropriate value and unit of measure for the LIMIT keyword. Note the following additional information about this value:

  - REORG PLUS rounds down to the nearest whole value.
  - If you specify a value that would cause REORG PLUS to translate to a value less than 1 cylinder, REORG PLUS builds the template with a LIMIT value of 1 CYL.

- For all other data sets, REORG PLUS ignores this option.

**Examples**
The following examples illustrate how to use THRESHLD for different types of scenarios. In example 1, all allocated data sets for this DDTYPE go to SYSDA:

**Figure 10 THRESHLD example 1**

| UNIT (SYSDA) |
| THRESHLD 0   |
In example 2, allocated data sets with a size greater than 720 MB for this DDTYPE go to unit LARGE. Smaller allocated data sets for this DDTYPE go to unit WORK:

**Figure 11  THRESHLD example 2**

```
UNIT(WORK,LARGE)
THRESHLD 720000
```

In example 3, data sets for this DDTYPE are SMS-managed. Allocated data sets with a size greater than 72 MB go to STORCLS2, MGMTCLS2, and DATACLs2. Smaller data sets go to the classes specified in the first parameter of each class type.

**Figure 12  THRESHLD example 3**

```
SMS YES
STORCLAS(STORCLS1,STORCLS2)
MGMTCLAS(MGMTCLS1,MGMTCLS2)
DATACLAS(DATACLs1,DATACLs2)
THRESHLD 72000
```

In example 4, REORG PLUS allocates data sets with the following properties for this DDTYPE:

- Data sets with a size less than or equal to 7,200 MB go to unit SYSDA. The maximum volume count is calculated based on an average volume usage of 30,000 tracks.
- Data sets with a size greater than 7,200 MB go to unit CART with no more than 25 volumes.

**Figure 13  THRESHLD example 4**

```
VOLCNT(AUTO,25)
AVGVOLSP((30000,TRK),(900000,TRK))
UNIT(SYSDA,CART)
THRESHLD 7200000
```

**Specifying the default**

You can specify the default for the THRESHLD command option in your installation options module by using the THRESHLD installation option (page 757). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.
The DSNPAT option specifies a particular pattern of variables and text that REORG PLUS uses to create data set names for dynamic data set allocation. The maximum total length that REORG PLUS allows for a data set name is 44 bytes.

When invoking DSNUTILB, REORG PLUS includes this pattern in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. Any variables that you include in your pattern for this type of reorganization must be either valid for the DB2 TEMPLATE control statement or translatable (as shown in Table 44 on page 316) to a valid TEMPLATE variable. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

Specifying the default
You can specify the default for the DSNPAT command option in your installation options module by using the DSNPAT installation option (page 745). REORG PLUS was shipped with a different default value for each DDTYPE. The command option overrides the default that is in the installation options module.

'pattern'

Specify the pattern of variables and text, with surrounding single quotes, to use to create data set names.

The pattern that you specify in your DSNPAT option must allow REORG PLUS to generate unique data set names. If REORG PLUS encounters non-unique data set names, processing terminates. Note the following considerations:

- For full and incremental copy data sets, you might need to include additional variables such as &VCAT, &DATEJ, and &TIME4 to generate unique names across multiple reorganizations.

- If you plan to run concurrent REORG PLUS jobs or run multiple REORG PLUS steps in a worklist environment, BMC recommends that you include additional variables in the default pattern to ensure unique names. For example, the following value ensures that REORG PLUS generates unique names:

    ```
    DSNPAT '&UID.&UTILPFX.&DDNAME..D&DATE..T&TIME'
    ```
Variables
You can use any of the symbolic variables listed in Table 44 to construct your pattern. In addition, you can use text or provide user-defined variables from a user exit. The REORG command option, DSNEXIT, is described on page 240. User exits are further discussed in Appendix C, “REORG PLUS user exits.”

Symbols for numeric variables (such as &DATE, &TIME, &PART, and &PART5) must be prefixed by a national character (alphabetic, #, @, or $). In the following example, the first statement causes an error, but the second is correct:

```
DSNPAT '&DB.&TSIX.&DATE'
DSNPAT '&DB.&TSIX..RP&DATE'
```

Although you can prefix a symbolic variable with an alphabetic character, you cannot append characters. For example, XX&DB is valid, but &DBXX is invalid. &DB.XX is valid.

User-defined variables must begin with an underscore character, as in _DEPT. User-defined variables are not valid for a DSNUTILB reorganization. For more information, see “DSNEXIT user-defined variables” on page 791.

Table 44  Symbolic variables for the DSNPAT command option (part 1 of 3)

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of result</th>
<th>DSNUTILB reorganization</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;DATE</td>
<td>current date (in the form MMDDYY)</td>
<td>6 bytes</td>
<td>variable passed</td>
</tr>
<tr>
<td>&amp;DATEJ</td>
<td>current Julian date (in the form YYYYDDD)</td>
<td>7 bytes</td>
<td>variable translated to the IBM &amp;JDATE variable</td>
</tr>
<tr>
<td>&amp;DB</td>
<td>database containing the space for this data set allocation</td>
<td>8 bytes maximum</td>
<td>variable passed</td>
</tr>
<tr>
<td>&amp;DDNAME</td>
<td>ddname for this data set allocation</td>
<td>8 bytes maximum</td>
<td>value passed</td>
</tr>
<tr>
<td>&amp;GRPNM</td>
<td>DB2 data sharing group name</td>
<td>4 bytes</td>
<td>value passed</td>
</tr>
<tr>
<td></td>
<td>In a non-data sharing environment, GRPNM contains the DB2 SSID.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp;JDATE</td>
<td>current Julian date (in the form YYDDD)</td>
<td>5 bytes</td>
<td>variable translated to the IBM &amp;JDATE(3,5) variable</td>
</tr>
<tr>
<td>&amp;JOBNAME</td>
<td>JOB name in the JCL</td>
<td>8 bytes maximum</td>
<td>variable passed</td>
</tr>
</tbody>
</table>
Table 44  Symbolic variables for the DSNPAT command option (part 2 of 3)

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of resulta</th>
<th>DSNUTILB reorganization</th>
</tr>
</thead>
</table>
| &PART             | partition for this data set allocation
You can use this variable for any data set. However, REORG PLUS substitutes the partition number for only copy and unload data sets. For all other data sets, REORG PLUS substitutes the value 000 or 0000. |
|                   |            | 3 bytes for table spaces with 999 partitions or less 4 bytes for table spaces with 1000 through 4096 partitions | variable passed |
| &PART5            | partition for this data set allocation
You can use this variable for any data set. However, REORG PLUS substitutes the partition number for only copy and unload data sets. For all other data sets, REORG PLUS substitutes the value 00000. |
|                   |            | 5 bytes for table spaces with 4096 partitions or less | variable passed |
| &RTYPE            | REORG type (TS or IX) | 2 bytes maximum | job terminated |
| &SSID             | DB2 subsystem ID | 4 bytes | variable passed |
| &STEPNAME         | STEP name in the JCL
REORG PLUS ignores PROC names. |
|                   | 8 bytes maximum | variable passed |
| &TIME             | current time (in the form HHMMSS) | 6 bytes | variable passed |
| &TIME4            | current time (in the form HHMM) | 4 bytes | variable passed |
| &TSIX             | table space or index space specified in your REORG command | 8 bytes maximum | variable translated to the IBM &SN variable |
Utility ID variable

If the utility ID has no special character delimiters, &UTILITY contains the first eight bytes of text and &UTILSUFFIX contains the remaining eight bytes of text.

If the utility ID has a special character delimiter within the first eight bytes of text, &UTILITY contains the bytes up to but not including the delimiter. &UTILSUFFIX contains the eight bytes following the first delimiter. The first delimiter is not included in either variable. Any delimiter after the first is treated as normal text and might be included in &UTILSUFFIX. Depending on the utility ID that is specified for this reorganization job, &UTILSUFFIX might be blank.

For example, if the utility ID is USER1//WORK1, the variables contain the following values:

```
&UTILITY = USER1/W0
&UTILITYPREFIX = USER1
&UTILSUFFIX = /WORK1
```

If the utility ID is USER1//WORK1, the variables contain the following values:

```
&UTILITY = USER1/W
&UTILITYPREFIX = USER1
&UTILSUFFIX = /WORK1
```
The following special delimiting characters tell REORG PLUS to split the utility ID:

### Table 45  Valid special delimiter characters for utility ID

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>period</td>
</tr>
<tr>
<td>+</td>
<td>plus sign</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td>semicolon</td>
</tr>
<tr>
<td>-</td>
<td>dash</td>
</tr>
<tr>
<td>/</td>
<td>slash</td>
</tr>
<tr>
<td>;</td>
<td>broken bar</td>
</tr>
<tr>
<td>_</td>
<td>underscore</td>
</tr>
<tr>
<td>:</td>
<td>colon</td>
</tr>
<tr>
<td>=</td>
<td>equal sign</td>
</tr>
</tbody>
</table>

**NOTE**
Any other special characters in the utility ID might cause REORG PLUS to generate invalid data set names.

**Name construction**
You can specify any or all nodes of a data set name by using variables or text. For example, the following example generates data set names that contain the ID of the user, the table or index space that is involved in the reorganization, and the name of the reorganization job:

`DSNPAT '&UID.&TSIX.&JOBNAME'`

The following example combines actual text with symbolic variables to generate a data set name:

`DSNPAT '&DB.&DDNAME..NEW'`

In certain cases, a period is required in your pattern as a node delimiter or to indicate the end of a variable name. Table 46 on page 320 illustrates concatenation in a data set name pattern and those instances in which a period is needed.

**NOTE**
Concatenation is not affected by a period contained in the value of a user variable.
REORG PLUS ignores trailing blanks and null value variables. However, REORG PLUS includes node-delimiting periods in the pattern, regardless of the variable’s value, which might result in an invalid data set name. For example, given that &UID=RDAB and the value of _DEPT is null, the following example results in an invalid data set name of RDAB..NEW:

```
&UID._DEPT..NEW
```

**GDG names**

You can also specify a pattern that contains a generation data group (GDG) name for your dynamically allocated SYSARC, SYSPUNCH, and full and incremental copy data sets. Each DDTYPE must have a different GDG base.

The GDG format that you use to construct data set names is the same as the format that you use in JCL when you use DD statements to allocate data sets. Simply append the generation number in parentheses. The open parenthesis tells REORG PLUS that the pattern is a GDG name. The generation number must be an integer from 1 through 255.

If the base does not exist, REORG PLUS creates it for you, using everything in the pattern up to the open parenthesis as the base name. For more information about GDG names and options, see “Generating data set names” on page 92.

The following example shows a valid GDG name:

```
'&UTILPFX.&DDNAME..COPY(+1)'
```
If you are using a substitution variable as the last variable before the open parenthesis, you must include a period before the open parenthesis, as in the following example:

'&UTILPFX.&DDNAME.(+1)'

If you specify COPYLVL PART on the REORG command, each partition must have a different GDG base. To specify a pattern that includes a partition, the partition must not be in parentheses. The following example shows a valid name:

'&UTILPFX.&DDNAME..P&PART.(+1)'

You cannot specify a pattern that contains a partitioned data set (PDS) name. The following example shows an invalid name:

'&UTILPFX.&DDNAME..(P&PART)'

'NONE' or ''

Specify 'NONE' or '' to indicate that you do not want to use any pattern to create data set names during dynamic data set allocation. This option is valid only with sort work data sets.

---

**WARNING**

If you specify NONE for SYSREC, SYSUT1, full copy data set, or incremental copy data set patterns, REORG PLUS will not allocate your data sets and will terminate your job.

---

In the DSNPAT command option, you must enclose the keyword NONE in single quotes. However, in the DSNPAT installation option, do not enclose the keyword NONE in single quotes.
**EXPDT**

Specify EXPDT to set an expiration date for any SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating. The value of `date` must be in the format `yyyyddd` or `yyyy/ddd`. The variable `yyyy` is the 4-digit year (1900 through 2155), and `ddd` is the 3-digit Julian day (000 through 366).

REORG PLUS ignores this option if you specify it for any of the other data sets that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

**Specifying the default**

You can specify the default for the EXPDT command option in your installation options module by using the EXPDT installation option (page 751). REORG PLUS was shipped with no default value for this option. The EXPDT command option overrides the RETPD command option and the EXPDT and RETPD (page 754) installation options.

**RETPD**

Specify RETPD to set a retention period (in days) for any SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating. The value of `integer` must be in the range 0 through 9999.

REORG PLUS ignores this option if you specify it for any `work` data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.
**Specifying the default**  
You can specify the default for the RETPD command option in your installation options module by using the RETPD installation option (page 754). REORG PLUS was shipped with no default value for this option. The RETPD command option overrides the default established at installation time for both the RETPD and EXPDT (page 751) installation options. However, if you specify the EXPDT command option, it takes precedence over the RETPD command option.

**GDGLIMIT**

Specify GDGLIMIT to indicate the number of generations to keep for the SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating as a GDG data set. The value of integer must be in the range 1 through 255.

REORG PLUS honors this option only when creating the GDG base. REORG PLUS ignores this option if you specify it for any other data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

**Specifying the default**  
You can specify the default for the GDGLIMIT command option in your installation options module by using the GDGLIMIT installation option (page 752). REORG PLUS was shipped with a default value of 5 for this option. The command option overrides the default that is in the installation options module.
These options allow you to specify the classes that REORG PLUS is to use for SMS allocations. You can repeat this block to specify more than one type of SMS class.

The class names for each of the following class types must be enclosed in parentheses and separated by a comma. Use the keyword NONE instead of the class name if you do not want to specify a class for dynamic data set allocation.

The THRESHLD installation option or command option determines how REORG PLUS selects the classes that you specify here. This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS). For more information, see the THRESHLD command option on page 312.

**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS includes these classes in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the secondary classes in that template. For information about when REORG PLUS builds a secondary template, see “THRESHLD” on page 312.

**Specifying the default**
You can specify the default for the SMS class block in your installation options module by using the DATACLAS (page 745), MGMTCLAS (page 754), or STORCLAS (page 756) installation options. REORG PLUS was shipped with default values of (NONE,NONE) for these options. The command options override the default that is in the installation options module.
DATACLAS

DATACLAS indicates the primary and secondary SMS data classes that REORG PLUS uses for allocating the specified data set type. For class1 and class2, specify valid SMS data class names that do not exceed eight characters each.

MGMTCLAS

MGMTCLAS indicates the primary and secondary SMS management classes that REORG PLUS uses for allocating the specified data set type. For class1 and class2, specify valid SMS management class names that do not exceed eight characters each.

STORCLAS

STORCLAS indicates the primary and secondary SMS storage classes that REORG PLUS uses for allocating the specified data set type. For class1 and class2, specify valid SMS storage class names that do not exceed eight characters each.
SHRLEVEL CHANGE options

You can perform an online reorganization by specifying SHRLEVEL CHANGE, along with its options, on the REORG command. You can specify SHRLEVEL CHANGE for two-phase or single-phase reorganization of a table space or index space. The benefit of SHRLEVEL CHANGE is that the data that you are reorganizing is available for read-write (RW) operations during most of the reorganization processing.

The following section provides the syntax of the options for SHRLEVEL CHANGE. For more information, see Chapter 7, “Online reorganization.”

| LOGTHRESHLD integer | MAXRO integer | DEADLINE NONE timeStamp time | DEFER |

LOGTHRESHLD

LOGTHRESHLD specifies the number of log records as a positive integer (0 or greater). When REORG PLUS determines that the number of log records remaining to be applied for all of the objects that you are reorganizing is less than or equal to this value, it begins the LOGFINAL phase. The LOGFINAL phase prevents updates to the objects and applies the remaining log records.

You can also use the XBM Utility Monitor function or the MVS operator console to dynamically change the value for this option while the reorganization is in progress. For more information, see “Using XBM to view and dynamically control the log apply process” on page 639.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 633.

Specifying the default

You can specify the default for the LOGTHRESHLD command option in your installation options module by using the LOGTHRSIH installation option (page 707). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.

MAXRO

MAXRO tells REORG PLUS when to end the LOGAPPLY phase and begin the LOGFINAL phase.
integer

This integer specifies the maximum number of seconds that REORG PLUS spends applying log records in the LOGFINAL phase. Specify this value as a positive integer (0 or greater). When REORG PLUS estimates that the number of seconds required to apply the remaining log records is less than this value, it ends the LOGAPPLY phase and begins the LOGFINAL phase. The estimated time does not include the time required to produce the incremental copy data sets or the time needed for the UTILTERM phase. In the LOGFINAL phase, REORG PLUS prevents updates to the objects and applies the remaining log records.

DEFER

This value tells REORG PLUS to continue applying log records indefinitely, regardless of the value set in any other SHRLEVEL CHANGE option except DEADLINE, which is still checked. The LOGFINAL phase will not begin until you change this value. If DEADLINE is reached first, the job terminates after issuing message BMC50784E.

You can also use the XBM Utility Monitor function or the MVS operator console to dynamically change the value for this option while the reorganization in progress. For more information, see “Using XBM to view and dynamically control the log apply process” on page 639.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 633.

Specifying the default

You can specify the default for the MAXRO command option in your installation options module by using the MAXRO installation option (page 710). REORG PLUS was shipped with a default value of 300 for this option. The command option overrides the default that is in the installation options module.

DEADLINE

DEADLINE specifies the time by which the LOGFINAL phase should finish applying log records. If REORG PLUS determines that the LOGFINAL phase will not finish by the deadline, it terminates the reorganization. When calculating the estimate, REORG PLUS does not include the time required to produce the incremental copy data sets (if creating them) or the time needed for the UTILTERM phase. REORG PLUS does not check the DEADLINE value until the LOGAPPLY phase begins.

If a timestamp or time value is specified on DEADLINE and LOGFINAL, the calculated LOGFINAL timestamp must be less than the calculated DEADLINE timestamp.
You can use the XBM Utility Monitor function or the MVS operator console to
dynamically change the DEADLINE values while the reorganization is in progress.
For more information, see “Using XBM to view and dynamically control the log
apply process” on page 639.

The DEADLINE option interacts with other options to control the log apply process.
For information, see “Control of the log apply process” on page 633.

Specifying the default
You can specify the default for the DEADLINE command option in your installation
options module by using the DEADLINE installation option (page 683). REORG
PLUS was shipped with a default value of NONE for this option. The command
option overrides the default that is in the installation options module.

NONE

The value NONE specifies no deadline by which the LOGFINAL phase must finish.

timestamp

Specify a timestamp value to indicate the time by which the LOGFINAL phase
should finish applying log records.

Specifying a time zone
You can specify a time zone with this timestamp, indicating its difference from
Coordinated Universal Time (UTC), by using a value from -12:59 through +14:00. If
you specify a time zone, REORG PLUS converts your timestamp to UTC plus or
minus the difference that you specify. The actual deadline that REORG PLUS
enforces depends on local time when the reorganization runs.

For example, if you specify a timestamp of 2012-09-25-18.00.00.000000+02:00, REORG
PLUS converts this value to 20:00 UTC. If the reorganization is running under North
American Central Time when daylight saving time is in effect, REORG PLUS enforces
a deadline of 15:00, based on the local time’s UTC offset.

For a DSNUTILB reorganization, REORG PLUS terminates if you specify a time zone
on your DEADLINE timestamp.

time

Specify a time of day to be used for DEADLINE. Specify the time value in hh:mm:ss
format, using a 24-hour clock.

REORG PLUS replaces the time portion of the current system timestamp (at
start-of-utility) with the time specified to create a DEADLINE timestamp. If the
calculated DEADLINE timestamp value is less than the current system timestamp,
REORG PLUS adds one day to the timestamp. REORG PLUS uses this calculated
value to determine when the LOGFINAL phase should finish applying log records.
LONGLOG

LONGLOG specifies the action to take if REORG PLUS determines that a longlog condition exists. A longlog condition exists when REORG PLUS determines over a period of time that the DB2 subsystem is generating log records for the objects that you are reorganizing faster than REORG PLUS is applying the records. After detecting that a longlog condition exists, REORG PLUS sends an action write-to-operator (WTO) notification (see page 644) and continues to apply the log records until the delay interval specified on the DELAY option (see page 330) expires. If the longlog condition still exists, REORG PLUS takes the action that you specify with one of the following options.

CONTINUE

CONTINUE indicates that REORG PLUS is to continue processing in the LOGAPPLY phase.

TERM

TERM indicates that REORG PLUS is to terminate the reorganization.

DRAIN

DRAIN indicates that REORG PLUS is to begin the LOGFINAL phase, which prevents updates to the objects that you are reorganizing and applies the remaining log records.

You can also use the XBM Utility Monitor function or the MVS operator console to dynamically change the value for this option while the reorganization is in progress. For more information, see “Using XBM to view and dynamically control the log apply process” on page 639.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 633.

Specifying the default

You can specify the default for the LONGLOG command option in your installation options module by using the LONGLOG installation option (page 708). REORG PLUS was shipped with a default value of CONTINUE for this option. The command option overrides the default that is in the installation options module.
DELAY

DELAY specifies the number of seconds that are to elapse from the time REORG PLUS detects a longlog condition until it performs the action specified by the LONGLOG option (page 329). If the longlog condition no longer exists at the end of the time period specified by DELAY, the timer is reset. When REORG PLUS detects the next longlog condition, it restarts the timer, using the original DELAY value. The value must be a positive integer (0 or greater).

During the window provided by the DELAY option, you can use the XBM Utility Monitor function or the MVS operator console to change various SHRLEVEL CHANGE options, such as the LONGLOG action or MAXRO. This capability allows you to dynamically change when LOGFINAL processing begins or terminate the reorganization. For more information, see “Using XBM to view and dynamically control the log apply process” on page 639.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 633.

Specifying the default
You can specify the default for the DELAY command option in your installation options module by using the DELAY installation option (page 684). REORG PLUS was shipped with a default value of 1200 for this option. The command option overrides the default that is in the installation options module.

SPILLUNIT

The SPILLUNIT option allows you to specify the DASD unit where REORG PLUS can allocate spill data sets. The unit name must be a valid DASD unit name not exceeding eight characters, or NONE. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.

For information about the interaction of the spill data set options and performance impacts, see “Allocation of spill data sets” on page 648 and “Performance considerations” on page 659.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the SPILLUNIT command option in your installation options module by using the SPILUNIT installation option (page 728). REORG PLUS was shipped with a default value of WORK for this option. The command option overrides the default that is in the installation options module.
SPILLSTORCLAS

The SPILLSTORCLAS option allows you to specify the SMS storage class that REORG PLUS uses to allocate spill data sets. You can specify a valid SMS storage class name not exceeding eight characters, or NONE. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates the spill data sets as they are needed. The spill data sets are VSAM objects.

For information about the interaction of the spill data set options and performance impacts, see “Allocation of spill data sets” on page 648 and “Performance considerations” on page 659.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the SPILLSTORCLAS command option in your installation options module by using the SPILCLS installation option (page 727). REORG PLUS was shipped with a default value of NONE for this option. The command option overrides the default that is in the installation options module.

SPILLDSNPAT

The SPILLDSNPAT option tells REORG PLUS to use a particular pattern of variables and text to create a prefix for the spill data set names. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates the spill data sets as they are needed. The spill data sets are VSAM objects.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the SPILLDSNPAT command option in your installation options module by using the SPIILDSNP installation option (page 726). REORG PLUS was shipped with a default value of &&UID for this option. The command option overrides the default that is in the installation options module.
'pattern'

Specify the pattern of variables and text, with surrounding single quotes, to use to create the data set names. The pattern must resolve to a prefix that is 22 bytes or less in length and not end in a period.

You can use any of the symbolic variables listed in Table 47 to construct your pattern. In addition, you can use text or provide user-defined variables from a user exit. The REORG command option, DSNUEXIT, is described on page 240, and user exits are further discussed in Appendix C, “REORG PLUS user exits.”

Table 47  Symbolic variables for the SPILLDSNPAT command option (part 1 of 2)

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of resulta</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;DATE</td>
<td>current date (in the form MMDDYY)</td>
<td>6 bytes</td>
</tr>
<tr>
<td>&amp;DATEJ</td>
<td>current Julian date (in the form YYYYDDD)</td>
<td>7 bytes</td>
</tr>
<tr>
<td>&amp;DB</td>
<td>database containing the space for this data set allocation</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;GRPNM</td>
<td>DB2 data sharing group name</td>
<td>4 bytes</td>
</tr>
<tr>
<td>&amp;JDATE</td>
<td>current Julian date (in the form YYDDD)</td>
<td>5 bytes</td>
</tr>
<tr>
<td>&amp;JOBNAME</td>
<td>JOB name in the JCL</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;RTYPE</td>
<td>REORG type (TS or IX)</td>
<td>2 bytes maximum</td>
</tr>
<tr>
<td>&amp;SSID</td>
<td>DB2 subsystem ID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>&amp;STEPNAME</td>
<td>STEP name in the JCL</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>current time (in the form HHMMSS)</td>
<td>6 bytes</td>
</tr>
<tr>
<td>&amp;TIME4</td>
<td>current time (in the form HHMM)</td>
<td>4 bytes</td>
</tr>
<tr>
<td>&amp;TSIX</td>
<td>table space or index space specified in the REORG command</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;USERID or &amp;UID</td>
<td>job user ID</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;UTIL</td>
<td>BMC utility ID</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;UTILPFX</td>
<td>BMC utility ID prefix</td>
<td>8 bytes maximum</td>
</tr>
</tbody>
</table>
User-defined variables must begin with an underscore character, as in _DEPT. For more information, see “DSNUEXIT user-defined variables” on page 791.

If the utility ID has no special character delimiters, &UTILPFX contains the first eight bytes of text and &UTILSFX contains the remaining eight bytes of text.

If the utility ID has a special character delimiter within the first eight bytes of text, &UTILPFX contains the bytes up to but not including the delimiter. &UTILSFX contains the eight bytes following the first delimiter. The first delimiter is not included in either variable. Any delimiter after the first is treated as normal text and will be included in &UTILSFX if it is contained within the eight-byte suffix. Depending on the utility ID specified for this reorganization, &UTILSFX might be blank. For example, if the utility ID is USER1/WORK1, the fields will have the following values:

\[
\begin{align*}
&\text{&UTIL = USER1/WO} \\
&\text{&UTILPFX = USER1} \\
&\text{&UTILSFX = WORK1}
\end{align*}
\]

If the utility ID is USER1//WORK1, the fields will have the following values:

\[
\begin{align*}
&\text{&UTIL = USER1/W} \\
&\text{&UTILPFX = USER1} \\
&\text{&UTILSFX = /WORK1}
\end{align*}
\]

The following special delimiting characters cause REORG PLUS to split the utility ID:

### Table 48  Valid special delimiter characters for utility ID (part 1 of 2)

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>period</td>
</tr>
<tr>
<td>+</td>
<td>plus sign</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>;</td>
<td>semicolon</td>
</tr>
<tr>
<td>-</td>
<td>dash</td>
</tr>
</tbody>
</table>
You can specify any or all nodes of a data set name prefix by using variables or text. The following example generates data set names containing the user ID, the table or index space, and the job name:

```
SPILDSNPAT '&UID.&TSIX'
```

The following example combines actual text with symbolic variables to generate a data set name prefix:

```
SPILDSNPAT '&DB.&SSID..NEW'
```

In certain cases, a period is required in your pattern as a node delimiter or to indicate the end of a variable name. Table 49 illustrates concatenation in a data set name pattern and instances in which a period is needed.

**NOTE**

Concatenation is not affected by a period contained in the value of a user variable.

### Table 48  Valid special delimiter characters for utility ID (part 2 of 2)

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>slash</td>
</tr>
<tr>
<td>!</td>
<td>broken bar</td>
</tr>
<tr>
<td>-</td>
<td>underscore</td>
</tr>
<tr>
<td>:</td>
<td>colon</td>
</tr>
<tr>
<td>=</td>
<td>equal sign</td>
</tr>
</tbody>
</table>

### Table 49  Variable concatenation examples (part 1 of 2)

<table>
<thead>
<tr>
<th>Task</th>
<th>Code</th>
<th>Result (where &amp;UID=RDAB and _DEPT=DEV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concatenate the values of two variables (no node delimiter).</td>
<td>&amp;UID.DEPT</td>
<td>RDABDEVP</td>
</tr>
<tr>
<td>Make two nodes from the values of two variables.</td>
<td>&amp;UID..DEPT</td>
<td>RDAB.DEVP</td>
</tr>
<tr>
<td>Concatenate the value of a variable with text (no node delimiter).</td>
<td>&amp;UID.NEW</td>
<td>RDABNEW</td>
</tr>
</tbody>
</table>
Table 49  Variable concatenation examples (part 2 of 2)

<table>
<thead>
<tr>
<th>Task</th>
<th>Code</th>
<th>Result (where &amp;UID=RDAB and _DEPT=DEV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concatenate text with the value of a variable (no node delimiter).</td>
<td>NEW&amp;UID</td>
<td>NEWRDAB</td>
</tr>
<tr>
<td>Make two nodes from the value of a variable followed by text.</td>
<td>&amp;UID..NEW</td>
<td>RDAB.NEW</td>
</tr>
<tr>
<td>Make two nodes from text followed by the value of a variable.</td>
<td>NEW.&amp;UID</td>
<td>NEW.RDAB</td>
</tr>
</tbody>
</table>

However, REORG PLUS includes node-delimiting periods in the pattern, regardless of the variable’s value, which might result in an invalid data set name. For example, given that &UID=RDAB and the value of _DEPT is null, &UID._DEPT..NEW results in an invalid data set name of RDAB..NEW.

Symbols for numeric variables (such as &DATE, and &TIME) must be prefixed by a national character (alphabetic, #, @, or $). In the following example, the following statement causes an error:

```
'SPILLOSPAT '&DB.&DATE'
```

The following statement is correct:

```
'SPILLOSPAT '&DB..RP&DATE'
```

Although you can prefix a symbolic variable with an alphabetic character, you cannot append characters. For example, XX&DB is valid, but &DBXX is invalid. &DB.XX is also valid.

For information about the interaction of the spill data set options and performance impacts, see “Allocation of spill data sets” on page 648 and “Performance considerations” on page 659.
RIDMAPMEM

The RIDMAPMEM option specifies the amount of data space memory (in kilobytes) that REORG PLUS can use to store the RID translation maps. The number must be 0 or a positive integer. BMC strongly recommends that you use 0.

If you specify 0, REORG PLUS automatically calculates the RIDMAPMEM value for you. To determine the value, REORG PLUS multiplies the value of the installation option RIDMMAXD by the value of the installation option RIDMDSSZ. The result is the maximum amount of storage that REORG PLUS can allocate. However, REORG PLUS uses only as much memory as needed to hold the RID map, up to the calculated value.

If you specify a nonzero value for RIDMAPMEM, REORG PLUS first ensures that the value that you supplied is sufficient to satisfy the minimum storage requirements. If the value is not sufficient, REORG PLUS issues message BMC50887I and changes the value for RIDMAPMEM to the minimum required storage amount.

To calculate the RIDMAPMEM value, use the formula described in “Using formulas for the calculations” on page 661. However, if you specify a value greater than the result of multiplying the RIDMMAXD value by the RIDMDSSZ value, REORG PLUS reduces the value that you specified to the product of those two installation options.

During execution, REORG PLUS allocates memory only as needed, up to the current value of RIDMAPMEM. If RIDMAPMEM is insufficient, REORG PLUS issues message BMC50885I and spills to the spill data set.

For more information, see the following sections:

- how the spill options work together to allocate the spill data sets, see “Allocation of spill data sets” on page 648
- calculating the data space size, see “Sizing memory for the RID translation map” on page 660
- RIDMMAXD and RIDMDSSZ installation options, see page 718

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.
**Specifying the default**
You can specify the default for the RIDMAPMEM command option in your installation options module by using the RMAPMEM installation option (page 718). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.

**LOGMEM**

The LOGMEM option specifies the amount of data space memory (in kilobytes) needed to hold the internal structures that REORG PLUS uses to store the log records. The number must be 0 or a positive integer. BMC strongly recommends that you specify 0.

**Additional considerations**
The following considerations apply to this option:

- When invoking DSNUTILB, REORG PLUS ignores this option.
- If you specify 0, REORG PLUS automatically computes the amount of data space memory that it needs. To do so, REORG PLUS uses the formula described in “Using formulas for the calculations” on page 663.
- If you specify a nonzero value for LOGMEM, during execution REORG PLUS first ensures that your specified value is sufficient to satisfy the minimum memory requirements. If the value is not sufficient, REORG PLUS issues message BMC50864I and changes the value for LOGMEM to the minimum required memory amount.
- For information about how the spill options work together to allocate the spill data sets, see “Allocation of spill data sets” on page 648. For performance information, see “Sizing memory for log records” on page 662.

**Specifying the default**
You can specify the default for the LOGMEM command option in your installation options module by using the LOGMEM installation option (page 706). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.
LOGSPILL

The LOGSPILL option determines the primary and secondary space allocation (in kilobytes) for the spill data sets that hold the log records. Each number can be a nonzero positive integer.

For information about how the spill options work together to allocate the spill data sets, see “Allocation of spill data sets” on page 648.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the LOGSPILL command option in your installation options module by using the LOGSPIL installation option (page 707). REORG PLUS was shipped with a default value of (20000,10000) for this option. The command option overrides the default that is in the installation options module.

LOGFINAL

The LOGFINAL option allows you to specify the time that you want REORG PLUS to start the LOGFINAL phase. This option enables you to control when updates are prevented to the table space or index that you are reorganizing.

If a timestamp or time value is specified on DEADLINE and LOGFINAL, the calculated LOGFINAL timestamp must be less than the calculated DEADLINE timestamp.

REORG PLUS does not check the LOGFINAL value until the LOGAPPLY phase begins. At that point, if the current time is greater than the time specified on the LOGFINAL option, REORG PLUS immediately begins the LOGFINAL phase.

You can use the XBM Utility Monitor function or the MVS operator console to dynamically change the LOGFINAL values while the reorganization is in progress. For more information, see “Using XBM to view and dynamically control the log apply process” on page 639.

The LOGFINAL option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 633.
Restriction
When invoking DSNUTILB, REORg PLUS ignores this option.

Specifying the default
You can specify the default for the LOGFINAL command option in your installation options module by using the LOGFINAL installation option (page 706). REORg PLUS was shipped with a default value of NONE for this option. The command option overrides the default that is in the installation options module.

NONE

The value of NONE specifies that there is no set time by which the LOGFINAL phase must start. Instead, the other log apply control options control the starting time. For more information, see “Hierarchy of log apply control options” on page 636.

timestamp

When the current timestamp exceeds the value specified on this option, REORg PLUS starts the LOGFINAL phase.

Specifying a time zone
You can specify a time zone with this timestamp, indicating its difference from Coordinated Universal Time (UTC), by using a value from -12:59 through +14:00. If you specify a time zone, REORg PLUS converts your timestamp to UTC plus or minus the difference that you specify. The actual time that triggers the LOGFINAL phase depends on local time when the reorganization runs.

For example, if you specify a timestamp of 2012-09-25-18.00.00.000000+02:00, REORg PLUS converts this value to 20:00 UTC. If the reorganization is running under North American Central Time when daylight saving time is in effect, the LOGFINAL phase will start at 15:00, based on the local time’s UTC offset.

time

This option enables you to specify a time to begin the LOGFINAL phase within the next 24 hours by specifying the time relative to the current time. Specify the time value in hh:mm:ss format, using a 24-hour clock.

REORg PLUS replaces the time portion of the current system timestamp with the time specified to create a new timestamp. If the new timestamp value is less than the current system timestamp, REORg PLUS adds one day to the timestamp. REORg PLUS then uses this calculated timestamp value to determine when to start the LOGFINAL phase.
DRAIN

The DRAIN option determines the type of drain that REORG PLUS issues when it enters the LOGFINAL phase, after reaching the MAXRO threshold.

- ALL tells REORG PLUS to drain all readers and writers.

  BMC recommends that you specify DRAIN ALL if either of the following conditions exists:
  
  — SQL queries might be running that do not commit often enough to allow a drain of the readers to successfully complete in a timely fashion.
  
  — SQL updaters might be running that require a minimum possible outage, and these updaters might time out if they are all held while awaiting the readers to be drained.

- WRITERS tells REORG PLUS to drain only writers.

Specifying the default
You can specify the default for the DRAIN command option in your installation options module by using the DRAINTYP installation option (page 686). REORG PLUS was shipped with a default value of ALL for this option. The command option overrides the default that is in the installation options module.

MAPPINGTABLE

This option specifies the mapping table that REORG PLUS provides to the IBM DB2 REORG utility when invoking DSNUTILB for SHRLEVEL CHANGE table space reorganizations. This option is meaningful only when REORG PLUS invokes DSNUTILB. Otherwise, REORG PLUS treats this option as a comment.

MAPPINGDATABASE

(DB2 Version 11 or later) This option specifies the mapping database that REORG PLUS provides to the IBM DB2 REORG utility as an override to the value of the DB2 subsystem parameter REORG_MAPPING_DATABASE. This option is meaningful only for SHRLEVEL CHANGE table space reorganizations when REORG PLUS invokes DSNUTILB. Otherwise, REORG PLUS treats this option as a comment.
Building and executing REORG PLUS jobs

This chapter presents the following topics:

Building the REORG PLUS job ............................................................... 341
   JOB statement ................................................................. 342
   EXEC statement ............................................................... 342
   REGION parameter ......................................................... 343
   Utility parameters on the EXEC statement .............. 343
   STEPLIB DD statement .................................................... 348
   REORG PLUS DD statements ............................................. 348
   ANALYZE option for estimating data set allocation ... 386
Running REORG PLUS jobs .............................................................. 390
   Invoking REORG PLUS ....................................................... 391
   Restarting REORG PLUS ..................................................... 391
   Terminating or canceling a job ......................................... 396
   Recovering the DB2 object after terminating or canceling a job ... 397
   Recovering from a failure .................................................... 398
   Not completing the reorganization in the UTILTERM phase .... 400

Building the REORG PLUS job

Building a job for the REORG PLUS product involves creating a set of JCL that includes the following elements:

- a JOB statement (page 342)
- an EXEC statement with the appropriate utility parameters (page 342)
- STEPLIB or JOBLIB DD statements as needed (page 348)
JOB statement

- DD statements as needed for the appropriate number and size of data sets for the data structure that you are reorganizing (page 348)

- REORG PLUS control statements using the appropriate command syntax (page 170)

For examples of REORG PLUS JCL and SYSPRINT output, see Chapter 5, “Examples of REORG PLUS jobs.”

JOB statement

Include a REORG PLUS JOB statement that conforms to your site’s standards. You can include the REGION parameter on either your JOB statement or your EXEC statement. For recommendations, see “REGION parameter” on page 343.

EXEC statement

The REORG PLUS EXEC statement specifies the module to be run for the REORG PLUS utility. The REORG PLUS module name is ARUUMAIN. The EXEC statement also specifies the utility parameters, which are described in “Utility parameters on the EXEC statement” on page 343.

You can include the REGION parameter on either your EXEC statement or your JOB statement. For recommendations, see “REGION parameter” on page 343.

When you are reorganizing a large number of partitions and dynamic allocation is active, BMC recommends that you specify DYNAMNBR=1600 on the EXEC statement of your execution JCL. Including this parameter enables the system to acquire adequate resources immediately before their use and to release them immediately after use.
REGION parameter

Include the REGION parameter on either your JOB statement or your EXEC statement to specify the region size (the amount of virtual storage used by the utility). For the best performance, BMC recommends that you specify REGION=0M to allocate the optimal amount available virtual storage to the REORG PLUS job. If your data center does not permit you to specify REGION=0M, specify the amount that allows the most virtual storage both above and below the 16-megabyte line. Using a region size that is less than optimal risks the following potential issues:

- running less efficiently, which could result in additional CPU and elapsed time
- encountering memory failures or jobs that fail when new versions implement changes that require additional memory

**NOTE**
If you specify a value for REGION other than 0M, ensure that you have an appropriate value set for the MEMLIMIT parameter, either as your site’s default SMF option or on your JOB statement or EXEC statement.

BMC makes the following recommendations for the MEMLIMIT option:

- Specify NOLIMIT to allow unlimited above-the-bar memory.
- If you are unable to specify NOLIMIT, specify at least 4 GB; if you are reorganizing LOB or XML data, specify at least 32 GB.

Utility parameters on the EXEC statement

The REORG PLUS EXEC statement includes the following utility parameters:

- DB2® subsystem ID or group attachment name
- utility ID
- restart parameter
- TSO user ID for notification of progress made on the reorganization
- message level
- installation options module parameter

The following illustration shows the format of the EXEC statement:

```bash
//stepname EXEC PGM=ARUUMAIN,
//   PARM='ssid,utilityID,restartParameter.userID,MSGLEVEL(n),optionsModule'
```
The REORG PLUS utility parameters are positional. If you do not specify a value for a parameter (thus allowing the default value), and additional parameters follow, you must substitute a comma for that parameter. The comma indicates that you omitted a parameter.

**DB2 subsystem identifier (SSID)**

This parameter specifies the four-character DB2 subsystem ID where the object resides.

If you do not specify the SSID, REORG PLUS uses the DB2 installation default from the DSNHDECP module. REORG PLUS depends on the application defaults module being named DSNHDECP. If you do not specify an SSID and REORG PLUS cannot find a module named DSNHDECP in your LINKLIST or STEPLIB, REORG PLUS terminates.

Note the following considerations when running REORG PLUS in a data sharing environment:

- REORG PLUS supports the DB2 group attachment name capability. When you supply a group attachment name as the SSID, REORG PLUS uses it to connect all plans. REORG PLUS then determines the actual DB2 SSID from within that group to use for the current reorganization.

- When you are restarting in a data sharing environment, REORG PLUS can use either the same member that was chosen in the original reorganization or any other member in the specified group.

- When both of the following conditions exist, specify a member SSID for your reorganization job instead of a group attachment name. Ensure that the DB2 version of the SSID that you specify corresponds to the version of the DB2 load library that you specify in your STEPLIB.

  — You are using table space compression.
  — The subsystems in your data sharing group are not all at the same DB2 version level.

**Utility identifier (utility ID)**

This parameter specifies the 1- to 16-character utility ID that gives a unique name to a utility job. If you omit this parameter, REORG PLUS uses the default, `userID.jobName`. Each BMC utility job should have a unique ID.
Restart parameter

The restart parameter can have one of the values described in this section. For more information about specific circumstances for which you should use a particular value, see “Restarting REORG PLUS” on page 391.

DSNUTILB reorganizations
After the UTILINIT phase, REORG PLUS passes the restart parameter to DSNUTILB. Restart processing is handled by the IBM DB2 REORG utility.

Blank or not specified

By not specifying a restart parameter, REORG PLUS initiates a new BMC utility job. The utility ID that you specify cannot currently exist in the BMCUTIL table.

RESTART

Specifying this value restarts a BMC utility from the last restart sync point. REORG PLUS takes restart sync points as each phase completes and as the processing of each DB2 object completes. The utility ID must exist in the BMCUTIL table. If the utility ID does not exist in the BMCUTIL table, REORG PLUS ends with return code 8.

RESTART(PHASE)

This value restarts a BMC utility at the beginning of the last incomplete phase. The utility ID must exist in the BMCUTIL table. If the utility ID does not exist in the BMCUTIL table, REORG PLUS ends with return code 8.

NEW

NEW initiates a new BMC utility job or replaces an existing utility ID. Specifying this value keeps you from having to end the utility ID separately and then start a utility specifying a blank restart parameter.

WARNING

Use care when you specify NEW. When you replace or terminate a utility ID for REORG PLUS, you might lose data from the UNLOAD or REORG phase, and you cannot restart the utility. For more information, see “Terminating or canceling a job” on page 396.
If you specify NEW and the utility ID has a status of X (executing), REORG PLUS issues error message BMC50012E and ends with return code 8.

**NEW/RESTART**

BMC recommends that you specify NEW/RESTART for the restart parameter value.

REORG PLUS takes restart sync points as it completes each phase and as it completes the processing of each DB2 object. If the utility ID exists, NEW/RESTART restarts the utility from the last restart sync point. Otherwise, this value starts the utility as NEW.

Using NEW/RESTART enhances restartability for a SHRLEVEL CHANGE reorganization. REORG PLUS treats NEW/RESTART the same as NEW if the previous run was a SHRLEVEL CHANGE reorganization that failed before the UTILTERM phase.

**NEW/RESTART(PHASE)**

This value restarts the utility at the beginning of the last incomplete phase if the utility ID exists. Otherwise, this value starts the utility as NEW.

**TERM**

Specifying this value terminates an existing utility, removes the utility ID from the BMCUTIL table, and removes the corresponding rows from the BMCSYNC and BMCDICT tables. After removing all sync point and restart information, REORG PLUS terminates without performing a reorganization. REORG PLUS terminates with return code 0, regardless of whether the utility ID exists.

When invoking DSNUTILB, REORG PLUS also terminates an existing DB2 utility ID.

The minimum JCL that is required when specifying TERM is the STEPLIB to the REORG PLUS load library and the SYSPRINT DD statement.

---

**WARNING**

Use care when you specify TERM. When you terminate a utility ID for REORG PLUS, you might lose existing data from the UNLOAD or REORG phase, and you cannot restart the utility. Similar consequences can occur in other BMC utilities. Consult the reference manual for the specific BMC utility. For more information for REORG PLUS, see “Terminating or canceling a job” on page 396.
MAINT

Specifying this value forces MSGLEVEL(1) and causes REORG PLUS to print the following information:

- an options module report that lists the values in the installation options module that you are using for this reorganization
- the values in the DSNHDECP module that REORG PLUS uses
- a summary report of all REORG PLUS fixes that you have applied

When you specify the MAINT parameter, the job ends without affecting any utility that is running.

For this parameter, you need only minimal JCL. Your JCL must include at least a SYSPRINT DD statement and STEPLIB to the REORG PLUS and DB2 load libraries.

User identifier (user ID)

This parameter specifies the TSO user ID that REORG PLUS notifies after it completes each phase and after it completes the command execution.

Message level (MSGLEVEL)

This parameter controls which messages REORG PLUS returns to the user in the SYSPRINT and SYSPRIN2 data sets. MSGLEVEL(0) returns minimal messages. MSGLEVEL(1) returns additional messages to help you diagnose problems and fine-tune performance.

You can use the MSGLEVEL installation option to set the default value of this parameter. For details, see Appendix A, “REORG PLUS installation options.”

Installation options module

The options module parameter allows you to identify which installation options module to use. If you include this parameter, you must specify the full name of the options module. If you omit this parameter, REORG PLUS uses the default installation options module, ARU$OPTS.

For more information about installation options, see Appendix A, “REORG PLUS installation options.” In addition, the Installation System User Guide explains how to create multiple installation options modules.
**STEPLIB DD statement**

The REORG PLUS STEPLIB DD statement must specify the following libraries, unless they are included in your system’s LINKLIST or in a JOBLIB statement:

- load library that contains the files (including the options modules) for the following BMC products and components:
  - REORG PLUS
  - BMCSORT (AUP)
  - Common Statistics (ATS)
  - DB2 Utilities Common Code (D2U)
  - DB2 Solution Common Code (SCC)

- SAS/C transient library, if you specify BMCSTATS YES, BMCSTATS REPORT, or UPDATEDB2STATS YES

- BMCPSWD and ASUBMAIN libraries, if you are updating statistics in the DASD MANAGER PLUS statistics tables

- libraries that contain any DB2 user exits (EDITPROCs, VALIDPROCs, FIELDPROCs, and user-written exit routines)

- DB2 load library

All load libraries in the STEPLIB or JOBLIB concatenation must be APF authorized.

**REORG PLUS DD statements**

REORG PLUS uses data sets specified by ddnames. This section provides specification guidelines, allocation information, and usage notes for each data set that REORG PLUS uses. Use Table 50 on page 349 to find information quickly about a specific data set. This table also provides a quick reference to associated command and installation options for each data set type.
### Table 50 Data set type descriptions and quick command reference (part 1 of 2)

<table>
<thead>
<tr>
<th>Data set type</th>
<th>Description reference</th>
<th>Default ddname(^a)</th>
<th>DDTYPE option keyword (page 294 or page 741)</th>
<th>ddname or prefix installation option</th>
<th>ddname or prefix command option</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER statement output</td>
<td>page 365</td>
<td>DDLOUT</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>archive</td>
<td>page 370</td>
<td>SYSARC</td>
<td>ARCHIVE</td>
<td>ARCHDDN (page 675)</td>
<td>ARCHDDN (page 185)</td>
</tr>
<tr>
<td>command input</td>
<td>page 376</td>
<td>SYSIN</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>copy, full</td>
<td>page 352</td>
<td></td>
<td>BCPY (local primary)</td>
<td>COPYDDN (primary copies) (page 680)</td>
<td>RECOVERYDDN (remote copies) (page 273)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BCPZ (local backup)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BRCY (remote primary)</td>
<td>RCVYDDN (remote copies) (page 716)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BRCZ (remote backup)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>copy, incremental</td>
<td>page 653</td>
<td></td>
<td>BICY (local primary)</td>
<td>ICDDN (page 696)</td>
<td>ICDDN (page 275)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BICZ (local backup)</td>
<td>RCVICDDN (page 716)</td>
<td>RECOVERYICDDN (page 277)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BIRY (remote primary)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BIRZ (remote backup)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>discard (DSNUTILB only)</td>
<td>page 370</td>
<td>SYSARC(^b)</td>
<td>ARCHIVE</td>
<td>ARCHDDN(^c) (page 675)</td>
<td>ARCHDDN(^c) (page 185)</td>
</tr>
<tr>
<td>error</td>
<td>page 373</td>
<td>SYSERR</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>REXX exits</td>
<td>page 373</td>
<td>SYSEXEC</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>IDCAMS input</td>
<td>page 374</td>
<td>SYSIDCIN</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>IDCDDN (page 214)</td>
</tr>
<tr>
<td>index work files</td>
<td>page 382</td>
<td>SYSUT1</td>
<td>WORK</td>
<td>WORKDDN(^c) (page 737)</td>
<td>WORKDDN(^c) (page 183)</td>
</tr>
<tr>
<td>input</td>
<td>page 360</td>
<td>DDLIN</td>
<td>Not applicable</td>
<td>DDLDDN (page 683)</td>
<td>DDLDDN (page 187)</td>
</tr>
<tr>
<td>LOAD statements (DSNUTILB only)</td>
<td>page 378</td>
<td>SYSPUNCH(^b)</td>
<td>SYSPUNCH</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>message output</td>
<td>page 382</td>
<td>SYSTEM</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>
Table 50  Data set type descriptions and quick command reference (part 2 of 2)

<table>
<thead>
<tr>
<th>Data set type</th>
<th>Description reference</th>
<th>Default ddname&lt;sup&gt;a&lt;/sup&gt;</th>
<th>DTYPE option keyword (page 294 or page 741)</th>
<th>ddname or prefix installation option</th>
<th>ddname or prefix command option</th>
</tr>
</thead>
<tbody>
<tr>
<td>message output</td>
<td>page 376</td>
<td>SYSPRINT</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>message output</td>
<td>page 377</td>
<td>SYSPRINT2</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>other</td>
<td>page 386</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>REXX statement output</td>
<td>page 382</td>
<td>SYSTSPRT</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>sort message output indicator</td>
<td>page 386</td>
<td>UTPRINT</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>sort work files</td>
<td>page 366</td>
<td>SORTWK</td>
<td>SORTWORK</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>statistics report output</td>
<td>page 351</td>
<td>ASUSRPRT</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>thread cancelation report output</td>
<td>page 352</td>
<td>BMCFORCE</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>unload</td>
<td>page 379</td>
<td>SYSREC</td>
<td>UNLOAD</td>
<td>UNLDDN&lt;sup&gt;c&lt;/sup&gt; (page 734)</td>
<td>UNLDDN&lt;sup&gt;c&lt;/sup&gt; (page 182)</td>
</tr>
</tbody>
</table>

<sup>a</sup> This book refers to the data sets by their default ddnames. The name that you use for the data sets will be different if you change them by using the ddname or prefix installation or command options.

<sup>b</sup> This value is the default ddname that REORG PLUS passes to the IBM DB2 REOR utility for the &DDNAME variable of the data set name pattern.

<sup>c</sup> For a DSNUTILB reorganization, REORG PLUS uses these options only to pass a value for the &DDNAME variable of the data set name pattern.
Methods for allocating copy and work data sets

You can use one of the following methods to allocate your copy and work data sets and determine the appropriate size for those data sets:

- (recommended) Have REORG PLUS dynamically allocate the following data sets by using the DDTYPE installation or command option (page 741 or page 294).

  **NOTE**
  Dynamic allocation of the work and copy data sets is required for a DSNUTILB reorganization.

  — copy (page 352 and page 653)
  — SORTWK (page 366)

  **NOTE**
  You can have either BMCSORT or REORG PLUS dynamically allocate your sort work data sets. Unless you are running a DSNUTILB reorganization, BMC recommends that you have BMCSORT dynamically allocate these files.

  — SYSREC (page 379)
  — SYSUT1 (page 382)

  For more information about dynamic allocation, see “Dynamic data set allocation” on page 90 and “Dynamic allocation options” on page 293.

- Specify ANALYZE PAUSE or ANALYZE ONLY to have REORG PLUS estimate the space that is needed for your work data sets. For more information about these options, see “ANALYZE option for estimating data set allocation” on page 386.

- Calculate the space allocation based on your knowledge of the table space that you are reorganizing and the information in this section.

ASUSRPRRT data sets

You can specify an ASUSRPRRT DD statement in your JCL to allocate a data set to contain the statistics reporting output from the Common Statistics component.

**NOTE**
The Common Statistics component writes informational and diagnostic messages to the REORG PLUS SYSPRINT, regardless of whether you specify this DD statement in your JCL.
REORG PLUS DD statements

Alternatively, you can use one of the following options for your statistics output:

- Send it to the REORG PLUS SYSPRINT.

To use this method, do not specify an ASUSRPRT DD statement in your JCL. REORG PLUS automatically sends the report to your REORG PLUS SYSPRINT.

**NOTE**

The Common Statistics component does not send this output to a REORG PLUS SYSPRINT2 data set.

- Suppress the output.

To suppress the output, specify `//ASUSRPRT DD DUMMY` in your JCL.

**BMCFORCE data sets**

REORG PLUS generates a thread cancelation report when you specify one of the following options:

- FORCE REPORTONLY
- FORCE_RPT YES with FORCE READERS or FORCE ALL

You can specify a BMCFORCE DD statement in your JCL to allocate a data set to contain the thread cancelation report output. If you do not specify a BMCFORCE DD statement in your JCL, REORG PLUS sends the report to your REORG PLUS SYSPRINT.

**Copy data sets**

REORG PLUS creates one or more output copy data sets when you specify COPY YES. These output data sets contain a DSN1COPY-type copy or image copy of a nonpartitioned table space, all partitions of a partitioned table space, or selected partitions of a partitioned table space.

For a two-phase reorganization, REORG PLUS makes the copies during the RELOAD phase. For a single-phase reorganization, REORG PLUS makes the copies during the REORG phase. REORG PLUS creates all copies except inline image copies with system pages at the beginning of the data set (in the same way that the IBM DB2 COPY utility creates image copies when you specify SYSTEMPAGES YES).
The copy ddnames identify the output data sets that will contain the copy (after the reorganization) either of the table space, of each partition in the table space, or of the specified subset of partitions. The number of copies that REORG PLUS makes when you specify COPY YES depends on the following information:

- *(with dynamic allocation)* the DDTYPE and COPYLVL specifications in your installation and command options
- *(without dynamic allocation)* the existence of the copy ddnames in your JCL

REORG PLUS provides the following default names for each type of copy data set. This book uses the default name to refer to that type of copy data set.

### Table 51 Default copy data set names

<table>
<thead>
<tr>
<th>Default name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCPYnn</td>
<td>local primary</td>
</tr>
<tr>
<td>BCPZnn</td>
<td>local backup</td>
</tr>
<tr>
<td>BRCYnn</td>
<td>remote primary</td>
</tr>
<tr>
<td>BRCZnn</td>
<td>remote backup</td>
</tr>
</tbody>
</table>
| BICYnn         | local primary
incremental |
| BICZnn         | local backup
incremental |
| BRCYnn         | remote primary
incremental |
| BRCZnn         | remote backup
incremental |

When you are reorganizing either all partitions or a single subset of contiguous partitions of a partitioned table space, you can also specify how REORG PLUS assigns the dynamically allocated full and incremental copy data sets, as described in “COPYLVL” on page 269.

### Requirements

REORG PLUS requires the following copy data sets when you specify COPY YES:

- For all other reorganization jobs, BCPY is required.
- For all reorganization jobs, if you specify BRCZ, BRCY is also required.
- For a SHRLEVEL CHANGE reorganization, REORG PLUS might require a matching incremental copy data set for each copy data set defined. For more information, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 653.

When invoking DSNUTILB, REORG PLUS requires that you dynamically allocate your copy data sets.
**Overriding the default ddnames**

To override the default copy ddnames or ddname prefixes that are specified in your installation options module, see the information in Table 50 on page 349.

**Registration**

The ddname itself controls the registration information that is placed in the ICBACKUP column of SYSIBM.SYSCOPY when you specify either ALL (the default) or specific ddnames for the REGISTER option.

---

**NOTE**

For a SHRLEVEL CHANGE reorganization, REORG PLUS forces COPY YES REGISTER ALL, even if you specify NONE or a list of ddnames.

---

**Methods for allocating copy data sets**

You can use one of the methods described in “Methods for allocating copy and work data sets” on page 351 to allocate your copy data sets.

---

**NOTE**

When invoking DSNUTILB, REORG PLUS requires that you dynamically allocate your copy data sets.

---

REORG PLUS determines the optimal block size of the copy data sets based on the device type that contains the data set. Copy data sets can be on different device types. However, if additional copy data sets (for example, remote backup copy data sets) for the same object are on different device types, the block size for all copy data sets for that object is the block size that REORG PLUS determined was optimal for the primary local copy.

**Dynamically allocating copy data sets**

Based on the information in Table 52 on page 356, specify dynamic allocation options that result in the same number and type of copy data sets as if you specified DD statements in your JCL.
Restriction
REORG PLUS does not allow dynamically allocated copy data sets to be stacked on tape.

Additional considerations
Note the following additional considerations:

- For full and incremental copy data sets, REORG PLUS determines the size of the dynamically allocated files based on the high-used RBA of the table space that you are reorganizing.

- When dynamically allocating copy data sets for partition-by-growth table spaces, REORG PLUS uses the DSSIZE value to estimate the size. For more information about copy data sets for partition-by-growth table spaces, see “Partition-by-growth table spaces” on page 358.

- If you specify DELETEFILES YES and the job terminates after a failure, REORG PLUS deletes any unregistered full and incremental copy data sets. For more information, see “DELETEFILES” on page 215.

- You can decide when to delete the dynamically allocated copy data sets by specifying an expiration date with the EXPDT option or by specifying a retention period with the RETPD option. For more information about these options, see “EXPDT” on page 322 and “RETPD” on page 322.

- For information about partition-level copies, see “Partition-level copies” on page 357 and “Partition-by-growth table spaces” on page 358.

For more information about dynamic allocation, see “Dynamic data set allocation” on page 90.

Allocating copy data sets in your JCL

If you do not use dynamic allocation, you must allocate the copy data sets in your JCL by specifying DD statements as described in Table 52 on page 356. Do not specify separate data sets for nonpartitioned, multi-data-set table spaces.
The size that REORG PLUS needs for the copy data sets depends on the number of pages required when the table space is reorganized. REORG PLUS calculates the number of required pages and issues message BMC51286I, which includes the number of pages. If you specify UNLOAD PAUSE, you can use information from the ANALYZE phase to allocate the primary amount as the total amount. No secondary amount is needed. Use Table 53 on page 357 to determine the value to use for your data set allocation.

---

### Table 52 DD statements required when allocating copy data sets in your JCL

<table>
<thead>
<tr>
<th>Table space being copied</th>
<th>Type of reorganization</th>
<th>Copy DD statements required</th>
</tr>
</thead>
<tbody>
<tr>
<td>nonpartitioned</td>
<td>any</td>
<td>Specify one DD statement for each copy type that you want REORG PLUS to make. Do not use ( nn ) in the ddname.</td>
</tr>
</tbody>
</table>
| partitioned, including partition-by-growth | full (entire table space) | - Specify one DD statement for each copy type that you want REORG PLUS to make. Do not use \( nn \) in the ddname.  
  
  or  
  
  - For each partition, specify one DD statement for each copy type that you want REORG PLUS to make. Use \( nn \) for all ddnames, where \( nn \) matches the partition number.  
  
  If necessary for partition-by-growth table spaces, specify additional DD statements as discussed in “Partition-by-growth table spaces” on page 358 |
| partial (selected partitions using PART option) | - Specify one DD statement for each copy type that you want REORG PLUS to make. Do not use \( nn \) in the ddname. This option is valid only when you specify a single subset of physically contiguous partitions.  
  
  or  
  
  - For each specified partition, specify one DD statement for each copy type that you want REORG PLUS to make. Use \( nn \) for all ddnames, where \( nn \) matches the partition number. (The \( nn \) is not required if you are performing a partial reorganization with only one partition.)  
  
  If necessary for partition-by-growth table spaces, specify additional DD statements as discussed in “Partition-by-growth table spaces” on page 358 |

\( a \) The value of the COPYSUBSET installation option must be YES.
Restrictions

Note the following restrictions when allocating copy data sets in your JCL:

- REORG PLUS does not support updating SMS-managed striped data sets for copy data sets if you specify the following options:
  - ICTYPE=AUTO (command or installation option)
  - ICTYPE=UPDATE (command or installation option)
  - INLINE NO (command option)
  - INLINECP=NO (installation option)

- Copy data sets cannot be temporary data sets. For information about how REORG PLUS defines a temporary data set, see “Work file validity and integrity checks” on page 96.

- BMC recommends that you do not specify DISP=MOD for an image copy data set because REORG PLUS does not reset the data set to empty but appends data to any data already present in the file.

Partition-level copies

Note the following additional information when allocating partition-level copies.

Naming partition-level copies

Use the $nn$ as part of the ddname only if you are reorganizing a partitioned table space and you want a separate data set for each partition.

JCL rules limit ddnames to eight characters. If you are making partition-level copies, the ddname has two parts: the ddname prefix (minimum of one character) and the partition number $nn$ (one to seven characters). The ddname that you specify in the installation options or the command options becomes the ddname prefix.

<table>
<thead>
<tr>
<th>Table space type</th>
<th>Number of copy data sets</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>nonpartitioned</td>
<td>Not applicable</td>
<td>Multiply the number of pages required for reorganizing by the page size of the table space.</td>
</tr>
<tr>
<td>partitioned</td>
<td>single</td>
<td>Multiply the total number of pages for all reorganized partitions by the page size of the table space.</td>
</tr>
<tr>
<td></td>
<td>multiple</td>
<td>For each data set, multiply the number of pages for that partition by the page size of the table space.</td>
</tr>
</tbody>
</table>
Note the following information about the partition number:

- If you are dynamically allocating copy or incremental copy data sets (and you specify COPYLVL PART to make partition-level copies), REORG PLUS appends the partition number to the ddname prefix. If you are reorganizing a table space that contains more than 99 partitions, use the copy or incremental copy options to specify a prefix that results in eight characters or less after REORG PLUS appends the highest partition number. For more information, see “Specifying ddname prefixes” on page 93.

- If you are allocating copy data sets in your JCL, you must append the partition number $nn$ to the ddname prefix. The $nn$ must match the partition that you are copying, and you must allow sufficient bytes for REORG PLUS to add the partition number to the prefix and still have a valid ddname (eight characters or less).

**Large number of partitions**

If you are reorganizing a large number of partitions, consider one of the following options to avoid encountering data set allocation restrictions of the operating system or REORG PLUS memory restrictions:

- Limit the number of copies per partition.
- If your site’s recovery strategy allows for full copies, use one of the following options:
  - If you are reorganizing all partitions, dynamically allocate your copy data sets and create a single copy by specifying COPYLVL FULL.
  - If you are reorganizing a subset of partitions, create a single copy by specifying the following options:
    - a single physically contiguous range of partitions
    - COPYSUBSET=YES in the installation options module
    - if you are dynamically allocating your copy data sets, COPYLVL FULL on your REORG command

**Partition-by-growth table spaces**

For partition-by-growth table spaces, REORG PLUS requires the allocation of additional data sets when you are making partition-level copies and either REORG PLUS can extend the table space or you are performing a full SHRLEVEL CHANGE table space reorganization.
For partition-by-growth table spaces, you need copy data sets as described in Table 52 on page 356, plus additional data sets equal to the smaller of the following values:

- MAXPARTITIONS minus the number of partitions in the table space
- MAXNEWPARTS

Although REORG PLUS requires that the additional data sets be allocated, it only uses them if partitions are added to the table space during the reorganization. REORG PLUS deletes unused data sets after the reorganization.

When dynamically allocating these data sets, REORG PLUS uses the DSSIZE value to allocate the data set for each partition. The following example illustrates the additional data set requirements using copy data sets.

**Example**
The following example illustrates the additional data set requirements using copy data sets. Assume the following scenario:

- You are performing a partial reorganization of a table space with four partitions.
- The value of MAXPARTITIONS is 6.
- Your SYSIN data set contains the following specifications:

  ```
  REORG TABLESPACE databaseName.tableSpaceName
  PART 3-LAST
  MAXNEWPARTS 3
  COPYDDN BMCCPY
  ```

In this example, REORG PLUS requires that four copy data sets be defined in the JCL or through dynamic allocation:

- one for each of the two partitions that are participating in the reorganization (partitions 3 and LAST)
- two additional for partitions that might be added

  That is, MAXPARTITIONS (6) minus total partitions in the table space (4) which equals 2. This value is smaller than the MAXNEWPARTS value (3).

For more information about partition-by-growth table spaces, see page 125.
**Improving performance**

When reorganizing a partitioned table space, REORG PLUS multitasks the image copy process whenever possible, thus decreasing the elapsed time needed to run the reorganization. To facilitate optimal multitasking, provide a unique full copy data set for each partition that you are reorganizing and ensure that each copy data set resides on DASD. If you are performing a SHRLEVEL CHANGE reorganization, use AUTO for the ICTYPE command or installation option.

If you decide to store your full copy data sets on tape, you will improve performance if you do not stack multiple files on a single tape.

You can improve performance by having REORG PLUS create inline image copies. The difference between inline image copies and standard image copies or DSN1COPY-type copies is that REORG PLUS creates inline copies as it reloads the table space, rather than after it reloads the table space. This can reduce the elapsed time of the reorganization. For details about how to create inline image copies, see “Copy options for REORG TABLESPACE” on page 264.

**DDLIN data set**

The DDLIN input data set contains the SQL ALTER INDEX statements or, for table-controlled partitioning, ALTER TABLE statements, with the new limit key values to use to rebalance partitions. REORG PLUS uses this optional data set only when performing a table space reorganization.

REORG PLUS does not explicitly execute the SQL statements in the DDLIN data set. Instead, it extracts the partition number and the associated limit key values from the statements. REORG PLUS reorganizes the data in the partitions based on the new key values and alters the limit keys after reloading the object. Using the DDLIN data set, you can rebalance up to 255 rebalance groups in a single execution of REORG PLUS.

The data set’s attributes must be specified as fixed length (RECFM is F, FB, or FBS), and the record length must be 80 columns (LRECL=80). REORG PLUS uses only columns 1 through 72.

For SHRLEVEL CHANGE and SHRLEVEL REFERENCE, you can use the ALTRFAIL installation option (page 674) to tell REORG PLUS what to do when failures occur during processing of the ALTER statements in your DDLIN data set.

**Guidelines for using the DDLIN data set**

REORG PLUS processes the DDLIN data set according to the following rules:

- REORG PLUS ignores any ALTER INDEX or ALTER TABLE statements that refer to an index that is not part of the reorganization.
REORG PLUS ignores any ALTER TABLE statements for an index-controlled table space.

If you specify the same partition number on more than one ALTER INDEX or ALTER TABLE statement, REORG PLUS uses the limit key value from the last statement that it found.

Character, hexadecimal, and graphic string constants are limited to a length of 256 bytes.

REORG PLUS does not support columns that use a FIELDPROC in SQL ALTER INDEX or ALTER TABLE statements in the DDLIN data set.

Restrictions

REORG PLUS ignores the rebalance request when reorganizing partition-by-growth table spaces.

REORG PLUS terminates when any of the following conditions exists:

- You specify any of the following items in the DDLIN data set:
  - the same partition number more than once on the same ALTER INDEX or ALTER TABLE statement, as shown in the following example:

```
ALTER INDEX USER1.TBL1INX
  PART 1 VALUES (X'11'),
  PART 2 VALUES (X'22'),
  PART 2 VALUES (X'11'),
  PART 3 VALUES (X'33'),
  PART 4 VALUES (X'44')
; 
```

- any of the following constants as a limit key value:
  - floating-point
  - decimal floating-point
  - XML
  - LOB
  - row ID
  - graphic types
  - binary strings

- a limit key constant that spans a line

- an invalid limit key value
REORG PLUS DD statements

— a limit key value in VARGRAPHIC format (‘G’ or ‘N’ in front of a double-byte string)

— a limit key value that is a timestamp with a precision greater than the precision defined on the column

■ You are using the DDLIN data set to rebalance a table space that has pending DDL changes, or an index on that table space has pending DDL changes.

■ You are using the DDLIN data set to rebalance one of the following types of table spaces:
  — on DB2 Version 11, table spaces that are defined with table-controlled partitioning
  — XML table spaces
  — LOB table spaces
  — table spaces that contain a LOB column
  — table spaces for clone objects or base objects that participate (or have participated) in a clone relationship

■ The table space that you are rebalancing contains an XML column, and an ALTER statement in the DDLIN data set would alter the last partition of one of the following types of table spaces:
  — a table space that uses table-controlled partitioning
  — a table space that is defined with the LARGE or DSSIZE attribute

■ The database containing the object to be reorganized is not in read-write (RW) status.

■ The DDLIN data set contains any of the following items:
  — any SQL statement other than an ALTER INDEX or ALTER TABLE statement
  — hexadecimal string constants UX’xxxx’ and GX’xxxx’
  — an SQL syntax error for any object, regardless of whether that object is part of the reorganization

**WARNING**

Do not issue an ALTER statement outside of REORG PLUS to alter the limit keys of an object if that object currently is participating in a reorganization that is waiting to be restarted.
Sample ALTER INDEX statements and resulting messages

In the following example, the DDLIN data set contains three ALTER INDEX statements. The first and third statements alter nine partitions of index USER1.TBL1INX, which is an index on table 1. The second statement alters three partitions of index USER1.TBL2INX, which is an index on table 2. The user specified to use the data set on the reorganization of the table space that contains table 1.

The statements in the DDLIN data set are as follows:

```
ALTER INDEX USER1.TBL1INX
    PART  1 VALUES (X'11'),
    PART  2 VALUES (X'22'),
    PART  3 VALUES (X'33'),
    PART  4 VALUES (X'44')
;
ALTER INDEX USER1.TBL2INX
    PART  1 VALUES (X'0B'),
    PART  2 VALUES (X'0C'),
    PART  3 VALUES (X'0D')
;
ALTER INDEX USER1.TBL1INX
    PART  5 VALUES (X'55'),
    PART  6 VALUES (X'56'),
    PART  7 VALUES (X'67'),
    PART  8 VALUES (X'78'),
    PART  9 VALUES (X'89')
;
```

REORG PLUS processed the ALTER statements successfully and sent the following messages to SYSPRINT:

```
BMC51291I A DDLIN DATASET HAS BEEN FOUND AND CONTAINS THE FOLLOWING STATEMENTS:

BMC50102I ALTER INDEX USER1.TBL1INX
    PART  1 VALUES (X'11'),
    PART  2 VALUES (X'22'),
    PART  3 VALUES (X'33'),
    PART  4 VALUES (X'44')
;
BMC50102I ALTER INDEX USER1.TBL2INX
    PART  1 VALUES (X'0B'),
    PART  2 VALUES (X'0C'),
    PART  3 VALUES (X'0D')
;
BMC50102I ALTER INDEX USER1.TBL1INX
    PART  5 VALUES (X'55'),
    PART  6 VALUES (X'56'),
    PART  7 VALUES (X'67'),
    PART  8 VALUES (X'78'),
    PART  9 VALUES (X'89')
;
```
Sample ALTER TABLE statements and resulting messages

In the following example, the DDLIN data set contains 10 ALTER TABLE statements. The user specified to use the data set on the reorganization of the table space that contains table F509085D.LART003.

The statements in the DDLIN data set are as follows:

```
ALTER TABLE F509085D.LART003
  ALTER PARTITION 1 ENDING AT (X'1F');
ALTER TABLE F509085D.LART003
  ALTER PARTITION 2 ENDING AT (X'2F');
ALTER TABLE F509085D.LART003
  ALTER PARTITION 3 ENDING AT (X'3F');
ALTER TABLE F509085D.LART003
  ALTER PARTITION 4 ENDING AT (X'4F');
ALTER TABLE F509085D.LART003
  ALTER PARTITION 5 ENDING AT (X'5F');
ALTER TABLE F509085D.LART003
  ALTER PARTITION 6 ENDING AT (X'6F');
ALTER TABLE F509085D.LART003
  ALTER PARTITION 7 ENDING AT (X'7F');
ALTER TABLE F509085D.LART003
  ALTER PARTITION 8 ENDING AT (X'8F');
ALTER TABLE F509085D.LART003
  ALTER PARTITION 9 ENDING AT (X'9F');
ALTER TABLE F509085D.LART003
  ALTER PARTITION 10 ENDING AT (X'FF');
```

REORG PLUS processed the ALTER statements successfully and sent the following messages to SYSPRINT:

```
BMC51291I A DDLIN DATASET HAS BEEN FOUND AND CONTAINS THE FOLLOWING STATEMENTS:
BMC50102I ALTER TABLE F509085D.LART003
BMC50102I ALTER PARTITION 1 ENDING AT (X'1F');
BMC50102I ALTER TABLE F509085D.LART003
BMC50102I ALTER PARTITION 2 ENDING AT (X'2F');
BMC50102I ALTER TABLE F509085D.LART003
BMC50102I ALTER PARTITION 3 ENDING AT (X'3F');
BMC50102I ALTER TABLE F509085D.LART003
BMC50102I ALTER PARTITION 4 ENDING AT (X'4F');
BMC50102I ALTER TABLE F509085D.LART003
BMC50102I ALTER PARTITION 5 ENDING AT (X'5F');
```
**DDLOUT data sets**

The DDLOUT output data set contains all SQL ALTER statements that REORG PLUS executes (in order of execution) during partition rebalancing. REORG PLUS opens this optional data set only if it executes an ALTER.

REORG PLUS opens the data set with the following attributes, regardless of any attributes that you specify:

- LRECL=80
- RECFM=FB
- BLKSIZE=27920

REORG PLUS writes the DDL to columns 1 through 72, and writes EBCDIC blanks to columns 73 through 80. This format enables you to use this data set as DDLIN input in a subsequent REORG PLUS job.
You can produce only a report of the ALTER statements, without completing the reorganization, by completing the following steps:

1. Include a DDLOUT DD statement in your JCL.

2. Specify the following options on your REORG command:
   
   — SHRLEVEL CHANGE
   — REBALANCE
   — MAXRO DEFER

3. When REORG PLUS issues message BMC50829I, indicating that the LOGAPPLY phase is starting, cancel the job. REORG PLUS backs out the reorganization, but leaves the information in the DDLOUT data set intact.

**SORTWK data sets**

SORTWK data sets are the work files that BMCSORT uses. For a single-phase reorganization, BMCSORT uses the SORTWK data sets as work files in the REORG phase. For a two-phase reorganization, BMCSORT uses the SORTWK data sets as follows:

- in the UNLOAD phase to sort the unloaded table space rows (if you do not specify ORDER NO)
- in the RELOAD phase to sort the non-data-sorting indexes (and the clustering indexes if you specify ORDER NO)

REORG PLUS does not require SORTWK data sets when any of the following conditions exists. In each of these cases, REORG PLUS does not perform a sort.

- when you are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect

- when either of the following conditions exists for a two-phase reorganization, or for a single-phase reorganization with SHRLEVEL REFERENCE or SHRLEVEL CHANGE specified:
  
  — You are performing an index reorganization with ORDER NO specified.
  — You are performing one of the following table space reorganizations:
    - a single-table table space (simple or segmented) with no indexes defined
    - a multitable simple table space with no indexes defined and ORDER NO specified

All other types of reorganizations perform a sort and require one or more SORTWK data sets.
Anytime that REORG PLUS performs a sort, you must allocate SORTWK data sets in one of the following ways:

- Have BMCSORT dynamically allocate SORTWK data sets.
- Have REORG PLUS dynamically allocate SORTWK data sets (by specifying the DDTYPE option).
- Explicitly specify SORTWK DD statements in your JCL. Use this method when you want to control the allocation of your SORTWK data sets. For more information, see “Allocating SORTWK data sets in your JCL” on page 369.

**Dynamically allocating SORTWK data sets**

Dynamic allocation takes place when any of the following actions occurs:

- You specifically request dynamic allocation through command or installation options.
- BMCSORT determines that it needs more sort work space than other allocation methods provide (allocated in your JCL, dynamically allocated by REORG PLUS, or both).

Several factors affect this dynamic allocation, as described in the following sections.

**REORG PLUS allocation (ACTIVE YES)**

If REORG PLUS dynamic allocation is active for sort work data sets, REORG PLUS calculates the optimal file size and number of files to allocate. REORG PLUS allocates at least 12 and up to 99 sort work data sets, depending on the total space required. REORG PLUS dynamic allocation is active when you specify ACTIVE YES for DDTYPE SORTWORK.

**NOTE**

Unless you are running a DSNUTILB reorganization, BMC recommends that you use BMCSORT to dynamically allocate your SORTWK data sets.

Note the following considerations:

- If you specify IFALLOC USE, REORG PLUS allocates sort work data sets, if needed, in addition to the ones specified in your JCL.
When REORG PLUS dynamic allocation is active for sort work data sets, any values that you specify for SORTNUM, SORTDEVT, and the BMCSORT DYNALOC installation option are not used for allocation unless BMCSORT determines that it needs more sort work data sets than REORG PLUS allocated. In this case, BMCSORT uses these options, but only for allocation of the additional data sets.

REORG PLUS deletes the sort work files when you specify DELETEFILES YES and the reorganization ends successfully. For more information, see “DELETEFILES” on page 215.

**BMCSORT allocation**

If BMCSORT allocation is active (see Table 54), BMCSORT dynamically allocates your sort work files in the following cases:

- REORG PLUS dynamic allocation is not enabled for sort work files (ACTIVE NO) and you have not specified SORTWK DD statements in your JCL.
- special cases when REORG PLUS dynamic allocation is enabled (see page 367)
- special cases when you have allocated sort work data sets in your JCL (see “Allocating SORTWK data sets in your JCL” on page 369)

Table 54 describes how BMCSORT allocates sort work files based on values for the SORTDEVT and SORTNUM options and the BMCSORT DYNALOC installation option.

### Table 54  Factors that affect BMCSORT dynamic allocation of sort work data sets

<table>
<thead>
<tr>
<th>SORTDEVT and SORTNUM</th>
<th>Third parameter of BMCSORT DYNALOC</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORTDEVT specified or</td>
<td>ON or OFF</td>
<td>If you specify a SORTNUM value greater than 32, BMCSORT allocates the number of data sets that it determines are needed, up to the specified number of data sets per sort task. Otherwise, BMCSORT allocates the number of data sets that it determines are needed, up to 32 per sort task.</td>
</tr>
<tr>
<td>SORTNUM n specified (where</td>
<td>If the value is OFF, specifying a value greater than 0 for SORTNUM or specifying SORTDEVT changes this value to ON.</td>
<td></td>
</tr>
<tr>
<td>n is greater than 0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SORTDEVT not specified and</td>
<td>ON</td>
<td>BMCSORT allocates the number of data sets that it determines are needed, up to 32 per sort task.</td>
</tr>
<tr>
<td>SORTNUM 0</td>
<td>OFF</td>
<td>BMCSORT does not allocate any sort work data sets and attempts to perform sort processing in memory.</td>
</tr>
</tbody>
</table>
Allocating SORTWK data sets in your JCL

Unless you specify ACTIVE YES IFALLOC FREE for REORG PLUS dynamic allocation of sort work data sets, REORG PLUS uses any SORTWK DD statements that you specify in your JCL.

To allocate SORTWK data sets in your JCL, determine the space needed for these data sets by specifying ANALYZE PAUSE or ANALYZE ONLY. The options tell REORG PLUS to provide an estimate of the space needed for your work data sets. For more information, see “ANALYZE option for estimating data set allocation” on page 386.

When determining the number of concurrent tasks to run, REORG PLUS checks the amount of sort work space that is allocated. In this calculation, REORG PLUS uses only the primary allocation. The secondary allocation is not guaranteed.

Additional considerations
Note the following information when allocating SORTWK data sets in your JCL:

- You cannot allocate a SORTWK data set as any of the following data set types:
  - VIO data set
  - tape data set
  - SMS-managed striped data set
  - multiple-volume data set
  - data set in an SMS storage group that specifies EXTENDED FORMAT YES

  BMCSORT does not support SORTWK data sets that extend beyond 65,535 tracks on a single volume.

- When you specify SORTWK DD statements in your JCL, BMCSORT dynamically allocates additional sort work data sets when both of the following conditions exist:
  - BMCSORT determines that it needs more sort work space than you have allocated.
  - The number of started sort tasks is such that at least one sort work data set can be allocated to each sort task.
SYSARC data set

REORG PLUS uses the SYSARC data set in one of the following ways:

- For native REORG PLUS jobs, SYSARC is an archive data set that contains discarded rows. REORG PLUS discards these rows when performing one of the following processes during a table space reorganization:
  - a SELECT or DELETE operation
  - rebalance of a table space where the last partition key has been altered and the table space is defined in one of the following ways:
    - LARGE (either by definition or default)
    - defined with DSSIZE
    - range-partitioned

- For DSNUTILB reorganization jobs, REORG PLUS passes the dynamic allocation information for this data set to the IBM DB2 REORG utility to allocate the data set that contains discarded rows.

Native REORG PLUS jobs

The following considerations and allocation information apply to non-DSNUTILB reorganization jobs.

Considerations

Note the following general information about SYSARC data sets:

- The archive data set is required when all of the following conditions exist:
  - REORG PLUS performs partition rebalancing as part of the reorganization
  - The last partition is in REORP status either going into the reorganization or changed to REORP status during the reorganization.
  - The table space is LARGE or was defined with DSSIZE specified, or uses table-controlled partitioning.

- The archive data set is optional if you are using a SELECT or DELETE statement. If you do not specify the archive data set, any discarded rows are lost.

- REORG PLUS writes to the archive data set during unload processing.
You can specify the ARCHFORMAT option to tell REORG PLUS whether you want to write the discarded rows to this data set in an internal format (ARCHFORMAT BMC) or in a format that both LOADPLUS and the DB2 LOAD utility can read (ARCHFORMAT DB2).

For ARCHFORMAT DB2, the format of the discarded rows is the same as that produced by the DB2 REORG utility when you specify UNLOAD ONLY. This format differs from the format of the discard records that the DB2 REORG utility produces.

You can reload the discarded rows in the SYSARC data set by using one of the following options:

— the FORMAT BMC option of any currently supported version of LOADPLUS (if ARCHFORMAT is BMC)

— the FORMAT UNLOAD option of LOADPLUS or the DB2 LOAD utility (if ARCHFORMAT is DB2)

The table that you are loading must have the same definition as the table from which the archive rows originated. For more information about using LOADPLUS to load these rows, see the *LOADPLUS for DB2 Reference Manual*.

**Dynamically allocating SYSARC data sets**

When dynamic allocation is active, REORG PLUS allocates the SYSARC data set only when performing the following processes:

- processing a SELECT or DELETE statement

- rebalancing the last partition of a table space that is LARGE or was defined with DSSIZE specified, or uses table-controlled partitioning

REORG PLUS uses information from the ANALYZE phase to analyze the requirements for the current execution and calculates the optimal file size and number of files to allocate.

Note the following additional information about dynamically allocating SYSARC data sets:

- You can use the ARCROWS command option to control the size of a dynamically allocated archive data set. For more information, see “ARCROWS” on page 229.

- You can decide when to delete the dynamically allocated archive (SYSARC) data set by specifying an expiration date with the EXPDT option or by specifying a retention period with the RETPD option. For more information about these options, see “EXPDT” on page 322 and “RETPD” on page 322.
Allocating SYSARC data sets in your JCL
The following restrictions apply to the SYSARC data set:

- To accommodate restarting REORG PLUS any time after the UNLOAD phase for a two-phase reorganization, the SYSARC should not be a temporary data set. For information about how REORG PLUS treats temporary data sets, see “Work file validity and integrity checks” on page 96.

- BMC recommends that you do not use either of the following DD specifications for the SYSARC data set:
  
  - DD DUMMY
  - DSN=NULLFILE

- Do not use the same SYSARC data set for two different jobs. If you specify DISP=MOD for an existing SYSARC data set, REORG PLUS treats it as if you had specified DISP=OLD. Therefore, you will lose the data that already exists in the data set.

- If you restart a job using the same SYSARC data set (DISP=MOD) as you used for the initial run, and the data set had an expiration date or retention period, respond U to the following message to continue processing:

  IEC507D REPLY 'U'-USE OR 'M'-UNLOAD

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes the dynamic allocation information for the optional SYSARC data set in the TEMPLATE control statement that REORG PLUS builds for the discard data set for the IBM DB2 REORG utility. This data set contains the rows that are discarded during the reorganization.

DB2 REORG requires the discard data set when certain conditions exist. If you do not enable dynamic allocation for this data set and DB2 REORG needs the data set, the reorganization terminates. For information about this data set, including requirements, see the DISCARD and SYSDISC information in the documentation for the DB2 REORG utility.

Allocation
To allocate this data set, you must enable dynamic allocation for the ARCHIVE DDTYPE. If you include a SYSARC DD statement in your JCL for a DSNUTILB reorganization, REORG PLUS ignores it.
Considerations
The following considerations apply to the SYSARC data set for DSNUTILB reorganization jobs:

- When invoking DSNUTILB to reorganize a LOB table space, REORG PLUS does not allocate this data set.

- The ARCROWS option is available for the SYSARC data set when you run a native REORG PLUS job, but is not available for a DSNUTILB job.

- You can decide when to delete this data set by specifying an expiration date with the EXPDT option, or by specifying a retention period with the RETPD option. For more information about these options, see “EXPDT” on page 322 and “RETPD” on page 322.

SYSERR data set

REORG PLUS recommends the SYSERR data set only if you specify SHRLEVEL CHANGE. This data set contains diagnostic messages that REORG PLUS might produce in error situations when running a SHRLEVEL CHANGE reorganization.

**NOTE**

You should not depend on the format of this proprietary data set; this format is subject to change without notice.

SYSEXEC data set

REORG PLUS always requires the SYSEXEC data set for the following user-written REXX exit routines:

- DSRSEXIT exits
- TERMEXIT exits
- MAPTEXIT exits that are contained in formatted data sets

**NOTE**

Alternatively for a MAPTEXIT REXX exit, you can compile the exit with the NOCEXEC options and link it into an unformatted PDS; you can then include the PDS in your STEPLIB concatenation. This option eliminates the need to include a SYSEXEC DD statement in your JCL. For more information, see “Supplying MAPTEXIT in your STEPLIB” on page 858.

This DD statement specifies the data set that contains these exits.
SYSIDCIN data set

SYSIDCIN is the input data set containing your IDCAMS command statements. REORG PLUS uses these statements to delete and redefine user-defined (VCAT-defined) data sets or the staging data sets for your VCAT-defined data sets. REORG PLUS issues the commands that the SYSIDCIN data set contains. You are responsible for command specifications and results.

REORG PLUS requires the SYSIDCIN data set if the value of the REDEFINE option is YES and one of the following conditions exists:

- You want to delete and redefine your existing VCAT-defined VSAM data sets as part of the reorganization for SHRLEVEL NONE (the default) or SHRLEVEL REFERENCE UNLOADONLY.

- You want to delete and define the staging data sets for your VCAT-defined VSAM data sets as part of the reorganization for SHRLEVEL REFERENCE or SHRLEVEL CHANGE.

If you omit the SYSIDCIN data set and specify REDEFINE YES, REORG PLUS performs the following actions, depending on the value of the PREFORMAT command or installation option:

- If PREFORMAT=NO, REORG PLUS changes the value of REDEFINE to NO and continues with the reorganization (without deleting and defining the data sets).

- If PREFORMAT=YES, REORG PLUS terminates the reorganization and issues message BMC50391E.

The UTILINIT phase reads, parses, and performs minimal verification checks on the commands in this data set. REORG PLUS issues the commands for each object during reload processing.

You still have the option of deleting and redefining VCAT-defined data sets when you specify UNLOAD PAUSE on the command. However, when you specify UNLOAD CONTINUE or UNLOAD RELOAD, the only way to delete and redefine VCAT-defined data sets as part of the reorganization is to also specify REDEFINE YES and provide the SYSIDCIN data set that contains the necessary IDCAMS command statements.

SYSIDCIN can be either a single or concatenated list of sequential data sets, partitioned data set members, or both. The data set must be specified as fixed length with blocked records (RECFM=FB), and the record length must be 80 bytes (LRECL=80). Only columns 1 through 72 are used.

For important restart information, see “Recovering from a failure” on page 398.
Guidelines for providing IDCAMS commands in the SYSIDCIN data set

REORG PLUS supports the following IDCAMS commands and their associated parameters as defined in the IBM DFSMS Access Method Services for Catalogs document:

- DELETE
- DEFINE
- SET
- IF-THEN-ELSE command sequence
- DO/END (with restrictions)

REORG PLUS does not allow the following IDCAMS command specifications:

- an IF statement without a DEFINE, DELETE, or SET command
- a nested IF statement
- different data set names specified in an IF statement
- more than one DO/END statement within a single THEN or ELSE clause
- a DO/END statement outside of an IF statement
- an embedded comment (a comment within a comment)
- a comment on a DO statement that continues to the next line
- a keyword that continues to the next line

REORG PLUS checks for these specifications during the UTILINIT phase and terminates with an error message if it finds any of them.

To avoid a failure during processing, ensure that each DEFINE has a corresponding DELETE specified before it, and that each DELETE has a corresponding DEFINE specified after it. Ensure that both commands specify the same data set name.

REORG PLUS terminates if IDCAMS returns a nonzero maximum condition code (MAXCC) value or encounters any syntax errors during command processing. You must correct the error and either restart or recover and then resubmit the job. If you do not want REORG PLUS to terminate in case of a failure during the RELOAD or REORG phase, use the SET command to reset the IDCAMS condition code. For more information about restarting during the RELOAD and REORG phases, see “Restarting REORG PLUS” on page 391.
When you specify SHRLEVEL NONE (default) or SHRLEVEL REFERENCE UNLOADONLY

Using the guidelines in “Guidelines for providing IDCAMS commands in the SYSIDCIN data set” on page 375, specify commands only for those VCAT-defined data sets that you want to delete and redefine. REORG PLUS reuses any data sets participating in the reorganization that have no corresponding IDCAMS commands. REORG PLUS ignores any command that references a data set that does not participate in the reorganization and issues message BMC50604I. REORG PLUS ignores any empty SYSIDCIN data set and continues processing.

When you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE

Using the guidelines in “Guidelines for providing IDCAMS commands in the SYSIDCIN data set” on page 375, specify commands that operate only on the staging data sets for your VCAT-defined VSAM data sets. Always use the naming conventions outlined in “Staging data sets” on page 101.

To define the staging data sets and avoid definition errors, provide a DELETE statement followed by a SET MAXCC=0 command (in case the delete process fails) before each DEFINE statement. REORG PLUS processes this set of statements (DELETE, SET MAXCC, DEFINE) for each object as the RELOAD phase begins for that object. REORG PLUS ignores any command that references one of the original VCAT-defined data sets and issues message BMC50604I.

SYSIN data set

REORG PLUS always requires the SYSIN data set. This input data set contains the REORG command. The UTILINIT phase reads, parses, and verifies the REORG command that is provided in this data set. The data set’s attributes must be specified as fixed length (RECFM is F, FB, or FBS), and the record length must be 80 columns (LRECL=80). REORG PLUS uses only columns 1 through 72.

SYSPRINT data set

REORG PLUS always requires the SYSPRINT data set. This output data set contains REORG PLUS messages.

NOTE

You should not depend on the content and format of this data set (for example, as input to user-defined processes). Message content and format are subject to change without notice.
REORG PLUS overrides any data control block attributes that you specify in your JCL with DCB=(RECFM=VBA,LRECL=137,BLKSIZE=141).

Note the following considerations if you direct SYSPRINT to a tape or disk data set:

- The DSN messages from DB2, such as those from a QUIESCE utility, are lost.
- BMC does not recommend using the BUFNO parameter on the DD statement for this data set.

Note the following restrictions on SYSPRINT data sets in your DSNUTILIB reorganizations:

- Do not specify FREE=CLOSE.
- You cannot use a PDS or PDSE.

For information about the level of messages that REORG PLUS displays and how to change the message level, see “Message level (MSGLEVEL)” on page 347.

**SYSPRIN2 data set**

The SYSPRIN2 data set is an optional output data set that contains the REORG PLUS messages. With the exceptions noted in this section, the content is identical to the content of the SYSPRINT data set.

SYSPRIN2 is not a substitute for SYSPRINT. If you include a SYSPRIN2 DD statement in your JCL, you must still include a SYSPRINT DD statement.

---

**NOTE**

You should not depend on the content and format of this data set (for example, as input to user-defined processes). Message content and format are subject to change without notice.

REORG PLUS overrides any data control block attributes that you specify in your JCL with DCB=(RECFM=VBA,LRECL=137,BLKSIZE=141).

In a worklist environment, you can specify SYSPRIN2 DD SYSOUT=* to view output in real time from any of the BMC Utility products that run in that worklist.

Note the following considerations if you direct SYSPRIN2 to a tape or disk data set:

- The DSN messages from DB2, such as those from a QUIESCE utility, are lost.
- BMC does not recommend using the BUFNO parameter on the DD statement for this data set.
SYSPRIN2 data sets have the following restrictions:

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

- When invoking DSNUTILB, REORG PLUS ignores the SYSPRIN2 DD statement.

For information about the level of messages that REORG PLUS displays and how to change the message level, see “Message level (MSGLEVEL)” on page 347.

**SYSPUNCH data set**

When invoking DSNUTILB, REORG PLUS includes the dynamic allocation information for the optional SYSPUNCH data set in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. This data set contains LOAD control statements for the rows that are discarded and written to the discard file during the reorganization.

DB2 REORG requires this data set when certain conditions exist. If you do not enable dynamic allocation for this data set and DB2 REORG determines that it needs the data set, the reorganization terminates. For more information about this data set, see the documentation for the DB2 REORG utility.

**Allocation**

To allocate this data set, you must enable dynamic allocation for the SYSPUNCH DDTYPE. If you include a SYSPUNCH DD statement in your JCL, REORG PLUS ignores it.

For non-DSNUTILB reorganizations, REORG PLUS does not allocate this data set.

**Considerations**

The following considerations apply to the SYSPUNCH data set for DSNUTILB reorganization jobs:

- When invoking DSNUTILB to reorganize a LOB table space, REORG PLUS does not allocate this data set.

- You can decide when to delete this data set by specifying an expiration date with the EXPDT option, or by specifying a retention period with the RETPD option. For more information about these options, see “EXPDT” on page 322 and “RETPD” on page 322.
SYSREC data sets

The SYSREC data set will contain the table space’s unloaded rows.

NOTE

You should not depend on the format of this proprietary data set; this format is subject to change without notice.

After loading the table space, REORG PLUS deallocates the data set to free the device. Table 55 lists the circumstances under which REORG PLUS requires a SYSREC data set.

Table 55  SYSREC usage

<table>
<thead>
<tr>
<th>Type of reorganization</th>
<th>SYSREC requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>single-phase table space reorganization when you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE</td>
<td>not used References to single-phase reorganizations in this section are for SHRLEVEL NONE only.</td>
</tr>
<tr>
<td>two-phase table space reorganization</td>
<td>required REORG PLUS uses this data set to pass information between the UNLOAD and RELOAD phases.</td>
</tr>
<tr>
<td>single-phase table space reorganization when you specify SHRLEVEL NONE</td>
<td>optional REORG PLUS writes information to this data set during the REORG phase for restart purposes only.</td>
</tr>
</tbody>
</table>

Warning: If you do not specify a SYSREC data set when performing a single-phase reorganization with SHRLEVEL NONE (the default), you might not be able to restart the job. For more information, see “Restarting REORG PLUS” on page 391.

For information about performance improvements for this data set, see page 575.

Methods for allocating SYSREC data sets

You can use one of the methods described in “Methods for allocating copy and work data sets” on page 351 to allocate your SYSREC data sets and determine the appropriate size for the data sets. If you do not use dynamic allocation, you must allocate the SYSREC data sets in your JCL by specifying DD statements.
**NOTE**
For a DSNUTILB reorganization, you must have REORG PLUS dynamically allocate your SYSREC data sets.

### Number of SYSREC data sets to allocate

Use the information in Table 56 to determine how many SYSREC data sets to allocate. Partition-by-growth table spaces have special requirements as described after the table.

#### Table 56   Number of SYSREC data sets to allocate

<table>
<thead>
<tr>
<th>Reorganization type</th>
<th>Object type</th>
<th>Number of SYSREC data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>Nonpartitioned table space</td>
<td>One</td>
</tr>
<tr>
<td>Two-phase</td>
<td>Partitioned table space</td>
<td>One or multiple</td>
</tr>
<tr>
<td>Single-phase</td>
<td>Partitioned table space</td>
<td>Multiple or none</td>
</tr>
<tr>
<td>Any</td>
<td>Partition-by-growth table space</td>
<td>See the section that follows</td>
</tr>
</tbody>
</table>

**Partition-by-growth table spaces**

For partition-by-growth table spaces, REORG PLUS requires the allocation of additional data sets when you are using partition-level SYSREC data sets and either REORG PLUS can extend the table space or you are performing a full SHRLEVEL CHANGE table space reorganization.

The number of additional data sets is equal to the smaller of the following values:

- MAXPARTITIONS minus the number of partitions in the table space
- MAXNEWPARTS

Although REORG PLUS requires that the additional data sets be allocated, it only uses them if partitions are added to the table space during the reorganization. REORG PLUS deletes unused data sets after the reorganization.

**Example**

The following example illustrates the additional data set requirements using SYSREC data sets. Assume the following scenario:

- You are performing a partial reorganization of a table space with four partitions.
- The value of MAXPARTITIONS is 6.
- Your SYSIN data set contains the following specifications:

```sql
REORG TABLESPACE databaseName.tableSpaceName PART 3:LAST MAXNEWPARTS 3 UNLDDN SYSREC
```
In this example, REORG PLUS requires four SYSREC data sets:

- one for each of the two partitions that are participating in the reorganization (partitions 3 and LAST)
- two additional for partitions that might be added

That is, MAXPARTITIONS (6) minus total partitions in the table space (4) which equals 2. This value is smaller than the MAXNEWPARTS value (3).

For more information about partition-by-growth table spaces, see page 125.

**Naming conventions for partition-level SYSREC data sets**

JCL rules limit ddnames to eight characters. When you use multiple SYSRECnn data sets, the ddname has two parts: the ddname prefix (minimum of one character) and the nn (one to seven characters). The ddname in the installation options module or on the REORG command becomes the prefix.

The ddname that you specify in the JCL must have the partition number nn appended to this prefix. The nn must match the partition that you are reorganizing, and you must allow sufficient bytes for the partition number to be added to the prefix and still have a valid ddname (eight characters or less).

**Dynamically allocating SYSREC data sets**

REORG PLUS uses information from the ANALYZE phase to analyze the requirements for the current execution and calculates the optimal file size and number of files to allocate.

Note the following information when dynamically allocating SYSREC data sets:

- When you use dynamic allocation of SYSREC data sets during partition rebalancing, REORG PLUS cannot predict how much data movement will occur between rebalanced partitions. Therefore, REORG PLUS might inadequately size the SYSREC data set for a partition that has a large amount of data rebalanced into it. BMC recommends that you code the required SYSRECnn DD statements for these partitions in the JCL and specify IFALLOC USE for DDTYPE UNLOAD. Continue to allow REORG PLUS to dynamically allocate all other SYSRECnn data sets. This consideration does not apply when you use the REBALANCE command option (page 179) to rebalance partitions.

- REORG PLUS deletes the SYSREC data sets when you specify DELETEFILES YES and the reorganization ends successfully. For more information, see “DELETEFILES” on page 215.
If you plan to have REORG PLUS allocate more than 99 SYSREC data sets, use the UNLDDN option to specify a ddname prefix that results in eight characters or less after REORG PLUS appends the highest data set number. For more information, see “Specifying ddname prefixes” on page 93.

Allocating SYSREC data sets in your JCL

Note the following restrictions when allocating SYSREC data sets in your JCL:

- To enable REORG PLUS to restart any time after the UNLOAD phase for a two-phase reorganization, SYSREC cannot be a temporary data set. For information about how REORG PLUS defines temporary data sets, see “Work file validity and integrity checks” on page 96.
- Do not specify FREE=CLOSE for SYSREC data sets.
- Do not use SMS extended sequential data sets as SYSREC data sets for multiple reorganizations running in a worklist environment (via the BMC Software DASD MANAGER PLUS, CHANGE MANAGER, or CATALOG MANAGER products).

When first using a data set, REORG PLUS defines the data set’s DCB attributes. With an SMS extended sequential data set, REORG PLUS cannot change these attributes. However, reusing the data set, (as with multiple reorganizations in a worklist environment), can require different DCB attribute values, such as a different block size or record length. In this case, REORG PLUS attempts to redefine the attributes, but fails.

SYSTEM data set

Use the SYSTEM data set if you are using REORG PLUS as part of the Database Performance solution. This output data set contains DASD MANAGER PLUS SQL and C runtime error messages.

SYSTSPRT data set

REORG PLUS always requires the SYSTSPRT data set for REXX exits. REXX routes all output from the REXX ‘SAY’ statements to this data set. It is usually coded as a SYSOUT DD statement.

For a MAPTEXIT user exit, REORG PLUS dynamically allocates this DD if your JCL does not include a SYSTSPRT DD statement.

SYSUT1 data sets

SYSUT1 represents the index work data set that contains the unloaded index keys and serves as input for BMCSORT.
For a two-phase table space or index reorganization, REORG PLUS uses the SYSUT1 data set to pass information between the UNLOAD and RELOAD phases. For a single-phase index reorganization, REORG PLUS writes information to this data set during the REORG phase for restart purposes only. After building the index, REORG PLUS deallocates the data set to free the device. Table 57 lists the circumstances under which REORG PLUS requires the SYSUT1 data set.

**Table 57  SYSUT1 usage**

<table>
<thead>
<tr>
<th>Reorganization type</th>
<th>SYSUT1 requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>single-phase index reorganization when you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE</td>
<td>not used</td>
</tr>
<tr>
<td>single-phase index reorganization when you specify SHRLEVEL NONE</td>
<td>required if you want to be able to restart the job</td>
</tr>
<tr>
<td>References to single-phase reorganizations in this section are for SHRLEVEL NONE only.</td>
<td></td>
</tr>
<tr>
<td>two-phase index reorganization</td>
<td>required</td>
</tr>
<tr>
<td>LOB table space reorganization</td>
<td>not used</td>
</tr>
<tr>
<td>table space reorganization where the following types of indexes exist:</td>
<td>required</td>
</tr>
<tr>
<td>- nonpartitioned index</td>
<td></td>
</tr>
<tr>
<td>- clustering index using ORDER NO</td>
<td></td>
</tr>
<tr>
<td>- data-partitioned secondary index</td>
<td></td>
</tr>
</tbody>
</table>

**Methods for allocating SYSUT1 data sets**

Use one of the methods described in “Methods for allocating copy and work data sets” on page 351 to allocate your SYSUT1 data sets and determine the appropriate size for the data sets. BMC recommends that you dynamically allocate your SYSUT1 data sets. If you do not use dynamic allocation, you must specify DD statements in your JCL for the SYSUT1 data sets.

**NOTE**

For a DSNUTILB reorganization, you must have REORG PLUS dynamically allocate your SYSUT1 data sets.
**Number of SYSUT1 data sets to allocate**

Use the information in Table 58 to determine how many SYSUT1 data sets to allocate.

<table>
<thead>
<tr>
<th>Type of reorganization</th>
<th>Number of SYSUT1 data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any table space reorganization</td>
<td>One data set for all indexes or one data set for each participating index</td>
</tr>
<tr>
<td>Two-phase index reorganization</td>
<td>One data set</td>
</tr>
<tr>
<td>Single-phase index reorganization</td>
<td>One data set or no data sets</td>
</tr>
</tbody>
</table>

**Specifying a single SYSUT1 data set**

If you want a single work data set for all participating indexes, specify one SYSUT1 DD statement.

**Specifying multiple SYSUT1 data sets**

When more than one non-data-sorting index exists, you can improve I/O performance by using multiple SYSUT1 data sets. Using multiple data sets allows I/O operations to overlap. For more information about performance and DASD considerations, see “SYSUT1 data set” on page 576.

If you specify more than one SYSUT1 data set, specify a DD statement named SYSUT1nn for each data set. Specifying nn creates a unique ddname; the nn has no relation to the index name.

Note the following additional considerations about the number of work data sets to allocate:

- If you specify more than one SYSUT1 data set, specify one data set for each participating index. When determining how many data sets to allocate, consider the following information:
  - When reorganizing a base table space that contains an XML column, ensure that you include a data set for the document ID index.
  - When reorganizing an XML table space, ensure that you include a data set for the node ID index.
  - When reorganizing a LOB table space, you do not need to include a data set for the auxiliary index.

- If you are reorganizing a table space with a large number of indexes, BMC recommends that you specify a single SYSUT1 data set to avoid data set allocation limitations of the operating system.
Dynamically allocating SYSUT1 data sets

REORG PLUS uses information from the ANALYZE phase to analyze the requirements for the current execution and calculates the optimal file size and number of files to allocate.

The following additional information applies when dynamically allocating SYSUT1 data sets:

- REORG PLUS deletes the SYSUT1 data sets when you specify DELETEFILES YES and the reorganization ends successfully. For more information, see “DELETEFILES” on page 215.

- If you plan to have REORG PLUS allocate more than 99 SYSUT1 data sets, use the WORKDDN option to specify a ddname prefix that results in eight characters or less after REORG PLUS appends the highest data set number. For more information, see “Specifying ddname prefixes” on page 93.

Allocating SYSUT1 data sets in your JCL

Note the following considerations for allocating SYSUT1 data sets:

- To avoid a failure in the RELOAD phase, do not specify DD DUMMY or DSN=NULLFILE for SYSUT1 data sets when running REORG PLUS in WARN mode.

- To enable REORG PLUS to restart any time after the UNLOAD phase for a two-phase reorganization, the SYSUT1 cannot be a temporary data set. For information about how REORG PLUS defines temporary data sets, see “Work file validity and integrity checks” on page 96.

- For multiple reorganizations running in a worklist environment (through DASD MANAGER PLUS, CHANGE MANAGER, or CATALOG MANAGER), do not use SMS extended sequential data sets as SYSUT1 data sets. When REORG PLUS first uses a data set, it defines the data set’s DCB attributes. With an SMS extended sequential data set, once these attributes are defined, they cannot be changed. If this data set is reused, as with multiple reorganizations in a worklist environment, it can require different DCB attribute values, such as a different block size or record length. REORG PLUS attempts to redefine the attributes if necessary. However, because MVS™ does not allow changes to these attributes after the file has been opened, REORG PLUS fails.
When using multiple SYSUT1 data sets, REORG PLUS assigns the index with the largest key length to the first SYSUT1 DD statement in the JCL, the index with the second-largest key length to the second SYSUT1 DD statement, and so on. This assignment of indexes to data sets by key length can assist you in allocating the sizes of your data sets and allows you to place data sets on different devices when needed.

Due to this assignment, you should allocate the space for the participating index with the largest key length in the first SYSUT1 DD statement in your JCL, space for the index with the second-largest key length in the second SYSUT1 DD statement, and so on.

**UTPRINT data set**

REORG PLUS always requires the UTPRINT data set if sorting is necessary. The presence of this data set tells REORG PLUS to report sort messages. However, the actual messages for each sort process appear in separate SYSnnnnn data sets, where nnnnn is a system-assigned sequential number. You cannot specify a sequential or partitioned data set for UTPRINT. UTPRINT supports only SYSOUT data sets.

**WARNING**

JES3 users should be aware of a limitation within JES3 that does not allow concurrent tasks to share SYSOUT data sets. (For a full description of this limitation, see IBM APAR OY23946.) This limitation means that you cannot use additional sort routine reporting DDs (other than UTPRINT) if they are defined as JES3 SYSOUT data sets and when REORG PLUS is multitasking its sort activity. If you attempt to use an unsupported DD, you risk S1FB abends when concurrent sort tasks are running. JES3 version 4.2.1 users should also refer to IBM APARs OW00111 and OY63725.

**Other data sets**

Because it uses BMCSORT, REORG PLUS ignores any traditional sort routine DD statements (such as $ORTPARM and DFSPARM) that you specify.

**ANALYZE option for estimating data set allocation**

If you specify ANALYZE PAUSE or ANALYZE ONLY, REORG PLUS gathers information about the objects that you are reorganizing. In addition to cardinality and average row size, the ANALYZE phase provides estimated data set sizes for the following data sets:

- unload (SYSREC)
- work (SYSUT1)
- sort (SORTWK)
REORG PLUS cannot take into account rows bypassed with SELECT or DELETE.

REORG PLUS writes these statistics to SYSPRINT. For information about the other statistical information messages that the ANALYZE phase issues, see “ANALYZE messages” on page 572.

If you specify ANALYZE ONLY and use the information to allocate your data sets, you can improve performance by changing the REORG command options to ANALYZE HURBA when you rerun the job. Specifying ANALYZE HURBA bypasses the ANALYZE phase. For the list of restrictions when using HURBA, see “HURBA” on page 202.

As an alternative to using ANALYZE PAUSE or ONLY to estimate sizes for data set allocation, you can have REORG PLUS dynamically allocate your data sets for you. To use dynamic allocation, specify ANALYZE (without PAUSE or ONLY). You must also have dynamic data set allocation active, either in your installation options or with the DDTYPE command option.

If you do not use the PAUSE or ONLY keywords with ANALYZE, REORG PLUS also gathers the information described in this section. However, instead of pausing or stopping, REORG PLUS continues processing. If dynamic allocation is enabled, REORG PLUS uses the ANALYZE phase information to dynamically allocate your data sets. In this case, the ANALYZE phase does not write the statistics to SYSPRINT.

For both optimum and minimum sort work file estimates, REORG PLUS uses the largest index to determine estimates. The optimum value is either the space required to sort the largest task (the task unloading the most data) or the space required to sort the largest index, whichever is greater. The space required for the task that unloads the most data can always be determined by sampling.

Table 59 on page 388 details the space estimates provided for both table space and index reorganizations and provides estimates for both single and multiple SYSUT1 and SYSREC data sets. Refer to the specification guidelines for each data set provided in the preceding pages to determine whether to specify single or multiple SYSUT1 and SYSREC data sets.
### Table 59: Estimates provided by the ANALYZE option

<table>
<thead>
<tr>
<th>Data sets for which estimates are provided</th>
<th>Reorganization type</th>
<th>Information provided</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table space</strong></td>
<td><strong>Index</strong></td>
<td></td>
</tr>
<tr>
<td>single SYSREC data set</td>
<td>yes</td>
<td>not applicable</td>
</tr>
<tr>
<td>multiple SYSREC data sets</td>
<td>yes</td>
<td>not applicable</td>
</tr>
<tr>
<td>single SYSUT1 data set</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>multiple SYSUT1 data sets</td>
<td>yes</td>
<td>not applicable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SORTWK data sets</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>single full or incremental image copy data set (BCPY, BRCY, BICY, and so on)</td>
<td>yes</td>
<td>not applicable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>multiple full or incremental image copy data sets (BCPYnn, BRCYnn, BICYnn, and so on)</td>
<td>yes</td>
<td>not applicable</td>
</tr>
</tbody>
</table>
REORG PLUS provides the estimated information in table format. Messages BMC51260I and multiple BMC51263I messages provide the estimates. A separate BMC51263I message for each data set provides the following information:

- data set name
- number of kilobytes
- primary and secondary 3380 cylinder quantities
- primary and secondary 3390 cylinder quantities
- index name, where applicable

**NOTE**

For several reasons, including rows that contain VARCHAR columns and tables that contain EDITPROCs, ANALYZE might report a secondary quantity for SYSREC that is too large. The reason is that the primary quantity is based on the average row length, and the secondary quantity is based on the maximum row length from the DB2 catalog. In this instance, BMC recommends that you provide a secondary quantity of approximately 25 percent of the primary quantity.

### Using ANALYZE with compressed table spaces

REORG PLUS uses the compressed row length to determine the size of the SYSREC and SORTWK data sets whenever possible.

#### SYSREC data set

REORG PLUS estimates the size of the SYSREC data set in the following manner:

- For compressed table spaces, REORG PLUS uses the average compressed row length.
- For noncompressed table spaces, REORG PLUS always uses the actual row length.
- For a multi-table table space, REORG PLUS averages the row length for the various tables.

Table 60 describes whether REORG PLUS uses compressed or expanded rows when KEEPDICTIONARY is in effect.

<table>
<thead>
<tr>
<th>Type of reorganization</th>
<th>KEEPDICTIONARY value</th>
<th>Row length used</th>
</tr>
</thead>
<tbody>
<tr>
<td>single phase</td>
<td>YES</td>
<td>compressed</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>expanded</td>
</tr>
<tr>
<td>two phase</td>
<td>YES</td>
<td>compressed</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>compressed</td>
</tr>
</tbody>
</table>
SORTWK data set

When estimating the size of the SORTWK data sets, REORG PLUS uses the average compressed row length only if all of the following criteria are true for a table or for all partitions of a table space:

- The value of the KEEPDICTIONARY command or installation option is YES (or is implied, as when you do a single-phase SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganization with ORDER NO).
- You did not add new columns to the table.
- You did not specify AMEND YES for the EDITPROC for this table.
- You did not specify UPDATE on the REORG command for the table.
- The table belongs to a table space with the COMPRESS YES attribute, or all of the partitions of the table space have the COMPRESS YES attribute.

For a partitioned table space, if only some of the partitions meet the preceding criteria, REORG PLUS uses the expanded row length to calculate the SORTWK data set size for all of the partitions.

For a multi-table table space, REORG PLUS uses the

- compressed row length for each table that meets all of the preceding criteria
- expanded row length for each table that does not meet the criteria

REORG PLUS then averages the row lengths to achieve the estimated data set size.

Running REORG PLUS jobs

After you have built your REORG PLUS job, the next step is to run the job. This section describes how to invoke a job, how to restart it, and how to terminate or cancel it and recover the DB2 objects.
Invoking REORG PLUS

You normally invoke REORG PLUS as a batch job by specifying execution of the module ARUUMAIN on the EXEC statement of your JCL, along with its required EXEC statement parameters. You must also specify any DD statements that REORG PLUS requires, as described in “REORG PLUS DD statements” on page 348. Ensure that all required libraries are available and APF-authorized as described in “STEPLIB DD statement” on page 348.

Restarting REORG PLUS

For a single-phase SHRLEVEL REFERENCE reorganization, you can restart REORG PLUS from a failure during any phase due to the nondestructive nature of this type of reorganization.

For a SHRLEVEL CHANGE (single- or two-phase) reorganization, you cannot restart any time before the beginning of the UTILTERM phase, unless you are restarting after ANALYZE PAUSE. Until UTILTERM begins, all user updates are made to the original data sets, which the reorganization has not yet changed. The data sets are exactly as they were before you ran the reorganization. After UTILTERM begins, restart works the same as it does for any other type of reorganization. For more information about restarting a SHRLEVEL CHANGE reorganization, see “Restart considerations for a SHRLEVEL CHANGE reorganization” on page 657.

For all other types of reorganizations, with exceptions described in the following sections, you can restart REORG PLUS from a failure during any phase as long as the SYSRECnn and SYSUT1nn data sets are present and are defined as cataloged data sets. Dynamically allocated SYSRECnn, SYSUT1nn, and copy data sets are automatically reallocated by REORG PLUS on restart.

You cannot make structural changes to objects, such as altering a column from VARCHAR to CHAR, before restarting a failed reorganization. REORG PLUS relies on the object structure to remain unchanged between restart and the original run. If you change the structure between runs, REORG PLUS might issue a user abend 3200 with reason code 5, or produce unpredictable results.

Specifying the RESTART and RESTART(PHASE) options

If a reorganization fails, correct the problem and restart the reorganization either with RESTART or RESTART(PHASE). REORG PLUS issues messages as it unloads, reloads, or rebuilds each DB2 object. The BMCSYNC table contains an entry for each DB2 object involved in the reorganization and its current status.
Specify RESTART without (PHASE) to restart REORG PLUS from the last restart sync point. REORG PLUS takes restart sync points as each phase completes and as the processing of each DB2 object completes. The utility ID must exist in the BMCUTIL table.

**NOTE**
The SYNC option (on the REORG command) controls the frequency with which rows are updated in the BMCSYNC table. SYNC does not control sync points that REORG PLUS uses when restarting. During the reload processing, restart sync points are established only after the last row is loaded in a table space, index, or partition.

Specify RESTART(PHASE) to restart REORG PLUS at the beginning of the last incomplete phase. The utility ID must exist in the BMCUTIL table.

**Additional restart considerations and restrictions**

This section describes additional considerations and restrictions that you should be familiar with before you restart a REORG PLUS job. For detailed instructions about dealing with a failure during the reorganization, see Table 62 on page 397.

**LOB table spaces**

If a failure occurs during the index rebuilding process of a LOB table space reorganization, restarting the job causes REORG PLUS to reorganize the LOB table space again.

**XML table spaces**

The following considerations apply when the table space contains a document ID index for which REORG PLUS has generated document ID values. REORG PLUS might generate document ID values if the original job is the first reorganization after adding the first XML column to the table.

- When both of the following conditions exist, you can restart the reorganization, but the index will be left in PSRBD status after the restarted job completes:
  - SHRLEVEL NONE is in effect for a partial table space reorganization.
  - The failure occurs after REORG PLUS has started updating the index.
When all of the following conditions exist, REORG PLUS changes
RESTART(PHASE) to RESTART and the table space is not reloaded again in the
restarted job:

— You specify RESTART(PHASE) to restart a partial table space reorganization for
which the following options were in effect:

■ SHRLLEVEL NONE
■ UNLOAD RELOAD

— During the original job or an earlier job, the document ID index was successfully
updated.

— The original job failed after the table space was reloaded.

**Partition-by-growth table spaces**

The following restrictions and considerations apply when you restart a
reorganization of a partition-by-growth table space:

■ REORG PLUS does not honor increases (ALTERs) to MAXPARTITIONS before a
restarted job. If REORG PLUS detects such an ALTER, it issues message
BMC50177I and continues with the reorganization as though the original
MAXPARTITIONS value were in effect.

■ You cannot change the value of the MAXNEWPARTS option; doing so causes the
REORG PLUS job to fail.

■ If a SHRLLEVEL NONE reorganization fails during reload processing in a
single-phase reorganization or unload processing in a two-phase reorganization
because insufficient space is available, (indicated by message BMC50174E or
message BMC51287E), BMC recommends that you perform an ALTER
TABLESPACE to either decrease PCTFREE or FREEPAGE or increase MAXROWS,
and then restart the job.

**Compressed indexes**

The following considerations apply when a compressed, non-unique, nonpartitioned
index is participating in the reorganization:

■ When both of the following conditions exist, you can restart the reorganization, but
the index will be left in PSRBD status after the restarted job completes:

— SHRLLEVEL NONE is in effect for a partial table space reorganization.
— The failure occurs after REORG PLUS has started updating the index.
When all of the following conditions exist, REORG PLUS changes
RESTART(PHASE) to RESTART and the table space is not reloaded again in the
restarted job:

— You specify RESTART(PHASE) to restart a partial table space reorganization for
which the following options were in effect:

  - SHRLEVEL NONE
  - UNLOAD RELOAD

— During the original job or an earlier job, the index was successfully updated.

— The original job failed after the table space was reloaded.

Non-data-sorting indexes

When restarting a failed SHRLEVEL REFERENCE partial reorganization with
non-data-sorting indexes, refer to Table 61 to determine whether non-data-sorting
indexes that were copied before the failure are recopied during restart processing. If
all data sets of a multi-data-set index are not copied before restart, the entire
multi-data-set index is recopied.

Table 61  Recopying of data sets for restart processing

<table>
<thead>
<tr>
<th>Type of reorganization</th>
<th>Phase for restart</th>
<th>Are data sets recopied?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RESTART</td>
</tr>
<tr>
<td>single-phase reorganization</td>
<td>REORG phase</td>
<td>no</td>
</tr>
<tr>
<td>two-phase reorganization</td>
<td>UNLOAD phase</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>RELOAD phase</td>
<td>no</td>
</tr>
</tbody>
</table>

Data sharing environment

On restart in a data sharing environment, REORG PLUS can use either the same
member chosen in the original reorganization or any other member in the specified
group.

SELECT and DELETE processing

You cannot restart a job that fails in the REORG phase when all of the following
conditions exist:

  - You are performing a SHRLEVEL NONE single-phase reorganization.
  - You allocated a SYSARC data set.
  - You are performing SELECT or DELETE processing.
Statistics

The following considerations apply to restarted jobs when you specify BMCSTATS YES or UPDATEDB2STATS YES:

- On restart, REORG PLUS does not update statistics if, in the original job, any participating table space partitions were completely loaded or any participating index partitions were completely built.

- You can change the TSSAMPLEPCT option when restarting a reorganization.

Failure due to inadequate space

Failure during the RELOAD or REORG phase can result in an unusable table space (not applicable for SHRLEVEL REFERENCE or SHRLEVEL CHANGE). The most likely cause of this failure is inadequate space in the DB2 data set. If the space is inadequate, either specify REDEFINE NO (command or installation option) and allocate new data sets for those that caused the failure, or increase the primary or secondary space values. Then restart the reorganization with the RESTART option. If you decide to reallocate any data sets that were successfully reloaded or rebuilt, however, you must restart the reorganization with RESTART(PHASE).

On any restart after UTILINIT, REORG PLUS does not use any changes to FREEPAGE, PCTFREE, MAXROWS, or PIECESIZE values. If REORG PLUS terminates with message BMC51287E, you must resubmit the job with an execution parameter of NEW.

CLONE option

You cannot add or remove the CLONE option when restarting a reorganization.

DELETEFILES

To restart your job during DELETEFILES processing, specify RESTART without (PHASE).

DSNUTILB reorganization

When restarting a DSNUTILB reorganization job, REORG PLUS passes the RESTART or RESTART(PHASE) parameter that you specified to DSNUTILB for processing.


**Dynamic allocation**

On restart, REORG PLUS automatically reallocates dynamically allocated data sets.

If you change any dynamic data set allocation option on restart and the change results in different ddnames or a different number of DDs than the original option had, you can receive an error. If you need to change the number of SYSREC or SYSUT1 data sets, you must resubmit the job with a parameter of NEW.

You cannot change the value for the ACTIVE option on any restart. To change the value of other dynamic data set allocation options, specify RESTART(PHASE).

**Inline image copies**

If you restart a SHRLEVEL NONE or SHRLEVEL REFERENCE table space reorganization job, REORG PLUS changes the value of the INLINE command to NO if all of the following statements are true:

- The table space is partitioned.
- You have a single image copy data set.
- At least one (but not all) of the partitions was reloaded before the failure.

This change occurs regardless of the value you specified for the INLINE command or the INLINECP installation option.

**Terminating or canceling a job**

If you want to end the utility immediately and want the ability to restart your job, cancel the job by using the MVS or TSO CANCEL command.

If you want to end the job and do not intend to restart, terminate the reorganization job by performing one of the following actions. If the job is currently running, it terminates at the next sync point.

- Delete the corresponding rows from the BMCUTIL, BMCSYNC, and BMCDICT tables.

- Specify TERM on the restart parameter of the utility. If you terminate a job, you cannot restart it. For more information, see “TERM” on page 346.
Recovering the DB2 object after terminating or canceling a job

After you terminate or cancel a reorganization job, you might need to perform steps to recover your DB2 objects. Table 62 lists the recovery steps for each phase and SHRLEVEL specification.

If REORG PLUS dynamically allocated your data sets, (by using the DDTYPE command option or through installation options), manually delete the data sets.

Table 62  Recovering objects after terminating or canceling jobs (part 1 of 2)

<table>
<thead>
<tr>
<th>Phase in which you terminated or canceled the job</th>
<th>SHRLEVEL specified</th>
<th>Recovery steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>terminated or canceled in UTILINIT or UNLOAD</td>
<td>any</td>
<td>All objects are usable and no additional steps are required.</td>
</tr>
<tr>
<td>DSNUTILB</td>
<td>SHRLEVEL NONE, SHRLEVEL REFERENCE, or SHRLEVEL CHANGE</td>
<td>For information about any recovery required, see the appropriate IBM DB2 documentation.</td>
</tr>
<tr>
<td>terminated in RELOAD or REORG</td>
<td>SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY</td>
<td>Recover any unusable objects.a</td>
</tr>
<tr>
<td></td>
<td>SHRLEVEL REFERENCE or SHRLEVEL CHANGE</td>
<td>Restart the objects in their original statuses to make them usable.</td>
</tr>
<tr>
<td>canceled in RELOAD or REORG</td>
<td>SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY</td>
<td>If you want the ability to restart the job:</td>
</tr>
<tr>
<td></td>
<td>SHRLEVEL REFERENCE or SHRLEVEL CHANGE</td>
<td>A. Reset the STATUS column in the BMCUTIL table to S (for Stopped).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B. Restart the job.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you do not plan to restart the job, recover any unusable objects.a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Restart the objects in their original statuses to make them usable.</td>
</tr>
</tbody>
</table>
Table 63  Recovering from a reorganization failure (part 1 of 2)

<table>
<thead>
<tr>
<th>Phase in which job fails</th>
<th>Recovery steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILINIT</td>
<td>Resubmit the job.</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>Correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>■ Correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td></td>
<td>■ For SHRLEVEL CHANGE, resubmit the job.</td>
</tr>
<tr>
<td>RELOAD - redefinition of VSAM data sets, reload and index build processing</td>
<td>■ Correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td></td>
<td>■ If the IDCAMS DEFINE failed, manually define the data set.</td>
</tr>
<tr>
<td></td>
<td>■ If you want to start your job at the beginning of the RELOAD phase, specify RESTART(PHASE).</td>
</tr>
<tr>
<td></td>
<td>■ For SHRLEVEL CHANGE, resubmit the job.</td>
</tr>
<tr>
<td>RELOAD - copy processing</td>
<td>■ Correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td></td>
<td>■ For SHRLEVEL CHANGE, resubmit the job.</td>
</tr>
</tbody>
</table>

Table 62  Recovering objects after terminating or canceling jobs (part 2 of 2)

<table>
<thead>
<tr>
<th>Phase in which you terminated or canceled the job</th>
<th>SHRLEVEL specified</th>
<th>Recovery steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>canceled or terminated in UTILTERM</td>
<td>SHRLEVEL NONE</td>
<td>All objects are usable and no additional steps are required.</td>
</tr>
<tr>
<td></td>
<td>or SHRLEVEL REFERENCE UNLOADONLY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHRLEVEL REFERENCE or SHRLEVEL CHANGE</td>
<td>BMC recommends that you restart your job rather than attempting to rename the staging data sets manually.</td>
</tr>
</tbody>
</table>

---

Recovering from a failure

Table 63 provides instructions for recovering after a reorganization job fails. The first column identifies the processing phases and, if applicable, the type of processing within the phase. The second column provides the steps to take when a reorganization job fails during that stage of processing.

Table 63  Recovering from a reorganization failure (part 1 of 2)

<table>
<thead>
<tr>
<th>Phase in which job fails</th>
<th>Recovery steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILINIT</td>
<td>Resubmit the job.</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>Correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>■ Correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td></td>
<td>■ For SHRLEVEL CHANGE, resubmit the job.</td>
</tr>
<tr>
<td>RELOAD - redefinition of VSAM data sets, reload and index build processing</td>
<td>■ Correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td></td>
<td>■ If the IDCAMS DEFINE failed, manually define the data set.</td>
</tr>
<tr>
<td></td>
<td>■ If you want to start your job at the beginning of the RELOAD phase, specify RESTART(PHASE).</td>
</tr>
<tr>
<td></td>
<td>■ For SHRLEVEL CHANGE, resubmit the job.</td>
</tr>
<tr>
<td>RELOAD - copy processing</td>
<td>■ Correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td></td>
<td>■ For SHRLEVEL CHANGE, resubmit the job.</td>
</tr>
</tbody>
</table>
Recovering from a failure

Chapter 4 Building and executing REORG PLUS jobs 399

For SHRLEVEL REFERENCE, correct the problem that caused the failure and restart the job with RESTART.

For SHRLEVEL NONE, take the appropriate action:

— If you specified a SYSREC data set for a table space reorganization, or a SYSUT1 data set for an index reorganization, correct the problem that caused the failure and restart the job with RESTART.

— If you did not specify a SYSREC data set for a table space reorganization, or a SYSUT1 data set for an index reorganization, and if REORG PLUS did not dynamically allocate these data sets, you will probably not be able to restart the job, but you can attempt to do so with RESTART.

If you receive message BMC50009S indicating that REORG PLUS is unable to restart, recover any objects in recovery pending state, manually delete any dynamically allocated data sets, and resubmit the job.

For SHRLEVEL CHANGE, correct the problem and resubmit the job.

<table>
<thead>
<tr>
<th>Phase in which job fails</th>
<th>Recovery steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG</td>
<td>For SHRLEVEL REFERENCE, correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td></td>
<td>For SHRLEVEL NONE, take the appropriate action:</td>
</tr>
<tr>
<td></td>
<td>— If you specified a SYSREC data set for a table space reorganization, or a SYSUT1 data set for an index reorganization, correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td></td>
<td>— If you did not specify a SYSREC data set for a table space reorganization, or a SYSUT1 data set for an index reorganization, and if REORG PLUS did not dynamically allocate these data sets, you will probably not be able to restart the job, but you can attempt to do so with RESTART.</td>
</tr>
<tr>
<td></td>
<td>If you receive message BMC50009S indicating that REORG PLUS is unable to restart, recover any objects in recovery pending state, manually delete any dynamically allocated data sets, and resubmit the job.</td>
</tr>
<tr>
<td></td>
<td>For SHRLEVEL CHANGE, correct the problem and resubmit the job.</td>
</tr>
<tr>
<td>LOGAPPLY</td>
<td>For SHRLEVEL CHANGE only, resubmit the job.</td>
</tr>
<tr>
<td>LOGFINAL</td>
<td>For SHRLEVEL CHANGE only, resubmit the job.</td>
</tr>
<tr>
<td>UTILTERM</td>
<td>Correct the problem that caused the failure and restart the job with RESTART. Note the following additional information:</td>
</tr>
<tr>
<td></td>
<td>— For SHRLEVEL REFERENCE and SHRLEVEL CHANGE, if the failure occurred while processing limit-key ALTER statements, the value of the ALTRFAIL installation option determines the action that REORG PLUS takes. This option applies to ALTER statements processed either from your DDLIN data set or as a result of the REBALANCE option.</td>
</tr>
<tr>
<td></td>
<td>— For SHRLEVEL CHANGE, REORG PLUS backs out the reorganization and leaves the spaces in their original statuses when any of the following conditions exists:</td>
</tr>
<tr>
<td></td>
<td>— The failure occurred while REORG PLUS was setting restrictive statuses and conditions exist that might cause a restarted reorganization to create invalid indexes.</td>
</tr>
<tr>
<td></td>
<td>— You are reorganizing a catalog object and the failure occurred during rename processing.</td>
</tr>
<tr>
<td></td>
<td>If you cannot restart the job, see “Not completing the reorganization in the UTILTERM phase” on page 400 for SHRLEVEL REFERENCE and SHRLEVEL CHANGE.</td>
</tr>
</tbody>
</table>
Not completing the reorganization in the UTILTERM phase

If you cannot restart a SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganization, you can either complete the reorganization manually or back it out (as if no reorganization had begun). For the steps to perform, see the following sections:

- “Completing when using a DDLIN data set or specifying REBALANCE” on page 400
- “Completing when not using a DDLIN data set or specifying REBALANCE” on page 400
- “Back out the reorganization” on page 401

Completing when using a DDLIN data set or specifying REBALANCE

If you are using a DDLIN data set or specifying REBALANCE, look for message BMC51297I in your SYSPRINT. If the message is displayed, the ALTER statement finished and you must complete the reorganization by issuing a DB2 START command with ACCESS(FORCE).

If message BMC51297I does not appear in SYSPRINT, the ALTER statement did not finish. You cannot complete the reorganization and must go back to the starting point. If you specified ALTRFAIL=TERM in your installation options, REORG PLUS performs the back out for you. For more information, see “Back out the reorganization” on page 401.

Completing when not using a DDLIN data set or specifying REBALANCE

If you are not using a DDLIN data set or specifying REBALANCE, and you want to complete the reorganization manually, first determine whether the FASTSWITCH or rename process completed. The appearance of message BMC50895I in SYSPRINT indicates that the FASTSWITCH or rename process completed.

FASTSWITCH or rename processing completed

If the FASTSWITCH or rename process completed, perform the following steps:

1. Issue a DB2 START command with ACCESS (UT) for the objects that you are reorganizing.

2. Remove the appropriate pending flags, such as RECP.
3 Run the MODIFY utility to remove SYSCOPY entries for prior utilities and image copies.

4 Take a full image copy.

5 Issue a DB2 START command to start your object in its original status.

**FASTSWITCH or rename processing did not complete**

If the FASTSWITCH or rename process did not complete, perform the following steps:

1 Issue a DB2 STOP command for the objects that you are going to rename.

2 Complete the renaming operations manually:
   - If you are using the rename process, rename the data sets that REORG PLUS did not finish renaming. For more information, see “Staging data sets and the rename process” on page 103.
   - If you are using the FASTSWITCH process, change the staging data set names to the original data set names, and change the original data set names to the staging data set names. For more information, see “Staging data sets and the FASTSWITCH process” on page 105.

3 Issue a DB2 START command with ACCESS (UT) for the objects that you are reorganizing.

4 Remove the appropriate pending flags, such as RECP.

5 Run the MODIFY utility to remove SYSCOPY entries for prior utilities and image copies.

6 Take a full image copy.

7 Issue a DB2 START command to start your object in its original status.

**Backing out the reorganization**

Use the information in this section when either of the following conditions exists:

- You are not using a DDLIN data set or specifying REBALANCE, and you want to back out the reorganization.

- You are using a DDLIN data set or specifying REBALANCE, the ALTER statement did not complete, and the value of the ALTRFAIL installation option is RCVRPEND.
Not completing the reorganization in the UTILTERM phase

NOTE

REORG PLUS backs out the reorganization for you in the following cases:

- if the failure occurs during processing of limit-key ALTER statements (rebalancing partitions) and the value of the ALTRFAIL installation option is TERM
- for SHRLEVEL CHANGE, if the failure occurs while REORG PLUS is setting restrictive statuses, and a restarted reorganization might create invalid indexes
- for SHRLEVEL CHANGE, if you are reorganizing a catalog object and the failure occurs during rename processing

The steps to follow depend on whether you are using the rename process or the FASTSWITCH process.

Backing out when using the rename process

To go back to the starting point, perform the following steps:

1. Back out the renames that REORG PLUS has already completed.
   
   For more information, see “Staging data sets and the rename process” on page 103.

2. Issue a DB2 START command with ACCESS (FORCE) to start your object with its original status.

Backing out when using the FASTSWITCH process

Perform the following steps, depending on whether the FASTSWITCH process completed.

- If the FASTSWITCH process completed, as indicated by message BMC50895I in SYSPRINT, complete the following steps:

  A. Issue a DB2 STOP command for the objects that you are going to rename.

  B. Change the staging data set names to the original data set names, and the original data set names to the staging data set names.

  C. Issue a DB2 START command with ACCESS (FORCE) for the objects that you are reorganizing.

- If the FASTSWITCH process did not complete, issue a DB2 START command with ACCESS (FORCE) for the objects that you are reorganizing.
Chapter 5 Examples of REORG PLUS jobs

This chapter presents the following reorganization examples:

Overview ................................................................. 403
Example 1: VCAT-defined segmented table space with SYSIDCIN ........ 407
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation ................................................. 416
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions ........................................... 441
Example 4: Index-only reorganization ........................................ 463
Example 5: Partial reorganization with UNLOAD PAUSE ............... 468
Example 6: Restart of a paused REORG job .................................. 474
Example 7: ANALYZE ONLY to generate space estimates ............. 480
Example 8: Selective unload with discards to archive data set .......... 491
Example 9: ON FAILURE with a user-specified return code .......... 499
Example 10: DSNUTILB reorganization and index that contains keys with random ordering ......................................................... 506
Example 11: Partition-by-growth table space ............................ 517
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning ................. 529
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY .... 542
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing ....................................................... 553

Overview

This chapter presents several examples of jobs that were run by using the REORG PLUS product. Each example includes the following information:

- a description of the reorganization job
- the REORG PLUS job stream
- the SYSPRINT from the job
Some examples include additional output that might be useful to understand a particular feature.

All of these examples have the following common properties:

- These examples use a single BMC product load library, assuming that this library contains the following files and libraries:
  - REORG PLUS load files
  - common component load files that REORG PLUS requires
  - SAS/C transient library (required for statistics processing)

- When data sets are allocated in the JCL, these examples use minimal data set space allocations.

- The value for the FILECHK installation option is set to WARN. This value prevents the jobs from failing when they encounter temporary data sets.

You can find copies of the JCL for these examples in member ARUEXnn in the HLQ.LLQCNTL installation data set (where nn is the number of the example, HLQ is the high-level qualifier specified during installation, and LLQ is the low-level qualifier or prefix set during installation).

For syntax details, see Chapter 3, “Syntax of the REORG command.” For information about JCL statement requirements, see Chapter 4, “Building and executing REORG PLUS jobs.”

Use Table 64 to locate an example with a specific reorganization, process, or object type, or one that uses a specific option. Use the chapter table of contents to find the example.

### Table 64  Cross-reference of examples by function  (part 1 of 3)

<table>
<thead>
<tr>
<th>Function</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reorganization type</strong></td>
<td></td>
</tr>
<tr>
<td>full (entire table space/all partitions)</td>
<td>1, 2, 7, 8, 9, 10-13</td>
</tr>
<tr>
<td>partial (selected partitions)</td>
<td>3, 5, 6</td>
</tr>
<tr>
<td>index only</td>
<td>4</td>
</tr>
<tr>
<td>two phase</td>
<td>8, 10-12</td>
</tr>
<tr>
<td>DSNUTILB reorganization</td>
<td>10</td>
</tr>
<tr>
<td><strong>Object type</strong></td>
<td></td>
</tr>
<tr>
<td>partitioned table space</td>
<td>2, 3, 5, 6, 12-14</td>
</tr>
<tr>
<td>segmented table space</td>
<td>1, 7-10</td>
</tr>
<tr>
<td>partition-by-growth table space</td>
<td>11</td>
</tr>
<tr>
<td>VCAT-defined table space</td>
<td>1</td>
</tr>
<tr>
<td>Function</td>
<td>Examples</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>clustering index</td>
<td>1-3, 5-7, 11-13</td>
</tr>
<tr>
<td>nonpartitioned secondary index</td>
<td>2-6, 8, 12-14</td>
</tr>
<tr>
<td>data-partitioned secondary index</td>
<td>14</td>
</tr>
<tr>
<td>index created with DEFER YES</td>
<td>1</td>
</tr>
<tr>
<td>index containing keys with random ordering</td>
<td>10</td>
</tr>
<tr>
<td>ASUSRPRT data set</td>
<td>3, 14</td>
</tr>
<tr>
<td>ASUSRPRT DD DUMMY</td>
<td>2, 6</td>
</tr>
<tr>
<td>DDLIN data set</td>
<td>12</td>
</tr>
<tr>
<td>DDOUT data set</td>
<td>14</td>
</tr>
<tr>
<td>SYSIDCIN data set</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command option</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE NO</td>
<td>1, 2, 9, 12</td>
</tr>
<tr>
<td>ACTIVE YES</td>
<td>3, 9, 10</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>12, 13</td>
</tr>
<tr>
<td>ANALYZE ONLY</td>
<td>7</td>
</tr>
<tr>
<td>ANALYZE SAMPLE</td>
<td>11, 14</td>
</tr>
<tr>
<td>ANALYZE SCAN</td>
<td>2</td>
</tr>
<tr>
<td>ARCHDDN</td>
<td>8</td>
</tr>
<tr>
<td>BMCSTATS YES</td>
<td>2, 3, 5, 6, 11, 14</td>
</tr>
<tr>
<td>COPY YES</td>
<td>1, 2, 3, 8-14</td>
</tr>
<tr>
<td>COPYDDN</td>
<td>2, 8, 10</td>
</tr>
<tr>
<td>COPYLVL</td>
<td>2, 3, 10, 11</td>
</tr>
<tr>
<td>DTYPE</td>
<td>1-6, 9-14</td>
</tr>
<tr>
<td>DEADLINE</td>
<td>12</td>
</tr>
<tr>
<td>DELAY</td>
<td>12-14</td>
</tr>
<tr>
<td>DELETEFILES NO</td>
<td>1</td>
</tr>
<tr>
<td>DSNPAT</td>
<td>2-6, 9-14</td>
</tr>
<tr>
<td>FASTSWITCH</td>
<td>12-14</td>
</tr>
<tr>
<td>IFALLOC</td>
<td>10</td>
</tr>
<tr>
<td>ICTYPE</td>
<td>14</td>
</tr>
<tr>
<td>INLINE YES</td>
<td>10</td>
</tr>
<tr>
<td>KEEPDICTIONARY NO</td>
<td>11</td>
</tr>
<tr>
<td>LOGMEM</td>
<td>13</td>
</tr>
<tr>
<td>LOGSPILL</td>
<td>12-14</td>
</tr>
<tr>
<td>LOGTHRESHLD</td>
<td>14</td>
</tr>
<tr>
<td>LONGLOG</td>
<td>13</td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td>10</td>
</tr>
<tr>
<td>Function</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>MAXRO</td>
<td>12, 13</td>
</tr>
<tr>
<td>MAXNEWPARTS</td>
<td>11</td>
</tr>
<tr>
<td>ON FAILURE</td>
<td>9</td>
</tr>
<tr>
<td>ORDER NO</td>
<td>7, 8</td>
</tr>
<tr>
<td>ORDER YES</td>
<td>7, 9-14</td>
</tr>
<tr>
<td>PART</td>
<td>3, 5, 6</td>
</tr>
<tr>
<td>PREFORMAT YES</td>
<td>10, 11, 14</td>
</tr>
<tr>
<td>REBALANCE</td>
<td>14</td>
</tr>
<tr>
<td>RECOVERYDDN</td>
<td>8, 10</td>
</tr>
<tr>
<td>REDEFINE NO</td>
<td>3, 4, 8, 10</td>
</tr>
<tr>
<td>REDEFINE YES</td>
<td>1</td>
</tr>
<tr>
<td>REGISTER</td>
<td>8, 11</td>
</tr>
<tr>
<td>REORG INDEX</td>
<td>4</td>
</tr>
<tr>
<td>RIDMAPMEM</td>
<td>13</td>
</tr>
<tr>
<td>SELECT (selective unload)</td>
<td>8</td>
</tr>
<tr>
<td>SET</td>
<td>2</td>
</tr>
<tr>
<td>SHRLEVEL NONE</td>
<td>10, 11</td>
</tr>
<tr>
<td>SHRLEVEL REFERENCE</td>
<td>2, 3</td>
</tr>
<tr>
<td>SHRLEVEL CHANGE</td>
<td>12-14</td>
</tr>
<tr>
<td>SIZEPCT</td>
<td>12-14</td>
</tr>
<tr>
<td>SORTDATA</td>
<td>9, 10</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>10</td>
</tr>
<tr>
<td>SORTNUM</td>
<td>10</td>
</tr>
<tr>
<td>SPILLDSNPAT</td>
<td>12-14</td>
</tr>
<tr>
<td>SPILLSTORCLAS</td>
<td>12-14</td>
</tr>
<tr>
<td>SPILLUNIT</td>
<td>12-14</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>10</td>
</tr>
<tr>
<td>UNLOAD PAUSE</td>
<td>5, 6</td>
</tr>
<tr>
<td>UNIT</td>
<td>10-14</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>10</td>
</tr>
<tr>
<td>UNLOAD CONTINUE</td>
<td>7, 8, 10-12</td>
</tr>
<tr>
<td>UPDATE</td>
<td>2</td>
</tr>
<tr>
<td>UPDATEDB2STATS YES</td>
<td>1, 5, 6, 10, 11, 14</td>
</tr>
<tr>
<td>VOLCNT</td>
<td>10</td>
</tr>
<tr>
<td>WHERE clause</td>
<td>2, 8</td>
</tr>
<tr>
<td>XBMID</td>
<td>12-14</td>
</tr>
</tbody>
</table>
Example 1: VCAT-defined segmented table space with SYSIDCIN

In this example, REORG PLUS reorganizes a segmented VCAT-defined table space. The table space has a clustering index that was defined using the DB2® CREATE INDEX DEFER command. The REORG PLUS job populates the clustering index as part of the reorganization.

REORG PLUS issues message BMC50482I in the SYSPRINT when it creates the index. Message BMC51276I indicates how many keys were loaded into the index. Because this example uses a VCAT-defined table space, a SYSIDCIN DD statement and its contents are included in the JCL.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 65 describes the key command options and DD statements for this job.

<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>COPY YES</td>
<td>creates a DB2 image copy of the table space</td>
</tr>
<tr>
<td></td>
<td>Because the default for the INLINECP installation option is YES, this copy is an inline image copy.</td>
</tr>
<tr>
<td></td>
<td>Because REGISTER is not specified, this statement defaults to REGISTER ALL. REORG PLUS makes four copies of the table space after reorganizing it, because four ddnames are specified in the JCL. The job registers all four copies in SYSIBM.SYSCOPY.</td>
</tr>
<tr>
<td>UPDATEDB2STATS YES</td>
<td>requests that statistics be updated in the DB2 catalog</td>
</tr>
<tr>
<td></td>
<td>The SYSPRINT shows the messages that the Common Statistics component displays for the statistics being updated in the DB2 catalog.</td>
</tr>
<tr>
<td>DELETEFILES NO</td>
<td>overrides the default and requests that REORG PLUS not delete the work files upon completion of the reorganization</td>
</tr>
<tr>
<td>DDTYPE ... ACTIVE NO</td>
<td>overrides the default and requests that REORG PLUS not enable dynamic allocation for the specified data sets</td>
</tr>
<tr>
<td>//SYSIDCIN</td>
<td>contains the IDCAMS command statements used to redefine VCAT-defined data sets</td>
</tr>
</tbody>
</table>
Table 65  Example 1 key command options and DD statements (part 2 of 2)

<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//SYSREC</td>
<td>contains the rows that you are reorganizing</td>
</tr>
<tr>
<td>//BMCCPY</td>
<td>the default ddnames that are used for the data sets that receive a full image copy of the table space that you are reorganizing</td>
</tr>
<tr>
<td>//BMCCPZ</td>
<td>The existence of the copy ddnames determines the number of copies made when COPY YES is specified.</td>
</tr>
<tr>
<td>//BMCRCY</td>
<td></td>
</tr>
<tr>
<td>//BMCRCZ</td>
<td></td>
</tr>
</tbody>
</table>

Figure 14 shows JCL for example 1.

Figure 14  JCL for example 1 (part 1 of 2)

```
//     JOB
//*
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
/* SEGMENTED TABLESPACE, VCAT DEFINED, INDEX CREATED WITH DEFER * */
/* YES. REORG TABLESPACE WITH REDEFINE YES AND SYSIDCIN IS USED, * */
/* 4 COPIES (TWO LOCALS & TWO REMOTES), */
/* REGISTER ALL COPIES (DEFAULT), UPDATE DB2 CATALOG WITH STATS */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='DEFF,ARU01,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB   DD  DISP=SHR,DSN=productLibraries
//          DD  DISP=SHR,DSN=DB2.DSNEXIT
//          DD  DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT DD  SYSOUT=*     
//SYSOUT   DD  SYSOUT=*     
//UTPRINT   DD  SYSOUT=*     
//SYSUDUMP  DD  SYSOUT=*     
//SYSTERM   DD  SYSOUT=*     
//*
//SYSIDCIN  DD  *            
DELETE -
   (DEFFCAT.DSNDBC.ARUB001.TS00111.I0001.A001) -
PURGE -
CLUSTER -
CATALOG(DEFFCAT)
DELETE -
   (DEFFCAT.DSNDBC.ARUB001.TS001.I0001.A001) -
PURGE -
CLUSTER -
CATALOG(DEFFCAT)
DEFINE CLUSTER(
   NAME(DEFFCAT.DSNDBC.ARUB001.TS001.I0001.A001) -
   LINEAR -
   REUSE -
   VOLUMES(TM0053 ARU4CD ARU112) -
```
Example 1: VCAT-defined segmented table space with SYSIDCIN

Chapter 5 Examples of REORG PLUS jobs

Figure 14 JCL for example 1 (part 2 of 2)

```
CYLINDERS (1 1) -
SHAREOPTIONS(3 3) -
)
DATA (-
    NAME(DEFFCAT.DSNDBD.ARUDBO01.TS001.I0001.A001) -
)
CATALOG(DEFFCAT)
DEFINE CLUSTER( -
    NAME(DEFFCAT.DSNDBC.ARUDBO01.TS00111.I0001.A001) -
    LINEAR -
    REUSE -
    VOLUMES(TM0055 ARU4CD ARU112) -
    CYLINDERS (10 5) -
    SHAREOPTIONS(3 3) -
)
DATA (-
    NAME(DEFFCAT.DSNDBD.ARUDBO01.TS00111.I0001.A001) -
)
CATALOG(DEFFCAT)
/*
//SYSREC    DD DSN=ARU.EXMPL01.SYSREC,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//*
//BMCCPY    DD DSN=ARU.EXMPL01.BMCCPY,
//          UNIT=WORK,SPACE=(CYL,(5,5)),
//          DISP=(,CATLG)
//BMCCPZ    DD DSN=ARU.EXMPL01.BMCCPZ,
//          UNIT=WORK,SPACE=(CYL,(5,5)),
//          DISP=(,CATLG)
//BMCRCY    DD DSN=ARU.EXMPL01.BMCRCY,
//          UNIT=WORK,SPACE=(CYL,(5,5)),
//          DISP=(,CATLG)
//BMCRCZ    DD DSN=ARU.EXMPL01.BMCRCZ,
//          UNIT=WORK,SPACE=(CYL,(5,5)),
//          DISP=(,CATLG)
//*/
//SYSIN     DD  *
REORG TABLESPACE ARUDBO01.TS001
COPY YES
UPDATEDB2STATS YES
REDEFINE YES
DELETEFILES NO
DDTYPE UNLOAD ACTIVE NO
DDTYPE WORK ACTIVE NO
DDTYPE LOCPFCPY ACTIVE NO
/*
```
Example 1: VCAT-defined segmented table space with SYSIDCIN

Figure 15 shows the SYSPRINT output for example 1.

### SYSPRINT for example 1 (part 1 of 7)

| BMC50471I | REORG PLUS FOR DB2--V11.01.00 |
| BMCS001I | UTILITY EXECUTION STARTING 10/14/2013 16:08:06 ... |
| BMCS002I | UTILITY ID = 'ARU01'. DB2 SUBSYSTEM ID = 'DEFF'. OPTION MODULE = 'ARU$OPTS'. |
| BMCS0471I | z/OS 2.1.0,PID=HBB7790,DFSMS FOR Z/OS=2.1.0,DB2=10.1.0 |
| BMCS0471I | REGION=ON.BELLOW 16M=8824K,ABOVE 16M=1407336K,EXUS1=10,CPUS=3 |
| BMCS0471I | MEMLIMIT=17592186040320M,AVAILABLE=17592186040306M,MEMLIMIT SET BY:REGION=0 |
| BMCS0471I | REORG PLUS FOR DB2--V11.01.00 |
| BMCS0471I | NO MAINTENANCE TO REPORT |
| BMCS0471I | DB2 UTILITIES COMMON CODE--V11.01.00 |
| BMCS0471I | NO MAINTENANCE TO REPORT |
| BMCS0471I | SOLUTION COMMON CODE--V11.01.00 |
| BMCS0471I | MAINT: BPJ0661 BPJ0667 BPJ0671 BPJ0674 BPJ0675 BPJ0676 BPJ0682 BPJ0686 BPJ0689 BPJ0697 |
| BMCS0471I | BMCSORT ENGINE--V02.04.01 |
| BMCS0471I | NO MAINTENANCE TO REPORT |
| BMCS0471I | BMCS堆 API--V11.01.00 |
| BMCS0471I | NO MAINTENANCE TO REPORT |
| BMCS0471I | BMCS堆 EXTENDED BUFFER MANAGER--V06.01.00 |
| BMCS0471I | NO MAINTENANCE TO REPORT |

| BMC50471I | ALTRFAIL=RCVRPEND | IDCNN=(BMCCY,BMCCIZ) | RRORMAX=300 |
| BMC50471I | ANALMAX=1000% | ICTYPE=AUTO | ROUTCDE=(11,1) |
| BMC50471I | ARC=NO | IDCACHE=1000 | SCYPMA=0 |
| BMC50471I | ARCHOND-SYSARC | INDFRM=10 | SQMP=NO |
| BMC50471I | ARCHFORMAT=BMC | INLNEP=NO | SHORTMEMORY=CONTINUE |
| BMC50471I | AVALPAGFCT=100 | IXONEX=NO | SIXSNP=NO |
| BMC50471I | BIGMAX=300% | IXRANDOM=NO | SMAX=0 |
| BMC50471I | BMCHIST=NO | KEPTDICT=NO | SYSZ=0 |
| BMC50471I | CBUFFFS=30 | LEAFDSLM=200 | SORTDFSY=(SYSSALDA) |
| BMC50471I | CONDEXEC=NO | LDB=NO | SORTNUM=32 |
| BMC50471I | CPDYON=(BMCCPY,BMCCPZ) | LOCKKRW=NO | SPLDLSPN=40ID |
| BMC50471I | COPYLYX=PART | LOGFINAL=NONE | SPLILSCL=NONE |
| BMC50471I | COPYMAX=1000% | LOGMEM=0 | SPLILN=WORK |
| BMC50471I | COPYSUBSET=NO | LOGSPIL=(20000,10000) | SOLDELAY=3 |
| BMC50471I | CPYFAIL=TERM | LOGTHRS=0 | SOLRETRY=100 |
| BMC50471I | DATACAP=NO | LOGLOG=CONTINUE | STAGGOS=BMCC |
| BMC50471I | DOLDON=DOLIN | LONGAMTRAN=MIDDLE | STORMS=BMCC |
| BMC50471I | DEADLINE=NONE | MAPEXIT=(NONE,REXX) | STOPDELAY=1 |
| BMC50471I | DELAY=1200 | MAXNEWPARTS=2 | STOPPREY=300 |
| BMC50471I | DELFILES=NO | MAXR=300 | TAPEDISP=DELETE |
| BMC50471I | DESCLOC=(3,7) | MAXSORTMEMORY=0 | TASKMAX=1000 |
| BMC50471I | DRAINTP=ALL | MAXTAPE=3 | TERMEXIT=(NONE,REXX) |
| BMC50471I | DNDDELAY=1 | MGEXT=CONTINUE | TIMEDOUT=TERM |
| BMC50471I | DRNRKEY=255 | MINSORTMEMORY=0 | TDAPLGAPFCT=0 |
| BMC50471I | DRNRWAIT=NONE | MSGLEVEL=1 | TDAPLGAPFCT=0 |
| BMC50471I | DSMEXIT=(NONE,ASM) | OFFPDSL=0 | TSAMPLEPCT=100 |
| BMC50471I | DSNUTILB=YES | OPDB2ID=YES | UBUFFS=20 |
| BMC50471I | DSNUTLBY=YES | ORIGDID=DELETE | UNDD=SYSREC |
| BMC50471I | DSPACECS=DSNFAIL | PENMOD=NO | UNLDMAX=300 |
| BMC50471I | DSPACECT=(NONE,ASM) | PENDEXIT=NO | UNLO=RELOAD |
| BMC50471I | DSPACEEX=(X37,X22,X06) | PREFORMAT=NO | UTILB=COLCSSID=UTILB |
| BMC50471I | FASTSYNC=NO | RCVIDON=(BMCCY,BMCCIZ) | UTILB_NULLIX=UTILB |
| BMC50471I | FILECHK=WARN | RCVYON=(BMCCY,BMCCIZ) | UXSTATE=SUP |
| BMC50471I | FORCE_AT=(START,3) | REDEFINE=NO | WRFUS=(10,10) |
| BMC50471I | FORCE_ATP=NO | REMMAP=30 | WORKCOM=SYSUT1 |
| BMC50471I | FORCE=NONE | RIMMDSZ=2097152 | WORKUNIT=SYSSALDA |
| BMC50471I | FSFAILBACK=NO | RIDMMAX=1 | XBMID= |
| BMC50471I | FSTHRESHOLD=0 | RMAPMEM=0 | ZILP=ENABLED |
| BMC50471I | HASHX=NO |

| BMC50471I | PLAN=A4UQA |

| BMC50471I | DOTYPE = UNLOAD | WORK | SORTrWORK |
| BMC50471I | ACTIVE = YES | YES | NO |
| BMC50471I | IFALLOC = USE | USE | USE |
| BMC50471I | ALLOC = N/A | N/A | ANY |
| BMC50471I | SMS = NO | NO | NO |
| BMC50471I | SMUNIT = NO | NO | NO |
| BMC50471I | SIZEPECT = (100,100) | (100,100) | (100,100) |
| BMC50471I | UNIT = (SYSSALDA,SYSSALDA) | (SYSSALDA,SYSSALDA) | (SYSSALDA,SYSSALDA) |
| BMC50471I | UNITCNT = (0,0) | (0) | N/A |
### Figure 15  SYSPRINT for example 1 (part 2 of 7)

<table>
<thead>
<tr>
<th>BMC50470I VOLTCL</th>
<th>(25,25)</th>
<th>(25,25)</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I AVGAVLPS</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(30000,TRK),(30000,TRK)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I THRESHOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAKETHS</td>
<td>((0,K),(0,K))</td>
<td>(0,K),(0,K)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I GDSSCRAT</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I ODTYPF</td>
<td>ARCHIVE</td>
<td>LOPCPFCPY</td>
<td>LOPCPFCPY</td>
</tr>
<tr>
<td>BMC50470I ACTIVF</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>BMC50470I IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SMUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I UNIT</td>
<td>(SYSALLDA)</td>
<td>(SYSALLDA)</td>
<td>(SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(5,100)</td>
</tr>
<tr>
<td>BMC50470I UNICTN</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I AVGAVLPS</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(30000,TRK),(30000,TRK)</td>
<td>(30000,TRK),(30000,TRK)</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I THRESHOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAKETHS</td>
<td>((0,K),(0,K))</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I GDSSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I ODTYPF</td>
<td>LOPCPFCPY</td>
<td>LOPCPFCPY</td>
<td>LOPCPFCPY</td>
</tr>
<tr>
<td>BMC50470I ACTIVF</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SMUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I UNICTN</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I AVGAVLPS</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(30000,TRK),(30000,TRK)</td>
<td>(30000,TRK),(30000,TRK)</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I THRESHOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAKETHS</td>
<td>((0,K),(0,K))</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td>REMPIF</td>
<td>REMPIF</td>
<td>REMPIF</td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I GDSSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

**Example 1: VCAT-defined segmented table space with SYSIDCIN**
### Figure 15  SYSPRINT for example 1 (part 3 of 7)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MAXEXTSZ</strong></td>
<td>((0,K),(0,K))</td>
<td><strong>MAXEXTSZ</strong></td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td><strong>EXPD</strong></td>
<td></td>
<td><strong>EXPD</strong></td>
<td></td>
</tr>
<tr>
<td><strong>RETPD</strong></td>
<td></td>
<td><strong>RETPD</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GOLIMIT</strong></td>
<td>5</td>
<td><strong>GOLIMIT</strong></td>
<td>5</td>
</tr>
<tr>
<td><strong>GGEMPT</strong></td>
<td>NO</td>
<td><strong>GGEMPT</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>GGSCRAT</strong></td>
<td>NO</td>
<td><strong>GGSCRAT</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>DTYPE</strong></td>
<td>SYSPUNCH</td>
<td><strong>DTYPE</strong></td>
<td>SYSPUNCH</td>
</tr>
<tr>
<td><strong>ACTIVE</strong></td>
<td>YES</td>
<td><strong>ACTIVE</strong></td>
<td>YES</td>
</tr>
<tr>
<td><strong>IFALLOC</strong></td>
<td>USE</td>
<td><strong>IFALLOC</strong></td>
<td>USE</td>
</tr>
<tr>
<td><strong>ALLOC</strong></td>
<td>N/A</td>
<td><strong>ALLOC</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>SMS</strong></td>
<td>NO</td>
<td><strong>SMS</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>SMUNIT</strong></td>
<td>NO</td>
<td><strong>SMUNIT</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>SIZEPCT</strong></td>
<td>(100,100)</td>
<td><strong>SIZEPCT</strong></td>
<td>(100,100)</td>
</tr>
<tr>
<td><strong>UNIT</strong></td>
<td>(SYSALLDA,SYSALLDA)</td>
<td><strong>UNIT</strong></td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td><strong>UNITCNT</strong></td>
<td>(0,0)</td>
<td><strong>UNITCNT</strong></td>
<td>(0,0)</td>
</tr>
<tr>
<td><strong>VOLCNT</strong></td>
<td>(25,25)</td>
<td><strong>VOLCNT</strong></td>
<td>(25,25)</td>
</tr>
<tr>
<td><strong>AVGOLSP</strong></td>
<td>((30000,TRK),(30000,TRK))</td>
<td><strong>AVGOLSP</strong></td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td><strong>DATACLAS</strong></td>
<td>(NONE,NONE)</td>
<td><strong>DATACLAS</strong></td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td><strong>STORCLAS</strong></td>
<td>(NONE,NONE)</td>
<td><strong>STORCLAS</strong></td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td><strong>THRESHLD</strong></td>
<td>0</td>
<td><strong>THRESHLD</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>MAXEXTSZ</strong></td>
<td>((0,K),(0,K))</td>
<td><strong>MAXEXTSZ</strong></td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td><strong>EXPD</strong></td>
<td></td>
<td><strong>EXPD</strong></td>
<td></td>
</tr>
<tr>
<td><strong>RETPD</strong></td>
<td></td>
<td><strong>RETPD</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GOLIMIT</strong></td>
<td>5</td>
<td><strong>GOLIMIT</strong></td>
<td>5</td>
</tr>
<tr>
<td><strong>GGEMPT</strong></td>
<td>NO</td>
<td><strong>GGEMPT</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>GGSCRAT</strong></td>
<td>NO</td>
<td><strong>GGSCRAT</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>DTYPE</strong></td>
<td>LOCLCPY</td>
<td><strong>DTYPE</strong></td>
<td>LOCLCPY</td>
</tr>
<tr>
<td><strong>ACTIVE</strong></td>
<td>NO</td>
<td><strong>ACTIVE</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>IFALLOC</strong></td>
<td>USE</td>
<td><strong>IFALLOC</strong></td>
<td>USE</td>
</tr>
<tr>
<td><strong>ALLOC</strong></td>
<td>N/A</td>
<td><strong>ALLOC</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>SMS</strong></td>
<td>NO</td>
<td><strong>SMS</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>SMUNIT</strong></td>
<td>NO</td>
<td><strong>SMUNIT</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>SIZEPCT</strong></td>
<td>(100,100)</td>
<td><strong>SIZEPCT</strong></td>
<td>(100,100)</td>
</tr>
<tr>
<td><strong>UNIT</strong></td>
<td>(SYSALLDA,SYSALLDA)</td>
<td><strong>UNIT</strong></td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td><strong>UNITCNT</strong></td>
<td>(0,0)</td>
<td><strong>UNITCNT</strong></td>
<td>(0,0)</td>
</tr>
<tr>
<td><strong>VOLCNT</strong></td>
<td>(25,25)</td>
<td><strong>VOLCNT</strong></td>
<td>(25,25)</td>
</tr>
<tr>
<td><strong>AVGOLSP</strong></td>
<td>((30000,TRK),(30000,TRK))</td>
<td><strong>AVGOLSP</strong></td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td><strong>DATACLAS</strong></td>
<td>(NONE,NONE)</td>
<td><strong>DATACLAS</strong></td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td><strong>STORCLAS</strong></td>
<td>(NONE,NONE)</td>
<td><strong>STORCLAS</strong></td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td><strong>THRESHLD</strong></td>
<td>0</td>
<td><strong>THRESHLD</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>MAXEXTSZ</strong></td>
<td>((0,K),(0,K))</td>
<td><strong>MAXEXTSZ</strong></td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td><strong>EXPD</strong></td>
<td></td>
<td><strong>EXPD</strong></td>
<td></td>
</tr>
<tr>
<td><strong>RETPD</strong></td>
<td></td>
<td><strong>RETPD</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GOLIMIT</strong></td>
<td>5</td>
<td><strong>GOLIMIT</strong></td>
<td>5</td>
</tr>
<tr>
<td><strong>GGEMPT</strong></td>
<td>NO</td>
<td><strong>GGEMPT</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>GGSCRAT</strong></td>
<td>NO</td>
<td><strong>GGSCRAT</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>DTYPE</strong></td>
<td>REMPLCPY</td>
<td><strong>DTYPE</strong></td>
<td>REMPLCPY</td>
</tr>
<tr>
<td><strong>ACTIVE</strong></td>
<td>NO</td>
<td><strong>ACTIVE</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>IFALLOC</strong></td>
<td>USE</td>
<td><strong>IFALLOC</strong></td>
<td>USE</td>
</tr>
<tr>
<td><strong>ALLOC</strong></td>
<td>N/A</td>
<td><strong>ALLOC</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>SMS</strong></td>
<td>NO</td>
<td><strong>SMS</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>SMUNIT</strong></td>
<td>NO</td>
<td><strong>SMUNIT</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>SIZEPCT</strong></td>
<td>(100,100)</td>
<td><strong>SIZEPCT</strong></td>
<td>(100,100)</td>
</tr>
<tr>
<td><strong>UNIT</strong></td>
<td>(SYSALLDA,SYSALLDA)</td>
<td><strong>UNIT</strong></td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td><strong>UNITCNT</strong></td>
<td>(0,0)</td>
<td><strong>UNITCNT</strong></td>
<td>(0,0)</td>
</tr>
<tr>
<td><strong>VOLCNT</strong></td>
<td>(25,25)</td>
<td><strong>VOLCNT</strong></td>
<td>(25,25)</td>
</tr>
<tr>
<td><strong>AVGOLSP</strong></td>
<td>((30000,TRK),(30000,TRK))</td>
<td><strong>AVGOLSP</strong></td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td><strong>DATACLAS</strong></td>
<td>(NONE,NONE)</td>
<td><strong>DATACLAS</strong></td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td><strong>STORCLAS</strong></td>
<td>(NONE,NONE)</td>
<td><strong>STORCLAS</strong></td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td><strong>THRESHLD</strong></td>
<td>0</td>
<td><strong>THRESHLD</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>MAXEXTSZ</strong></td>
<td>((0,K),(0,K))</td>
<td><strong>MAXEXTSZ</strong></td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td><strong>EXPD</strong></td>
<td></td>
<td><strong>EXPD</strong></td>
<td></td>
</tr>
<tr>
<td><strong>RETPD</strong></td>
<td></td>
<td><strong>RETPD</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GOLIMIT</strong></td>
<td>5</td>
<td><strong>GOLIMIT</strong></td>
<td>5</td>
</tr>
<tr>
<td><strong>GGEMPT</strong></td>
<td>NO</td>
<td><strong>GGEMPT</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>GGSCRAT</strong></td>
<td>NO</td>
<td><strong>GGSCRAT</strong></td>
<td>NO</td>
</tr>
</tbody>
</table>

**UNLOAD** DSNPAT=&UID..BMC.&TSIX.&DDNAME
Example 1: VCAT-defined segmented table space with SYSIDCIN

Chapter 5 Examples of REORG PLUS jobs

BMC50483I WORK DSNPAT=UID..BMC..TSIX..ADNAME
BMC50483I SORTWORK DSNPAT=UID..AUTILPFX..ADNAME
BMC50483I ARCHIVE DSNPAT=UID..AUTILPFX..ADNAME
BMC50483I SYSPUNCH DSNPAT=UID..AUTILPFX..ADNAME
BMC50483I LOCPFCPY DSNPAT=UID..ADNAME..TSIX..T&PART..T&TIME
BMC50483I LOCPICPY DSNPAT=UID..ADNAME..TSIX..T&PART..T&TIME
BMC50483I LOCBFCPY DSNPAT=UID..ADNAME..TSIX..T&PART..T&TIME
BMC50483I LOCBICPY DSNPAT=UID..ADNAME..TSIX..T&PART..T&TIME
BMC50483I LOCPLCPY DSNPAT=UID..ADNAME..TSIX..T&TIME
BMC50483I LOCBLCPY DSNPAT=UID..ADNAME..TSIX..T&TIME
BMC50483I REMPFCPY DSNPAT=UID..ADNAME..TSIX..T&PART..T&TIME
BMC50483I REMPICPY DSNPAT=UID..ADNAME..TSIX..T&PART..T&TIME
BMC50483I REMBFCPY DSNPAT=UID..ADNAME..TSIX..T&PART..T&TIME
BMC50483I REMPLCPY DSNPAT=UID..ADNAME..TSIX..T&TIME
BMC50483I REMBLCPY DSNPAT=UID..ADNAME..TSIX..T&TIME

BMC50028I DB2 MODE = NFM

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION = 1010
BMC50471I SUBSYSTEM DEFAULT = DEFF
BMC50471I CHARACTER SET = ALPHANUM
BMC50471I DATE FORMAT = USA
BMC50471I TIME FORMAT = USA
BMC50471I LOCAL DATE LENGTH = 0
BMC50471I LOCAL TIME LENGTH = 0
BMC50471I DECIMAL POINT = PERIOD
BMC50471I DECIMAL ARITHMETIC = 15
BMC50471I DELIMITER = DEFAULT
BMC50471I SQL DELIMITER = DEFAULT
BMC50471I ENCODING SCHEME = EBCDIC
BMC50471I APPL. ENCODING SCHEME = EBCDIC
BMC50471I MIXED = NO
BMC50471I EBCDIC CCSID = (37,65534,65534
BMC50471I ASCII CCSID = (819,65534,65534
BMC50471I UNICODE CCSID = (367,1208,1200
BMC50471I IMPLICIT TIME ZONE = CURRENT (-05:00)

BMC51265I ESTIMATED CARDINALITY OF SPACE = 9819  AVG SORTWK ROW LENGTH = 173  AVG UNLOAD ROW LENGTH = 173
BMC51264I UNLOAD WILL READ 617 DATA PAGES FROM SPACE 'ARUDB001.TS001'

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMB
BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMB
BMC51301I 1: SAMPLING STATISTICS: PART=0,TP=619,SP=95,SR=1512,AVGR=173,SO=65,SE=0,AVGF=1591,SO=65,SE=59,EP=11
BMC51265I ESTIMATED CARDINALITY OF SPACE = 9819  AVG SORTWK ROW LENGTH = 173  AVG UNLOAD ROW LENGTH = 173
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0011 = 9819  AVG SORTWK ROW LENGTH = 173  AVG UNLOAD ROW LENGTH = 173
BMC51264I UNLOAD WILL READ 617 DATA PAGES FROM SPACE 'ARUDB001.TS001'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMB
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

Figure 15  SYSPRINT for example 1 (part 4 of 7)
Example 1: VCAT-defined segmented table space with SYSIDCIN

Figure 15  SYSPRINT for example 1 (part 5 of 7)
### Figure 15 SYSPRINT for example 1 (part 6 of 7)

<table>
<thead>
<tr>
<th>FIELD</th>
<th>RESULT</th>
<th>FIELD</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIAPFAR</td>
<td>3</td>
<td>PSUPQ_DEL_RIDS</td>
<td>0</td>
</tr>
<tr>
<td>REDGRSPACE_KB</td>
<td>720</td>
<td>PCTFREE</td>
<td>20</td>
</tr>
<tr>
<td>DATAREPEAT</td>
<td>556</td>
<td>FREEPAGE</td>
<td>9</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td>1</td>
<td>SPACE_KB</td>
<td>7200</td>
</tr>
<tr>
<td>EXTENTS</td>
<td>1</td>
<td>NUMDATASETS</td>
<td>1</td>
</tr>
<tr>
<td>COLUMN</td>
<td>EMPNAME</td>
<td>UTILCODE</td>
<td>R</td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>ARUDB001.TS001</td>
<td>LOCATION</td>
<td>DEFF</td>
</tr>
<tr>
<td>ATTRIBUTES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLNO</td>
<td>3</td>
<td>COLTYPE</td>
<td>CHAR</td>
</tr>
<tr>
<td>LENGTH</td>
<td>40</td>
<td>NULLS</td>
<td>Y</td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLCARD</td>
<td>1</td>
<td>COLNULLS</td>
<td>0</td>
</tr>
<tr>
<td>COLAVG</td>
<td>41</td>
<td>COLMIN</td>
<td>41</td>
</tr>
<tr>
<td>KEYCARD</td>
<td></td>
<td>EMPNAME</td>
<td></td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>ARUDB001.TS001</td>
<td>LOCATION</td>
<td>(NULL)</td>
</tr>
<tr>
<td>ATTRIBUTES AND STATISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN</td>
<td>EMPNAME</td>
<td>COLCARD</td>
<td>10000</td>
</tr>
<tr>
<td>COLUMN</td>
<td>EMPNO</td>
<td>COLCARD</td>
<td>4</td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NACTIVE</td>
<td>619</td>
<td>REDGRSPACE</td>
<td>60</td>
</tr>
<tr>
<td>ROWMAXFOUND</td>
<td>179</td>
<td>REDGRSPACE_KB</td>
<td>2880</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td>90</td>
<td>SPACE_KB</td>
<td>4320</td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>ARUDB001.TS001</td>
<td>LOCATION</td>
<td>(NULL)</td>
</tr>
<tr>
<td>HISTORY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAVESTATS</td>
<td>N</td>
<td>UPDATEDDB2</td>
<td>Y</td>
</tr>
<tr>
<td>SAMPLING</td>
<td>N</td>
<td>UTILCODE</td>
<td>R</td>
</tr>
<tr>
<td>ATTRIBUTES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POTY</td>
<td>-1</td>
<td>SQTY</td>
<td>1</td>
</tr>
<tr>
<td>ALLOCUNIT</td>
<td>0</td>
<td>COMPRESS</td>
<td>(BLANK)</td>
</tr>
<tr>
<td>ROW_FORMAT</td>
<td>RRF</td>
<td>BASIC</td>
<td></td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CARD</td>
<td>-10000</td>
<td>NACTIVE</td>
<td>619</td>
</tr>
<tr>
<td>ROMMAXFOUND</td>
<td>179</td>
<td>ROWMAXFOUND</td>
<td>179</td>
</tr>
<tr>
<td>FULL</td>
<td>0</td>
<td>FARDIR</td>
<td>0</td>
</tr>
<tr>
<td>PERCROP</td>
<td>0</td>
<td>REDGRSPACE</td>
<td>60</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td>66</td>
<td>POTYRWS</td>
<td>2073</td>
</tr>
<tr>
<td>IPREFIX</td>
<td>1</td>
<td>SPACE_KB</td>
<td>4320</td>
</tr>
<tr>
<td>EXTENTS</td>
<td>3</td>
<td>DEVTYPE</td>
<td>-3390</td>
</tr>
<tr>
<td>TABLE</td>
<td>ARUDB001.TS001</td>
<td>LOCATION</td>
<td>(NULL)</td>
</tr>
<tr>
<td>HISTORY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAVESTATS</td>
<td>N</td>
<td>UPDATEDDB2</td>
<td>Y</td>
</tr>
<tr>
<td>SAMPLING</td>
<td>N</td>
<td>UTILCODE</td>
<td>R</td>
</tr>
<tr>
<td>ATTRIBUTES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POTY</td>
<td>-1</td>
<td>SQTY</td>
<td>1</td>
</tr>
<tr>
<td>ALLOCUNIT</td>
<td>0</td>
<td>COMPRESS</td>
<td>(BLANK)</td>
</tr>
<tr>
<td>ROW_FORMAT</td>
<td>RRF</td>
<td>BASIC</td>
<td></td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CARD</td>
<td>-10000</td>
<td>NACTIVE</td>
<td>619</td>
</tr>
<tr>
<td>ROMMAXFOUND</td>
<td>179</td>
<td>ROWMAXFOUND</td>
<td>179</td>
</tr>
<tr>
<td>FULL</td>
<td>0</td>
<td>FARDIR</td>
<td>0</td>
</tr>
<tr>
<td>PERCROP</td>
<td>0</td>
<td>REDGRSPACE</td>
<td>60</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td>66</td>
<td>POTYRWS</td>
<td>2073</td>
</tr>
<tr>
<td>IPREFIX</td>
<td>1</td>
<td>SPACE_KB</td>
<td>4320</td>
</tr>
<tr>
<td>EXTENTS</td>
<td>3</td>
<td>DEVTYPE</td>
<td>-3390</td>
</tr>
</tbody>
</table>

---

**Chapter 5 Examples of REORG PLUS jobs** 415
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Both cases in this example reorganize a 16-partition, storage-group-defined table space. The table has three indexes: a clustering index, a unique nonpartitioned secondary index, and a non-unique nonpartitioned secondary index.

Both cases combine unload and reload processing into a single REORG phase. One of the differences between the two cases, however, is the data availability during the job as specified by the SHRLEVEL option:

- Case 1 uses SHRLEVEL NONE (the default). Therefore, the data is unavailable during the job.
- Case 2 specifies SHRLEVEL REFERENCE, which allows the data to be available in read-only (RO) status during the job. REORG PLUS messages BMC50890I and BMC50891I detail the rename and delete activity associated with the staging data sets.

In case 1, dynamic data set allocation by REORG PLUS is inactive, although BMCSORT dynamically allocates the sort work files. With the exception of sort work files, dynamic allocation is inactivated using the DDTYPE \texttt{ddtype} ACTIVE NO command option, overriding the ACTIVE YES installation option values.

Case 2 also illustrates dynamic data set allocation by REORG PLUS for unload, work, and sort work data sets and the full copy data set. The dynamic data set allocation report (messages BMC50445I through BMC50448I) shows the data set allocations.

In both cases, REORG PLUS creates the inline copy data set while reloading the table space. Table 66 on page 417 describes the key command options and DD statements for this job.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.
Table 66  Key command options and DD statements used in example 2 (part 1 of 2)

<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>UPDATE</td>
<td>resets a specific value to a constant</td>
</tr>
<tr>
<td></td>
<td>In this case, the EMPDEPT column of table ARU.TS0071 is set to the constant DP where EMPNO is equal to 011110009.</td>
</tr>
<tr>
<td>COPY YES</td>
<td>creates a DB2 image copy of the table space</td>
</tr>
<tr>
<td></td>
<td>Because the default for the INLINECP installation option is YES, this copy is an inline image copy.</td>
</tr>
<tr>
<td></td>
<td>Because REGISTER is not specified, this statement defaults to REGISTER ALL. REORG PLUS makes one copy of the table space after reorganizing it and registers the copy in SYSIBM.SYSCOPY.</td>
</tr>
<tr>
<td>COPYDDN (FULLCP)</td>
<td>specifies a ddname of FULLCP for the data set that receives a full image copy of the table space that you are reorganizing</td>
</tr>
<tr>
<td></td>
<td>This option overrides the default name of BMCCPY (used in case 1 only).</td>
</tr>
<tr>
<td>COPYLVL FULL</td>
<td>In case 1, COPYLVL FULL overrides the default of COPYLVL PART, allocating a single image copy data set for all partitions being reorganized.</td>
</tr>
<tr>
<td>COPYLVL PART</td>
<td>In case 2, COPYLVL PART allocates an image copy data set for each partition that you are reorganizing.</td>
</tr>
<tr>
<td>BMCSTATS YES</td>
<td>tells REORG PLUS to save the BMC statistics in the DASD MANAGER PLUS database statistics tables</td>
</tr>
<tr>
<td></td>
<td>The JCL for this example also includes the DD statement //ASUSRPRRT DD DUMMY. This DD statement tells REORG PLUS not to produce a report of the statistics.</td>
</tr>
<tr>
<td>SHRLEVEL NONE</td>
<td>In case 1, SHRLEVEL NONE is the default, so is not specified. The objects that you are reorganizing are stopped and unavailable during the job.</td>
</tr>
<tr>
<td>SHRLEVEL REFERENCE</td>
<td>In case 2, SHRLEVEL REFERENCE is specified. The objects that you are reorganizing are available in read-only (RO) status during the job.</td>
</tr>
<tr>
<td>ANALYZE SCAN</td>
<td>calculates optimal data set sizes for dynamic data set allocation (used in case 2 only)</td>
</tr>
<tr>
<td>//SYSREC01</td>
<td>data sets that contain the rows that you are reorganizing</td>
</tr>
<tr>
<td></td>
<td>This example specifies multiple SYSRECnn data sets, one for each partition where nn corresponds to the partition number.</td>
</tr>
<tr>
<td></td>
<td>In case 1, the unload data sets are not required for single-phase reorganization. However, to ensure restartability, they are included in this example.</td>
</tr>
<tr>
<td></td>
<td>In case 2, the unload data sets are not specified in the JCL because dynamic data set allocation is active.</td>
</tr>
</tbody>
</table>
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Table 66  Key command options and DD statements used in example 2 (part 2 of 2)

<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//SYSUT101&lt;br&gt; //SYSUT102</td>
<td>In case 1, two SYSUT1nn data sets are allocated for the two nonpartitioned secondary index. In case 2, no SYSUT1nn data sets are specified in the JCL because dynamic data set allocation is active.</td>
</tr>
<tr>
<td>//FULLCP</td>
<td>ddbname that is used for the data set that receives a full image copy of the table space that you are reorganizing (used in case 1 only)</td>
</tr>
</tbody>
</table>

Figure 16 shows the JCL for case 1 of example 2.

Figure 16  JCL for example 2, case 1 (part 1 of 3)

```
// JOB
// *
/ * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/ * PARTITIONED TABLESPACE, 16 PARTS, 3 INDEXES, STOGROUP DEFINED           *
/ * -CLUSTERING INDEX, UNIQUE                                              *
/ * -SECONDARY INDEX, UNIQUE                                               *
/ * -SECONDARY INDEX, NON-UNIQUE                                          *
/ *                                                                         *
/ * REORG TABLESPACE USING SINGLE PHASE REORG (DEFAULT),                   *
/ * UPDATE OPTION USED.                                                   *
/ * USING MULTIPLE JCL ALLOCATED SYSREC'S.                                *
/ * REALLOCATE SPACE VIA THE REDEFINE YES OPTION (DEFAULT),               *
/ * MAKE IMAGE COPY ON CARTRIDGE FOR ALL PARTS,                           *
/ * UPDATE DASD MANAGER STATS.                                            *
/ * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/ BMCREORG EXEC PGM=ARUUMAIN,                                            *
/     PARM='DEFF,EXMPL02A.NEW,.MSGLEVEL(1),ARU$OPTS'                      *
/ STEPLIB DD DISP=SHR,DSN=productLibraries                                *
/ DD DISP=SHR,DSN=DB2.DSNEXIT                                             *
/ DD DISP=SHR,DSN=DB2.DSNLOAD                                              *
/ *                                                                         *
/ SYSPRINT DD SYSOUT=*                                                   *
/ SYSOUT DD SYSOUT=*                                                     *
/ UTPRINT DD SYSOUT=*                                                    *
/ SYSPDUMP DD SYSOUT=*                                                   *
/ SYSTERM DD SYSOUT=*                                                    *
/ ASUSRPRT DD DUMMY                                                      *
/ *                                                                         *
/ SYSEXEC01 DD DSN=ARU.EXMPL02A.SYSEXEC01,                                 *
/     UNIT=WORK,SPACE=(CYL,(20,10)),                                    *
/     DISP=(MOD,CATLG,CATLG)                                             *
/ SYSEXEC02 DD DSN=ARU.EXMPL02A.SYSEXEC02,                                 *
/     UNIT=WORK,SPACE=(CYL,(20,10)),                                    *
/     DISP=(MOD,CATLG,CATLG)                                             *
```
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 16  JCL for example 2, case 1 (part 2 of 3)

```plaintext
//SYSREC03 DD DSN=ARU.EXMPL02A.SYSREC03,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC04 DD DSN=ARU.EXMPL02A.SYSREC04,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC05 DD DSN=ARU.EXMPL02A.SYSREC05,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC06 DD DSN=ARU.EXMPL02A.SYSREC06,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC07 DD DSN=ARU.EXMPL02A.SYSREC07,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC08 DD DSN=ARU.EXMPL02A.SYSREC08,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC09 DD DSN=ARU.EXMPL02A.SYSREC09,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC10 DD DSN=ARU.EXMPL02A.SYSREC10,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC11 DD DSN=ARU.EXMPL02A.SYSREC11,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC12 DD DSN=ARU.EXMPL02A.SYSREC12,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC13 DD DSN=ARU.EXMPL02A.SYSREC13,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC14 DD DSN=ARU.EXMPL02A.SYSREC14,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC15 DD DSN=ARU.EXMPL02A.SYSREC15,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
//SYSREC16 DD DSN=ARU.EXMPL02A.SYSREC16,
   // UNIT=WORK,SPACE=(CYL,(20,10)),
   // DISP=(MOD,CATLG,CATLG)
/*
//SYSUT101 DD DSN=ARU.EXMPL02A.SYSUT101,
   // UNIT=WORK,SPACE=(CYL,(20,20)),
   // DISP=(MOD,CATLG,CATLG)
//SYSUT102 DD DSN=ARU.EXMPL02A.SYSUT102,
   // UNIT=WORK,SPACE=(CYL,(20,20)),
   // DISP=(MOD,CATLG,CATLG)
/*
//FULLCP DD DSN=ARU.EXMPL02A.FULLCPY,
```
Figure 16  JCL for example 2, case 1 (part 3 of 3)

//          UNIT=CART,
//          DISP=(,CATLG)
// *
//SYSIN     DD  *
REORG TABLESPACE ARUDB007.TS007
UPDATE ARU.TS0071 SET EMPDEPT = 'DP' WHERE EMPNO = '011110009'
COPY YES
COPYLVL FULL
COPYDDN(FULLCP)
BMCSTATS YES
DDTYPE UNLOAD ACTIVE NO
DDTYPE WORK ACTIVE NO
DDTYPE LOCPFCPY ACTIVE NO
/* ----

Figure 17 shows the JCL for case 2 of example 2.

Figure 17  JCL for example 2 (part 1 of 2)

//          JOB
// *
//** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//** PARTITIONED TABLESPACE, 16 PARTS, 3 INDEXS, STOGROUP DEFINED  *
//** -CLUSTERING INDEX, UNIQUE *
//** -SECONDARY INDEX, UNIQUE *
//** -SECONDARY INDEX, NON-UNIQUE *
//** *
//** REORG TABLESPACE USING SINGLE PHASE REORG (DEFAULT), *
//** USING SHRLEVEL REFERENCE OPTION, *
//** USING DYNAMIC WORKFILE ALLOCATIONS FOR WORK AND COPY, *
//** DELETE WORK FILES VIA THE DELETEFILES YES OPTION (DEFAULT), *
//** REALLOCATE SPACE VIA THE REDEFINE YES OPTION (DEFAULT), *
//** MAKE IMAGE COPY ON DASD OF ALL PARTS, *
//** UPDATE DASD MANAGER STATS. *
//** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN, *
//         PARM='DEFF,EXMPL02B,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB   DD  DISP=SHR,DSN=ProductLibraries
//          DD  DISP=SHR,DSN=DB2.DSNEXIT
//          DD  DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT DD  SYSOUT=* 
//SYSOUT  DD  SYSOUT=* 
//UTPRINT  DD  SYSOUT=* 
//SYSUDUMP  DD  SYSOUT=* 
//SYSTYPE  DD  SYSOUT=* 
//ASUSRPRTR  DD  DUMMY 
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB007.TS007
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 18 shows the SYSPRINT output for case 1 of example 2.

Figure 17  JCL for example 2 (part 2 of 2)
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 18  SYSPRINT for example 2, case 1 (part 2 of 8)
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

<table>
<thead>
<tr>
<th>BMS504701</th>
<th>GDEMPRTY</th>
<th>NO</th>
<th>NO</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMS504701</td>
<td>GDESCTR</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS504701</td>
<td>D&amp;type</td>
<td>REMPICPY</td>
<td>REMBCPY</td>
<td>REMBCPY</td>
</tr>
<tr>
<td>BMS504701</td>
<td>ACTV</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS504701</td>
<td>IFALOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMS504701</td>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS504701</td>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS504701</td>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS504701</td>
<td>SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMS504701</td>
<td>UNIT</td>
<td>(S/YSALDA,S/YSALDA)</td>
<td>(S/YSALDA,S/YSALDA)</td>
<td>(S/YSALDA,S/YSALDA)</td>
</tr>
<tr>
<td>BMS504701</td>
<td>UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMS504701</td>
<td>AVGOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMS504701</td>
<td>DATATYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS504701</td>
<td>DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS504701</td>
<td>MGMTCLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS504701</td>
<td>STORCLASS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS504701</td>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMS504701</td>
<td>MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>BMS504701</td>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMS504701</td>
<td>RETPD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMS504701</td>
<td>GDGLIMIT</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMS504701</td>
<td>GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS504701</td>
<td>GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

| BMS504701 | D&type    | SYSUPNCH | | |
| BMS504701 | ACTV      | YES | | |
| BMS504701 | IFALOC    | USE | | |
| BMS504701 | ALLOC     | N/A | | |
| BMS504701 | SMS       | NO | | |
| BMS504701 | SMSUNIT   | NO | | |
| BMS504701 | SIZEPCT   | (100,100) | | |
| BMS504701 | UNIT      | (S/YSALDA,S/YSALDA) | | |
| BMS504701 | UNITCNT   | (0,0) | | |
| BMS504701 | VOLCNT    | (25,25) | | |
| BMS504701 | AVGOLSP   | ((30000,TRK),(30000,TRK)) | | |
| BMS504701 | DATATYPE  | (NONE,NONE) | | |
| BMS504701 | DATACLAS  | (NONE,NONE) | | |
| BMS504701 | MGMTCLASS | (NONE,NONE) | | |
| BMS504701 | STORCLASS | (NONE,NONE) | | |
| BMS504701 | THRESHLD  | 0 | | |
| BMS504701 | MAXEXTSZ  | ((0,K),(0,K)) | | |
| BMS504701 | EXPDT     | | | |
| BMS504701 | RETPD     | | | |
| BMS504701 | GDGLIMIT  | 0 | | |
| BMS504701 | GDGEMPTY  | NO | | |
| BMS504701 | GDGSCRAT  | NO | | |

| BMS504701 | D&type    | LOCPLCPY | LOCBLCPY | |
| BMS504701 | ACTV      | NO | NO | |
| BMS504701 | IFALOC    | USE | USE | |
| BMS504701 | ALLOC     | N/A | N/A | |
| BMS504701 | SMS       | NO | NO | |
| BMS504701 | SMSUNIT   | NO | NO | |
| BMS504701 | SIZEPCT   | (100,100) | (100,100) | |
| BMS504701 | UNIT      | (S/YSALDA,S/YSALDA) | (S/YSALDA,S/YSALDA) | |
| BMS504701 | UNITCNT   | (0,0) | (0,0) | |
| BMS504701 | VOLCNT    | (25,25) | (25,25) | |
| BMS504701 | AVGOLSP   | ((30000,TRK),(30000,TRK)) | ((30000,TRK),(30000,TRK)) | |
| BMS504701 | DATATYPE  | (NONE,NONE) | (NONE,NONE) | |
| BMS504701 | DATACLAS  | (NONE,NONE) | (NONE,NONE) | |
| BMS504701 | MGMTCLASS | (NONE,NONE) | (NONE,NONE) | |
| BMS504701 | STORCLASS | (NONE,NONE) | (NONE,NONE) | |
| BMS504701 | THRESHLD  | 0 | 0 | |
| BMS504701 | MAXEXTSZ  | ((0,K),(0,K)) | ((0,K),(0,K)) | |
| BMS504701 | EXPDT     | | | |
| BMS504701 | RETPD     | | | |
| BMS504701 | GDGLIMIT  | 5 | 5 | 5 |
| BMS504701 | GDGEMPTY  | NO | NO | NO |
| BMS504701 | GDGSCRAT  | NO | NO | NO |
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 18  SYSPRINT for example 2, case 1 (part 4 of 8)

<table>
<thead>
<tr>
<th>BMC504701</th>
<th>SMSUNIT = NO</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I</td>
<td>SIZEPECT = (100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT = (SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNITCNT = (0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT = (25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AVGVOLSP = ((30000, TRK), (30000, TRK))</td>
<td>((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DSNTYPE = (NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS = (NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MGMTCLAS = (NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>STORCLAS = (NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>THRESHLD = 0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ = ((0, K), (0, K))</td>
<td>((0, K), (0, K))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNLOAD DSNPAT=&amp;UID..BMC.&amp;TSIX.&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>WORK DSNPAT=&amp;UID..BMC.&amp;TSIX.&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SORTWORK DSNPAT=&amp;UID..BTILPPF.XXNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>ARCHIVE DSNPAT=&amp;UID..BTILPPF.XXNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SYSPRINT DSNPAT=&amp;UID..BTILPPF.XXNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCFCPY DSNPAT=&amp;UID..&amp;DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCICPY DSNPAT=&amp;UID..&amp;DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LCLFCPY DSNPAT=&amp;UID..&amp;DDNAME..TSIX..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LCLICPY DSNPAT=&amp;UID..&amp;DDNAME..TSIX..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMFCPY DSNPAT=&amp;UID..&amp;DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMPICPY DSNPAT=&amp;UID..&amp;DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMBFCPY DSNPAT=&amp;UID..&amp;DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMBLCPY DSNPAT=&amp;UID..&amp;DDNAME..TSIX..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>VERSION = 1010</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>SUBSYSTEM DEFAULT = DEFF</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>CHARACTER SET = ALPHANUM</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DATE FORMAT = USA</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>TIME FORMAT = USA</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LOCAL DATE LENGTH = 0</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LOCAL TIME LENGTH = 0</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DECIMAL POINT = PERIOD</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DECIMAL ARITHMETIC = 15</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DELIMITER = DEFAULT</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>SQL DELIMITER = DEFAULT</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ENCODING SCHEME = EBCDIC</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>APPL. ENCODING SCHEME = EBCDIC</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>MIXED = NO</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>EBCDIC CCSID = (37,65534,65534)</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ASCII CCSID = (819,65534,65534)</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>UNICODE CCSID = (367,1208,1200)</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>IMPLICIT TIME ZONE = CURRENT (-05:00)</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DASD MANAGER PLUS TABLES:</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DB2 DSNHDECP MODULE SETTINGS:</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>BMCUTIL='BMCUTIL.CMN_BMCUTIL'</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>BMCSYNC='BMCUTIL.CMN_BMCSYNC'</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>BMCHIST='BMCUTIL.CMN_BMCIST'</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>BMCDICT='BMCUTIL.CMN_BMCDCIT'</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>BMCMCOPY='BMCUTIL.CMN_BMCXCOPY'</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DASD MANAGER PLUS TABLES:</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>BMC50493I</td>
<td>&quot;ATS102.RS_COLUMNSTATS&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50493I</td>
<td>&quot;ATS102.RS_COLUMNS&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50493I</td>
<td>&quot;ATS102.RS_INDEXES&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50493I</td>
<td>&quot;ATS102.RS_INDEXPART&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50493I</td>
<td>&quot;ATS102.RS_TABLEPART&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50493I</td>
<td>&quot;ATS102.RS_TABLESPACE&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50493I</td>
<td>&quot;ATS102.RS_TSPART_DIST&quot;</td>
<td></td>
</tr>
</tbody>
</table>
Figure 18  SYSPRINT for example 2, case 1 (part 5 of 8)

Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Chapter 5  Examples of REORG PLUS jobs  425
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 18  SYSPRINT for example 2, case 1 (part 6 of 8)
Figure 18  SYSPRINT for example 2, case 1 (part 7 of 8)

Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Chapter 5  Examples of REORG PLUS jobs  427
Figure 18  SYSPRINT for example 2, case 1 (part 8 of 8)

Figure 19 shows the SYSPRINT output for case 2 of example 2.
Figure 19  SYSPRINT for example 2, case 2 (part 2 of 14)

<table>
<thead>
<tr>
<th>BMC50471I</th>
<th>ALTFAIL=RCVRPEND</th>
<th>ICDON=(BCMCY,BCMCZ)</th>
<th>RORMAX=3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471I</td>
<td>ALNMAX=10000</td>
<td>ICTYPE=AUTO</td>
<td>ROUTCOME=(11,1)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ARC=NO</td>
<td>IDAACHE=10000</td>
<td>SCPPMAX=8</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ARCHDVP=SYSARC</td>
<td>INDFRM=10</td>
<td>SQMN=YES</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ARCHFORM=BMX</td>
<td>INLNECP=YES</td>
<td>SHTGTHMEM=CONTINUE</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>AVAILPAGECPET=100</td>
<td>IXDRES=NO</td>
<td>SIXSNAP=NO</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>BELOMAX=3000</td>
<td>IXRANDOM=NO</td>
<td>SMAX=0</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>BMCHST=YES</td>
<td>KEEPPOCITORY=NO</td>
<td>SMORE=(0K,0K)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>CBUFFS=30</td>
<td>LEAFSILM=200</td>
<td>SORTDST=(,SYSALLDA)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>CONDEXC=NO</td>
<td>LOG=YES</td>
<td>SORTNUM=32</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>COPYDOM=(BMCCPY,BMCCPZ)</td>
<td>LOCKNOW=YES</td>
<td>SPIFDSM=YES</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>COPYL=PART</td>
<td>LOGFNL=NONE</td>
<td>SPISSCIS=NONE</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>COPYMAX=10000</td>
<td>LOGMEM=0</td>
<td>SPILNOT=WORK</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>COPYSUBET=NO</td>
<td>LOGSPIL=(20000,10000)</td>
<td>SPILDELAY=3</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>CYPFAIL=TERM</td>
<td>LOGTHSK=0</td>
<td>SOLRETRY=100</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DATACAP=NO</td>
<td>LONGLOG=CONTINUE</td>
<td>STAGEDON=BMX</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DOLLOM=30</td>
<td>LONGLAUGR=CONTINUE</td>
<td>STOPMEM=NO</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DEADLINE=NONE</td>
<td>MAPTEXTI=(NONE,REXX)</td>
<td>STOPDELAY=1</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DELAY=1200</td>
<td>MAXNEWPARTS=2</td>
<td>STOPCITY=300</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DELFILES=YES</td>
<td>MAXLOG=500</td>
<td>TAPEDISP=DELETE</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSCD=(3,7)</td>
<td>MAXSORT=1000</td>
<td>TASKMEM=1000</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSCNTPY=ALL</td>
<td>MAXTAPE=3</td>
<td>TIERБ=E (NONE,REXX)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DRNDelay=1</td>
<td>MGEXTENT=CONTINUE</td>
<td>TIMOUT=TERM</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DRNRTRY=255</td>
<td>MINSORTMEM=0</td>
<td>TOTALPAGEPCT=0</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DRNMA=NONE</td>
<td>MSGLEVEL=1</td>
<td>TSSAMPLEPCT=100</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSLN=NONE,ASM</td>
<td>OFFPFSL=0</td>
<td>UBUTF=20</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSLUTILE=YES</td>
<td>OPNBGID=NO</td>
<td>UNLDD=SYSREC</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSPLOK=RNFAIL</td>
<td>ORIGDISP=DELETE</td>
<td>UNLDMAX=3000</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSRSXET=(NONE,REXX)</td>
<td>PENORD=NO</td>
<td>UNLOAD=PLOAD</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>EXCLDUMP=(X37,X22,X06)</td>
<td>PFSF=NO</td>
<td>UTILITY=COLLCSID=UTILITY</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FASTSWITCH=NO</td>
<td>RVCDSTATE=(BMCR Y,BMCRZ)</td>
<td>UTILITY=NUXILI=UTILITY</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSNUTILB=YES</td>
<td>RCYDON=(BMCRY,BMCRZ)</td>
<td>USTKATE=SUP</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSNUEXIT=(NONE,ASM)</td>
<td>REDF=NO</td>
<td>WBUFFS=(20,10)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSP=NONE</td>
<td>REMM=30</td>
<td>WORKDOM=SYSUTI</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSF=NONE</td>
<td>RIDMMESSSZ=2097152</td>
<td>WORKUNIT=SYSALLDA</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>TSHRESH=E</td>
<td>RMDMA=0</td>
<td>XBMID=0</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>TSHRESH=0</td>
<td>RXMAPMEM=0</td>
<td>ZIP=ENABLED</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>HASH=E</td>
<td>RXMAPMEM=0</td>
<td>ZIP=ENABLED</td>
</tr>
</tbody>
</table>

Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Chapter 5  Examples of REORG PLUS jobs 429
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

### Figure 19: SYSPRINT for example 2, case 2 (part 3 of 14)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
</tr>
<tr>
<td>DSTYPE</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>DATAclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MAXEXTsz</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
</tr>
<tr>
<td>DSTYPE</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>DATAclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MAXEXTsz</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
</tr>
<tr>
<td>DSTYPE</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>DATAclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MAXEXTsz</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
</tr>
<tr>
<td>DSTYPE</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>DATAclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MAXEXTsz</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
</tr>
<tr>
<td>DSTYPE</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>DATAclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MAXEXTsz</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
<td>(30000, TRK), (30000, TRK)</td>
</tr>
<tr>
<td>DSTYPE</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>DATAclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MAXEXTsz</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

#### Chapter 5 Examples of REORG PLUS jobs

**Figure 19** SYSPRINT for example 2, case 2 (part 4 of 14)

<table>
<thead>
<tr>
<th>BMC504701</th>
<th>AWGOLSP</th>
<th>((30000,TRK),(30000,TRK))</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC504701</td>
<td>DATACLS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>MGMTCLS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>STORCLS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>THRESHLD</td>
<td>0</td>
</tr>
<tr>
<td>BMC504701</td>
<td>MAXETSZ</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>BMC504701</td>
<td>EXPDT</td>
<td></td>
</tr>
<tr>
<td>BMC504701</td>
<td>RETPD</td>
<td></td>
</tr>
<tr>
<td>BMC504701</td>
<td>GDGLIMIT</td>
<td>0</td>
</tr>
<tr>
<td>BMC504701</td>
<td>GDGEMPTY</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>GGDSRCT</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>ODTYPE</td>
<td>= LOCPCLPY</td>
</tr>
<tr>
<td>BMC504701</td>
<td>ACTIVE</td>
<td>= NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>IFALLOC</td>
<td>= USE</td>
</tr>
<tr>
<td>BMC504701</td>
<td>ALLOC</td>
<td>= N/A</td>
</tr>
<tr>
<td>BMC504701</td>
<td>SMS</td>
<td>= NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>SMSUNIT</td>
<td>= NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>SIZEPCT</td>
<td>= (100,100)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>UNIT</td>
<td>= (SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>UNITCNT</td>
<td>= (0,0)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>VOLCNT</td>
<td>= (25,25)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>AVGWOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC504701</td>
<td>DSNTYPE</td>
<td>= (NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>DATACLS</td>
<td>= (NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>MGMTCLS</td>
<td>= (NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>STORCLS</td>
<td>= (NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>THRESHLD</td>
<td>0</td>
</tr>
<tr>
<td>BMC504701</td>
<td>MAXETSZ</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>BMC504701</td>
<td>EXPDT</td>
<td></td>
</tr>
<tr>
<td>BMC504701</td>
<td>RETPD</td>
<td></td>
</tr>
<tr>
<td>BMC504701</td>
<td>GDGLIMIT</td>
<td>= 5</td>
</tr>
<tr>
<td>BMC504701</td>
<td>GDGEMPTY</td>
<td>= NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>GGDSRCT</td>
<td>= NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>ODTYPE</td>
<td>= REMPLCPY</td>
</tr>
<tr>
<td>BMC504701</td>
<td>ACTIVE</td>
<td>= NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>IFALLOC</td>
<td>= USE</td>
</tr>
<tr>
<td>BMC504701</td>
<td>ALLOC</td>
<td>= N/A</td>
</tr>
<tr>
<td>BMC504701</td>
<td>SMS</td>
<td>= NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>SMSUNIT</td>
<td>= NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>SIZEPCT</td>
<td>= (100,100)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>UNIT</td>
<td>= (SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>UNITCNT</td>
<td>= (0,0)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>VOLCNT</td>
<td>= (25,25)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>AVGWOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC504701</td>
<td>DSNTYPE</td>
<td>= (NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>DATACLS</td>
<td>= (NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>MGMTCLS</td>
<td>= (NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>STORCLS</td>
<td>= (NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701</td>
<td>THRESHLD</td>
<td>0</td>
</tr>
<tr>
<td>BMC504701</td>
<td>MAXETSZ</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>BMC504701</td>
<td>EXPDT</td>
<td></td>
</tr>
<tr>
<td>BMC504701</td>
<td>RETPD</td>
<td></td>
</tr>
<tr>
<td>BMC504701</td>
<td>GDGLIMIT</td>
<td>= 5</td>
</tr>
<tr>
<td>BMC504701</td>
<td>GDGEMPTY</td>
<td>= NO</td>
</tr>
<tr>
<td>BMC504701</td>
<td>GGDSRCT</td>
<td>= NO</td>
</tr>
<tr>
<td>BMC504831</td>
<td>UNLOAD</td>
<td>DSNPAT=&amp;UID..BMC.&amp;TSIX..ADDNAME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>WORK</td>
<td>DSNPAT=&amp;UID..BMC.&amp;TSIX..ADDNAME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>SORTWORK</td>
<td>DSNPAT=&amp;UID,AUTILPFX,ADDNAME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>ARCHIVE</td>
<td>DSNPAT=&amp;UID,AUTILPFX,ADDNAME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>SYSPUNCH</td>
<td>DSNPAT=&amp;UID,AUTILPFX,ADDNAME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>LOCPFCPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>LOCPICPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>LOCBFCPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>LOCBICPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>LOCPLCPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;TIME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>LOCBLCPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;TIME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>REMPFCPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>REMPICPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>REMBFCPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>REMPLCPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;TIME</td>
</tr>
<tr>
<td>BMC504831</td>
<td>REMBLCPY</td>
<td>DSNPAT=&amp;UID,ADDNAME,ATSIX..&amp;TIME</td>
</tr>
</tbody>
</table>
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 19  SYSPRINT for example 2, case 2 (part 5 of 14)

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION  = 1010
BMC50471I SUBSYSTEM DEFAULT = DEF
BMC50471I CHARACTER SET = ALPHANUM
BMC50471I DATE FORMAT  = USA
BMC50471I TIME FORMAT = USA
BMC50471I LOCAL DATE LENGTH = 0
BMC50471I LOCAL TIME LENGTH = 0
BMC50471I DECIMAL POINT = PERIOD
BMC50471I DECIMAL ARITHMETIC = 15
BMC50471I DELIMITER = DEFAULT
BMC50471I SQL DELIMITER = DEFAULT
BMC50471I ENCODING SCHEME = EBCDIC
BMC50471I APPL. ENCODING SCHEME = EBCDIC
BMC50471I MIXED = NO
BMC50471I EBCDIC CCSID = (37,65534,65534)
BMC50471I ASCII CCSID = (819,65534,65534)
BMC50471I UNICODE CCSID = (367,1208,1200)
BMC50471I IMPLICIT TIME ZONE = CURRENT (-05:00)
BMC50471I DB2 MODE = NFM
BMC50471I BMCUTIL = 'BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC = 'BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST = 'BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT = 'BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXPYP = 'BMCUTIL.CMN_BMCXPYP'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...'ATS102.RS_COLDIST'
BMC50471I ...'ATS102.RS_COLSTATS'
BMC50471I ...'ATS102.RS_COLUMNS'
BMC50471I ...'ATS102.RS_INDEXES'
BMC50471I ...'ATS102.RS_INDEXPART'
BMC50471I ...'ATS102.RS_IXPART_DIST'
BMC50471I ...'ATS102.RS_STOGROUP'
BMC50471I ...'ATS102.RS_TABLEPART'
BMC50471I ...'ATS102.RS_TABLESPACE'
BMC50471I ...'ATS102.RS_TSPART_DIST'

BMC50102I REORG TABLESPACE ARUDB007.TS007
BMC50102I SHRLEVEL REFERENCE
BMC50102I COPY YES
BMC50102I CPGVL PART
BMC50102I ANALYZE SCAN
BMC50102I BMCSSTATS YES
BMC50102I DTYPE WORK
BMC50102I DSNPAT 'ARU.&UTILPFX.&DDNAME..A&PART5'
BMC50102I DTYPE LOCPFCPY
BMC50102I DSNPAT 'ARU.&UTILPFX.&DDNAME.(+1)'

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:00
BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XMB
BMC51301I 15: SAMPLING STATISTICS: PART=3,TP=1,SP=10,AVG=100,SD=64,SE=0,AVGF=100,SD=64,SE=0,EP=0
BMC51301I 16: SAMPLING STATISTICS: PART=4,TP=1,SP=10,AVG=100,SD=64,SE=0,AVGF=100,SD=64,SE=0,EP=0
BMC51266I CARDINALITY OF SPACE = 2044  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I CARDINALITY OF PART 0001 = 1610  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I CARDINALITY OF PART 0002 = 10  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I CARDINALITY OF PART 0003 = 159  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I CARDINALITY OF PART 0004 = 112  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I CARDINALITY OF PART 0005 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0006 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0007 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0008 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0009 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0010 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0011 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0012 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0013 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0014 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0015 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0016 = 404  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 19  SYSPRINT for example 2, case 2 (part 6 of 14)

Table 5-15

| BDMSQ446I DNAME DNAME | UNITS OR KBYTES KBYTES ALOC ALOC |
|----------------------|-------------------|-------------------|
| BMCCPY01 ARU.ARUCPY01.ARUCPY01.G001V00 | SYSSALDA | 376 | 37 | 0 | Trk |
| BMCCPY02 ARU.ARUCPY02.ARUCPY02.G001V00 | SYSSALDA | 1488 | 148 | 31 | 4 Trk |
| BMCCPY03 ARU.ARUCPY03.ARUCPY03.G001V00 | SYSSALDA | 1488 | 148 | 31 | 4 Trk |
| BMCCPY04 ARU.ARUCPY04.ARUCPY04.G001V00 | SYSSALDA | 8 | 0 | 1 | Trk |
| BMCCPY05 ARU.ARUCPY05.ARUCPY05.G001V00 | SYSSALDA | 8 | 0 | 1 | Trk |
| BMCCPY06 ARU.ARUCPY06.ARUCPY06.G001V00 | SYSSALDA | 8 | 0 | 0 | Trk |
| BMCCPY07 ARU.ARUCPY07.ARUCPY07.G001V00 | SYSSALDA | 8 | 0 | 1 | Trk |
| BMCCPY08 ARU.ARUCPY08.ARUCPY08.G001V00 | SYSSALDA | 8 | 0 | 1 | Trk |
| BMCCPY09 ARU.ARUCPY09.ARUCPY09.G001V00 | SYSSALDA | 8 | 0 | 1 | Trk |
| BMCCPY10 ARU.ARUCPY10.ARUCPY10.G001V00 | SYSSALDA | 8 | 0 | 1 | Trk |
| BMCCPY11 ARU.ARUCPY11.ARUCPY11.G001V00 | SYSSALDA | 8 | 0 | 1 | Trk |
| BMCCPY12 ARU.ARUCPY12.ARUCPY12.G001V00 | SYSSALDA | 8 | 0 | 1 | Trk |
| BMCCPY13 ARU.ARUCPY13.ARUCPY13.G001V00 | SYSSALDA | 1488 | 148 | 31 | 4 Trk |
| BMCCPY14 ARU.ARUCPY14.ARUCPY14.G001V00 | SYSSALDA | 1488 | 148 | 31 | 4 Trk |
| BMCCPY15 ARU.ARUCPY15.ARUCPY15.G001V00 | SYSSALDA | 1488 | 148 | 31 | 4 Trk |
| BMCCPY16 ARU.ARUCPY16.ARUCPY16.G001V00 | SYSSALDA | 1488 | 148 | 31 | 4 Trk |

BMC50877I POINT OF CONSISTENCY ESTABLISHED AT RBA/LRSN = 0000000000C43621DCB7

Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Chapter 5 Examples of REORG PLUS jobs 433
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 19  SYSPRINT for example 2, case 2 (part 7 of 14)
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 19  SYSPRINT for example 2, case 2 (part 8 of 14)

Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Chapter 5 Examples of REORG PLUS jobs 435
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

![Figure 19 SYSPRINT for example 2, case 2 (part 9 of 14)](image)
Figure 19 SYSPRINT for example 2, case 2 (part 10 of 14)

Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Message Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A008' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A008'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A011' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A011'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A014' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A014'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A010' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A010'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A006' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A006'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A009' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A009'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A003' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A003'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A002' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A002'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A001' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A001'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A013' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A013'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A016' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A016'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A010' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A010'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A007' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A007'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A012' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A012'...</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A011' IS RENAMED</td>
</tr>
<tr>
<td>BMC50891 I</td>
<td>ATTEMPTING TO RENAME DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.10001.A011'...</td>
</tr>
</tbody>
</table>

Figure 19 SYSPRINT for example 2, case 2 (part 11 of 14)
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 19  SYSPRINT for example 2, case 2 (part 12 of 14)

Chapter 5  Examples of REORG PLUS jobs 439
Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Figure 19  SYSPRINT for example 2, case 2 (part 13 of 14)
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

This example illustrates a partial reorganization of a 16-partition storage-group-defined table space. The table has three indexes: the clustering index, a unique nonpartitioned secondary index (NPSI), and a non-unique nonpartitioned secondary index.

In case 1, REORG PLUS reorganizes partitions 1 through 5, 8, and 16. In case 2, only partitions 1 through 5 are reorganized. This case illustrates creating a single copy of a subset of partitions. The following attributes of this job enable this copy:

- A contiguous subset of partitions is specified with the PART option.
- COPYLVL FULL is specified.
- A different options module, JG9$OPTS, is specified on the EXEC statement. This options module changes the default value for the COPYSUBSET installation option to YES.

Another difference between the two cases is the data availability during the job as specified by the SHRLEVEL option:

- Case 1 uses the default, SHRLEVEL NONE. Therefore, the data is unavailable during the job.
- Case 2 specifies SHRLEVEL REFERENCE, which allows the data to be available in read-only status during the job.

REORG PLUS messages BMC50890I and BMC50891I detail the rename and delete activity associated with the staging data sets.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.
Table 67 describes the key command options for this job.

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>PART 1:5, 8, 16 PART 1:5</td>
<td>reorganizes partitions 1 through 5, 8, and 16 (case 1)</td>
</tr>
<tr>
<td></td>
<td>reorganizes contiguous partitions 1-5 to enable a single copy of a subset of partitions (case 2)</td>
</tr>
<tr>
<td></td>
<td>For case 2, COPYSUBSET=YES is also set in the installation options for these copies.</td>
</tr>
<tr>
<td>COPYLVL FULL</td>
<td>requests a full image copy for a partitioned table space</td>
</tr>
<tr>
<td></td>
<td>This option is required to create a copy of the subset of partitions in case 2.</td>
</tr>
<tr>
<td>COPY YES</td>
<td>creates DB2 image copies of table space partitions 1 through 5, 8, and 16 (case 1)</td>
</tr>
<tr>
<td></td>
<td>creates a single image copy of table space partitions 1 through 5 (case 2)</td>
</tr>
<tr>
<td></td>
<td>Because REGISTER is not specified, this statement defaults to REGISTER ALL. REORG PLUS makes four copies of the table space after reorganizing it registering all four copies in SYSIBM.SYSCOPY.</td>
</tr>
<tr>
<td>REDEFINE NO REDEFINE YES (default)</td>
<td>In case 1, REDEFINE NO tells REORG PLUS not to delete and redefine the VSAM data sets containing the DB2 objects. Instead, REORG PLUS will reset the high-used RBA (HURBA).</td>
</tr>
<tr>
<td></td>
<td>In case 2, the default of REDEFINE YES tells REORG PLUS to delete and redefine the VSAM data sets containing the DB2 objects.</td>
</tr>
<tr>
<td>BMCSTATS YES</td>
<td>tells REORG PLUS to save the BMC statistics in the DASD MANAGER PLUS database statistics tables</td>
</tr>
<tr>
<td></td>
<td>The JCL for this example also includes a DD statement for the ASUSRPRT data set. This DD statement tells REORG PLUS not to print the statistics from the Common Statistics component to SYSPRINT, but to send them to a separate data set. Figure 24 on page 460 shows a portion of this ASUSRPRT data set for case 2.</td>
</tr>
<tr>
<td>SHRLEVEL NONE SHRLEVEL REFERENCE</td>
<td>In case 1, the default SHRLEVEL NONE is used. The objects that you are reorganizing are stopped and unavailable during the job.</td>
</tr>
<tr>
<td></td>
<td>In case 2, SHRLEVEL REFERENCE is specified. The objects that you are reorganizing are available in read-only status during the job.</td>
</tr>
<tr>
<td>DDTYPE LOCBFCPY ACTIVE YES DDTYPE REMPFCPY ACTIVE YES DDTYPE REMBFCPY ACTIVE YES</td>
<td>activates dynamic allocation for the local backup copy and the remote copies, overriding the installation defaults</td>
</tr>
</tbody>
</table>
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Figure 20 shows the JCL for case 1 of example 3.

**Figure 20  JCL for example 3, case 1**

```
// JOB
/**
** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
** PARTITIONED TABLESPACE, 16 PARTS, 3 INDEXES, STOGROUP DEFINED *
** -CLUSTERING INDEX, UNIQUE *
** -SECONDARY INDEX, UNIQUE *
** -SECONDARY INDEX, NONUNIQUE *
** *
** REORG PART 1-5, 8 AND 16, *
** ONE SYSREC PER PART, *
** DO NOT REDEFINE SPACE *
** MAKE 4 COPIES OF EACH PART REORGED. *
** * *
** BMCREORG EXEC PGM=ARUUMAIN,
** PARM='DEFF.EXMPL03A.NEW,,MSGLEVEL(1),ARU$OPTS'
** STEPLIB DD DISP=SHR,DSN=ProductLibraries
** DD DISP=SHR,DSN=DB2.DSNEXIT
** DD DISP=SHR,DSN=DB2.DSNLOAD
** *
** SYSPRINT DD SYSOUT=* 
** SYSOUT DD SYSOUT=* 
** UTPRINT DD SYSOUT=* 
** SYSUDUMP DD SYSOUT=* 
** SYSTERM DD SYSOUT=* 
** ASUSRPRT DD SYSOUT=* 
** *
** SYSSIN DD *
REORG TABLESPACE ARUDB007.TS007 PART 1:5, 8, 16
COPY YES
REDEFINE NO
BMCSTATS YES
DDTYPE UNLOAD
  DSNPAT 'ARU.EXMPL03A.&DDNAME'
DDTYPE WORK
  DSNPAT 'ARU.EXMPL03A.&DDNAME'
DDTYPE LOCPFCPY
  DSNPAT 'ARU.EXMPL03A.&DDNAME'
DDTYPE LOCBFCPY ACTIVE YES
  DSNPAT 'ARU.EXMPL03A.&DDNAME'
DDTYPE REMPFCPY ACTIVE YES
  DSNPAT 'ARU.EXMPL03A.&DDNAME'
DDTYPE REMBFCPY ACTIVE YES
  DSNPAT 'ARU.EXMPL03A.&DDNAME'
*/
```
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Figure 21 shows the JCL for case 2 of example 3.

Figure 21  JCL for example 3, case 2

```
// JOB
/*
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/* PARTITIONED TABLESPACE, 16 PARTS, 3 INDEXES, STOGROUP-DEFINED *
/* -CLUSTERING INDEX, UNIQUE *
/* -SECONDARY INDEX, UNIQUE *
/* -SECONDARY INDEX, NONUNIQUE *
/* *
/* REORG PARTS 1-5, SHRLEVEL REFERENCE *
/* ONE SYSREC PER PART. *
/* REDEFINE SPACE (USING REDEND YES OPTION), *
/* MAKE 4 COPIES OF EACH PART REORGED (LOCAL AND REMOTE COPIES), *
/* ONE IMAGE COPY DATASET FOR ALL PARTS (COPYSUBSET=YES). *
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
/* PARM='DEFF,EXMPL03B,NW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB DD DISP=SHR,DSN=ProductLibraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
/*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=* 
//UTPRINT DD SYSOUT=* 
//SYSDUMP DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSTEM DD SYSOUT=* 
//ASUSRPRDD DD SYSOUT=* 
/*
//SYSIN DD *
REORG TABLESPACE ARUDB007.TS007 PART 1:5
SHRLEVEL REFERENCE
COPY YES
COPYLVL FULL
BMCSTATS YES
DDTYPE WORK
    DSNPAT 'ARU.EXMPL03B.&DDNAME'
DDTYPE LOCPFCPY
    DSNPAT 'ARU.EXMPL03B.&DDNAME'
DDTYPE LOCBFCPY ACTIVE YES
    DSNPAT 'ARU.EXMPL03B.&DDNAME'
DDTYPE REMPFCPY ACTIVE YES
    DSNPAT 'ARU.EXMPL03B.&DDNAME'
DDTYPE REMBFCPY ACTIVE YES
    DSNPAT 'ARU.EXMPL03B.&DDNAME'
/*
```
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Figure 22 shows the SYSPRINT output for case 1 of example 3.

### Figure 22 SYSPRINT for example 3, case 1 (part 1 of 8)

<table>
<thead>
<tr>
<th>BMC50471I</th>
<th>REDR</th>
<th>DB2 UTILITIES COMMON CODE : V11.01.00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I NO MAINTENANCE TO REPORT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I EXTENDED BUFFER MANAGER -- V06.01.00</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I BMC STATS API -- V11.01.00</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I BMCSORT ENGINE -- V02.04.01</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I BMC50471I ALTRFAIL = RCVRPEND</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I ANALMA = 1000</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I ARC = NO</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I ARCHDDN = SYSARC</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I ARCH = NO</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I ANALMA = 1000</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I ARCHDDN = SYSARC</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I ARCH = NO</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I ANALMA = 1000</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I ARCHDDN = SYSARC</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>BMC50471I ARCH = NO</strong></td>
</tr>
</tbody>
</table>

---

Chapter 5 Examples of REORG PLUS jobs 445
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Figure 22  SYSPRINT for example 3, case 1 (part 2 of 8)

<table>
<thead>
<tr>
<th>BMS0470I</th>
<th>DTYPE</th>
<th>ARCHIVE</th>
<th>LDPFPCPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMS0470I</td>
<td>ACTIVE</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>IFALOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>UNIT</td>
<td>SYSALLDA,SYSALLDA</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>UNITCNT</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>VOLCNT</td>
<td>25,25</td>
<td>25,25</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>VOLCPSK</td>
<td>(30000,TRK),(30000,TRK)</td>
<td>(30000,TRK),(30000,TRK)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>DSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>DATAclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>MGMTclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>STORclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>EXPDT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>RETPD</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>GDDLIMIT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>GDDEMPTY</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>GDDSCAT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>IFALOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>UNIT</td>
<td>SYSALLDA,SYSALLDA</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>UNITCNT</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>VOLCNT</td>
<td>25,25</td>
<td>25,25</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>VOLCPSK</td>
<td>(30000,TRK),(30000,TRK)</td>
<td>(30000,TRK),(30000,TRK)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>DSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>DATAclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>MGMTclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>STORclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>EXPDT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>RETPD</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>GDDLIMIT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>GDDEMPTY</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>GDDSCAT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>IFALOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>UNIT</td>
<td>SYSALLDA,SYSALLDA</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>UNITCNT</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>VOLCNT</td>
<td>25,25</td>
<td>25,25</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>VOLCPSK</td>
<td>(30000,TRK),(30000,TRK)</td>
<td>(30000,TRK),(30000,TRK)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>DSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>DATAclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>MGMTclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>STORclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>EXPDT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>RETPD</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>GDDLIMIT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>GDDEMPTY</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>GDDSCAT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>IFALOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>UNIT</td>
<td>SYSALLDA,SYSALLDA</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>UNITCNT</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>VOLCNT</td>
<td>25,25</td>
<td>25,25</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>VOLCPSK</td>
<td>(30000,TRK),(30000,TRK)</td>
<td>(30000,TRK),(30000,TRK)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>DSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>DATAclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>MGMTclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>STORclas</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMS0470I</td>
<td>MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
</tbody>
</table>

REORG PLUS for DB2 Reference Manual

446
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

**Figure 22** SYSPRINT for example 3, case 1 (part 3 of 8)

<table>
<thead>
<tr>
<th>BMC504701 MAXEXTSZ</th>
<th>(0,K),(0,K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC504701 EXPDT</td>
<td>-</td>
</tr>
<tr>
<td>BMC504701 RETPD</td>
<td>-</td>
</tr>
<tr>
<td>BMC504701 GDGLIMIT</td>
<td>5</td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 GDGSCRAT</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 DTYPE</td>
<td>SYSPUNCH</td>
</tr>
<tr>
<td>BMC504701 ACTIVE</td>
<td>YES</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 ALLOC</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 SMS</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMSUNIT</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SIZEPCT</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSALLDA,SYSCALLDA)</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 VOLCNT</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC504701 AWGVOLSZ</td>
<td>(30000,TRK),(30000,TRK)</td>
</tr>
<tr>
<td>BMC504701 DATACLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 MGMTCLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 STORCLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 THRESHLD</td>
<td>0</td>
</tr>
<tr>
<td>BMC504701 MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMC504701 EXPDT</td>
<td>-</td>
</tr>
<tr>
<td>BMC504701 RETPD</td>
<td>-</td>
</tr>
<tr>
<td>BMC504701 GDGLIMIT</td>
<td>5</td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 GDGSCRAT</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 DTYPE</td>
<td>REPLCPY</td>
</tr>
<tr>
<td>BMC504701 ACTIVE</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 ALLOC</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 SMS</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMSUNIT</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SIZEPCT</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 VOLCNT</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC504701 AWGVOLSZ</td>
<td>(30000,TRK),(30000,TRK)</td>
</tr>
<tr>
<td>BMC504701 DATACLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 MGMTCLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 STORCLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 THRESHLD</td>
<td>0</td>
</tr>
<tr>
<td>BMC504701 MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMC504701 EXPDT</td>
<td>-</td>
</tr>
<tr>
<td>BMC504701 RETPD</td>
<td>-</td>
</tr>
<tr>
<td>BMC504701 GDGLIMIT</td>
<td>5</td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 GDGSCRAT</td>
<td>NO</td>
</tr>
</tbody>
</table>

BMC504831 UNLOAD DSNPAT=&UID..BMC.&TSIX.&DDNAME
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

**Figure 22** SYSPRINT for example 3, case 1 (part 4 of 8)

<table>
<thead>
<tr>
<th></th>
<th>DSNPAT='UID..BM..ATSIX..DDNAME'</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50483I WORK</td>
<td></td>
</tr>
<tr>
<td>BMC50483I SORTWORK</td>
<td>DSNPAT='UID..UTILPFX..DDNAME'</td>
</tr>
<tr>
<td>BMC50483I ARCHIVE</td>
<td>DSNPAT='UID..UTILPFX..DDNAME'</td>
</tr>
<tr>
<td>BMC50483I SYSPUNCH</td>
<td>DSNPAT='UID..UTILPFX..DDNAME'</td>
</tr>
<tr>
<td>BMC50483I LCPCPCPY</td>
<td>DSNPAT='UID..DDNAME..ATSIX..F&amp;PART..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I LCPCIPCY</td>
<td>DSNPAT='UID..DDNAME..ATSIX..F&amp;PART..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I LCPCBCPY</td>
<td>DSNPAT='UID..DDNAME..ATSIX..F&amp;PART..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I LCPLIPCY</td>
<td>DSNPAT='UID..DDNAME..ATSIX..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I LCBLIPCY</td>
<td>DSNPAT='UID..DDNAME..ATSIX..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I REMPFCPY</td>
<td>DSNPAT='UID..DDNAME..ATSIX..F&amp;PART..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I REMPICPY</td>
<td>DSNPAT='UID..DDNAME..ATSIX..F&amp;PART..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I REMBCFCPY</td>
<td>DSNPAT='UID..DDNAME..ATSIX..F&amp;PART..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I REMMCPHY</td>
<td>DSNPAT='UID..DDNAME..ATSIX..F&amp;PART..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I REMBLCPY</td>
<td>DSNPAT='UID..DDNAME..ATSIX..T&amp;TIME'</td>
</tr>
</tbody>
</table>

BMC50471I DB2 DSNHDECP MODULE SETTINGS:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERSION</td>
<td>1010</td>
</tr>
<tr>
<td>SUBSYSTEM DEFAULT</td>
<td>DEFF</td>
</tr>
<tr>
<td>CHARACTER SET</td>
<td>ALPHANUM</td>
</tr>
<tr>
<td>DATE FORMAT</td>
<td>USA</td>
</tr>
<tr>
<td>TIME FORMAT</td>
<td>USA</td>
</tr>
<tr>
<td>LOCAL DATE LENGTH</td>
<td>0</td>
</tr>
<tr>
<td>LOCAL TIME LENGTH</td>
<td>0</td>
</tr>
<tr>
<td>DECIMAL POINT</td>
<td>15</td>
</tr>
<tr>
<td>DECIMAL ARITHMETIC</td>
<td>15</td>
</tr>
<tr>
<td>DELIMITER</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>SQL DELIMITER</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>ENCODING SCHEME</td>
<td>EBCDIC</td>
</tr>
<tr>
<td>APPL. ENCODING SCHEME</td>
<td>EBCDIC</td>
</tr>
<tr>
<td>MIXED</td>
<td>NO</td>
</tr>
<tr>
<td>EBCDIC CCSID</td>
<td>(37,65534,65534)</td>
</tr>
<tr>
<td>ASCII CCSID</td>
<td>(819,65534,65534)</td>
</tr>
<tr>
<td>UNICODE CCSID</td>
<td>(367,1208,1200)</td>
</tr>
<tr>
<td>IMPLICIT TIME ZONE</td>
<td>CURRENT (-05:00)</td>
</tr>
</tbody>
</table>

BMC50028I DB2 MODE = NFM

BMC50471I BMCUTIL = "BMCUTIL.CMN_BMCUTIL"
| BMC50471I BMCSYNC | "BMCUTIL.CMN_BMCSYNC" |
| BMC50471I BMCHIST | "BMCUTIL.CMN_BMCCHIST" |
| BMC50471I BMCDICT | "BMCUTIL.CMN_BMCDICT" |
| BMC50471I BMCCOPY | "BMCUTIL.CMN_BMCCOPY" |

BMC50471I DASD MANAGER PLUS TABLES:

...'ATS102.RS Coldist'
...'ATS102.RS Coldist Stats'
...'ATS102.RS Columns'
...'ATS102.RS Indexes'
...'ATS102.RS Indexpart'
...'ATS102.RS IX Part Dist'
...'ATS102.RS Stogroup'
...'ATS102.RS Tablepart'
...'ATS102.RS Tables'
...'ATS102.RS Tablespace'
...'ATS102.RS Tp Part Dist'

BMC50102I REORG TABLESPACE ARUDB007.T5007 PART 1:5, 8, 16

BMC50102I COPY YES
| BMC50102I REDUCE NO  |
| BMC50102I BMCSAS YES |
| BMC50102I DTYPE UNLOAD |
| BMC50102I DSNPAT 'ARU.EXMPL03A..DDNAME' |
| BMC50102I DSNPAT 'ARU.EXMPL03A..DDNAME' |
| BMC50102I DSNPAT 'ARU.EXMPL03A..DDNAME' |
| BMC50102I DSNPAT 'ARU.EXMPL03A..DDNAME' |
| BMC50102I DSNPAT 'ARU.EXMPL03A..DDNAME' |
| BMC50102I DSNPAT 'ARU.EXMPL03A..DDNAME' |
| BMC50102I DSNPAT 'ARU.EXMPL03A..DDNAME' |

BMC50004I UTILINIT PHASE COMPLETE. ELAPSED TIME = 00:00:00
Figure 22: SYSPRINT for example 3, case 1 (part 5 of 8)

Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Figure 22  SYSPRINT for example 3, case 1 (part 6 of 8)

| BMC51271I UNLOAD STATISTICS: 10 ROWS KEYS UNLOADED FROM PARTITION 3 |
| BMC51281I UNLOAD STATISTICS: X'0000000000C43659ABD0' IS THE HIGHEST LOGRBA FOR PARTITION 3 |
| BMC50481I 3: SORT COMPLETE. ELAPSED TIME = 00:00:00 |
| BMC50482I 4: BUILD COMPLETE. ELAPSED TIME = 00:00:00 DSN = 'DEFFCAT.DSNDBD.ARUDB007.TS00711.I0001.A016' |
| BMC50482I 4: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE |
| BMC50477I 4: PARTITION = 16, ROWS/KEYS = 10, I/O WAITS = 1, DDNAME = SYS00054 |
| BMC50375I INLINE COPY STARTED FOR DATASET = 'DEFFCAT.DSNDBD.ARUDB007.TS00711.I0001.A004' |
| BMC50481I 3: RELOAD COMPLETE. ELAPSED TIME = 00:00:00 DSN = 'DEFFCAT.DSNDBD.ARUDB007.TS00711.I0001.A002' |
| BMC51271I UNLOAD STATISTICS: 10 ROWS KEYS UNLOADED FROM PARTITION 2 |
| BMC51281I UNLOAD STATISTICS: X'0000000000C43659ABD0' IS THE HIGHEST LOGRBA FOR PARTITION 2 |
| BMC50481I 2: SORT COMPLETE. ELAPSED TIME = 00:00:00 |
| BMC50482I 2: BUILD COMPLETE. ELAPSED TIME = 00:00:00 DSN = 'DEFFCAT.DSNDBD.ARUDB007.TS00711.I0001.A003' |
| BMC50482I 2: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE |
| BMC50477I 2: PARTITION = 8, ROWS/KEYS = 10, I/O WAITS = 1, DDNAME = SYS00053 |
| BMC50481I 1: REORG TASK COMPLETE. ELAPSED TIME = 00:00:01 |
| BMC50477I 1: SORT COMPLETE. ELAPSED TIME = 00:00:00 |
| BMC50477I 1: PARTITION = 1, ROWS/KEYS = 1610, I/O WAITS = 1, DDNAME = SYS00050 |
| BMC51271I UNLOAD STATISTICS: 10 ROWS KEYS UNLOADED FROM PARTITION 1 |
| BMC50482I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE |
| BMC50477I 1: PARTITION = 1, ROWS/KEYS = 1610, I/O WAITS = 1, DDNAME = SYS00057 |
| BMC51281I UNLOAD STATISTICS: X'0000000000C4364F8A5D' IS THE HIGHEST LOGRBA FOR PARTITION 1 |
| BMC50481I 1: SORT COMPLETE. ELAPSED TIME = 00:00:00 |
| BMC50482I 1: BUILD COMPLETE. ELAPSED TIME = 00:00:00 DSN = 'DEFFCAT.DSNDBD.ARUDB007.TS00711.I0001.A001' |
| BMC50482I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE |
| BMC50477I 1: PARTITION = 1, ROWS/KEYS = 1610, I/O WAITS = 1, DDNAME = SYS00050 |
| BMC51281I UNLOAD STATISTICS: X'0000000000C4364F51A9' IS THE HIGHEST LOGRBA FOR PARTITION 5 |

450  **REORG PLUS for DB2 Reference Manual**
Figure 22  SYSPRINT for example 3, case 1 (part 7 of 8)

Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Chapter 5  Examples of REORG PLUS jobs  451
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Figure 22  SYSPRINT for example 3, case 1 (part 8 of 8)

Figure 23  SYSPRINT for example 3, case 2 (part 1 of 9)

Figure 23 shows the SYSPRINT output for case 2 of example 3.
### Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

#### Figure 23  SYSPRINT for example 3, case 2 (part 2 of 9)

<table>
<thead>
<tr>
<th>BMC50471I</th>
<th>DESCCOL=(3,7)</th>
<th>MAXSORTMEMORY=0</th>
<th>TASKMAX=10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471I</td>
<td>DRAINTPY=ALL</td>
<td>MAXTAPE=3</td>
<td>TERMEXIT=(NONE,REXX)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DRAINCLAY=1</td>
<td>MGEXTENT=CONTINUE</td>
<td>TIMEOUT=TERM</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DRAINTRY=255</td>
<td>MINSORTMEMORY=0</td>
<td>TOTALPAGEPCT=0</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DRAINAIT=NONE</td>
<td>MSGLEVEL=1</td>
<td>TSSAMPLEPCT=100</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSNUEXIT=(NONE,ASM)</td>
<td>OFFPOSILM=10</td>
<td>UBUFFS=20</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSNUUTILYES=OPOW82ID=YES</td>
<td>UPLDOND=SYSREC</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSNEXIT=(NONE,REXX)</td>
<td>ORIGDSIP=DELETE</td>
<td>UNLXMA=300</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSNEXIT=(NONE,REXX)</td>
<td>PENDSL=YES</td>
<td>UNLXLOAD=RELOAD</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>EXITNUM=(X37,X22,X06)</td>
<td>PREFORMAT=NO</td>
<td>UTILB_COLCCS=ID=UTILB</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FASTSWITCH=NO</td>
<td>RCVIDCNUM=(BMCIRY,BMCIRZ)</td>
<td>UTLI_B_NULLIX=UTILB</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FILECN=WXRN</td>
<td>RCVFIDCNUM=(BMCIRY,BMCIRZ)</td>
<td>UXSTAT=SUPE</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FORCE_AT=(START,3)</td>
<td>REDCREASE=YES</td>
<td>WBUFFS=(20,10)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FORCE_RPT=NO</td>
<td>REXECMAK=30</td>
<td>WORKDDN=SYSUT1</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FORCE=NONE</td>
<td>RINDOSSZ=2097152</td>
<td>WORKUNIT=SYSSALDA</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FSALLBACK=NO</td>
<td>RIMAXD=1</td>
<td>XBMID=</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FSTHRESHOLD=0</td>
<td>RMAPLEN=0</td>
<td>ZIIP=ENABLED</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>PLAN=ARUQA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| BMC50470I | DOTYPE=UNLOAD | WORK=WORK | SORTWORK= | |
| BMC50470I | ACTIVE=YES | YES | NO | |
| BMC50470I | IFALLOC=USE | USE | USE | |
| BMC50470I | ALLOC=N/A | N/A | ANY | |
| BMC50470I | SMS=NO | NO | NO | |
| BMC50470I | SMSUNIT=NO | NO | NO | |
| BMC50470I | SIZEPCT=(100,100) | (100,100) | (100,100) | |
| BMC50470I | UNIT=(SYSSALDA,SYSSALDA) | (SYSSALDA,SYSSALDA) | (SYSSALDA,SYSSALDA) | |
| BMC50470I | UNITCNT=(0,0) | N/A | N/A | |
| BMC50470I | VOLCNT=(25,25) | (25,25) | N/A | |
| BMC50470I | AVGOLSP=((30000,TRK),(30000,TRK)) | ((30000,TRK),(30000,TRK)) | N/A | |
| BMC50470I | DTSTYPE=(NONE,NONE) | (NONE,NONE) | (NONE,NONE) | |
| BMC50470I | DATACLASS=(NONE,NONE) | (NONE,NONE) | (NONE,NONE) | |
| BMC50470I | MTMCLASS=(NONE,NONE) | (NONE,NONE) | (NONE,NONE) | |
| BMC50470I | STORAGECLASS=(NONE,NONE) | (NONE,NONE) | (NONE,NONE) | |
| BMC50470I | THRESHOLD=0 | 0 | 0 | |
| BMC50470I | MAKESETZ=(0,K),(0,K) | (0,K),(0,K) | N/A | |
| BMC50470I | EXPOT=N/A | N/A | N/A | |
| BMC50470I | RETPD=N/A | N/A | N/A | |
| BMC50470I | GDGLIMIT=N/A | N/A | N/A | |
| BMC50470I | GDGEMPTY=N/A | N/A | N/A | |
| BMC50470I | GDGSCAST=N/A | N/A | N/A | |

| BMC50470I | DOTYPE=ARCHIVE | LOCPFTP | LOCPFTP | |
| BMC50470I | ACTIVE=NO | YES | YES | |
| BMC50470I | IFALLOC=USE | USE | USE | |
| BMC50470I | ALLOC=N/A | N/A | N/A | |
| BMC50470I | SMS=NO | NO | NO | |
| BMC50470I | SMSUNIT=NO | NO | NO | |
| BMC50470I | SIZEPCT=(100,100) | (100,100) | (5,100) | |
| BMC50470I | UNIT=(SYSSALDA,SYSSALDA) | (SYSSALDA,SYSSALDA) | (SYSSALDA,SYSSALDA) | |
| BMC50470I | UNITCNT=(0,0) | (0,0) | (0,0) | |
| BMC50470I | AVGOLSP=((30000,TRK),(30000,TRK)) | ((30000,TRK),(30000,TRK)) | ((30000,TRK),(30000,TRK)) | |
| BMC50470I | DTSTYPE=(NONE,NONE) | (NONE,NONE) | (NONE,NONE) | |
| BMC50470I | DATACLASS=(NONE,NONE) | (NONE,NONE) | (NONE,NONE) | |
| BMC50470I | MTMCLASS=(NONE,NONE) | (NONE,NONE) | (NONE,NONE) | |
| BMC50470I | STORAGECLASS=(NONE,NONE) | (NONE,NONE) | (NONE,NONE) | |
| BMC50470I | THRESHOLD=0 | 0 | 0 | |
| BMC50470I | MAKESTZ=(0,K),(0,K) | (0,K),(0,K) | (0,K),(0,K) | |
| BMC50470I | EXPOT= |  |  | |
| BMC50470I | RETPD= |  |  | |
| BMC50470I | GDGLIMIT=5 | 5 | 5 | |
| BMC50470I | GDGEMPTY=NO | NO | NO | |
| BMC50470I | GDGSCAST=NO | NO | NO | |

<p>| BMC50470I | DOTYPE=LOGBCPY | LOGBCPY | LOGBCPY | |
| BMC50470I | ACTIVE=NO | NO | NO | |
| BMC50470I | IFALLOC=USE | USE | USE | |
| BMC50470I | ALLOC=N/A | N/A | N/A | |
| BMC50470I | SMS=NO | NO | NO | |
| BMC50470I | SMUNIT=NO | NO | NO | |
| BMC50470I | SIZEPCT=(100,100) | (100,100) | (100,100) | |
| BMC50470I | UNIT=(SYSSALDA,SYSSALDA) | (SYSSALDA,SYSSALDA) | (SYSSALDA,SYSSALDA) | |
| BMC50470I | UNITCNT=(0,0) | (0,0) | (0,0) | |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVGVOLSP</td>
<td>(((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
</tr>
<tr>
<td>DSNTYPE</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>DATAclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>MGMTclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>STORclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>MAXEXTsz</td>
<td>(0, K), (0, K)</td>
<td>(0, K), (0, K)</td>
<td>(0, K), (0, K)</td>
</tr>
<tr>
<td>THRESHld</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>DDTYPE</td>
<td>REMPICAL</td>
<td>REMBFCPY</td>
<td>REMBFCPY</td>
</tr>
<tr>
<td>VID</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>SIZEPCT</td>
<td>(100, 100)</td>
<td>(100, 100)</td>
<td>(100, 100)</td>
</tr>
<tr>
<td>UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>(((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
</tr>
<tr>
<td>DATAclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>MGMTclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>STORclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>MAXEXTsz</td>
<td>(0, K), (0, K)</td>
<td>(0, K), (0, K)</td>
<td>(0, K), (0, K)</td>
</tr>
<tr>
<td>THRESHld</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>DDTYPE</td>
<td>SYSPUNCH</td>
<td>SYSPUNCH</td>
<td>SYSPUNCH</td>
</tr>
<tr>
<td>VID</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>SIZEPCT</td>
<td>(100, 100)</td>
<td>(100, 100)</td>
<td>(100, 100)</td>
</tr>
<tr>
<td>UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>(((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
</tr>
<tr>
<td>DATAclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>MGMTclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>STORclas</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>MAXEXTsz</td>
<td>(0, K), (0, K)</td>
<td>(0, K), (0, K)</td>
<td>(0, K), (0, K)</td>
</tr>
<tr>
<td>THRESHld</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Figure 23  SYSPRINT for example 3, case 2 (part 3 of 9)
### Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

#### Figure 23  SYSPRINT for example 3, case 2 (part 4 of 9)

<table>
<thead>
<tr>
<th>BMC50470I EXPDT</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I RETPD</td>
<td>-</td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>NO</td>
</tr>
<tr>
<td>BMCS0470I GDGSCRAT</td>
<td>NO</td>
</tr>
<tr>
<td>BMCS0470I DDTYPE</td>
<td>REMPLCPY</td>
</tr>
<tr>
<td>BMCS0470I ACTIVE</td>
<td>NO</td>
</tr>
<tr>
<td>BMCS0470I IFALLOC</td>
<td>USE</td>
</tr>
<tr>
<td>BMCS0470I ALLOC</td>
<td>N/A</td>
</tr>
<tr>
<td>BMCS0470I SMS</td>
<td>NO</td>
</tr>
<tr>
<td>BMCS0470I SMSUNIT</td>
<td>NO</td>
</tr>
<tr>
<td>BMCS0470I SIZEPCT</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMCS0470I UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMCS0470I UNITCNT</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMCS0470I VOLCNT</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMCS0470I AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMCS0470I DSNTYPE</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMCS0470I DATACLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMCS0470I MGMTCLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMCS0470I STORCLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMCS0470I THRESHLD</td>
<td>0</td>
</tr>
<tr>
<td>BMCS0470I MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>BMCS0470I UD</td>
<td>DSNPATS=UID..BMC.&amp;TSIX..&amp;DDNAME</td>
</tr>
<tr>
<td>BMCS0470I WORK</td>
<td>DSNPATS=UID..BMC.&amp;TSIX..&amp;DDNAME</td>
</tr>
<tr>
<td>BMCS0470I SORTWORK</td>
<td>DSNPATS=UID..UTILPFX..&amp;DDNAME</td>
</tr>
<tr>
<td>BMCS0470I ARCHIVE</td>
<td>DSNPATS=UID..UTILPFX..&amp;DDNAME</td>
</tr>
<tr>
<td>BMCS0470I SYSPUNCH</td>
<td>DSNPATS=UID..UTILPFX..&amp;DDNAME</td>
</tr>
<tr>
<td>BMCS0470I LOCPCPY</td>
<td>DSNPATS=UID..&amp;DDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMCS0470I LORPCPY</td>
<td>DSNPATS=UID..&amp;DDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMCS0470I REMPCPY</td>
<td>DSNPATS=UID..&amp;DDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMCS0470I REMLCPY</td>
<td>DSNPATS=UID..&amp;DDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMCS0470I OB2 DSNHDECP MODULE SETTINGS:</td>
<td></td>
</tr>
<tr>
<td>BMCS0471I VERSION</td>
<td>1010</td>
</tr>
<tr>
<td>BMCS0471I SUBSYSTEM DEFAULT</td>
<td>DEFF</td>
</tr>
<tr>
<td>BMCS0471I CHARACTER SET</td>
<td>ALPHANUM</td>
</tr>
<tr>
<td>BMCS0471I DATE FORMAT</td>
<td>USA</td>
</tr>
<tr>
<td>BMCS0471I LOCAL DATE LENGTH</td>
<td>0</td>
</tr>
<tr>
<td>BMCS0471I LOCAL TIME LENGTH</td>
<td>0</td>
</tr>
<tr>
<td>BMCS0471I DECIMAL POINT</td>
<td>PERIOD</td>
</tr>
<tr>
<td>BMCS0471I DECIMAL ARITHMETIC</td>
<td>15</td>
</tr>
<tr>
<td>BMCS0471I QELIMIT</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>BMCS0471I ASCIICSID</td>
<td>(37,65534,65534)</td>
</tr>
<tr>
<td>BMCS0471I EBCII CSSID</td>
<td>(819,65534,65534)</td>
</tr>
<tr>
<td>BMCS0471I ENCODING SCHEME</td>
<td>EBCDIC</td>
</tr>
<tr>
<td>BMCS0471I APPL. ENCODING SCHEME</td>
<td>EBCDIC</td>
</tr>
<tr>
<td>BMCS0471I DASD MANAGER PLUS TABLES:</td>
<td></td>
</tr>
<tr>
<td>BMCS0471I DASD MANAGER PLUS TABLES:</td>
<td></td>
</tr>
<tr>
<td>BMCS0471I DASD MANAGER PLUS TABLES:</td>
<td></td>
</tr>
</tbody>
</table>
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Figure 23 SYSPRINT for example 3, case 2 (part 5 of 9)

```
BMC50471I "ATS102.RS_COLSTATS"
BMC50471I "ATS102.RS_COLUMNS"
BMC50471I "ATS102.RS_INDEXES"
BMC50471I "ATS102.RS_INDEXDAPRT"
BMC50471I "ATS102.RS_EXPART_DIST"
BMC50471I "ATS102.RS_STOGROUP"
BMC50471I "ATS102.RS_TABLEPART"
BMC50471I "ATS102.RS_TABLES"
BMC50471I "ATS102.RS_TABLESPACE"
BMC50471I "ATS102.RS_EXPART_DIST"

BMC50102I REORG TABLESPACE ARUDBO07.TS007 PART 1:5
BMC50102I SHRLEVEL REFERENCE
BMC50102I COPY YES
BMC50102I COPYLVL FULL
BMC50102I BENCHMARK YES
BMC50102I DOTYPE WORK
BMC50102I DSNPAT 'ARU.EXMPL03B.&DDNAME'
BMC50102I DOTYPE LOCPCPY
BMC50102I DSNPAT 'ARU.EXMPL03B.&DDNAME'
BMC50102I DOTYPE LOCDPCPY ACTIVE YES
BMC50102I DSNPAT 'ARU.EXMPL03B.&DDNAME'
BMC50102I DOTYPE REMPCPY ACTIVE YES
BMC50102I DSNPAT 'ARU.EXMPL03B.&DDNAME'
BMC50102I DOTYPE REMDPCPY ACTIVE YES
BMC50102I DSNPAT 'ARU.EXMPL03B.&DDNAME'

BMC50138I 'COPYLVL FULL' CHANGED TO 'COPYLVL PART' DUE TO PARTIAL REDRGR AND COPYSUBSET NOT TURNED ON
BMC50044I UTILINIT PHASE COMPLETE. ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP ENABLED (0) USING XBM SUBSYSTEM XBM
BMC50130I 5: SAMPLING STATISTICS: PART=1, TP=64, SR=62, SR=1610, AVG=101, SD=65, SE=0, AVG=2596, SD=65, SE=110, EP=6
BMC50130I 4: SAMPLING STATISTICS: PART=3, TP=1, SR=10, AVG=133, SD=64, SE=0, AVG=1000, SD=64, SE=0, EP=0
BMC50130I 3: SAMPLING STATISTICS: PART=3, TP=3, SR=10, AVG=159, SD=64, SE=0, AVG=1000, SD=64, SE=0, EP=0
BMC50130I 5: SAMPLING STATISTICS: PART=4, TP=3, SR=10, AVG=112, SD=64, SE=0, AVG=1000, SD=64, SE=0, EP=0
BMC50130I 5: ESTIMATED CARDINALITY OF SPACE = 1640 AVG SORTWK ROW LENGTH = 102 AVG UNLOAD ROW LENGTH = 102
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0001 = 1610 AVG SORTWK ROW LENGTH = 101 AVG UNLOAD ROW LENGTH = 101
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0002 = 10 AVG SORTWK ROW LENGTH = 133 AVG UNLOAD ROW LENGTH = 133
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0003 = 10 AVG SORTWK ROW LENGTH = 159 AVG UNLOAD ROW LENGTH = 159
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0004 = 159 AVG UNLOAD ROW LENGTH = 112
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0005 = 0 AVG SORTWK ROW LENGTH = 0 AVG UNLOAD ROW LENGTH = 0
BMC50130I 5: ESTIMATED CARDINALITY OF SPACE = 1640 AVG SORTWK ROW LENGTH = 102 AVG UNLOAD ROW LENGTH = 102
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0001 = 1610 AVG SORTWK ROW LENGTH = 101 AVG UNLOAD ROW LENGTH = 101
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0002 = 10 AVG SORTWK ROW LENGTH = 133 AVG UNLOAD ROW LENGTH = 133
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0003 = 10 AVG SORTWK ROW LENGTH = 159 AVG UNLOAD ROW LENGTH = 159
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0004 = 159 AVG UNLOAD ROW LENGTH = 112
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0005 = 0 AVG SORTWK ROW LENGTH = 0 AVG UNLOAD ROW LENGTH = 0
BMC50130I 5: ESTIMATED CARDINALITY OF SPACE = 1640 AVG SORTWK ROW LENGTH = 102 AVG UNLOAD ROW LENGTH = 102
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0001 = 1610 AVG SORTWK ROW LENGTH = 101 AVG UNLOAD ROW LENGTH = 101
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0002 = 10 AVG SORTWK ROW LENGTH = 133 AVG UNLOAD ROW LENGTH = 133
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0003 = 10 AVG SORTWK ROW LENGTH = 159 AVG UNLOAD ROW LENGTH = 159
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0004 = 159 AVG UNLOAD ROW LENGTH = 112
BMC50130I 5: ESTIMATED CARDINALITY OF PART 0005 = 0 AVG SORTWK ROW LENGTH = 0 AVG UNLOAD ROW LENGTH = 0

BMC50877I POINT OF CONSISTENCY ESTABLISHED AT RBA/LRNS = 0000000000C36805C61
BMC50446I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50447I DNAME DSNAME UNIT OR DATACLAS MGMTCLAS STORCLAS KBYTES KBYTES PRI SEC ALOC PRI ALOC
BMC50448I BMCCPY01 ARU.EXMPL03B.BMCCPY01 SYSAIDA 256 25 6 1 TRK
BMC50448I BMCCPY02 ARU.EXMPL03B.BMCCPY02 SYSAIDA 12 1 1 1 TRK
BMC50448I BMCCPY03 ARU.EXMPL03B.BMCCPY03 SYSAIDA 12 1 1 1 TRK
BMC50448I BMCCPY04 ARU.EXMPL03B.BMCCPY04 SYSAIDA 12 1 1 1 TRK
BMC50448I BMCCPY05 ARU.EXMPL03B.BMCCPY05 SYSAIDA 8 0 1 0 TRK
BMC50448I BMCCPY01 ARU.EXMPL03B.BMCCPY01 SYSAIDA 256 25 6 1 TRK
BMC50448I BMCCPY02 ARU.EXMPL03B.BMCCPY02 SYSAIDA 12 1 1 1 TRK
BMC50448I BMCCPY03 ARU.EXMPL03B.BMCCPY03 SYSAIDA 12 1 1 1 TRK
BMC50448I BMCCPY04 ARU.EXMPL03B.BMCCPY04 SYSAIDA 12 1 1 1 TRK
BMC50448I BMCCPY05 ARU.EXMPL03B.BMCCPY05 SYSAIDA 8 0 1 0 TRK
BMC50448I BMCCPY01 ARU.EXMPL03B.BMCCPY01 SYSAIDA 256 25 6 1 TRK
BMC50448I BMCCPY02 ARU.EXMPL03B.BMCCPY02 SYSAIDA 12 1 1 1 TRK
BMC50448I BMCCPY03 ARU.EXMPL03B.BMCCPY03 SYSAIDA 12 1 1 1 TRK
BMC50448I BMCCPY04 ARU.EXMPL03B.BMCCPY04 SYSAIDA 12 1 1 1 TRK
BMC50448I BMCCPY05 ARU.EXMPL03B.BMCCPY05 SYSAIDA 8 0 1 0 TRK
456 REORG PLUS for DB2 Reference Manual

```
### Figure 23  SYSPRINT for example 3, case 2 (part 6 of 9)

<table>
<thead>
<tr>
<th>BMC50448I</th>
<th>BRCRC02S ARU.EXEMPL03B.BMCRC02S</th>
<th>SYSSIDLA 8 0 1 0 TRK</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50448I</td>
<td>SYSSUT01 ARU.EXEMPL03B.SYSSUT01</td>
<td>SYSSIDLA 45 5 1 1 TRK</td>
</tr>
<tr>
<td>BMC50448I</td>
<td>SYSSUT02 ARU.EXEMPL03B.SYSSUT02</td>
<td>SYSSIDLA 34 4 1 1 TRK</td>
</tr>
</tbody>
</table>

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50509I STATISTICS COLLECTION FOR INDEX SPACE ARU.TS00712 BYPASSED DUE TO PARTIAL TABLE SPACE REORG
BMC50574I BELOW 16M = 8192K, ABOVE 16M = 1396396K, CPUS = 3
BMC50388I 18: COPY OF SECONDARY INDEX: 'DEFFCAT.DSNDBD.ARUB007.TS00712.1001.001' STARTED
BMC50389I 18: COPY OF SECONDARY INDEX: 'DEFFCAT.DSNDBD.ARUB007.TS00713.1001.001' STARTED
BMC50390I 17: COPY OF SECONDARY INDEX: 'DEFFCAT.DSNDBD.ARUB007.TS00713.1001.001' COMPLETE
BMC50469I TOTAL PAGES: 394398, ALLOWED: 0; AVAILABLE PAGES: 394398, ALLOWED: 394398

**Chapter 5  Examples of REORG PLUS jobs**
Figure 23  SYSPRINT for example 3, case 2 (part 7 of 9)

Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Figure 23  SYSPRINT for example 3, case 2 (part 8 of 9)

BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS007.I0001.A002'...
BMC50891I 1: DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.I0001.A002' IS RENAMED
BMC50891I 5: DATASET 'DEFFCAT.BMCDBC.ARUDB007.TS007.I0001.A005' IS RENAMED
BMC50890I 6: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A004'...
BMC50891I 6: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50890I 7: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00712.I0001.A001'...
BMC50891I 7: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00712.I0001.A001' IS RENAMED
BMC50890I 8: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A003'...
BMC50891I 8: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A003' IS RENAMED
BMC50890I 9: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A004'...
BMC50891I 9: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50890I 10: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A004'...
BMC50891I 10: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50890I 11: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A001'...
BMC50891I 11: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A001' IS RENAMED
BMC50890I 12: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00713.I0001.A001'...
BMC50891I 12: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00713.I0001.A001' IS RENAMED
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A005'...
BMC50891I 5: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A005' IS RENAMED
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A004'...
BMC50891I 4: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A003'...
BMC50891I 3: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A003' IS RENAMED
BMC50890I 2: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A002'...
BMC50891I 2: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A002' IS RENAMED
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A001'...
BMC50891I 1: DATASET 'DEFFCAT.DSNDBC.ARUDB007.TS00711.I0001.A001' IS RENAMED

Chapter 5  Examples of REORG PLUS jobs  459
Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

Figure 24 shows a portion of the statistics report sent to ASUSRPRT for case 2 of example 3.

This figure shows one set of statistics for each object type. Ellipses (...) at the beginning of a line indicate the omitted statistics sections.

---

**Figure 23**  SYSPRINT for example 3, case 2 (part 9 of 9)

```
BMC50891I 3: DATASET 'DEFCCAT.BMCDBD.ARUDB007.TS007.10001.A003' IS RENAMED
BMC50891I 2: DATASET 'DEFCCAT.BMCDBD.ARUDB007.TS007.10001.A002' IS RENAMED
BMC50891I 4: DATASET 'DEFCCAT.BMCDBD.ARUDB007.TS007.10001.A004' IS RENAMED
BMC50891I 5: DATASET 'DEFCCAT.BMCDBD.ARUDB007.TS007.10001.A005' IS RENAMED
BMC50891I 10: DATASET 'DEFCCAT.BMCDBD.ARUDB007.TS00711.10001.A005' IS RENAMED
BMC50891I 11: DATASET 'DEFCCAT.BMCDBD.ARUDB007.TS00712.10001.A001' IS RENAMED
BMC50891I 1: DATASET 'DEFCCAT.BMCDBD.ARUDB007.TS007.10001.A001' IS RENAMED
BMC50891I 12: DATASET 'DEFCCAT.BMCDBD.ARUDB007.TS00713.10001.A001' IS RENAMED
BMC50891I RENAME PROCESS COMPLETE. ELAPSED TIME = 00:00:00

BMC50890I 8: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00711.10001.A003'...
BMC50890I 4: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS007.10001.A004'...
BMC50890I 9: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00711.10001.A004'...
BMC50890I 6: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00711.10001.A001'...
BMC50890I 5: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS007.10001.A005'...
BMC50890I 10: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00711.10001.A005'...
BMC50890I 1: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS007.10001.A001'...
BMC50890I 3: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS007.10001.A003'...
BMC50890I 2: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS007.10001.A002'...
BMC50890I 12: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00713.10001.A001'...
BMC50890I 11: ATTEMPTING TO DELETE DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00712.10001.A001'...
BMC50890I 8: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00711.10001.A003' IS DELETED
BMC50891I 4: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS007.10001.A004' IS DELETED
BMC50891I 10: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00711.10001.A005' IS DELETED
BMC50891I 6: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00711.10001.A001' IS DELETED
BMC50891I 5: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS007.10001.A005' IS DELETED
BMC50891I 9: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00711.10001.A004' IS DELETED
BMC50891I 7: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00711.10001.A002' IS DELETED
BMC50891I 1: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS007.10001.A001' IS DELETED
BMC50891I 3: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS007.10001.A003' IS DELETED
BMC50891I 2: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS007.10001.A002' IS DELETED
BMC50891I 12: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00713.10001.A001' IS DELETED
BMC50891I 11: DATASET 'DEFCCAT.OLDDBC.ARUDB007.TS00712.10001.A001' IS DELETED
BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

---

**Figure 24**  ASUSRPR for example 3, case 2 (part 1 of 4)

```
BMCSTATS V11.01.00 REPORT FOR DEFF V101        TIME 2013-10-14-16.18.44.818872                                          1
INDEX --------------- ARU.TS00711
ASSOCIATIONS
TABLESPACE ------ ARUDB007.TS007
TABLE ---------------- ARU.TS0071
HISTORY
SAVESTATS-----------------Y  UPDATEDB2-------------------N  UTILCODE---------------------R  LOCATION-------------------DEFF
SAMPLING-------------------N
ATTRIBUTES
PIECESIZE--------------------0  TYPE TS----------------(BLANK)  UNIQUERULE-------------------U  KEYLENGTH-------------------34
SUBPAGES---------------------0  INDEXTYPE--------------------2  CLUSTERING-------------------Y  COLCOUNT---------------------3
INDEXSPACE-------------TS00711  PGSIZE-----------------------4  EXTTYPE----------------(BLANK)  COMPRESS---------------------N
COLNAME------------EMPHIREDATE
STATISTICS
FIRSTKEYCARD----------------89  FULLKEYCARD---------------2084  NACTIVE--------------------105  TBCARDF-------------------2084
NUMNONLEAF--------------------16  NLEVELS----------------------2  CLUSTERRATIO----------------96  NLEAF-----------------------31
AVGKEYLEN-------------------34  REORGSPACE------------------18  REORGSPACE_KB--------------864  REORGNLEVELS-----------------2
HIGH2K-----------X'0022000101'  HIGH2K (CHAR)------------.....  LOW2K------------X'0014690308'  LOW2K (CHAR)-------------....
DATAREPEAT------------------77
ALLOCATION
SPACE-----------------------46  SPACE_KB------------------2208
```

---
### Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

**Figure 24** ASUSRPRT for example 3, case 2 (part 2 of 4)

<table>
<thead>
<tr>
<th>INDEXPART</th>
<th>PART 001</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSOCIATIONS</td>
<td></td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>ARUDB007.TS007</td>
</tr>
<tr>
<td>TABLE</td>
<td>ARU.TS0071</td>
</tr>
<tr>
<td>HISTORY</td>
<td></td>
</tr>
<tr>
<td>SAVESTATS</td>
<td>Y</td>
</tr>
<tr>
<td>UPDATEDB2</td>
<td>N</td>
</tr>
<tr>
<td>UTILCODE</td>
<td>R</td>
</tr>
<tr>
<td>LOCATION</td>
<td>(NULL)</td>
</tr>
<tr>
<td>SAMPLING</td>
<td>N</td>
</tr>
<tr>
<td>ATTRIBUTES</td>
<td></td>
</tr>
<tr>
<td>POTY</td>
<td>-1</td>
</tr>
<tr>
<td>ALLUNIT</td>
<td>T</td>
</tr>
<tr>
<td>STORUNIT</td>
<td>-1</td>
</tr>
<tr>
<td>PCTFREE</td>
<td>15</td>
</tr>
<tr>
<td>FREEPAGE</td>
<td>15</td>
</tr>
<tr>
<td>RBA_FORMAT</td>
<td>BASIC</td>
</tr>
<tr>
<td>STORNAME</td>
<td>ARULARSG</td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>FIRSTKEYCARD</td>
<td>17</td>
</tr>
<tr>
<td>FULLKEYCARD</td>
<td>1610</td>
</tr>
<tr>
<td>CARD</td>
<td>1610</td>
</tr>
<tr>
<td>NACTIVE</td>
<td>26</td>
</tr>
<tr>
<td>NLEVELS</td>
<td>20</td>
</tr>
<tr>
<td>FAROFFPOS</td>
<td>0</td>
</tr>
<tr>
<td>NEAROFFPOS</td>
<td>6</td>
</tr>
<tr>
<td>LEAFDIST</td>
<td>15</td>
</tr>
<tr>
<td>NLEAF</td>
<td>20</td>
</tr>
<tr>
<td>FREE</td>
<td>17</td>
</tr>
<tr>
<td>FULL</td>
<td>0</td>
</tr>
<tr>
<td>AVGKEYLEN</td>
<td>9</td>
</tr>
<tr>
<td>NLAVGKEYLEN</td>
<td>15</td>
</tr>
<tr>
<td>VCATNAME</td>
<td>DEFFCAT</td>
</tr>
<tr>
<td>RBA_FORMAT</td>
<td>BASIC</td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>FIRSTKEYCARD</td>
<td>17</td>
</tr>
<tr>
<td>FULLKEYCARD</td>
<td>1610</td>
</tr>
<tr>
<td>CARD</td>
<td>1610</td>
</tr>
<tr>
<td>NACTIVE</td>
<td>26</td>
</tr>
<tr>
<td>NLEVELS</td>
<td>20</td>
</tr>
<tr>
<td>FAROFFPOS</td>
<td>0</td>
</tr>
<tr>
<td>NEAROFFPOS</td>
<td>6</td>
</tr>
<tr>
<td>LEAFDIST</td>
<td>15</td>
</tr>
<tr>
<td>NLEAF</td>
<td>20</td>
</tr>
<tr>
<td>FREE</td>
<td>17</td>
</tr>
<tr>
<td>FULL</td>
<td>0</td>
</tr>
<tr>
<td>AVGKEYLEN</td>
<td>9</td>
</tr>
<tr>
<td>NLAVGKEYLEN</td>
<td>15</td>
</tr>
<tr>
<td>VCATNAME</td>
<td>DEFFCAT</td>
</tr>
<tr>
<td>RBA_FORMAT</td>
<td>BASIC</td>
</tr>
</tbody>
</table>

**COLUMN** EMPHIREDATE

| ASSOCIATIONS |         |
| TABLESPACE | ARUDB007.TS007 |
| TABLE | ARU.TS0071 |
| ATTRIBUTES |         |
| COLNO | 6 |
| COLTYPE | DATE |
| LENGTH | 4 |
| N | Y |
| STATISTICS |         |
| COLCARD | 89 |
| COLNULLS | 0 |
| COLAVG | 5 |
| COLMIN | 5 |
| COLMAX | 5 |

**MOST FREQUENT VALUES**

<table>
<thead>
<tr>
<th>FREQUENCY</th>
<th>COLVAL</th>
<th>COLTYPE</th>
<th>LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.383877</td>
<td>X'0014881209' (CHAR)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.195777</td>
<td>X'0022000101' (CHAR)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.191939</td>
<td>X'0014030101' (CHAR)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.191939</td>
<td>X'0014690308' (CHAR)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.191939</td>
<td>X'0014721112' (CHAR)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.191939</td>
<td>X'0014820101' (CHAR)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.191939</td>
<td>X'0014850131' (CHAR)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.191939</td>
<td>X'0022000228' (CHAR)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.000960</td>
<td>X'0019900324' (CHAR)</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**KEYCARD** EMPHIREDATE

| ASSOCIATIONS |         |
| TABLESPACE | ARUDB007.TS007 |
| TABLE | ARU.TS0071 |
| ATTRIBUTES AND STATISTICS |         |
| COLUMN | EMPHIREDATE PART 001 |
| COLNO | 6 |
| COLTYPE | DATE |
| LENGTH | 4 |
| N | Y |
| STATISTICS |         |
| COLCARD | 89 |
| COLNULLS | 0 |
| COLAVG | 5 |
| COLMIN | 5 |
| COLMAX | 5 |
### Example 3: Partial reorganization, SHRLEVEL REFERENCE, and a single copy of a subset of partitions

**Figure 24** ASUSRPRT for example 3, case 2 (part 3 of 4)

<table>
<thead>
<tr>
<th>KEYCARD</th>
<th>EMPHIREDATE PART 001</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTES AND STATISTICS</td>
<td></td>
</tr>
<tr>
<td>COLUMN</td>
<td>EMPHIREDATE COLCARD</td>
</tr>
<tr>
<td>COLUMN</td>
<td>EMPNO COLCARD</td>
</tr>
<tr>
<td>COLUMN</td>
<td>EMPDEPTNAME COLCARD</td>
</tr>
</tbody>
</table>

**BMCSTATS V11.01.00 REPORT FOR DEFF V101**

**TABLESPACE** ARUDB007.TS007

**HISTORY**

<table>
<thead>
<tr>
<th>SAVESTATS</th>
<th>Y</th>
<th>LOCATION</th>
<th>ARUDB007.TS007</th>
<th>PART 001</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PARTITIONS</td>
<td>16</td>
<td>NTABLES</td>
<td>-1</td>
<td>PGSIZE</td>
</tr>
<tr>
<td>DISSIZE</td>
<td>-0</td>
<td>MAXROWS</td>
<td>-125</td>
<td>TYPE</td>
</tr>
<tr>
<td>LOB</td>
<td>N</td>
<td>XML</td>
<td>-N</td>
<td>MEMBER_CLUSTER</td>
</tr>
</tbody>
</table>

**STATISTICS**

| NACTIVE | -116 | REORGSPACE | -31 | REORGSPACE_KB | 1488 | ROMMAXFOUND | 165 |
| ROWMINFOUND | -107 | ROWAVG | 103 |
| ALLOCATION | | SPACE | -53 | SPACE_KB | -2544 |

**TABLEPART** ARUDB007.TS007

**HISTORY**

<table>
<thead>
<tr>
<th>SAVESTATS</th>
<th>Y</th>
<th>LOCATION</th>
<th>ARUDB007.TS007</th>
<th>PART 001</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PQTY</td>
<td>2</td>
<td>SQTY</td>
<td>1</td>
<td>PCTFREE</td>
</tr>
<tr>
<td>ALLOCUNIT</td>
<td>C</td>
<td>STORtype</td>
<td>-1</td>
<td>COMpress</td>
</tr>
<tr>
<td>ROW_FORMAT</td>
<td>RRF</td>
<td>RBA_FORMAT</td>
<td>BASIC</td>
<td>VCATNAME</td>
</tr>
</tbody>
</table>

**STATISTICS**

| CARD | 1610 | NACTIVE | -64 | NPAGES | -56 | ROMAVG | 102 |
| ROWMINFOUND | -118 | AVGNONCOMROWLEN | -1 | DIRTY | -0 |
| FULL | 0 | FARINDREF | 0 | NEARINDREF | 0 | PERCACTIVE | -65 |
| PCTPAGES | 87 | PERCDCROP | 0 | REORGSPACE | 15 | REORGSPACE_KB | 720 |
| PCUSED | 50 | POTYROWS | -9343 |

**ALLOCATION**

| IPREFIX | SPACE | -30 | SPACE_KB | -1440 | NUMDATASETS | -1 |
| EXTENTS | 1 | VOLCOUNT | -1 | DEVTYPE | -3390 | VOLUME | ARU382 |

...
Example 4: Index-only reorganization

In this example, a non-unique, nonpartitioned secondary index is reorganized. Because this reorganization processes only an index, a SYSREC data set is not required and no image copy is requested. The default DDTYPE option enables REORG PLUS to dynamically allocate one SYSUT1 data set. This data set will contain the unloaded index key-RID pairs and serve as input for BMCSORT. Statistics for the reorganized index are reported, but are not recorded in the DB2 catalog or DASD MANAGER PLUS tables.

Message 500411 indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 68 describes the key command options for this job.

Table 68  Key command options used in example 4

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG INDEX</td>
<td>specifies that the index named in the statement is to be reorganized</td>
</tr>
<tr>
<td>REDEFINE NO</td>
<td>tells REORG PLUS not to delete and redefine the VSAM data sets containing the DB2 objects. Instead, REORG PLUS will reset the HURBA.</td>
</tr>
</tbody>
</table>

Figure 25 shows the JCL for example 4.

Figure 25  JCL for example 4 (part 1 of 2)
Example 4: Index-only reorganization

Figure 25  JCL for example 4 (part 2 of 2)

```
//UTPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//SYSTEM DD SYSOUT=*  
//*  
//SYSIN DD *  
REORG INDEX ARU.TS00211  
REDEFINE NO  
DDTYPE WORK  
DSNPAT 'ARU.EXMPL04.&DDNAME'  
/*
```

Figure 26 shows the SYSPRINT output for example 4.

Figure 26  SYSPRINT for example 4 (part 1 of 5)

```
***** BMC REORG PLUS FOR DB2 V11R1.00 *****  
(REOR PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762)
BMC5000I UTILITY EXECUTION STARTING 10/14/2013 16:19:08 ...
BMC5002I UTILITY ID = 'EXMPL04'. DB2 SUBSYSTEM ID = 'DEFF'. OPTION MODULE = 'ARU$OPTS'.
BMC50471I z/OS 2.1.0,PID=HBB7790,DFSMS FOR Z/OS=2.1.0,DB2=10.1.0
BMC50471I REGION=(16M,256M),ABOVE 16M=166924K,16M=83836K,16M=95851K,CPUS=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040306M,MEMLIMIT SET BY:REGION=0
BMC50471I REORG PLUS FOR DB2--V11.01.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V11.01.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V11.01.00
BMC50471I MAINT: BPJ0661  BPJ0667  BPJ0671  BPJ0674  BPJ0675  BPJ0682  BPJ0686  BPJ0688  BPJ0697
BMC50471I BMGSORT ENGINE--V02.04.01
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V11.01.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V11.01.00
BMC50471I EXTENDED BUFFER MANAGER--V06.01.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I ALTFAIL=RCVPEND  
ICDDON=(BMCICY,BMCICZ)  
RODOMAX=300%
BMC50471I ANALMAx=1000%
BMC50471I ARC=NO  
IDCACHE=10000  
SCPYMAx=N
BMC50471I ARCHDND=SYSARC  
INSDRFIL=10  
SDUMP=YES
BMC50471I ARCHFORMAT=BMX  
INLINDEX=YES  
SHORTMEMORY=CONTINUE
BMC50471I AVAILPAGECPT=100  
IXONEX=NO  
SISSNP=NO
BMC50471I BIDMAX=3005  
IRANDONX=NO  
SHAR=0
BMC50471I BMXHIST=YES  
KEEPDICTIONARY=NO
BMC50471I BMXHIST=NO  
MSMORE=(OK,OK)
BMC50471I BMXHIST=NO  
SORTDEV=(,SYSALLDA)
BMC50471I CONEXEC=NO  
SQL=NO  
SORTNUM=32
BMC50471I CMPYD=(BCPCCP,BMCCP2)  
LDCKRN=YES
BMC50471I CPGYLV=PART  
LDGFN=NONE  
SPILSCT=NONE
BMC50471I CPGY=1000%x  
LDGM=NONE  
SPILINIT=WORK
BMC50471I CPGYD=(BCPCCP,BMCCP2)  
LDGSPI=(200000,100000)  
SOLDLY=3
BMC50471I CPGYFAIL=TERM  
LDHTK=0  
SQRTRY=100
BMC50471I DATACAP=NO  
LONGLOG=CONTINUE  
STAGEDSM=0
BMC50471I DLDNO=DOLINE  
LONGNAMEPRINT=MIDDLE  
STOPCMT=YES
BMC50471I DEADLINE=NONE  
MAPIEXT=TERM,REXX
BMC50471I DELAY=1200  
MAXNEWPARTS=2  
STOPTRY=300
BMC50471I DELFILES=YES  
MAXR=1000  
TAPEDISP=DELETE
BMC50471I DESCCDE=(3,7)  
MAXSORTMEM=0  
TASKMAX=1000%
BMC50471I DRAINTYP=ALL  
MXTAPE=3  
TERMEXIT=(NONE,REXX)
BMC50471I DRAIND=1  
MMEXTEND=CONTINUE  
TIMEOUT=TERM
BMC50471I DNNTRY=255  
MINSORTMEM=0  
TOTALPAGECPT=0
BMC50471I DSNWAIT=NONE  
MSGLEVEL=1  
TSSAMPLECPT=100
BMC50471I DSNUEXIT=(NONE,ASM)  
OFFPDSIN=10  
UBUFF=20
BMC50471I DNSUTF3=YES  
ORIGDISP=DELETE  
UNLDMAX=300%
BMC50471I DSNUTLQ=NONE  
PREFORMAT=NO
BMC50471I DSNUTLQ=NONE  
UNLOAD=RELOAD
BMC50471I DSNUTLQ=NONE  
UNLOAD=RELOAD
BMC50471I EXCLUD=(X37,X32,X06)  
UTILITY_CLCICSID=UTILITY
BMC50471I FASTSWITCH=NO  
RXVICDON=(BMCICY,BMCICZ)  
UTILITY_NULLINIT=UTILITY
```
### Figure 26  SYSPRINT for example 4 (part 2 of 5)

<table>
<thead>
<tr>
<th>BMC50471I</th>
<th>FILECHK=WANN</th>
<th>RVYDDN=(BMCRLY, BMCR2)</th>
<th>USESTATE=SUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471I</td>
<td>FORCE_AT=(START,3)</td>
<td>REDEFINE=YES</td>
<td>WBUFSS=(20,10)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FORCE_RPT=NO</td>
<td>RENMMAX=50</td>
<td>WORKDDN=SYSUT1</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FORCE=NONE</td>
<td>RIMMDSZ=2097152</td>
<td>WORKUNIT=SYSALLDA</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FSFAILBACK=NO</td>
<td>RIMMAD=1</td>
<td>XBMID=</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FSTHRESHOLD=0</td>
<td>RMAPMEN=0</td>
<td>ZM=</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>HASHAX=YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>PLAN=AIRQA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BMC50470I</th>
<th>ODTYPE = UNLOAD</th>
<th>WORK</th>
<th>SORTWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I</td>
<td>ACTIVE = NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>IFALLOC = USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>ALLOC = N/A</td>
<td>N/A</td>
<td>ANY</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SMS = NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SMSUNIT = NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNITCT = (0,0)</td>
<td>(0,0)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT = (25,25)</td>
<td>(25,25)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AGVLCP = ((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DSNTYPE = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT = 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ = ((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT = (SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNITCT = (0,0)</td>
<td>(0,0)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AGVLCP = ((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DSNTYPE = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT = 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ = ((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT = (SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNITCT = (0,0)</td>
<td>(0,0)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AGVLCP = ((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DSNTYPE = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT = 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ = ((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT = (SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNITCT = (0,0)</td>
<td>(0,0)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AGVLCP = ((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DSNTYPE = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT = 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ = ((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT = (SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNITCT = (0,0)</td>
<td>(0,0)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AGVLCP = ((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DSNTYPE = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT = 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ = ((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT = (SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNITCT = (0,0)</td>
<td>(0,0)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AGVLCP = ((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DSNTYPE = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT = 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ = ((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT = (SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNITCT = (0,0)</td>
<td>(0,0)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AGVLCP = ((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DSNTYPE = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT = 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ = ((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Chapter 5  Examples of REORG PLUS jobs  465
**Example 4: Index-only reorganization**

<table>
<thead>
<tr>
<th>BMC50470I DDGEMPTY</th>
<th>DDGSCRAT</th>
<th>DDTYPE</th>
<th>ACTIVE</th>
<th>IFALLOC</th>
<th>ALLOC</th>
<th>SMS</th>
<th>SMUNIT</th>
<th>SIZEPECT</th>
<th>UNIT</th>
<th>UNITCNT</th>
<th>VOLCNT</th>
<th>AVGVOLS</th>
<th>DATACLAS</th>
<th>DATACLAS</th>
<th>MGMTCLAS</th>
<th>STORCLAS</th>
<th>THRESHLD</th>
<th>MAXEXTSZ</th>
<th>EXPDT</th>
<th>RETPD</th>
<th>GDGLIMIT</th>
<th>GDGEMPTY</th>
<th>GDGSCRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>NO</td>
<td>REMPICPY</td>
<td>NO</td>
<td>USE</td>
<td>N/A</td>
<td>NO</td>
<td>NO</td>
<td>(100,100)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(0,0)</td>
<td>(25,25)</td>
<td>(30000,TRK)</td>
<td>(0,0,TRK)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>0</td>
<td>((0,K),(0,K))</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
<td>REMBCPY</td>
<td>NO</td>
<td>USE</td>
<td>N/A</td>
<td>NO</td>
<td>NO</td>
<td>(100,100)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(0,0)</td>
<td>(25,25)</td>
<td>(30000,TRK)</td>
<td>(0,0,TRK)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>0</td>
<td>((0,K),(0,K))</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
<td>REMPICPY</td>
<td>NO</td>
<td>USE</td>
<td>N/A</td>
<td>NO</td>
<td>NO</td>
<td>(100,100)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(0,0)</td>
<td>(25,25)</td>
<td>(30000,TRK)</td>
<td>(0,0,TRK)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>0</td>
<td>((0,K),(0,K))</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
<td>REMBCPY</td>
<td>NO</td>
<td>USE</td>
<td>N/A</td>
<td>NO</td>
<td>NO</td>
<td>(100,100)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(0,0)</td>
<td>(25,25)</td>
<td>(30000,TRK)</td>
<td>(0,0,TRK)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>0</td>
<td>((0,K),(0,K))</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

**Figure 26** SYSPRINT for example 4 (part 3 of 5)
**Figure 26**  SYSPRINT for example 4 (part 4 of 5)

<table>
<thead>
<tr>
<th>BMC50470I</th>
<th>SRUNIT</th>
<th>NO</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I</td>
<td>SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT</td>
<td>(25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AVGOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>STORCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNLOAD</td>
<td>DSNPAT=UID..BMC..TSIX..DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>WORK</td>
<td>DSNPAT=UID..BMC..TSIX..DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SORTWORK</td>
<td>DSNPAT=UID..UTILPFX..DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>ARCHIVE</td>
<td>DSNPAT=UID..UTILPFX..DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SYSPUNCH</td>
<td>DSNPAT=UID..UTILPFX..DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>LOCPFCPY</td>
<td>DSNPAT=UID..DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>LOCPICPY</td>
<td>DSNPAT=UID..DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>LOCBFCPY</td>
<td>DSNPAT=UID..DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>LOCBICPY</td>
<td>DSNPAT=UID..DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>LOCPLCPY</td>
<td>DSNPAT=UID..DDNAME..TSIX..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>LOCBLCPY</td>
<td>DSNPAT=UID..DDNAME..TSIX..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>REMPFCPY</td>
<td>DSNPAT=UID..DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>REMPICPY</td>
<td>DSNPAT=UID..DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>REMBFCPY</td>
<td>DSNPAT=UID..DDNAME..TSIX..F&amp;PART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>REMBLCPY</td>
<td>DSNPAT=UID..DDNAME..TSIX..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DB2 VERSION</td>
<td>1010</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>SUBSYSTEM DEFAULT</td>
<td>DEFF</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>CHARACTER SET</td>
<td>ALPHANUM</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DATE FORMAT</td>
<td>USA</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LOCAL DATE LENGTH</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LOCAL TIME LENGTH</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DECIMAL POINT</td>
<td>PERIOD</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DECIMAL ARITHMETIC</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DECIMLIMITER</td>
<td>DEFAULT</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>SQL DELIMITER</td>
<td>DEFAULT</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ENCODING SCHEME</td>
<td>EBCDIC</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>APPL. ENCODING SCHEME</td>
<td>EBCDIC</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>MIXED</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>EBCDIC CCSID</td>
<td>(37,65534,65534)</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ASCII CCSID</td>
<td>(819,65534,65534)</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>UNICODE CCSID</td>
<td>(367,1208,1200)</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>IMPLICIT TIME ZONE</td>
<td>CURRENT (-05:00)</td>
<td></td>
</tr>
<tr>
<td>BMC50472I</td>
<td>DB2 MODE</td>
<td>NFM</td>
<td></td>
</tr>
<tr>
<td>BMC50472I</td>
<td>BMCUTIL</td>
<td>&quot;BMCUTIL.CMN_BMCUTIL&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50472I</td>
<td>BMCSYNC</td>
<td>&quot;BMCUTIL.CMN_BMCSYNC&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50472I</td>
<td>BMCHIST</td>
<td>&quot;BMCUTIL.CMN_BMCHIST&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50472I</td>
<td>BMCDICT</td>
<td>&quot;BMCUTIL.CMN_BMCDICT&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50472I</td>
<td>BMXCOPY</td>
<td>&quot;BMCUTIL.CMN_BMXCOPY&quot;</td>
<td></td>
</tr>
<tr>
<td>BMC50472I</td>
<td>DASD MANAGER PLUS TABLES:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50472I</td>
<td>ATS102_RS_COOLDIST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50472I</td>
<td>ATS102_RS_TABLESPACE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chapter 5  Examples of REORG PLUS jobs  467
Example 5: Partial reorganization with UNLOAD PAUSE

In this example, REORG PLUS reorganizes partition 1 of a partitioned table space. The table has one clustering index and two nonpartitioned secondary indexes.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 69 on page 469 describes the key command options for this job.
Table 69  Key command options used in example 5

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>PART 01</td>
<td>limits the reorganization to partition 1</td>
</tr>
<tr>
<td>UNLOAD PAUSE</td>
<td>instructs REORG PLUS to pause after the UNLOAD phase</td>
</tr>
<tr>
<td></td>
<td>The index space taking part in the reorganization will be stopped at the</td>
</tr>
<tr>
<td></td>
<td>end of unload processing. REORG PLUS completes with return code 4.</td>
</tr>
<tr>
<td>BMCSTATS YES</td>
<td>tells REORG PLUS to update the statistics in the DASD MANAGER PLUS</td>
</tr>
<tr>
<td>UPDATEDB2STATS YES</td>
<td>database statistics tables and the DB2 catalog</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS does not save the statistics in this case, because the job is</td>
</tr>
<tr>
<td></td>
<td>paused.</td>
</tr>
</tbody>
</table>

Figure 27 shows the JCL for example 5.

**Figure 27  JCL for example 5**

```plaintext
//            JOB
//             /*
//             /* ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//             /* PARTITIONED TABLESPACE, ONE CLUSTERING INDEX, TWO SECONDARY    */
//             /* INDEXES REORG UNLOAD PAUSE PART 01 OF PARTITIONED TABLESPACE.  */
//             /* ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//             //BMCREORG EXEC PGM=ARUUMAIN,
//             //         PARM='DEFF,EXMPL05,NEW,,MSGLEVEL(1),ARU$OPTS'
//             //STEPLIB   DD  DISP=SHR,DSN=ProductLibraries
//             //         DD  DISP=SHR,DSN=DB2.DSNEXIT
//             //         DD  DISP=SHR,DSN=DB2.DSNLOAD
//             //*
//             //SYSPRINT  DD  SYSOUT=*                                                
//             //SYSOUT   DD  SYSOUT=*                                                
//             //UPTRNDRIT DD  SYSOUT=*                                                
//             //SYSUDUMP  DD  SYSOUT=*                                               
//             //SYSTEM    DD  SYSOUT=*                                               
//             //*********
//             //SYSSIN   DD  *
//             //REORG TABLESPACE ARUDB032.TS032 PART 01
//             //UNLOAD PAUSE
//             //BMCSTATS YES
//             //UPDATEDB2STATS YES
//             //DDTYPE UNLOAD
//             //   DSNPAT 'ARU.EXMPL05.&DDNAME'
//             //DDTYPE WORK
//             //   DSNPAT 'ARU.EXMPL05.&DDNAME'
//             /*
```

Figure 28 on page 470 shows the SYSPRINT output for example 5.
Example 5: Partial reorganization with UNLOAD PAUSE

Table 28 SYSPRINT for example 5 (part 1 of 5)
Figure 28  SYSPRINT for example 5 (part 2 of 5)

<table>
<thead>
<tr>
<th>BMC50470I DATACLAS</th>
<th>(NONE,NONE)</th>
<th>(NONE,NONE)</th>
<th>(NONE,NONE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
<td>0</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I GDGSCRAT</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I OTYPE</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BMC50470I ACTIVE</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(5,100)</td>
</tr>
<tr>
<td>BMC50470I UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>(30000,TRK),(30000,TRK)</td>
<td>(30000,TRK),(30000,TRK)</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Example 5: Partial reorganization with UNLOAD PAUSE
Example 5: Partial reorganization with UNLOAD PAUSE

**Figure 28** SYSPRINT for example 5 (part 3 of 5)

<table>
<thead>
<tr>
<th>BMC50470I</th>
<th>DDTYPE</th>
<th>SYSPUNCH</th>
<th>BMC50470I</th>
<th>DDTYPE</th>
<th>LOCPCLPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I</td>
<td>ACTI...</td>
<td>USE</td>
<td>BMC50470I</td>
<td>AC...</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>IFALO...</td>
<td>USE</td>
<td>BMC50470I</td>
<td>ALLOC...</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SMS...</td>
<td>NO</td>
<td>BMC50470I</td>
<td>SM...</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SMSUNIT...</td>
<td>NO</td>
<td>BMC50470I</td>
<td>SMSUNIT...</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SIZEPCT...</td>
<td>(100,100)</td>
<td>BMC50470I</td>
<td>SIZEPCT...</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT...</td>
<td>(SYSALLDA,SY...</td>
<td>BMC50470I</td>
<td>UNIT...</td>
<td>(SYSALLDA,SY...</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNI...</td>
<td>(0,0)</td>
<td>BMC50470I</td>
<td>UNI...</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLC...</td>
<td>(25,25)</td>
<td>BMC50470I</td>
<td>VOLC...</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AVGVOLSP...</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>BMC50470I</td>
<td>AVGVOLSP...</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS...</td>
<td>(NONE,...</td>
<td>BMC50470I</td>
<td>DATACLAS...</td>
<td>(NONE,...</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MMGCTL...</td>
<td>(NONE,...</td>
<td>BMC50470I</td>
<td>MMGCTL...</td>
<td>(NONE,...</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>STORC...</td>
<td>(NONE,...</td>
<td>BMC50470I</td>
<td>STORC...</td>
<td>(NONE,...</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>THRES...</td>
<td>0</td>
<td>BMC50470I</td>
<td>THRES...</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ...</td>
<td>(0,K),(0,...</td>
<td>BMC50470I</td>
<td>MAXEXTSZ...</td>
<td>(0,K),(0,...</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>EXPDT...</td>
<td></td>
<td>BMC50470I</td>
<td>EXPDT...</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>RETPD...</td>
<td></td>
<td>BMC50470I</td>
<td>RETPD...</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DDTYPE</td>
<td>REMPLCPY</td>
<td>BMC50470I</td>
<td>DDTYPE</td>
<td>REMBLCPY</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>ACTI...</td>
<td>NO</td>
<td>BMC50470I</td>
<td>ACTI...</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>IFALO...</td>
<td>USE</td>
<td>BMC50470I</td>
<td>IFALO...</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>ALLOC...</td>
<td>N/A</td>
<td>BMC50470I</td>
<td>ALLOC...</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SMS...</td>
<td>NO</td>
<td>BMC50470I</td>
<td>SMS...</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SMSUNIT...</td>
<td>NO</td>
<td>BMC50470I</td>
<td>SMSUNIT...</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SIZEPCT...</td>
<td>(100,100)</td>
<td>BMC50470I</td>
<td>SIZEPCT...</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT...</td>
<td>(SYSALLDA,SY...</td>
<td>BMC50470I</td>
<td>UNIT...</td>
<td>(SYSALLDA,SY...</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNI...</td>
<td>(0,0)</td>
<td>BMC50470I</td>
<td>UNI...</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLC...</td>
<td>(25,25)</td>
<td>BMC50470I</td>
<td>VOLC...</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AVGVOLSP...</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>BMC50470I</td>
<td>AVGVOLSP...</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS...</td>
<td>(NONE,...</td>
<td>BMC50470I</td>
<td>DATACLAS...</td>
<td>(NONE,...</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MMGCTL...</td>
<td>(NONE,...</td>
<td>BMC50470I</td>
<td>MMGCTL...</td>
<td>(NONE,...</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>STORC...</td>
<td>(NONE,...</td>
<td>BMC50470I</td>
<td>STORC...</td>
<td>(NONE,...</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>THRES...</td>
<td>0</td>
<td>BMC50470I</td>
<td>THRES...</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ...</td>
<td>(0,K),(0,...</td>
<td>BMC50470I</td>
<td>MAXEXTSZ...</td>
<td>(0,K),(0,...</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>EXPDT...</td>
<td></td>
<td>BMC50470I</td>
<td>EXPDT...</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>RETPD...</td>
<td></td>
<td>BMC50470I</td>
<td>RETPD...</td>
<td></td>
</tr>
</tbody>
</table>

REORG PLUS for DB2 Reference Manual
Example 5: Partial reorganization with UNLOAD PAUSE

![Figure 28 SYSPRINT for example 5 (part 4 of 5)]
In this example, REORG PLUS restarts the reorganization job from example 5 in the RELOAD phase. Because this job dynamically allocates the unload and index work files, REORG PLUS manages allocation of these data sets (including their disposition) appropriately for a restarted job. The command statement options are not changed for this restart.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Because no copy option is specified in the command statement, message BMC50387W is produced. This message indicates that the table space state is set to copy pending after the utility completes. The job completes with return code 4.

Table 70 on page 475 describes the key command options for this job.
Table 70  Key command options used in example 6

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>PART 01</td>
<td>limits the reorganization to partition 1</td>
</tr>
<tr>
<td>BMCSTATS YES</td>
<td>tells REORG PLUS to update statistics in the DASD MANAGER PLUS database statistics tables and the DB2 catalog</td>
</tr>
<tr>
<td>UPDATEDB2STATS YES</td>
<td>tells REORG PLUS to update statistics in the DASD MANAGER PLUS database statistics tables and the DB2 catalog</td>
</tr>
</tbody>
</table>

The JCL for this example also includes the DD statement //ASUSRPRRT DD DUMMY. This DD statement tells REORG PLUS not to produce a report of the statistics.

Figure 29 shows the JCL for example 6.

Figure 29  JCL for example 6

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  
//* PARTITIONED TABLESPACE, ONE CLUSTERING INDEX, TWO SECONDARY        * 
//* INDEXES.  *                                                        
//* RESTART FROM UNLOAD PAUSE OF PART 01 - UTILID MUST BE THE SAME     * 
//*                                                            * 
//* NOTE: UNLOAD PAUSE OPTION IN REORG COMMAND DOES NOT HAVE TO BE    * 
//* DELETED IN RESTART JOB.  ALSO, NOTE MESSAGE FROM                  * 
//* UPDATEDB2STATS YES.                                               * 
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  
//BMCREORG EXEC PGM=ARUUMAIN,                                       
//             PARM='DEFF,EXMPL05,RESTART,,MSGLEVEL(1),ARU$OPTS'        
//STEPLIB   DD  DISP=SHR,DSN=ProductLibraries                        
//          DD  DISP=SHR,DSN=DB2.DSNEXIT                              
//          DD  DISP=SHR,DSN=DB2.DSNLOAD                              
//*                                                                 
//SYSPRINT  DD  SYSOUT=*                                           
//SYSOUT   DD  SYSOUT=*                                           
//UTPRINT   DD  SYSOUT=*                                           
//SYSDUMP   DD  SYSOUT=*                                           
//SYSTEM    DD  SYSOUT=*                                           
//ASUSRPRRT DD  DUMMY                                             
//*                                                                 
//SYSIN     DD  *                                               
REORG TABLESPACE ARUDB032.TS032 PART 01                           
UNLOAD PAUSE                                                      
BMCSTATS YES                                                     
UPDATEDB2STATS YES                                               
DDTYPE UNLOAD                                                    
  DSNPAT 'ARU.EXMPL05.&DDNAME'                                  
DDTYPE WORK                                                      
  DSNPAT 'ARU.EXMPL05.&DDNAME'                                  
/*
```
Example 6: Restart of a paused REORG job

Figure 30 shows the SYSPRINT output for example 6.

** Figure 30 SYSPRINT for example 6 (part 1 of 5) **
### Example 6: Restart of a paused REORG job

**Figure 30**  SYSPRINT for example 6 (part 2 of 5)

<table>
<thead>
<tr>
<th>BMC504701 VOLCNT</th>
<th>(25,25)</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC504701 AVGWOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 STORCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC504701 MAKERTSZ</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 EXPDT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 ACTIVE</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>BMC504701 DDTYPE</td>
<td>REMPICPY</td>
<td>LOCPICPY</td>
</tr>
<tr>
<td>BMC504701 GDGSCRAT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC504701 UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 VOLCNT</td>
<td>(25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC504701 AVGWOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 STORCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC504701 MAKERTSZ</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 EXPDT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 ACTIVE</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>BMC504701 DDTYPE</td>
<td>LOCBICPY</td>
<td>REMPICPY</td>
</tr>
<tr>
<td>BMC504701 GDGSCRAT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC504701 UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 VOLCNT</td>
<td>(25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC504701 AVGWOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 STORCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC504701 MAKERTSZ</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 EXPDT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 ACTIVE</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>BMC504701 DDTYPE</td>
<td>ARCHIVE</td>
<td>LOCPFPCPY</td>
</tr>
<tr>
<td>BMC504701 GDGSCRAT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC504701 UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 VOLCNT</td>
<td>(25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC504701 AVGWOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 STORCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC504701 MAKERTSZ</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 EXPDT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 ACTIVE</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>BMC504701 DDTYPE</td>
<td>ARCHIVE</td>
<td>LOCPFPCPY</td>
</tr>
<tr>
<td>BMC504701 GDGSCRAT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC504701 UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 VOLCNT</td>
<td>(25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC504701 AVGWOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 STORCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC504701 THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Example 6: Restart of a paused REORG job

Figure 30  SYSPRINT for example 6 (part 3 of 5)

```
BMC50470I MAXEXTSZ = ((0,K),(0,K))  ((0,K),(0,K))  ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I GDGLIMIT = 5  5  5
BMC50470I GDGEMPTY = NO  NO  NO
BMC50470I GDGSCRAT = NO  NO  NO
BMC50470I DDTYPE = SYSPUNCH
BMC50470I ACTIVE = YES
BMC50470I IFALLOC = USE
BMC50470I ALLOC = N/A
BMC50470I SMS = NO
BMC50470I SMSUNIT = NO
BMC50470I SIZEPCT = (100,100)
BMC50470I UNIT = (SYSALLDA,SYSALLDA)
BMC50470I UNITENT = (0,0)
BMC50470I VOLCNT = (25,25)
BMC50470I AVGVOLDP = ((30000,TRK),(30000,TRK))
BMC50470I DATAclas = (NONE,NONE)
BMC50470I MGMTclas = (NONE,NONE)
BMC50470I STORclas = (NONE,NONE)
BMC50470I THRESHld = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I GDGLIMIT = 5
BMC50470I GDGEMPTY = NO
BMC50470I GDGSCRAT = NO
BMC50470I DDTYPE = LOCPLCPY  LOCBLCPY
BMC50470I ACTIVE = NO  NO
BMC50470I IFALLOC = USE  USE
BMC50470I ALLOC = N/A  N/A
BMC50470I SMS = NO  NO
BMC50470I SMSUNIT = NO  NO
BMC50470I SIZEPCT = (100,100)  (100,100)
BMC50470I UNIT = (SYSALLDA,SYSALLDA)  (SYSALLDA,SYSALLDA)
BMC50470I UNITENT = (0,0)  (0,0)
BMC50470I VOLCNT = (25,25)  (25,25)
BMC50470I AVGVOLDP = ((30000,TRK),(30000,TRK))  ((30000,TRK),(30000,TRK))
BMC50470I DSNTYPE = (NONE,NONE)  (NONE,NONE)
BMC50470I DATAclas = (NONE,NONE)  (NONE,NONE)
BMC50470I MGMTclas = (NONE,NONE)  (NONE,NONE)
BMC50470I STORclas = (NONE,NONE)  (NONE,NONE)
BMC50470I THRESHld = 0  0
BMC50470I MAXEXTSZ = ((0,K),(0,K))  ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I DDTYPE = REMPLCPY  REMBLCPY
BMC50470I ACTIVE = NO  NO
BMC50470I IFALLOC = USE  USE
BMC50470I ALLOC = N/A  N/A
BMC50470I SMS = NO  NO
BMC50470I SMSUNIT = NO  NO
BMC50470I SIZEPCT = (100,100)  (100,100)
BMC50470I UNIT = (SYSALLDA,SYSALLDA)  (SYSALLDA,SYSALLDA)
BMC50470I UNITENT = (0,0)  (0,0)
BMC50470I VOLCNT = (25,25)  (25,25)
BMC50470I AVGVOLDP = ((30000,TRK),(30000,TRK))  ((30000,TRK),(30000,TRK))
BMC50470I DSNTYPE = (NONE,NONE)  (NONE,NONE)
BMC50470I DATAclas = (NONE,NONE)  (NONE,NONE)
BMC50470I MGMTclas = (NONE,NONE)  (NONE,NONE)
BMC50470I STORclas = (NONE,NONE)  (NONE,NONE)
BMC50470I THRESHld = 0  0
BMC50470I MAXEXTSZ = ((0,K),(0,K))  ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I DDTYPE = USE  USE
BMC50470I ALLOC = N/A  N/A
BMC50470I SMS = NO  NO
BMC50470I SMSUNIT = NO  NO
BMC50483I UNLOAD  DSNPAT=UID..BMC..ATSIX...ADNAME
```
Example 6: Restart of a paused REORG job

**Figure 30** SYSPRINT for example 6 (part 4 of 5)
Example 7: ANALYZE ONLY to generate space estimates

In this example, two jobs are run with ANALYZE ONLY. The jobs generate space estimates for a subsequent REORG execution that reorganizes a segmented table space with one clustering index and two secondary indexes. The only difference between the two jobs is that case 1 specifies ORDER NO and case 2 specifies ORDER YES.

No SYSREC or SYSUT1 data set is specified in either job because the jobs are run with ANALYZE ONLY. The messages that the ANALYZE phase issues provide data set size information needed to allocate the unload and work files for the subsequent REORG execution.
Example 7: ANALYZE ONLY to generate space estimates

Also, in message BMC51263I, REORG PLUS assigns indexes to SYSUT1nn data sets based on index key length. The index with the largest key length will be assigned to the first SYSUT1nn data set, the index with the second largest key length will be assigned to the second SYSUT1nn, and so on.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 71 describes the key command options for these jobs.

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>UNLOAD CONTINUE</td>
<td>In case 1, this option overrides the installation option default of UNLOAD=RELOAD, telling REORG PLUS to continue the reorganization with two-phase processing after the UNLOAD phase has unloaded the data. UNLOAD CONTINUE is required when ORDER NO is in effect.</td>
</tr>
<tr>
<td>ORDER NO</td>
<td>In case 1, ORDER NO performs all other functions of the reorganization without the overhead of sorting the data rows. Also, the clustering index is processed together with the secondary indexes.</td>
</tr>
<tr>
<td>ORDER YES</td>
<td>In case 2, with ORDER YES specified, the clustering index is embedded in the SYSREC.</td>
</tr>
<tr>
<td>ANALYZE ONLY</td>
<td>terminates utility execution after the ANALYZE phase completes, and removes utility ID entries from the BMC Software BMCUTIL and BMCSYNC tables</td>
</tr>
</tbody>
</table>

Figure 31 shows the JCL for case 1 of example 7.

Figure 31  JCL for example 7, case 1 (part 1 of 2)

```
//        JOB
//*
//*    **   **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **
//** SEGMENTED TABLESPACE, 3 INDEXES (ONE CLUSTERED, TWO NONCLUSTERED) *
//**     **
//** CASE 1: REORG TABLESPACE, ORDER NO, ANALYZE ONLY.     *
//**     **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **    **
//BMCREORG EXEC PGM=ARUUMAIN,
//    PARM='DEFF,EXMPL07A,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB DD DISP=SHR,DSN=ProductLibraries
//    DD DISP=SHR,DSN=DB2.DSNEXIT
//    DD DISP=SHR,DSN=DB2.DSNLOAD
//**
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
```
Example 7: ANALYZE ONLY to generate space estimates

Figure 31  JCL for example 7, case 1 (part 2 of 2)

```
//UTPRINT   DD  SYSOUT=*  
//SYSUDUMP  DD  SYSOUT=*  
//SYSTERM   DD  SYSOUT=*  
//*  
//SYSIN   DD  *  
REORG TABLESPACE ARUDB250.TS250  
UNLOAD CONTINUE  
ORDER NO  
ANALYZE ONLY  
/*
```

Figure 32 shows the JCL for case 2 of example 7.

Figure 32  JCL for example 7, case 2

```
//  
//  
// JOB  
//  
// **********  **********  **********  **********  **********  **********  **********  **********  **********  
// CASE 2: REORG TABLESPACE, ORDER YES, ANALYZE ONLY.  
// **********  **********  **********  **********  **********  **********  **********  **********  
//BMCREORG EXEC PGM=ARUUMAIN,  
// PARM='DEFF,EXMPL07B,NEW,,MSGLEVEL(1),ARU$OPTS'  
//STEPLIB   DD  DISP=SHR,DSN=ProductLibraries  
// DD DISP=SHR,DSN=DB2.DSNEXIT  
// DD DISP=SHR,DSN=DB2.DSNLOAD  
//  
//SYSPRINT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//UTPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//SYSTERM DD SYSOUT=*  
//  
//SYSIN DD *  
REORG TABLESPACE ARUDB250.TS250  
ORDER YES  
ANALYZE ONLY  
/*
```

Figure 33 shows the SYSPRINT output for case 1 of example 7.

Figure 33  SYSPRINT for example 7, case 1 (part 1 of 6)

```
*****  B M C   R E O R G   P L U S   F O R   D B 2    V11R1.00  *****  
(C) COPYRIGHT 1988 - 2013 BMC SOFTWARE, INC.  
REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762  
BMC50001I UTILITY EXECUTION STARTING   10/14/2013   16:20:36   ...  
BMC50471I REGION=0M,BELOW 16M=8836K,ABOVE 16M=1406872K,IEFUSI=NO,CPUS=3  
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040306M,MEMLIMIT SET BY:REGION=0
```
Figure 33  SYSPRINT for example 7, case 1 (part 2 of 6)

Example 7: ANALYZE ONLY to generate space estimates

Chapter 5  Examples of REORG PLUS jobs 483
Example 7: ANALYZE ONLY to generate space estimates

**Figure 33  SYSPRINT for example 7, case 1 (part 3 of 6)**

<table>
<thead>
<tr>
<th>BMC50470I DDTYPE</th>
<th>DTCFCPY</th>
<th>DTCFCPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I ACTIVE</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>BMC50470I IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(5,100)</td>
</tr>
<tr>
<td>BMC50470I UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>(25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I AGVGOLSP</td>
<td>((30000, TRK), (30000, TRK))</td>
<td>((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I DSNTP</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>(25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I AGVGOLSP</td>
<td>((30000, TRK), (30000, TRK))</td>
<td>((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I DSNTP</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>(25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I AGVGOLSP</td>
<td>((30000, TRK), (30000, TRK))</td>
<td>((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I DSNTP</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 7: ANALYZE ONLY to generate space estimates

Chapter 5 Examples of REORG PLUS jobs

| BMC50470I SIZEPCT   | = (100,100)         | LOCBLCPY   |
| BMC50470I UNIT      | = (SYSALLDA, SYSALLDA) |             |
| BMC50470I UNITCNT   | = (0,0)             |             |
| BMC50470I VOLCNT    | = (25,25)           |             |
| BMC50470I AVGWOLSP  | = ((30000, TRK), (30000, TRK)) |             |
| BMC50470I DATACLAS  | = (NONE, NONE)      |             |
| BMC50470I MMTCCLAS  | = (NONE, NONE)      |             |
| BMC50470I STORCLAS  | = (NONE, NONE)      |             |
| BMC50470I THRESHLD  | = 0                 |             |
| BMC50470I MAXEXTSZ  | = ((0, K), (0, K))  |             |
| BMC50470I EXPOT     |                     |             |
| BMC50470I RETPD     |                     |             |
| BMC50470I GDGEMPTY  | = NO                |             |
| BMC50470I GDGSCRAT  | = NO                |             |

| BMC50470I ODTYPE    | = LOCLCPY           | LOCLCPY    |
| BMC50470I ACTIVE    | = NO                |             |
| BMC50470I IFALLOC   | = USE               |             |
| BMC50470I ALLOC     | = N/A               |             |
| BMC50470I SMS       | = NO                |             |
| BMC50470I SMSUNIT   | = NO                |             |
| BMC50470I SIZEPCT   | = (100,100)         | (100,100)   |
| BMC50470I UNIT      | = (SYSALLDA, SYSALLDA) | (SYSALLDA, SYSALLDA) |
| BMC50470I UNITCNT   | = (0,0)             | (0,0)       |
| BMC50470I VOLCNT    | = (25,25)           | (25,25)     |
| BMC50470I AVGWOLSP  | = ((30000, TRK), (30000, TRK)) | (30000, TRK), (30000, TRK) |
| BMC50470I DATACLAS  | = (NONE, NONE)      | (NONE, NONE) |
| BMC50470I MMTCCLAS  | = (NONE, NONE)      | (NONE, NONE) |
| BMC50470I STORCLAS  | = (NONE, NONE)      | (NONE, NONE) |
| BMC50470I THRESHLD  | = 0                 |             |
| BMC50470I MAXEXTSZ  | = ((0, K), (0, K))  | ((0, K), (0, K)) |
| BMC50470I EXPOT     |                     |             |
| BMC50470I RETPD     |                     |             |
| BMC50470I GDGEMPTY  | = 5                 |             |
| BMC50470I GDGSCRAT  | = 5                 |             |

Figure 33 SYSPRINT for example 7, case 1 (part 4 of 6)
Example 7: ANALYZE ONLY to generate space estimates

### Figure 33  SYSPRINT for example 7, case 1 (part 5 of 6)

<table>
<thead>
<tr>
<th>BMC50483I</th>
<th>REMPLCPY DSNPAT=UID.&amp;DDNAME.&amp;TSIX...F&amp;PART..T&amp;TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50483I</td>
<td>REMBFCPY DSNPAT=UID.&amp;DDNAME.&amp;TSIX...F&amp;PART..T&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMPLCPY DSNPAT=UID.&amp;DDNAME.&amp;TSIX...T&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMBLCPY DSNPAT=UID.&amp;DDNAME.&amp;TSIX...T&amp;TIME</td>
</tr>
</tbody>
</table>

BMC50471I DB2 DSNDEEP MODULE SETTINGS:

- **VERSION** = 1010
- **SUBSYSTEM DEFAULT** = DEFF
- **CHARACTER SET** = ALPHANUM
- **DATE FORMAT** = USA
- **TIME FORMAT** = USA
- **LOCAL DATE LENGTH** = 0
- **LOCAL TIME LENGTH** = 0
- **DECIMAL POINT** = PERIOD
- **DECIMAL ARITHMETIC** = 15
- **SQL DELIMITER** = DEFAULT
- **ENCODING SCHEME** = EBCDIC
- **APPL. ENCODING SCHEME** = EBCDIC
- **MIXED** = NO
- **EBCDIC CCSID** = (37,65534,65534)
- **ASCII CCSID** = (819,65534,65534)
- **UNICODE CCSID** = (367,1208,1200)
- **IMPLICIT TIME ZONE** = CURRENT (-05:00)

BMC50028I DB2 MODE = NFM

BMC50471I BMCUTIL = 'BMCUTIL.CMN_BMCUTIL'

BMC50471I BMCSYNC = 'BMCUTIL.CMN_BMCSYNC'

BMC50471I BMCHIST = 'BMCUTIL.CMN_BMCHIST'

BMC50471I BMCDICT = 'BMCUTIL.CMN_BMCDICT'

BMC50471I BMCCOPY = 'BMCUTIL.CMN_BMCCOPY'

BMC50471I BMCCPYZ = 'BMCUTIL.CMN_BMCCPYZ'

BMC50471I BMCCPYZ = 'BMCUTIL.CMN_BMCCPYZ'

BMC50471I DASD MANAGER PLUS TABLES:

- **'ATS102.RS_COLDIST'**
- **'ATS102.RS_COLSTATS'**
- **'ATS102.RS_COLUMNS'**
- **'ATS102.RS_INDEXES'**
- **'ATS102.RS_INDEXPART'**
- **'ATS102.RS_INDEXPART_DIST'**
- **'ATS102.RS_STOGROUP'**
- **'ATS102.RS_TABLEPART'**
- **'ATS102.RS_TABLESPACE'**
- **'ATS102.RS_TABLESPACE'**

BMC50004I UTILINIT PHASE COMPLETE. ELAPSED TIME = 00:00:00

BMC50041I ORDER NO 0: ZIIP ENABLED (0) USING XBM SUBSYSTEM XBM

BMC51301I 1: SAMPLING STATISTICS: PART=0, TP=13, SP=11, SR=333, AVG=104, SD=65, SE=0, AVGF=3027, SD=64, SE=184, EP=0

BMC51265I ESTIMATED CARDINALITY OF SPACE = 333 AVG SORTWK ROW LENGTH = 104 AVG UNLOAD ROW LENGTH = 104

BMC51264I UNLOAD WILL READ 11 DATA PAGES FROM SPACE 'ARUDB250.TS250'

BMC51260I REORG PLUS DASD REQUIREMENT ESTIMATES

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>KBYTES</th>
<th>PRI</th>
<th>SEC</th>
<th>PRI</th>
<th>SEC</th>
<th>INDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSREC</td>
<td>37</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SYSUT101</td>
<td>24</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>ARU.TS25012</td>
</tr>
<tr>
<td>SYSUT102</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>ARU.TS25013</td>
</tr>
<tr>
<td>SYSUT103</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>ARU.TS25011</td>
</tr>
<tr>
<td>SYSUT1</td>
<td>72</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SORTWK (OPT)</td>
<td>72</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SORTWK (MIN)</td>
<td>24</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>BMCCPY</td>
<td>52</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>BMCCPYZ</td>
<td>52</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

BMC51260I REORG TABLESPACE ARUDB250.TS250

BMC5102I REORG CONTINUE

BMC5102I ORDER NO

BMC5102I ANALYZE ONLY

BMC50004I UTILINIT PHASE COMPLETE. ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP ENABLED (0) USING XBM SUBSYSTEM XBM

BMC51301I 1: SAMPLING STATISTICS: PART=0, TP=13, SP=11, SR=333, AVG=104, SD=65, SE=0, AVGF=3027, SD=64, SE=184, EP=0

BMC51265I ESTIMATED CARDINALITY OF SPACE = 333 AVG SORTWK ROW LENGTH = 104 AVG UNLOAD ROW LENGTH = 104

BMC51264I UNLOAD WILL READ 11 DATA PAGES FROM SPACE 'ARUDB250.TS250'

BMC51260I REORG PLUS DASD REQUIREMENT ESTIMATES

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>3380 CYLS</th>
<th>3390 CYLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNAME</td>
<td>PRI</td>
<td>SEC</td>
</tr>
<tr>
<td>SYSREC</td>
<td>37</td>
<td>1</td>
</tr>
<tr>
<td>SYSUT101</td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>SYSUT102</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>SYSUT103</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>SYSUT1</td>
<td>72</td>
<td>1</td>
</tr>
<tr>
<td>SORTWK (OPT)</td>
<td>72</td>
<td>1</td>
</tr>
<tr>
<td>SORTWK (MIN)</td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>BMCCPY</td>
<td>52</td>
<td>1</td>
</tr>
<tr>
<td>BMCCPYZ</td>
<td>52</td>
<td>1</td>
</tr>
</tbody>
</table>
Example 7: ANALYZE ONLY to generate space estimates

Chapter 5 Examples of REORG PLUS jobs

Figure 33 SYSPRINT for example 7, case 1 (part 6 of 6)

| BMC51263I | BMCRCY | 52 | 1 | 1 | 1 |
| BMC51263I | BMCRCZ | 52 | 1 | 1 | 1 |

BMC50041I: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XMB
BMC50041I: ANALYZE PHASE COMPLETE. ELAPSED TIME = 00:00:00
BMC50006I: UTILITY EXECUTION COMPLETE. RETURN CODE = 0

Figure 34 SYSPRINT for example 7, case 2 (part 1 of 5)

***** BMC REORG PLUS FOR DB2 VI1R1.00 *****
(C) COPYRIGHT 1988 - 2013 BMC SOFTWARE, INC.
REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50001I: UTILITY EXECUTION STARTING 10/14/2013 16:20:46 ...
BMC50002I: UTILITY ID = 'EXMPL07B'. DB2 SUBSYSTEM ID = 'DEFF'. OPTION MODULE = 'ARU$OPTS'.
BMC50471I: REGION=0M,PID=HBB7790,DFSMS FOR Z/OS=2.1.0
BMC50471I: DB2 UTILITIES COMMON CODE--V11.01.00
BMC50471I: SOLUTION COMMON CODE--V11.01.00
BMC50471I: UTILITY EXECUTION COMPLETE, RETURN CODE = 0
BMC50471I: UTILITY EXECUTION COMPLETE, RETURN CODE = 0

Figure 34 shows the SYSPRINT output for case 2 of example 7.

Figure 34 SYSPRINT for example 7, case 2 (part 1 of 5)
### Example 7: ANALYZE ONLY to generate space estimates

<table>
<thead>
<tr>
<th>BMC50471I Force_Rpt=NO</th>
<th>RmmMax=30</th>
<th>Workdom=SYS01</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471I Force=NONE</td>
<td>RmdMx5sz=2097152</td>
<td>Workunit=SYSALLDA</td>
</tr>
<tr>
<td>BMC50471I FsaLlback=NO</td>
<td>RmdMrk=1</td>
<td>Xbrd=</td>
</tr>
<tr>
<td>BMC50471I FstHshld=20</td>
<td>RmAPSMEM=0</td>
<td>Zfs=ENABLED</td>
</tr>
<tr>
<td>BMC50471I HsHaX=YES</td>
<td>RmmX=30</td>
<td>Workdom=SYS01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BMC50471I Plan-AruQa</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I DdType = UNLOAD</td>
</tr>
<tr>
<td>BMC50470I Active = YES</td>
</tr>
<tr>
<td>BMC50470I FalloC = USE</td>
</tr>
<tr>
<td>BMC50470I AlloC = N/A</td>
</tr>
<tr>
<td>BMC50470I Sms = ND</td>
</tr>
<tr>
<td>BMC50470I SmsUnit = ND</td>
</tr>
<tr>
<td>BMC50470I SziePcT = (100,100)</td>
</tr>
<tr>
<td>BMC50470I Unit = (SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I UnitCnt = (0,0)</td>
</tr>
<tr>
<td>BMC50470I VolCt = (25,25)</td>
</tr>
<tr>
<td>BMC50470I AvgVolSp = ((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I Dsntype = (NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I Dataclas = (NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I Mgmtclas = (NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I StoRclas = (NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I Threshld = 0</td>
</tr>
<tr>
<td>BMC50470I MaxExtsz = (0,K), (0,K)</td>
</tr>
<tr>
<td>BMC50470I Expot = N/A</td>
</tr>
<tr>
<td>BMC50470I Retpd = N/A</td>
</tr>
<tr>
<td>BMC50470I GdGEmptY = N/A</td>
</tr>
<tr>
<td>BMC50470I GdGScr = N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BMC50470I Plan-AruQa</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I DdType = ARCHIVE</td>
</tr>
<tr>
<td>BMC50470I Active = ND</td>
</tr>
<tr>
<td>BMC50470I FalloC = USE</td>
</tr>
<tr>
<td>BMC50470I AlloC = N/A</td>
</tr>
<tr>
<td>BMC50470I Sms = ND</td>
</tr>
<tr>
<td>BMC50470I SmsUnit = ND</td>
</tr>
<tr>
<td>BMC50470I SziePcT = (100,100)</td>
</tr>
<tr>
<td>BMC50470I Unit = (SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I UnitCnt = (0,0)</td>
</tr>
<tr>
<td>BMC50470I AvgVolSp = ((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I Dsntype = (NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I Dataclas = (NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I Mgmtclas = (NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I StoRclas = (NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I Threshld = 0</td>
</tr>
<tr>
<td>BMC50470I MaxExtsz = (0,K), (0,K)</td>
</tr>
<tr>
<td>BMC50470I Expot =</td>
</tr>
<tr>
<td>BMC50470I Retpd =</td>
</tr>
<tr>
<td>BMC50470I GdGEmptY = 5</td>
</tr>
<tr>
<td>BMC50470I GdGScr = ND</td>
</tr>
</tbody>
</table>

**Figure 34** SYSPRINT for example 7, case 2 (part 2 of 5)
### Example 7: ANALYZE ONLY to generate space estimates

#### Chapter 5 Examples of REORG PLUS jobs

<table>
<thead>
<tr>
<th>BMC50470I DDTYPE</th>
<th>REMPICPY</th>
<th>REMBFCPY</th>
<th>REMBICPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I ACTIVE</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I AVGVOLSP</td>
<td>((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>(0,K)</td>
<td>(0,K)</td>
<td>(0,K)</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I DDTYPE</td>
<td>SYSPUNCH</td>
<td>SYSPUNCH</td>
<td>SYSPUNCH</td>
</tr>
<tr>
<td>BMC50470I ACTIVE</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>BMC50470I IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I AVGVOLSP</td>
<td>((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>(0,K)</td>
<td>(0,K)</td>
<td>(0,K)</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I DDTYPE</td>
<td>LOCPLCPY</td>
<td>LOCBLCPY</td>
<td>LOCBLCPY</td>
</tr>
<tr>
<td>BMC50470I ACTIVE</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I AVGVOLSP</td>
<td>((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
<td>((30000, TRK),(30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>(0,K)</td>
<td>(0,K)</td>
<td>(0,K)</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I DDTYPE</td>
<td>REMPLCPY</td>
<td>REMBLCPY</td>
<td>REMBLCPY</td>
</tr>
<tr>
<td>BMC50470I ACTIVE</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
</tbody>
</table>

Figure 34 SYSPRINT for example 7, case 2 (part 3 of 5)
Example 7: ANALYZE ONLY to generate space estimates

<table>
<thead>
<tr>
<th>BMC50470I UNIT</th>
<th>(SYSALLDA,SYSALLDA)</th>
<th>(SYSALLDA,SYSALLDA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I VOCNT</td>
<td>(25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I ANGOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I DNSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50471I UNLOAD</td>
<td>DSNPAT=&amp;UID..BMC.&amp;TSIX.&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I WORK</td>
<td>DSNPAT=&amp;UID..BMC.&amp;TSIX.&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I SORTWORK</td>
<td>DSNPAT=&amp;UID.&amp;UTILPFX.&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I ARCHIVE</td>
<td>DSNPAT=&amp;UID.&amp;UTILPFX.&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I SYSFUNC</td>
<td>DSNPAT=&amp;UID.&amp;UTILPFX.&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I LOCFCPY</td>
<td>DSNPAT=&amp;UID.&amp;DONAME.&amp;ATSIX..FAPART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I LOCPICPY</td>
<td>DSNPAT=&amp;UID.&amp;DONAME.&amp;ATSIX..FAPART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I LOCBFCPY</td>
<td>DSNPAT=&amp;UID.&amp;DONAME.&amp;ATSIX..FAPART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I LOCPICPY</td>
<td>DSNPAT=&amp;UID.&amp;DONAME.&amp;ATSIX..FAPART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I LOCBICPY</td>
<td>DSNPAT=&amp;UID.&amp;DONAME..ATSIX..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I REMFCPY</td>
<td>DSNPAT=&amp;UID.&amp;DONAME..ATSIX..FAPART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I REMICPY</td>
<td>DSNPAT=&amp;UID.&amp;DONAME..ATSIX..FAPART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I REMBFCPY</td>
<td>DSNPAT=&amp;UID.&amp;DONAME..ATSIX..FAPART..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I RELICPY</td>
<td>DSNPAT=&amp;UID.&amp;DONAME..ATSIX..T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I DB2 DSNHECPS MODULE SETTINGS:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50471I VERSION</td>
<td>1010</td>
<td></td>
</tr>
<tr>
<td>BMC50471I SUBSYSTEM DEFAULT</td>
<td>DEF</td>
<td></td>
</tr>
<tr>
<td>BMC50471I CHARACTER SET</td>
<td>ALPHANUM</td>
<td></td>
</tr>
<tr>
<td>BMC50471I DATE FORMAT</td>
<td>USA</td>
<td></td>
</tr>
<tr>
<td>BMC50471I TIME FORMAT</td>
<td>USA</td>
<td></td>
</tr>
<tr>
<td>BMC50471I LOCAL DATE LENGTH</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>BMC50471I DECIMAL POINT</td>
<td>PERIOD</td>
<td></td>
</tr>
<tr>
<td>BMC50471I DECIMAL ARITHMETIC</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>BMC50471I DELIMITER</td>
<td>DEFAULT</td>
<td></td>
</tr>
<tr>
<td>BMC50471I SQL DELIMITER</td>
<td>DEFAULT</td>
<td></td>
</tr>
<tr>
<td>BMC50471I APPL. ENCODING SCHEME</td>
<td>EBCDIC</td>
<td></td>
</tr>
<tr>
<td>BMC50471I IMPLICIT TIME ZONE</td>
<td>CURRENT (-05:00)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 34  SYSPRINT for example 7, case 2 (part 4 of 5)
Example 8: Selective unload with discards to archive data set

In this example, REORG PLUS reorganizes a segmented table space containing only one table. The table has one nonpartitioned secondary index.

Although the installation options enable dynamic allocation for the unload, work, and primary copy data sets, they also specify IFALLOC USE for these data sets. Therefore, the job does not have to specify ACTIVE NO to have REORG PLUS use the data sets allocated in the JCL.

In the SYSPRINT output, message BMC50477I (issued during the UNLOAD phase) indicates the total number of rows in the table space before reorganization. Message BMC51272I (issued at the end of the UNLOAD phase) indicates the number of rows selected to be unloaded, number of rows discarded, and number of rows updated.
Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 72 describes the key command options and DD statements for this job.

<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>UNLOAD CONTINUE</td>
<td>tells REORG PLUS to continue the reorganization with two-phase processing after the UNLOAD phase has unloaded the data</td>
</tr>
<tr>
<td>COPY YES</td>
<td>creates a DB2 image copy of the table space</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS makes four copies of the table space after reorganizing it.</td>
</tr>
<tr>
<td>REGISTER (ARUCPY1, ARUCPY3)</td>
<td>specifies that only two of the four copies, ARUCPY1 and ARUCPY3, are to be registered in SYSIBM.SYSCOPY</td>
</tr>
<tr>
<td>COPYDDN (ARUCPY1, ARUCPY2)</td>
<td>specifies ddnames of ARUCPY1 and ARUCPY2 for the two local data sets that receive full local image copies of the table space that you are reorganizing</td>
</tr>
<tr>
<td></td>
<td>overrides the default names of the BMCCPY and BMCCPZ</td>
</tr>
<tr>
<td>RECOVERYDDN (ARUCPY3, ARUCPY4)</td>
<td>specifies ddnames of ARUCPY3 and ARUCPY4 for the two data sets that receive full remote image copies of the table space that you are reorganizing</td>
</tr>
<tr>
<td></td>
<td>overrides the default names of BMCRCY and BMCRCZ</td>
</tr>
<tr>
<td>ARCHDDN (ARUARC)</td>
<td>specifies a ddname of ARUARC for the archive data set</td>
</tr>
<tr>
<td></td>
<td>overrides the default name of SYSARC</td>
</tr>
<tr>
<td>REDEFINE NO</td>
<td>tells REORG PLUS not to delete and redefine the VSAM data sets containing the DB2 objects</td>
</tr>
<tr>
<td></td>
<td>Instead, REORG PLUS will reset the HURBA.</td>
</tr>
<tr>
<td>ORDER NO</td>
<td>performs no ordering at all, and the rows retain the order of the table before reorganization</td>
</tr>
<tr>
<td>SELECT * FROM</td>
<td>specifies the table name that contains the rows and that only rows that meet the specified WHERE clause are to be unloaded and reloaded</td>
</tr>
<tr>
<td>WHERE</td>
<td>specifies the conditions that must be true for rows to be unloaded and reloaded</td>
</tr>
<tr>
<td>//SYSREC</td>
<td>contains the rows that you are reorganizing</td>
</tr>
</tbody>
</table>
Example 8: Selective unload with discards to archive data set

Chapter 5 Examples of REORG PLUS jobs

Table 72 Command options and DD statements used in example 8 (part 2 of 2)

<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//ARUCPY1</td>
<td>ddnames that are used for the data sets that receive full image copies of the table space that you are reorganizing</td>
</tr>
<tr>
<td>//ARUCPY2</td>
<td>ddname that is used for the data set that will contain the rows not selected</td>
</tr>
<tr>
<td>//ARUCPY3</td>
<td>the SYSUT1 data set to be allocated for the nonpartitioned secondary index</td>
</tr>
<tr>
<td>//SYSUT1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 35 shows the JCL for example 8.

**Figure 35** JCL for example 8 (part 1 of 2)

// ** JOB
/*
** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
** SEGMENTED TABLESPACE, ONE TABLE, ONE INDEX *
** REORG TABLESPACE RETAINING ONLY SELECTED ROWS *
** ALL DELETED ROWS GO TO THE ARCHIVE DATASET *
** MAKE 4 COPIES (TWO LOCALS, TWO REMOTES) *
** REGISTER ONLY THE PRIMARY COPIES. *
** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
// PARM='DEFF,EXMPL08,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB DD DISP=SHR,DSN=ProductLibraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//UTPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//SYSTEM DD SYSOUT=*  
//*
//SYSREC DD DSN=ARU.EXMPL08.SYSREC,
// DISP=(MOD,CATLG,CATLG),
// UNIT=WORK,SPACE=(CYL,(30,10))
//*
//ARUCPY1 DD DSN=ARU.EXMPL08.ARUCPY1,
// DISP=(,CATLG),
// SPACE=(CYL,(1,1),RLSE),UNIT=WORK
//ARUCPY2 DD DSN=ARU.EXMPL08.ARUCPY2,
// DISP=(,CATLG),
// SPACE=(CYL,(1,1),RLSE),UNIT=WORK
//ARUCPY3 DD DSN=ARU.EXMPL08.ARUCPY3,
// DISP=(,CATLG),
// SPACE=(CYL,(1,1),RLSE),UNIT=WORK
Example 8: Selective unload with discards to archive data set

Figure 36 shows the SYSPRINT output for example 8.
Example 8: Selective unload with discards to archive data set

Figure 36  SYSPRINT for example 8 (part 2 of 6)

<table>
<thead>
<tr>
<th>BMC50471</th>
<th>DSMXIX1=(NONE,NONE)</th>
<th>PLNUM001=YES</th>
<th>UNLOADPRELOAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471</td>
<td>EXCLUDOM=(X37,X22,X06)</td>
<td>PREFORMAT=NO</td>
<td>UTILP_COLCSD=UTILB</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FASTSWITCH=NO</td>
<td>RCVIDDDM=(BMCRY,BMCIRZ)</td>
<td>UTILP_MULTIX=UTILB</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FILEX=WAR</td>
<td>RCVIDDDM=(BMCRY,BMCIRZ)</td>
<td>UTSTATE=SUP</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FORCE_AT=(START,3)</td>
<td>REMPLDDM=(BMCRY,BMCIRZ)</td>
<td>WBUFSS=(20,10)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FORCE_RPT=NO</td>
<td>REMPLDDM=(2097152)</td>
<td>WORKDDM=SYSPU1</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FORCE=NONE</td>
<td>REMPLDDM=(12)</td>
<td>WORKUNIT=SYSPALLDA</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FSSFALLBACK=NO</td>
<td>REMPLDDM=(12)</td>
<td>XBLD=(1)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FSFERROR=0</td>
<td>REMPLDDM=(0)</td>
<td>ZILP=ENABLED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BMC50471</th>
<th>PLAN=ARUQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471</td>
<td>ODTYPE = UNLOAD</td>
</tr>
<tr>
<td>BMC50471</td>
<td>ACTIVE = YES</td>
</tr>
<tr>
<td>BMC50471</td>
<td>IFALLOC = USE</td>
</tr>
<tr>
<td>BMC50471</td>
<td>ALLOC = N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>SMS = NO</td>
</tr>
<tr>
<td>BMC50471</td>
<td>SMSNIT = NO</td>
</tr>
<tr>
<td>BMC50471</td>
<td>SIZEPCT = (100,100)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNIT = (SYSALLDA,SYSSALLDA)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNITCNT = (0,0)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>VOLCNT = (25,25)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = ((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DSNNAME = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DATACLAS = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MGMTCLASS = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = (100,100)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNIT = (SYSALLDA,SYSSALLDA)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNITCNT = (0,0)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = ((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DSNNAME = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DATACLAS = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MGMTCLASS = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = (100,100)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNIT = (SYSALLDA,SYSSALLDA)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNITCNT = (0,0)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = ((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DSNNAME = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DATACLAS = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MGMTCLASS = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = (100,100)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNIT = (SYSALLDA,SYSSALLDA)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNITCNT = (0,0)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = ((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DSNNAME = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DATACLAS = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MGMTCLASS = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = (100,100)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNIT = (SYSALLDA,SYSSALLDA)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNITCNT = (0,0)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = ((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DSNNAME = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DATACLAS = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MGMTCLASS = (NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = (100,100)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNIT = (SYSALLDA,SYSSALLDA)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNITCNT = (0,0)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGWOLSP = ((30000,TRK),(30000,TRK))</td>
</tr>
</tbody>
</table>

Chapter 5  Examples of REORG PLUS jobs 495
### Example 8: Selective unload with discards to archive data set

#### SYSPRINT for example 8 (part 3 of 6)

```plaintext
| BMS04701 | EXPDT     | -        |
| BMS04701 | RETPD     | -        |
| BMS04701 | GDGLIMIT  | = 5      |
| BMS04701 | GDGEMPTY  | = NO     |
| BMS04701 | GDGSCRAT  | = NO     |

| BMS04701 | DDTYPE    | = REMPICPY |
| BMS04701 | ACTIVE    | = NO       |
| BMS04701 | IFALLOC   | = USE      |
| BMS04701 | ALLOC     | = N/A      |
| BMS04701 | SMS       | = NO       |
| BMS04701 | SMSUNIT   | = NO       |
| BMS04701 | SIZEPCT   | = (100,100) |
| BMS04701 | UNIT      | = SYSSALLDA,SYSSALLDA |
| BMS04701 | UNITCNT   | = (0,0)    |
| BMS04701 | VOLCNT    | = (25,25)  |
| BMS04701 | AVGVOLSP  | = ((30000,TRK),(30000,TRK)) |
| BMS04701 | DATACLAS  | = (NONE,NONE) |
| BMS04701 | MGMTCLAS  | = (NONE,NONE) |
| BMS04701 | STORCLAS  | = (NONE,NONE) |
| BMS04701 | THRESHLD  | = 0        |
| BMS04701 | MAXEXTSZ  | = (0,K),(0,K) |
| BMS04701 | EXPDT     | =          |
| BMS04701 | RETPD     | =          |
| BMS04701 | GDGLIMIT  | = 5        |
| BMS04701 | GDGEMPTY  | = NO       |
| BMS04701 | GDGSCRAT  | = NO       |

| BMS04701 | DDTYPE    | = SYSPUNCH |
| BMS04701 | ACTIVE    | = YES      |
| BMS04701 | IFALLOC   | = USE      |
| BMS04701 | ALLOC     | = N/A      |
| BMS04701 | SMS       | = NO       |
| BMS04701 | SMSUNIT   | = NO       |
| BMS04701 | SIZEPCT   | = (100,100) |
| BMS04701 | UNIT      | = SYSSALLDA,SYSSALLDA |
| BMS04701 | UNITCNT   | = (0,0)    |
| BMS04701 | VOLCNT    | = (25,25)  |
| BMS04701 | AVGVOLSP  | = ((30000,TRK),(30000,TRK)) |
| BMS04701 | DATACLAS  | = (NONE,NONE) |
| BMS04701 | MGMTCLAS  | = (NONE,NONE) |
| BMS04701 | STORCLAS  | = (NONE,NONE) |
| BMS04701 | THRESHLD  | = 0        |
| BMS04701 | MAXEXTSZ  | = (0,K),(0,K) |
| BMS04701 | EXPDT     | =          |
| BMS04701 | RETPD     | =          |
| BMS04701 | GDGLIMIT  | = 0        |
| BMS04701 | GDGEMPTY  | = NO       |
| BMS04701 | GDGSCRAT  | = NO       |

| BMS04701 | DDTYPE    | = LDCPLCPY |
| BMS04701 | ACTIVE    | = NO       |
| BMS04701 | IFALLOC   | = USE      |
| BMS04701 | ALLOC     | = N/A      |
| BMS04701 | SMS       | = NO       |
| BMS04701 | SMSUNIT   | = NO       |
| BMS04701 | SIZEPCT   | = (100,100) |
| BMS04701 | UNIT      | = SYSSALLDA,SYSSALLDA |
| BMS04701 | UNITCNT   | = (0,0)    |
| BMS04701 | VOLCNT    | = (25,25)  |
| BMS04701 | AVGVOLSP  | = ((30000,TRK),(30000,TRK)) |
| BMS04701 | DATACLAS  | = (NONE,NONE) |
| BMS04701 | MGMTCLAS  | = (NONE,NONE) |
| BMS04701 | STORCLAS  | = (NONE,NONE) |
| BMS04701 | THRESHLD  | = 0        |
| BMS04701 | MAXEXTSZ  | = (0,K),(0,K) |
| BMS04701 | EXPDT     | =          |
| BMS04701 | RETPD     | =          |
| BMS04701 | GDGLIMIT  | = 0        |
| BMS04701 | GDGEMPTY  | = NO       |
| BMS04701 | GDGSCRAT  | = NO       |
```

---

**Figure 36** SYSPRINT for example 8 (part 3 of 6)

496 **REORG PLUS for DB2 Reference Manual**
Example 8: Selective unload with discards to archive data set

### Example 8 (part 4 of 6)

<table>
<thead>
<tr>
<th>BMC50470I IFALLOC</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I ALLOC</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I SMS</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SMSUNIT</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I UNICNT</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I THRESHLD</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td></td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td></td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470I GDGEMPTY</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I GDGSCRAT</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50471I UNLOAD</td>
<td>DSNPAT=&amp;UID..BMC.&amp;TSIX..ADDNAME</td>
</tr>
<tr>
<td>BMC50483I WORK</td>
<td>DSNPAT=&amp;UID..BMC.&amp;TSIX..ADDNAME</td>
</tr>
<tr>
<td>BMC50483I SORTWORK</td>
<td>DSNPAT=&amp;UID..UTILPFX..ADDNAME</td>
</tr>
<tr>
<td>BMC50483I ARCHIVE</td>
<td>DSNPAT=&amp;UID..UTILPFX..ADDNAME</td>
</tr>
<tr>
<td>BMC50483I SYSPPUNCH</td>
<td>DSNPAT=&amp;UID..UTILPFX..ADDNAME</td>
</tr>
<tr>
<td>BMC50483I LOCPFCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I LOGICPCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I LOGBPFCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I LOGBIPCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I LOGCPCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I LOGBCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I LOGPBCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I LOGPCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I REMPFCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I REMPICPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I REMBFCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I REMPLCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50483I REMBLCPY</td>
<td>DSNPAT=&amp;UID..ADDNAME..&amp;TSIX..&amp;PART..&amp;TIME</td>
</tr>
<tr>
<td>BMC50471I DB2 DSNHDECP MODULE SETTINGS:</td>
<td></td>
</tr>
<tr>
<td>BMC50471I VERSION</td>
<td>1010</td>
</tr>
<tr>
<td>BMC50471I SUBSYSTEM DEFAULT</td>
<td>'DEFF'</td>
</tr>
<tr>
<td>BMC50471I CHARACTER SET</td>
<td>'ALPHANUM'</td>
</tr>
<tr>
<td>BMC50471I DATE FORMAT</td>
<td>'USA'</td>
</tr>
<tr>
<td>BMC50471I TIME FORMAT</td>
<td>'USA'</td>
</tr>
<tr>
<td>BMC50471I LOCAL DATE LENGTH</td>
<td>'0'</td>
</tr>
<tr>
<td>BMC50471I LOCAL TIME LENGTH</td>
<td>'0'</td>
</tr>
<tr>
<td>BMC50471I DECIMAL POINT</td>
<td>'PERIOD'</td>
</tr>
<tr>
<td>BMC50471I DECIMAL ARITHMETIC</td>
<td>'15'</td>
</tr>
<tr>
<td>BMC50471I DELIMITER</td>
<td>'DEFAULT'</td>
</tr>
<tr>
<td>BMC50471I SQL DELIMITER</td>
<td>'DEFAULT'</td>
</tr>
<tr>
<td>BMC50471I ENCODING SCHEME</td>
<td>'EBCDIC'</td>
</tr>
<tr>
<td>BMC50471I APPL ENCODING SCHEME</td>
<td>'EBCDIC'</td>
</tr>
<tr>
<td>BMC50471I MIXED</td>
<td>'NO'</td>
</tr>
<tr>
<td>BMC50471I EBCDIC CCSID</td>
<td>(37,65534,65534)</td>
</tr>
<tr>
<td>BMC50471I ASCII CCSID</td>
<td>(319,65534,65534)</td>
</tr>
<tr>
<td>BMC50471I UNICODE CCSID</td>
<td>(367,1208,1200)</td>
</tr>
<tr>
<td>BMC50471I IMPLICIT TIME ZONE</td>
<td>'CURRENT (-05:00)'</td>
</tr>
<tr>
<td>BMC50028I DB2 MODE</td>
<td>'NFM'</td>
</tr>
<tr>
<td>BMC50471I BMCUTIL</td>
<td>'BMCUTIL.CMN_BMCUTIL'</td>
</tr>
<tr>
<td>BMC50471I BMCSYNC</td>
<td>'BMCUTIL.CMN_BMCSYNC'</td>
</tr>
<tr>
<td>BMC50471I BMCHIST</td>
<td>'BMCUTIL.CMN_BMCHIST'</td>
</tr>
<tr>
<td>BMC50471I BMCDICT</td>
<td>'BMCUTIL.CMN_BMCDICT'</td>
</tr>
<tr>
<td>BMC50471I BMCCOPY</td>
<td>'BMCUTIL.CMN_BMCCOPY'</td>
</tr>
<tr>
<td>BMC50471I DASD MANAGER PLUS TABLES:</td>
<td></td>
</tr>
<tr>
<td>BMC50471I &quot;ATS102_RS_COLDIST&quot;</td>
<td>&quot;ATS102_RS_COLDIST&quot;</td>
</tr>
<tr>
<td>BMC50471I &quot;ATS102_RS_COLDIST&quot;</td>
<td>&quot;ATS102_RS_COLDIST&quot;</td>
</tr>
<tr>
<td>BMC50471I &quot;ATS102_RS_COLDIST&quot;</td>
<td>&quot;ATS102_RS_COLDIST&quot;</td>
</tr>
<tr>
<td>BMC50471I &quot;ATS102_RS_INDEXES&quot;</td>
<td>&quot;ATS102_RS_INDEXES&quot;</td>
</tr>
<tr>
<td>BMC50471I &quot;ATS102_RS_INDEXES&quot;</td>
<td>&quot;ATS102_RS_INDEXES&quot;</td>
</tr>
<tr>
<td>BMC50471I &quot;ATS102_RS_INDEXES&quot;</td>
<td>&quot;ATS102_RS_INDEXES&quot;</td>
</tr>
<tr>
<td>BMC50471I &quot;ATS102_RS_TABLEPART&quot;</td>
<td>&quot;ATS102_RS_TABLEPART&quot;</td>
</tr>
<tr>
<td>BMC50471I &quot;ATS102_RS_TABLEPART&quot;</td>
<td>&quot;ATS102_RS_TABLEPART&quot;</td>
</tr>
</tbody>
</table>

Figure 36 SYSPRINT for example 8 (part 4 of 6)
Example 8: Selective unload with discards to archive data set

Figure 36  SYSPRINT for example 8 (part 5 of 6)
Example 9: ON FAILURE with a user-specified return code

This example shows a full table space reorganization of a segmented table space that contains multiple tables. The job fails with a space failure (abend D37) because the SYSREC data set has insufficient space. The reorganization terminates with return code 31 as specified with the ON FAILURE TERMINATE option. The job is not restartable because of the ON FAILURE TERMINATE option, but it can be resubmitted from the beginning at a later time. For more information about the ON FAILURE option, see page 224.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 73 describes the key command options and DD statements for this job.

Table 73  Key command options and DD statements used in example 9 (part 1 of 2)

<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>COPY YES</td>
<td>tells REORG PLUS to create a DB2 image copy of the table space</td>
</tr>
<tr>
<td>REDEFINE YES</td>
<td>tells REORG PLUS to delete and redefine the VSAM data sets containing the DB2 objects</td>
</tr>
<tr>
<td>ORDER YES</td>
<td>tells REORG PLUS to sort the rows during the reorganization</td>
</tr>
<tr>
<td>BYTABLE is the default if you specify ORDER YES. It tells REORG PLUS to sort the rows by table as well as by each table’s clustering key. If no clustering key exists, X’00’ are used.</td>
<td></td>
</tr>
<tr>
<td>SORTDATA</td>
<td>enables this job to be compatible if run with the IBM DB2 REORG utility</td>
</tr>
</tbody>
</table>
**Example 9: ON FAILURE with a user-specified return code**

500

REORG PLUS for DB2 Reference Manual

Figure 37 shows the JCL for example 9.

**Figure 37** JCL for example 9 (part 1 of 2)

```plaintext
//        JOB
//*
//*  SEGMENTED TABLESPACE, MULTIPLE TABLES,
//*  REORG ENTIRE TABLESPACE, ALLOCATE ONLY ONE TRACK FOR THE SYSREC
//*  TO FORCE REORG ABEND W/ D37 DURING THE UNLOAD PHASE.
//*  ON FAILURE TESTING
//*     ON FAILURE UNLOAD TERMINATE UTILITY RETCODE 31
//*     NOTE: WITH ON FAILURE UNLOAD TERMINATE UTILITY RETCODE
//*     SPECIFIED IN REORG, REORG STARTED ALL TABLESPACES
//*     AND TERMED THE UTILID WITH A RETURN CODE 31.
//*     IN THIS CASE, RESTART IS NOT ALLOWED.
//BMCREORG EXEC PGM=ARUUMAIN,
//      PARM='DEFF,EXMPL09,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD  DISP=SHR,DSN=ProductLibraries
//          DD  DISP=SHR,DSN=DB2.DSNEXIT
//          DD  DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT DD  SYSOUT=*  
//SYSOUT DD  SYSOUT=*  
//UTPRINT DD  SYSOUT=*  
//SYSUDUMP DD  SYSOUT=*  
//SYSTERM  DD  SYSOUT=*  
```

| **Table 73** Key command options and DD statements used in example 9  (part 2 of 2) |
|------------------------------|----------------------------------------------------------------------------------|
| **Command options and DD statements used in JCL** | **Description** |
| ON FAILURE | when the job abends in the UNLOAD phase, tells REORG PLUS to start the table space and index space and delete the row containing the utility ID from the BMCUTIL table. |
| UNLOAD TERMINATE | The job is not restartable but can be resubmitted from the beginning at a later time. In addition, return code 31 is issued, as specified. |
| UTILITY RETCODE 31 | activates dynamic allocation for the local backup copy and the remote copies, overriding the installation defaults. |
| DSNPAT | specifies data set names for dynamic allocation that override the default patterns. |
| //SYSREC | contains the rows that you are reorganizing. |
| | The SYSREC data set is allocated with only one track (SPACE=(TRK,(1))) causing the job to abend during the UNLOAD phase. |

The SYSREC data set is allocated with only one track (SPACE=(TRK,(1))) causing the job to abend during the UNLOAD phase.
Example 9: ON FAILURE with a user-specified return code

Figure 37  JCL for example 9 (part 2 of 2)

```bash
//X37IGN    DD  DUMMY
//SYSREC    DD  DSN=ARU.EXMPL09.SYSREC,
 //DISP=(MOD,CATLG,CATLG),
 //UNIT=3390,SPACE=(TRK,(1,0))
//SYSIN    DD  *
REORG TABLESPACE ARUDB006.TS006
COPY YES
ORDER YES
SORTDATA
ON FAILURE UNLOAD TERMINATE UTILITY RETCODE 31
DDTYPE UNLOAD ACTIVE NO
DDTYPE WORK
   DSNUMAT 'ARU.EXMPL09.&DDNAME'
DDTYPE LOCPFCPY
   DSNUMAT 'ARU.EXMPL09.&DDNAME'
DDTYPE REMPFCPY ACTIVE YES
   DSNUMAT 'ARU.EXMPL09.&DDNAME'
DDTYPE REMBFCPY ACTIVE YES
   DSNUMAT 'ARU.EXMPL09.&DDNAME'
END
```

Figure 38  SYSPRINT for example 9 (part 1 of 6)

```
***** BMC REORG PLUS FOR DB2 V11R1.00 *****
(C) COPYRIGHT 1988 - 2013 BMC SOFTWARE, INC.
REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE
OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50001I UTILITY EXECUTION STARTING   10/14/2013   16:22:40 ...
BMC50002I UTILITY ID = 'EXMPL09'.  DB2 SUBSYSTEM ID = 'DEFF'.  OPTION MODULE = 'ARU$OPTS'.
BMC50471I z/OS 2.1.0,PID=HBB7790,DFSMS FOR Z/OS=2.1.0,DB2=10.1.0
BMC50471I REGION=0M,BELOW 16M=8832K,ABOVE 16M=1408944K,IEFUSI=NO,CPUS=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040304M,MEMLIMIT SET BY:REGION=0
BMC50471I REORG PLUS FOR DB2--V11.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMS UTILITIES COMMON CODE--V11.01.00
BMC50471I BMCSORT ENGINE--V02.04.01
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V11.01.00
BMC50471I MAINT: BPJ0661 BPJ0667 BPJ0671 BPJ0675 BPJ0682 BPJ0686 BPJ0689 BPJ0697
BMC50471I BMCSORT ENGINE--V02.04.01
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V11.01.00
BMC50471I MAINT: BPJ0661 BPJ0667 BPJ0671 BPJ0675 BPJ0682 BPJ0686 BPJ0689 BPJ0697
BMC50471I BMCSORT ENGINE--V02.04.01
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I EXTENDED BUFFER MANAGER--V06.01.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I ALTRFAIL=RCVRPEND ICOW=(BMCICY,BMCICZ) RORGMAX=300%
BMC50471I ANALMAX=10000 ICTYPE=AUTO ROUTCDE=(11,1)
BMC50471I ARC=NO IDECACHE=10000 SCPYM/B
BMC50471I ARCHDDN=SYSARC INDEFLM=10 SDUMP=NO
BMC50471I ARCHFORMAT=BMC INLNECP=YES SHORTMEMORY=CONTINUE
BMC50471I AVAILPAGEPCT=100 IKNEX=NO SIXSNAP=NO
BMC50471I BLMAX=300% 1KRANDOM=NO SMAX=0
BMC50471I BMCHST=YES KEEPDICTIONARY=NO SMORE=(0K,0K)
BMC50471I CBUFFS=30 LEAFSLM=200 SORTDEV=(,SYSAIDDA)
BMC50471I CONEXEC=NO LGB=YES SORTNUM=32
```
Example 9: ON FAILURE with a user-specified return code

Figure 38  SYSPRINT for example 9 (part 2 of 6)
### Example 9: ON FAILURE with a user-specified return code

#### Chapter 5 Examples of REORG PLUS jobs

<table>
<thead>
<tr>
<th>BMC50470</th>
<th>DDTYPE</th>
<th>LOCBLCPY</th>
<th>LOCBLCPY</th>
<th>REMPFCPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470</td>
<td>ACTIVE</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470</td>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470</td>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470</td>
<td>UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470</td>
<td>UNICNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470</td>
<td>AVGVOLS</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470</td>
<td>DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470</td>
<td>MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470</td>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470</td>
<td>MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>BMC50470</td>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470</td>
<td>RETPD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470</td>
<td>GDGLIMIT</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470</td>
<td>GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>GDSSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BMC50470</th>
<th>DDTYPE</th>
<th>REMPFCPY</th>
<th>REMBFCPY</th>
<th>REMPFCPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470</td>
<td>ACTIVE</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470</td>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470</td>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470</td>
<td>UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470</td>
<td>UNICNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470</td>
<td>AVGVOLS</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470</td>
<td>DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470</td>
<td>MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470</td>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470</td>
<td>MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>BMC50470</td>
<td>EXPDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470</td>
<td>RETPD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470</td>
<td>GDGLIMIT</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470</td>
<td>GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>GDSSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BMC50470</th>
<th>DDTYPE</th>
<th>LOCBLCPY</th>
<th>LOCBLCPY</th>
<th>REMPFCPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470</td>
<td>ACTIVE</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470</td>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470</td>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470</td>
<td>SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
</tbody>
</table>

---

**Figure 38**  
SYSPRINT for example 9 (part 3 of 6)
Example 9: ON FAILURE with a user-specified return code

| BMC50470I UNIT     | (SYSALLDA, SYSALLDA) | (SYSALLDA, SYSALLDA) |
| BMC50470I UNITCNT  | (0, 0)               | (0, 0)               |
| BMC50470I VOLLNT   | (25, 25)             | (25, 25)             |
| BMC50470I AVGOLSP  | ((30000, TRK), (30000, TRK)) | ((30000, TRK), (30000, TRK)) |
| BMC50470I DSNTYPE  | (NONE, NONE)         | (NONE, NONE)         |
| BMC50470I DATACLAS | (NONE, NONE)         | (NONE, NONE)         |
| BMC50470I MGMTCLAS | (NONE, NONE)         | (NONE, NONE)         |
| BMC50470I STORCLAS | (NONE, NONE)         | (NONE, NONE)         |
| BMC50470I THRESHLD | 0                    | 0                    |
| BMC50470I MAXEXTSZ | ((0, K), (0, K))     | ((0, K), (0, K))     |
| BMC50470I EXPDT    |                       |                       |
| BMC50470I RETPD    |                       |                       |
| BMC50470I GDGLIMIT | 5                    | 5                    |
| BMC50470I GDGEMPTY | NO                   | NO                   |
| BMC50470I GDGSCRAT | NO                   | NO                   |
| BMC50470I DTYPE    | REPLCPY              | REPLCPY              |
| BMC50470I ACTIVE   | NO                   | NO                   |
| BMC50470I IFALLOC  | USE                  | USE                  |
| BMC50470I ALLOC    | N/A                  | N/A                  |
| BMC50470I SMS      | NO                   | NO                   |
| BMC50470I OMUNIT   | NO                   | NO                   |
| BMC50470I SIZEPCT  | (100, 100)           | (100, 100)           |
| BMC50470I UNIT     | (SYSALLDA, SYSALLDA) | (SYSALLDA, SYSALLDA) |
| BMC50470I UNITCNT  | (0, 0)               | (0, 0)               |
| BMC50470I VOLLNT   | (25, 25)             | (25, 25)             |
| BMC50470I AVGOLSP  | ((30000, TRK), (30000, TRK)) | ((30000, TRK), (30000, TRK)) |
| BMC50470I DSNTYPE  | (NONE, NONE)         | (NONE, NONE)         |
| BMC50470I DATACLAS | (NONE, NONE)         | (NONE, NONE)         |
| BMC50470I MGMTCLAS | (NONE, NONE)         | (NONE, NONE)         |
| BMC50470I STORCLAS | (NONE, NONE)         | (NONE, NONE)         |
| BMC50470I THRESHLD | 0                    | 0                    |
| BMC50470I MAXEXTSZ | ((0, K), (0, K))     | ((0, K), (0, K))     |
| BMC50470I EXPDT    |                       |                       |
| BMC50470I RETPD    |                       |                       |
| BMC50470I GDGLIMIT | 5                    | 5                    |
| BMC50470I GDGEMPTY | NO                   | NO                   |
| BMC50470I GDGSCRAT | NO                   | NO                   |
| BMC50483I UNLOAD   | DSNPAT=UID..BMC..ATSIX..ADDNAME | DSNPAT=UID..BMC..ATSIX..ADDNAME |
| BMC50483I WORK     | DSNPAT=UID..BMC..ATSIX..ADDNAME | DSNPAT=UID..BMC..ATSIX..ADDNAME |
| BMC50483I SORTWORK | DSNPAT=UID..UTILPFX..ADDNAME | DSNPAT=UID..UTILPFX..ADDNAME |
| BMC50483I ARCHIVE  | DSNPAT=UID..UTILPFX..ADDNAME | DSNPAT=UID..UTILPFX..ADDNAME |
| BMC50483I SYSPUNCH | DSNPAT=UID..UTILPFX..ADDNAME | DSNPAT=UID..UTILPFX..ADDNAME |
| BMC50483I LOCPFCPY | DSNPAT=UID..ATSIX..FAPART..T&TTIME | DSNPAT=UID..ATSIX..FAPART..T&TTIME |
| BMC50483I LOCPICPY | DSNPAT=UID..ATSIX..FAPART..T&TTIME | DSNPAT=UID..ATSIX..FAPART..T&TTIME |
| BMC50483I LOCBFCPY | DSNPAT=UID..ATSIX..FAPART..T&TTIME | DSNPAT=UID..ATSIX..FAPART..T&TTIME |
| BMC50483I LOCBICPY | DSNPAT=UID..ATSIX..FAPART..T&TTIME | DSNPAT=UID..ATSIX..FAPART..T&TTIME |
| BMC50483I LOCPLCPY | DSNPAT=UID..ATSIX..T&TTIME | DSNPAT=UID..ATSIX..T&TTIME |
| BMC50483I LOCBLCPY | DSNPAT=UID..ATSIX..T&TTIME | DSNPAT=UID..ATSIX..T&TTIME |
| BMC50483I REMPPFCPY | DSNPAT=UID..ATSIX..FAPART..T&TTIME | DSNPAT=UID..ATSIX..FAPART..T&TTIME |
| BMC50483I REMPPICPY | DSNPAT=UID..ATSIX..FAPART..T&TTIME | DSNPAT=UID..ATSIX..FAPART..T&TTIME |
| BMC50483I REMBBFCPY | DSNPAT=UID..ATSIX..FAPART..T&TTIME | DSNPAT=UID..ATSIX..FAPART..T&TTIME |
| BMC50483I REMBBICPY | DSNPAT=UID..ATSIX..FAPART..T&TTIME | DSNPAT=UID..ATSIX..FAPART..T&TTIME |
| BMC50483I REMBLCPY | DSNPAT=UID..ATSIX..T&TTIME | DSNPAT=UID..ATSIX..T&TTIME |
| BMC50483I REMBLICPY | DSNPAT=UID..ATSIX..T&TTIME | DSNPAT=UID..ATSIX..T&TTIME |

**Figure 38**  SYSPRINT for example 9 (part 4 of 6)
Example 9: ON FAILURE with a user-specified return code

Chapter 5 Examples of REORG PLUS jobs

BMC50471I IMPLICIT TIME ZONE = CURRENT (-05:00)
BMC50028I DB2 MODE = NFM
BMC50471I BMCSYNC = BMCSYNC
BMC50471I BMCHIST = BMCHIST
BMC50471I BMCDICT = BMCDICT
BMC50471I BMKCOPY = BMKCOPY
BMC50471I BMCMANAGER PLUS TABLES:
BMC50471I ...'ATS102.RS_COLDIST'
BMC50471I ...'ATS102.RS_COLSTATS'
BMC50471I ...'ATS102.RS_COLUMNS'
BMC50471I ...'ATS102.RS_INDEXES'
BMC50471I ...'ATS102.RS_INDEXPART'
BMC50471I ...'ATS102.RS_IXPART_DIST'
BMC50471I ...'ATS102.RS_TABLEPART'
BMC50471I ...'ATS102.RS_TABLESPACE'
BMC50471I ...'ATS102.RS_TSPART_DIST'

BMC50102I REORG TABLESPACE ARUDB006.TS006
BMC50102I COPY YES
BMC50102I ORDER YES
BMC50102I SORTDATA
BMC50102I ON FAILURE UNLOAD TERMINATE UTILITY RETCODE 31
BMC50102I DDTYPE UNLOAD ACTIVE NO
BMC50102I DDTYPE WORK
BMC50102I DSNPAT 'ARU.EXMPL09.&DDNAME'
BMC50102I DDTYPE LOCPFCPY
BMC50102I DSNPAT 'ARU.EXMPL09.&DDNAME'
BMC50102I DDTYPE REMBFCPY ACTIVE YES
BMC50102I DSNPAT 'ARU.EXMPL09.&DDNAME'
BMC50102I DDTYPE REMBFCPY ACTIVE YES
BMC50102I DSNPAT 'ARU.EXMPL09.&DDNAME'

BMC50004I UTILINIT PHASE COMPLETE. ELAPSED TIME = 00:00:00
BMC50041I 0: ZIIP ENABLED (0) USING XBM SUBSYSTEM XBMX
BMC51301I 1: SAMPLING STATISTICS: PART=0, TP=766, SP=310, SR=6614, AVG=10.50, SD=66, SE=0, AVGF=2133, SD=66, SE=65, EP=64
BMC51265I ESTIMATED CARDINALITY OF SPACE = 16296  AVG SORTWK ROW LENGTH = 111
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0061 = 1907  AVG SORTWK ROW LENGTH = 172
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0062 = 4657  AVG SORTWK ROW LENGTH = 98
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0063 = 4410  AVG SORTWK ROW LENGTH = 100
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0064 = 5322  AVG SORTWK ROW LENGTH = 109

BMC51264I UNLOAD WILL READ 764 DATA PAGES FROM SPACE 'ARUDB006.TS006'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMX
BMC50004I ANALYZE PHASE COMPLETE. ELAPSED TIME = 00:00:00
BMC50041I 0: ZIIP ENABLED (0) USING XBM SUBSYSTEM XBMX
BMC50004I ANALYZE PHASE COMPLETE. ELAPSED TIME = 00:00:00
BMC51301I 1: SAMPLING STATISTICS: PART=0, TP=766, SP=310, SR=6614, AVG=10.50, SD=66, SE=0, AVGF=2133, SD=66, SE=65, EP=64
BMC51265I ESTIMATED CARDINALITY OF SPACE = 16296  AVG SORTWK ROW LENGTH = 111
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0061 = 1907  AVG SORTWK ROW LENGTH = 172
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0062 = 4657  AVG SORTWK ROW LENGTH = 98
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0063 = 4410  AVG SORTWK ROW LENGTH = 100
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0064 = 5322  AVG SORTWK ROW LENGTH = 109

BMC50425I &JOBNAME = JRGAEXM9  &STEPNAME = BMCREORG  &DB = ARUDB006  &TSIX = TS006  &RTYPE = TS
BMC50425I &UID = RDAJRG3  &DATE = 101413  &TIME = 162240  &SSID = DEFF  &UTIL = EXMPL09
BMC50425I &UTILPFX = EXMPL09  &UTILSFX =  &DATE8 = 10142013  &GRPNM = DEFF  &VCAT = DEFFCAT
BMC50425I &TIME4 = 1622  &DATEJ = 2013287  &JDATE = 13287

BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I REDRG PLUS DYNAMIC FILE ALLOCATION REPORT

Figure 38  SYSPRINT for example 9 (part 5 of 6)
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

In this example, REORG PLUS invokes DSNUTILB to reorganize a segmented table space that has an associated index that contains keys with random ordering. This job is a two-phase table space reorganization. This job specifies a different options module on the EXEC statement that activates DSNUTILB when random key indexes are participating in the reorganization (IXRANDOM=YES).

Because this job invokes DSNUTILB, ACTIVE YES must be specified for all work files that the reorganization job requires. ACTIVE YES is required for at least the primary local copy data set (DDTYPE LOCPFCPY) when COPY YES is specified.

**NOTE**

All copy data sets for a DSNUTILB reorganization are dynamically allocated, even if you specify ACTIVE YES for only the primary local copy data set.

When REORG PLUS must invoke DSNUTILB, the SYSPRINT displays message BMC50182I, indicating the reason for invoking DSNUTILB.
DSNU050I messages show the REORG command and the TEMPLATE statements that are created from the REORG PLUS command and installation option values.

Table 74 describes the key command options for this job.

### Table 74  Key command options used in example 10 (part 1 of 2)

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>COPY YES INLINE YES</td>
<td>creates an inline DB2 image copy of the table space</td>
</tr>
</tbody>
</table>
| COPYDDN BMCPY | specifies the ddname BMCPY for the data set that receives the full local image copy of the table space that you are reorganizing  
| | overrides the default name of BMCCPY |
| RECOVERYDDN BMRPY | specifies a ddname of BMRPY for data set that receives the full remote image copy of the table space that you are reorganizing  
| | overrides the default names of BMCRCY |
| ORDER YES | tells REORG PLUS to sort the rows  
| | ORDER YES is passed to DSNUTILB as SORTDATA YES, and DSNUTILB will sort the clustering keys |
| REDEFINE NO | tells REORG PLUS not to delete and redefine the VSAM data sets for the table space  
| | Because this job invokes DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility as REUSE. |
| SORTDEVT | specifies the device type for the sort work files that are allocated dynamically  
| | Because this job invokes DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility for processing. |
| SORTNUM | affects the allocation of sort work files  
| | Because this job invokes DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility as the number of sort work files to allocate dynamically. |
| DDTYPE LOCBFCPY ACTIVE YES  
DDTYPE REMPFCPY ACTIVE YES  
DDTYPE REMBFCPY ACTIVE YES | activates dynamic data set allocation for the specified data set types |
| IFALLOC | Because this job invokes DSNUTILB, REORG PLUS ignores this option. |
| DSNPAT | specifies a pattern of variables and text that REORG PLUS uses to create data set names for dynamic data set allocation, overriding the default patterns  
| | Because this job invokes DSNUTILB, REORG PLUS includes this pattern in the TEMPLATE control statement that it builds for the DB2 REORG utility. |
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

Table 74  Key command options used in example 10 (part 2 of 2)

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIT MAXEXTSZ UNITCNT VOLCNT THRESHLD</td>
<td>specifies allocation information for the data sets that you are dynamically allocating. Because this job invokes DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the DB2 REORG utility.</td>
</tr>
<tr>
<td>UPDATEDB2STATS YES</td>
<td>requests that statistics be updated in the DB2 catalog. Because this job invokes DSNUTILB, REORG PLUS passes this option to the STATISTICS option of the DB2 REORG utility as STATISTICS TABLE (ALL) INDEX (ALL) REPORT YES UPDATE ALL.</td>
</tr>
</tbody>
</table>

Figure 39 shows the JCL for example 10.

**Figure 39**  JCL for example 10 (part 1 of 2)

```plaintext
//        JOB
/*.*/
//**  SEGMENTED TABLESPACE WITH AN ASSOCIATED INDEX CONTAINING A  */
//**  RANDOM INDEX KEY. DSNUTILB INVOKED DUE TO RANDOM INDEX KEY.  */
//**  NOTE THE BMC OPTIONS TRANSLATED TO DSNUTILB OPTIONS.        */
//**  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
//LARREORG EXEC PGM=ARUUMAIN,REGION=0M,COND=(7,LT),
// PARM='DEFF,ARUDB070.RG1,NEW,,MSGLEVEL(1),JG9$OPTS'
//STEPLIB   DD  DISP=SHR,DSN=ProductLibraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//**
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//UTPRINT DD SYSOUT=* 
//SYSTEM DD SYSOUT=* 
//**
//SYSIN DD *
REORG TABLESPACE ARUDB070.LARS$JBA
   SHRLEVEL NONE
   UNLOAD CONTINUE
   COPY YES
   INLINE YES
   COPYLVL FULL
   COPYDDN BMCPY
   RECOVERYDDN BMRPY
   PREFORMAT YES
   REDEFINE NO
   ORDER YES
   UPDATEDB2STATS YES
   SORTDEVT 3390
```
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

Figure 39  JCL for example 10 (part 2 of 2)

```
SORTNUM 12
SORTDATA
DDTYPE UNLOAD UNIT (WORK) IFALLOC USE
DSNPAT 'ARU.LARREORG.DEFF.ARUDB070.&DDNAME'
MAXEXTSZ 100 UNITCNT (1,15) VOLCNT AUTO
DDTYPE WORK UNIT (WORK) IFALLOC USE
DSNPAT 'ARU.LARREORG.DEFF.ARUDB070.&DDNAME'
DDTYPE LOCPCOPY IFALLOC USE DSNPAT
'ARU.LARREORG.DEFF.&DB.&TSIX.&DDNAME'
UNIT (WORK) THRESHLD 10000
DDTYPE LOCBFCPY ACTIVE YES IFALLOC USE DSNPAT
'ARU.LARREORG.DEFF.&DB.&TSIX.&DDNAME' UNIT (WORK)
DDTYPE REMPFCPY ACTIVE YES UNIT (WORK) IFALLOC USE
DSNPAT 'ARU.LARREORG.DEFF.&DB.&TSIX.&DDNAME'
DDTYPE REMBFCPY ACTIVE YES UNIT (WORK) IFALLOC USE
DSNPAT 'ARU.LARREORG.DEFF.&DB.&TSIX.&DDNAME'
*
```

Figure 40 shows the SYSPRINT output for example 10.

Figure 40  SYSPRINT for example 10 (part 1 of 8)

```
***** BMC REORG PLUS FOR DB2 V11R1.00 *****
(C) COPYRIGHT 1988 - 2013 BMC SOFTWARE, INC.
REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50001I UTILITY EXECUTION STARTING 10/14/2013 16:29:16 ...
BMC50002I UTILITY ID = 'ARUDB070.RG1'. DB2 SUBSYSTEM ID = 'DEFF'. OPTION MODULE = 'JG9$OPTS'.
BMC50471I z/OS 2.1.0,PID=HBB7790,DFSMS FOR Z/OS=2.1.0,DSB=2.1.0 ...
BMC50471I REGION=OM,BELLOW 16M=8844K,ABOVE 16M=1412652K,IEFUSI=NO,CPU=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040310M,MEMLIMIT SET BY:REGION=0
BMC50471I REORG PLUS FOR DB2--V11.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.04.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V11.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I EXTENDED BUFFER MANAGER--V06.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I ALTRFAIL=RCVRPEND   ICDDN=(BMCICY,BMCI)   RORGMAX=300%
BMC50471I ANALMAX=1000%   ICETYPE=AUTO   ROUTING=(-11,1)
BMC50471I ARCHNO=NO   IDCACHE=10000   SCPYM=Rh
BMC50471I ARCHDDN=DDLIN   INDEF=10   $DUMP=Yes
BMC50471I ARCHFORM=BMIC   INLINEXP=Yes   $SHORTMEM=CONTINUE
BMC50471I AVAILPAGEEXIT=100   IXONEX=No   SIXSNAP=No
BMC50471I BILOMAX=300%   IXRANDOM=Yes   $SMAX=0
BMC50471I BMCSFREE=YES   KEEPPOSITIONARY=No   $SMORE=(OK,OK)
BMC50471I CFUFFS=30   LEAFDSLM=200   SORTDEVT=(.,SYSAILDA)
BMC50471I CONDOMEM=No   LOB=Yes   SORTNUM=32
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)   LOCKROW=Yes   $SPILOSM=SU0
BMC50471I COPYMAX=1000%   LOGFINAL=NONE   $SPILICIS=NONE
BMC50471I COPYLVL=PART   LOGMEM=0   $SPILINIT=WORK
BMC50471I COPYMAX=1000%   LOGSPIL=(20000,10000)   $SQLDELAY=3
BMC50471I COPYPFAIL=TERM   LOGTHRES=0   $SQLRETRY=100
BMC50471I DATAFCAP=NONE   LOGTHRED=CONTINUE   STAGEDSN=BMIC
BMC50471I DDOLDD=DDLIN   LONGNAMETRUNC=BASE   STOPQM=Yes
BMC50471I DDOLDD=DDLIN   LOGNAMETRUNC=MIDDLE   STOPSH=Yes
BMC50471I DEADLINE=None   MAPTEXIT=(NONE,REXX)   STOPDB=300
BMC50471I DELAY=1200   MAXNEWPARTS=2   STOPRETRY=300
BMC50471I DELFILE=Yes   MAXRO=300   TAPEDISP=DELETE
```
### Example 10: DSNUTILB reorganization and index that contains keys with random ordering

#### Figure 40  SYSPRINT for example 10 (part 2 of 8)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRCM0471I DSNUEXIT</td>
<td>(NONE, ASM)</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I UNITCNT</td>
<td>(0, 0)</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMSUNIT</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>YES</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>ARCHIVE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>YES</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>YES</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>YES</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>YES</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>YES</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>YES</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I DDTYPE</td>
<td>UNLOAD</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ALLOC</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I SMS</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I IFALLOC</td>
<td>USE</td>
<td>-</td>
</tr>
<tr>
<td>BRCM0471I ACTIVE</td>
<td>YES</td>
<td>-</td>
</tr>
</tbody>
</table>

**Note:** The table above contains the SYSPRINT output for the example 10 reorganization and index creation process, showing various parameters and values. The output includes details on unit counts, SMS usage, allocation methods, and other related configurations.
Figure 40  SYSPRINT for example 10 (part 3 of 8)
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

Figure 40    SYSPRINT for example 10 (part 4 of 8)
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

Chapter 5 Examples of REORG PLUS jobs

Figure 40  SYSPRINT for example 10 (part 5 of 8)

BMC50471I ...'ATS102.RS_COLSTATS'
BMC50471I ...'ATS102.RS_COLUMNS'
BMC50471I ...'ATS102.RS_INDEXES'
BMC50471I ...'ATS102.RS_INDEXPART'
BMC50471I ...'ATS102.RS_INDEXPART_DIST'
BMC50471I ...'ATS102.RS_STOGROUP'
BMC50471I ...'ATS102.RS_TABLEPART'
BMC50471I ...'ATS102.RS_TABLESPACE'
BMC50471I ...'ATS102.RS_TSPART_DIST'

BMC50102I REORG TABLESPACE ARUDB070.LARS$JBA
BMC50102I SHLEVEL NONE
BMC50102I UNLOAD CONTINUE
BMC50102I COPY YES
BMC50102I INLINC YES
BMC50102I COPYLUL FULL
BMC50102I COPYUDN BMCPY
BMC50102I RECOVERY DDN BMCPY
BMC50102I PREFORMAT YES
BMC50102I REDEFINE NO
BMC50102I UDATEDB2STATS YES
BMC50102I SORTDEVT 3390
BMC50102I SORTNUM 12
BMC50102I SORTDATA
BMC50102I DTYPE UNLOAD UNIT (WORK) IFALLOCC USE
BMC50102I DSNPAT 'ARU.LARREORG.DEFF.ARUB070.'&DDNAME'
BMC50102I MAXEXTSZ 100 UNITCNT (1,15) VOLCNT
BMC50102I DTYPE WORK UNIT (WORK) IFALLOCC USE
BMC50102I DSNPAT 'ARU.LARREORG.DEFF.ARUB070.'&DDNAME'
BMC50102I DTYPE LOCPFCPY IFALLOCC USE
BMC50102I 'ARU.LARREORG.DEFF.'&DB.'&TSIX.'&DDNAME'
BMC50102I UNIT (WORK) THRESHLD 1000
BMC50102I DTYPE LOCPFCPY ACTIVE YES IFALLOCC USE
BMC50102I 'ARU.LARREORG.DEFF.'&DB.'&TSIX.'&DDNAME'
BMC50102I DDTYPE REMBFCPY ACTIVE YES UNIT (WORK)
BMC50102I DSNPAT 'ARU.LARREORG.DEFF.'&DB.'&TSIX.'&DDNAME'
BMC50102I DSNPAT 'ARU.LARREORG.DEFF.'&DB.'&TSIX.'&DDNAME'
BMC50102I DSNPAT 'ARU.LARREORG.DEFF.'&DB.'&TSIX.'&DDNAME'

BMC50182I DSNUTILB REQUIRED: INDEX CONTAINS A RANDOM KEY INDEX 'ARUDB070.'.'LARX001D'
BMC50004I UTILINIT PHASE COMPLETE. ELAPSED TIME = 00:00:00
DSNU000I 287 16:29:16.88 DSNUUGUTC - OUTPUT START FOR UTILITY, UTILID = ARUDB070.RG1
DSNU0144I 287 16:29:16.96 DSNUUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU005I 287 16:29:16.96 DSNUUGUTC - TEMPLATE BMC00001 DSN 'ARU.LARREORG.DEFF.&DB..&SN..BMCRCZ' UNIT WORK VOLCNT 25 PCTPRIME 100
DSNU1035I 287 16:29:16.97 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU005I 287 16:29:16.97 DSNUUGUTC - TEMPLATE BMC00002 DSN 'ARU.LARREORG.DEFF.&DB..&SN..BMRPY' UNIT WORK VOLCNT 25 PCTPRIME 100
DSNU1035I 287 16:29:16.97 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU005I 287 16:29:16.97 DSNUUGUTC - TEMPLATE BMC00003 DSN 'ARU.LARREORG.DEFF.&DB..&SN..BMCCPZ' UNIT WORK VOLCNT 25 PCTPRIME 100
DSNU1035I 287 16:29:16.97 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU005I 287 16:29:16.97 DSNUUGUTC - TEMPLATE BMC00004 DSN 'ARU.LARREORG.DEFF.&DB..&SN..BMCPY' LIMIT(9 MB, BMC00005)
DSNU1035I 287 16:29:16.97 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU005I 287 16:29:16.97 DSNUUGUTC - TEMPLATE BMC00005 DSN 'ARU.LARREORG.DEFF.&DB..&SN..BMCPY' UNIT SYSALLDA VOLCNT 25 PCTPRIME 100
DSNU1035I 287 16:29:16.97 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU005I 287 16:29:16.97 DSNUUGUTC - TEMPLATE BMC00006 DSN 'ARU.LARREORG.DEFF.ARUB070.SYSREC' UNIT UNCNT 1
DSNPIN(NEW, DELETE, CATLG) PCTPRIME 100 MAXPRIME 100
DSNU035I 287 16:29:16.97 DSNUUGUTC - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU005I 287 16:29:16.97 DSNUUGUTC - TEMPLATE BMC00007 DSN 'ARU.LARREORG.DEFF.ARUB070.SYSREC' UNIT SYSALDA VOLCNT 25 PCTPRIME 100
DSNU1035I 287 16:29:16.97 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU005I 287 16:29:16.97 DSNUUGUTC - REORG TABLESPACE ARUDB070.LARS$JBA REUSE COPYDDN(BMC00004, BMC00003)
RECOVERY(BMC00002, BMC00001) UNLDDN(BMC00006) SHLEVEL NONE UNLOAD CONTINUE STATISTICS TABLE ALL INDEX ALL
REPORT YES UPDATE ALL FORCEROLLUP YES PUNCHDDN BMC00007 SORTDEVT 3390 SORTNUM 12 PREFORMAT
DSNU038I 287 16:29:17.07 DSNUUGDYN - DATASET ALLOCATED. TEMPLATE=BMC00004 DONAME=SYS00003
DSN=ARU.LARREORG.DEFF.ARUB070.LARS$JBA.BMCPY
DSNU038I 287 16:29:17.10 DSNUUGDYN - DATASET ALLOCATED. TEMPLATE=BMC00003 DONAME=SYS00004
DSN=ARU.LARREORG.DEFF.ARUB070.LARS$JBA.BMCPY

Chapter 5  Examples of REORG PLUS jobs  513
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

**Figure 40**  SYSPRINT for example 10 (part 6 of 8)

![Image of SYSPRINT output](image-url)
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

Figure 40  SYSPRINT for example 10 (part 7 of 8)
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

```
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAROFFPOS</td>
<td>240</td>
</tr>
<tr>
<td>FAROFFPOSF</td>
<td>2.4E+02</td>
</tr>
<tr>
<td>LEAFDIST</td>
<td>18</td>
</tr>
<tr>
<td>LEAFNEAR</td>
<td>2</td>
</tr>
<tr>
<td>LEAFFAR</td>
<td>0</td>
</tr>
<tr>
<td>SPACE</td>
<td>4320</td>
</tr>
<tr>
<td>SPACEF</td>
<td>4.32E+03</td>
</tr>
<tr>
<td>DSNUM</td>
<td>1</td>
</tr>
<tr>
<td>EXTENTS</td>
<td>1</td>
</tr>
<tr>
<td>P_DEL_ENT</td>
<td>0</td>
</tr>
<tr>
<td>POTY</td>
<td>1000</td>
</tr>
<tr>
<td>SDTY</td>
<td>500</td>
</tr>
</tbody>
</table>
```

**Figure 40**  SYSPRINT for example 10 (part 8 of 8)

```
DSNU610I  *DEFF 287 16:29:19.07 DSNUSUUP - SYSSNIPRT CATALOG UPDATE FOR ARUDB070.LARX001D SUCCESSFUL
DSNU615I  *DEFF 287 16:29:19.07 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR CO09_TIMESTAMP
  COLCARD = 2048
  COLCARDF = 2.048E+03
  HIGHWKEY = 'X'44600702000000000000'
  LOWHWKEY = X'19100202000000000000'
DSNU610I  *DEFF 287 16:29:19.07 DSNUSUCO - SYSCOLUMNS CATALOG UPDATE FOR ARUDB070.LARX001A SUCCESSFUL
DSNU615I  *DEFF 287 16:29:19.07 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR CO09_TIMESTAMP
  COLCARD = 2048
  COLCARDF = 2.048E+03
  HIGHWKEY = 'X'44600702000000000000'
  LOWHWKEY = X'19100202000000000000'
DSNU610I  *DEFF 287 16:29:19.08 DSNUSUUP - SYSSNIPRT CATALOG UPDATE FOR ARUDB070.LARX001D SUCCESSFUL
DSNU615I  *DEFF 287 16:29:19.08 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR CO09_TIMESTAMP
  COLCARD = 2048
  COLCARDF = 2.048E+03
  HIGHWKEY = 'X'44600702000000000000'
  LOWHWKEY = X'19100202000000000000'
DSNU610I  *DEFF 287 16:29:19.08 DSNUSUCO - SYSCOLUMNS CATALOG UPDATE FOR ARUDB070.LARX001D SUCCESSFUL
DSNU615I  *DEFF 287 16:29:19.08 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR CO09_TIMESTAMP
  COLCARD = 2048
  COLCARDF = 2.048E+03
  HIGHWKEY = 'X'44600702000000000000'
  LOWHWKEY = X'19100202000000000000'
DSNU610I  *DEFF 287 16:29:19.08 DSNUSUUP - SYSSNIPRT CATALOG UPDATE FOR ARUDB070.LARX001D SUCCESSFUL
DSNU615I  *DEFF 287 16:29:19.08 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR CO09_TIMESTAMP
  COLCARD = 2048
  COLCARDF = 2.048E+03
  HIGHWKEY = 'X'44600702000000000000'
  LOWHWKEY = X'19100202000000000000'
DSNU610I  *DEFF 287 16:29:19.08 DSNUSUUP - SYSSNIPRT CATALOG UPDATE FOR ARUDB070.LARX001D SUCCESSFUL
DSNU615I  *DEFF 287 16:29:19.08 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR CO09_TIMESTAMP
  COLCARD = 2048
  COLCARDF = 2.048E+03
  HIGHWKEY = 'X'44600702000000000000'
  LOWHWKEY = X'19100202000000000000'
```

```
<table>
<thead>
<tr>
<th>FREQUENCY</th>
<th>COLVALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.8828125E-04</td>
<td>'X'44500702000000000000'</td>
</tr>
<tr>
<td>4.8828125E-04</td>
<td>'X'40500302000000000000'</td>
</tr>
<tr>
<td>4.8828125E-04</td>
<td>'X'41500202000000000000'</td>
</tr>
<tr>
<td>4.8828125E-04</td>
<td>'X'42500102000000000000'</td>
</tr>
<tr>
<td>4.8828125E-04</td>
<td>'X'37200402000000000000'</td>
</tr>
<tr>
<td>4.8828125E-04</td>
<td>'X'36200502000000000000'</td>
</tr>
<tr>
<td>4.8828125E-04</td>
<td>'X'32200102000000000000'</td>
</tr>
<tr>
<td>4.8828125E-04</td>
<td>'X'31200202000000000000'</td>
</tr>
</tbody>
</table>
```

**Figure 40**  SYSPRINT for example 10 (part 8 of 8)
Example 11: Partition-by-growth table space

In this example, REORG PLUS reorganizes a partition-by-growth table space. It is a SHRLEVEL NONE, two-phase, full table space reorganization. During the reorganization, REORG PLUS adds new partitions, indicated by message BMC50175I.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 75 describes the key command options for this job.

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>UNLOAD CONTINUE</td>
<td>tells REORG PLUS to continue the reorganization with two-phase processing after the UNLOAD phase has unloaded the data</td>
</tr>
<tr>
<td>MAXNEWPARTS</td>
<td>specifies the maximum number of partitions that REORG PLUS can add during a reorganization and the maximum number of partitions added by DB2 that REORG PLUS can support</td>
</tr>
<tr>
<td>COPY YES</td>
<td>creates a DB2 image copy of the table space</td>
</tr>
<tr>
<td></td>
<td>Because the default for the INLINECP installation option is YES, this copy is an inline image copy.</td>
</tr>
<tr>
<td>COPYLVL PART</td>
<td>allocates an image copy data set for each partition that you are reorganizing</td>
</tr>
<tr>
<td>REDEFINE YES</td>
<td>tells REORG PLUS to delete and redefine the VSAM data sets containing the DB2 objects</td>
</tr>
<tr>
<td>ORDER YES</td>
<td>sorts the data rows</td>
</tr>
<tr>
<td>BMCSTATS YES</td>
<td>tells REORG PLUS to update statistics in the DASD MANAGER PLUS database statistics tables and the DB2 catalog</td>
</tr>
<tr>
<td>UPDATEDB2STATS YES</td>
<td>The SYSPRINT shows the messages that the Common Statistics component displays for the statistics being updated.</td>
</tr>
</tbody>
</table>

Figure 41 shows the JCL for example 11.
Example 11: Partition-by-growth table space

Figure 41  JCL for example 11 (part 2 of 2)

```
//** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//LARREORG EXEC PGM=ARUUMAIN,REGION=0M,COND=(7,LT),
//PARM='DEFF,RDAJRGD3.RG1,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB DD DISP=SHR,DSN=ProductLibraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT DD SYSOUT=*  
//SYSOUT DD SYSOUT=* 
//SYSTERM DD SYSOUT=* 
//*
//SYSIN DD *

REORG TABLESPACE RDAJRGD3.LARSZABA
SHRLEVEL NONE
UNLOAD CONTINUE
MAXNEWPARTS 10
COPY YES
REGISTER ALL
COPYLVL PART
ORDER YES
ANALYZE SAMPLE
KEEPDICTIONARY NO
PREFORMAT YES
UPDATEDB2STATS YES
BMCSTATS YES
DDTYPE UNLOAD UNIT (WORK)
DSNPAT 'ARU.EXMPL11.&DB.&TSIX.&DDNAME'
DDTYPE WORK UNIT (WORK)
DSNPAT 'ARU.EXMPL11.&DB.&TSIX.&DDNAME'
DDTYPE LOCPFCPY UNIT (WORK) SIZEPCT(200,200)
DSNPAT 'ARU.EXMPL11.&DB.&TSIX.&DDNAME'
/*
```

Figure 42 shows the SYSPRINT output for example 11.

Figure 42  SYSPRINT for example 11 (part 1 of 12)

```
***** BMC REORG PLUS FOR DB2 V11R1.00 *****
(C) COPYRIGHT 1988 - 2013 BMC SOFTWARE, INC.
REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50011I UTILITY EXECUTION STARTING 10/14/2013 17:01:21 ...
BMC50011I UTILITY ID = 'RDAJRGD3.RG1'. DB2 SUBSYSTEM ID = 'DEFF'. OPTION MODULE = 'ARU$OPTS'.
BMC50471I z/OS 2.1.0,PID=HBB7790,DFSMS FOR Z/OS=2.1.0,PARM=DEFF.
BMC50471I REGION=0M,BELOW 16M=8836K,ABOVE 16M=1406780K,IEFUSI=NO,CPUS=3
BMC50471I REORG PLUS FOR DB2--V11.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V11.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V11.01.00
BMC50471I MAINT: BPJ0661 BPJ0667 BPJ0671 BPJ0674 BPJ0675 BPJ0682 BPJ0686 BPJ0689 BPJ0697
BMC50471I BMCSORT ENGINES--V02.04.01
BMC50471I NO MAINTENANCE TO REPORT
```

518  REORG PLUS for DB2 Reference Manual
Figure 42  SYSPRINT for example 11 (part 2 of 12)
**Figure 42** SYSPRINT for example 11 (part 3 of 12)

<table>
<thead>
<tr>
<th>BMC50470I UNIT</th>
<th>((SYSALLDA, SYSALLDA), (SYSALLDA, SYSALLDA), (SYSALLDA, SYSALLDA))</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I UNITCNT</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I AVGVOLSP</td>
<td>((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I STORCLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0, K), (0, K))</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td>-</td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td>-</td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I AVGVOLSP</td>
<td>((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0, K), (0, K))</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td>-</td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td>-</td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I AVGVOLSP</td>
<td>((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0, K), (0, K))</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td>-</td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td>-</td>
</tr>
<tr>
<td>BMC50470I DSNTYPE</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I AVGVOLSP</td>
<td>((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC50470I VOLCNT</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I MAXEXTSZ</td>
<td>((0, K), (0, K))</td>
</tr>
<tr>
<td>BMC50470I EXPDT</td>
<td>-</td>
</tr>
<tr>
<td>BMC50470I RETPD</td>
<td>-</td>
</tr>
</tbody>
</table>

**Example 11: Partition-by-growth table space**
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I GDGLIMIT = 0
BMC50470I GDEMPY = NO
BMC50470I GDESCRAT = NO
BMC50470I ODTYPE = LOCPFCPY
BMC50470I ACTIVE = NO
BMC50470I IFALLOC = USE
BMC50470I ALLOC = N/A
BMC50470I SMS = NO
BMC50470I SMSUNIT = NO
BMC50470I SIZEPCT = (100,100)
BMC50470I UNIT = (SYSALLDA, SYSALLDA)
BMC50470I UNITCNT = (0,0)
BMC50470I VOLCNT = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DSNTYPE = DSNHDECP
BMC50470I DATACLAS = DSNHDECP
BMC50470I MGMTCLAS = DSNHDECP
BMC50470I STORCLAS = DSNHDECP
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I GDGLIMIT = 5
BMC50470I GDEMPY = NO
BMC50470I GDESCRAT = NO
BMC50470I ODTYPE = REMPLCPY
BMC50470I ACTIVE = NO
BMC50470I IFALLOC = USE
BMC50470I ALLOC = N/A
BMC50470I SMS = NO
BMC50470I SMSUNIT = NO
BMC50470I SIZEPCT = (100,100)
BMC50470I UNIT = (SYSALLDA, SYSALLDA)
BMC50470I UNITCNT = (0,0)
BMC50470I VOLCNT = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DSNTYPE = DSNHDECP
BMC50470I DATACLAS = DSNHDECP
BMC50470I MGMTCLAS = DSNHDECP
BMC50470I STORCLAS = DSNHDECP
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I GDEMPY = NO
BMC50470I GDESCRAT = NO
BMC50470I UNLOAD DSNPAT=&UID..BMC.&TSIX.&DDNAME
BMC50483I WORK DSNPAT=&UID..BMC.&TSIX.&DDNAME
BMC50483I SORTWORK DSNPAT=&UID..AUTILPFX..ADDNAME
BMC50483I ARCHIVE DSNPAT=&UID..AUTILPFX..ADDNAME
BMC50483I SYSUNCH DSNPAT=&UID..AUTILPFX..ADDNAME
BMC50483I LOCFPCPY DSNPAT=&UID..ADDNAME..ATSIX..F&PART..T&TIME
BMC50483I LOCPFCPY DSNPAT=&UID..ADDNAME..ATSIX..F&PART..T&TIME
BMC50483I LORBFPCPY DSNPAT=&UID..ADDNAME..ATSIX..F&PART..T&TIME
BMC50483I LORBFCPY DSNPAT=&UID..ADDNAME..ATSIX..F&PART..T&TIME
BMC50483I LOCLFCPY DSNPAT=&UID..ADDNAME..ATSIX..F&PART..T&TIME
BMC50483I LOCLFCPY DSNPAT=&UID..ADDNAME..ATSIX..F&PART..T&TIME
BMC50483I REMPLFCPY DSNPAT=&UID..ADDNAME..ATSIX..F&PART..T&TIME
BMC50483I REMPFCPY DSNPAT=&UID..ADDNAME..ATSIX..F&PART..T&TIME
BMC50483I REMPLFCPY DSNPAT=&UID..ADDNAME..ATSIX..T&TIME
BMC50483I REMPFCPY DSNPAT=&UID..ADDNAME..ATSIX..T&TIME
BMC50471I 0R2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION = 1010
BMC50471I SUBSYSTEM DEFAULT = DEFF
Figure 42  SYSPRINT for example 11 (part 5 of 12)
Figure 42  SYSPRINT for example 11 (part 6 of 12)

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>User/Owner</th>
<th>Action</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>170121</td>
<td>DEFF</td>
<td>RELOAD</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>170121</td>
<td>DEFF</td>
<td>UNLOAD</td>
<td>STATISTICS</td>
</tr>
<tr>
<td>170121</td>
<td>DEFF</td>
<td>RELOAD</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>170121</td>
<td>DEFF</td>
<td>UNLOAD</td>
<td>STATISTICS</td>
</tr>
<tr>
<td>170121</td>
<td>DEFF</td>
<td>RELOAD</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>170121</td>
<td>DEFF</td>
<td>UNLOAD</td>
<td>STATISTICS</td>
</tr>
<tr>
<td>170121</td>
<td>DEFF</td>
<td>RELOAD</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>170121</td>
<td>DEFF</td>
<td>UNLOAD</td>
<td>STATISTICS</td>
</tr>
<tr>
<td>170121</td>
<td>DEFF</td>
<td>RELOAD</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>170121</td>
<td>DEFF</td>
<td>UNLOAD</td>
<td>STATISTICS</td>
</tr>
</tbody>
</table>

Chapter 5 Examples of REORG PLUS jobs 523
Example 11: Partition-by-growth table space

Figure 42  SYSPRINT for example 11 (part 7 of 12)
Example 11: Partition-by-growth table space

Figure 42  SYSPRINT for example 11 (part 8 of 12)

<table>
<thead>
<tr>
<th>BMC50376</th>
<th>262997 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY01'</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50476</td>
<td>DATASET SUCCESSFULLY DELETED, DONNAME = 'BMCCPY01', I/OS = 2242, I/O WAITS = 855, RDB LOCK WAITS = 0</td>
</tr>
<tr>
<td>BMC50376</td>
<td>262997 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY02'</td>
</tr>
<tr>
<td>BMC50476</td>
<td>DATASET SUCCESSFULLY DELETED, DONNAME = 'BMCCPY02', I/OS = 2242, I/O WAITS = 855, RDB LOCK WAITS = 0</td>
</tr>
<tr>
<td>BMC50376</td>
<td>262997 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY03'</td>
</tr>
<tr>
<td>BMC50476</td>
<td>DATASET SUCCESSFULLY DELETED, DONNAME = 'BMCCPY03', I/OS = 2242, I/O WAITS = 855, RDB LOCK WAITS = 0</td>
</tr>
<tr>
<td>BMC50376</td>
<td>262997 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY04'</td>
</tr>
<tr>
<td>BMC50476</td>
<td>DATASET SUCCESSFULLY DELETED, DONNAME = 'BMCCPY04', I/OS = 2242, I/O WAITS = 855, RDB LOCK WAITS = 0</td>
</tr>
<tr>
<td>BMC50376</td>
<td>262997 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY05'</td>
</tr>
<tr>
<td>BMC50476</td>
<td>DATASET SUCCESSFULLY DELETED, DONNAME = 'BMCCPY05', I/OS = 2242, I/O WAITS = 855, RDB LOCK WAITS = 0</td>
</tr>
<tr>
<td>BMC50376</td>
<td>262997 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY06'</td>
</tr>
<tr>
<td>BMC50476</td>
<td>DATASET SUCCESSFULLY DELETED, DONNAME = 'BMCCPY06', I/OS = 2242, I/O WAITS = 855, RDB LOCK WAITS = 0</td>
</tr>
<tr>
<td>BMC50376</td>
<td>262997 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY07'</td>
</tr>
<tr>
<td>BMC50476</td>
<td>DATASET SUCCESSFULLY DELETED, DONNAME = 'BMCCPY07', I/OS = 2242, I/O WAITS = 855, RDB LOCK WAITS = 0</td>
</tr>
<tr>
<td>BMC50376</td>
<td>262997 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY08'</td>
</tr>
<tr>
<td>BMC50476</td>
<td>DATASET SUCCESSFULLY DELETED, DONNAME = 'BMCCPY08', I/OS = 2242, I/O WAITS = 855, RDB LOCK WAITS = 0</td>
</tr>
</tbody>
</table>

Chapter 5  Examples of REORG PLUS jobs  525
Figure 42  SYSPRINT for example 11 (part 9 of 12)
### Example 11: Partition-by-growth table space

**Figure 42**  SYSPRINT for example 11 (part 10 of 12)

<table>
<thead>
<tr>
<th>TABLEPART</th>
<th>RDAJRGD3.LARSZABA PART 002</th>
</tr>
</thead>
<tbody>
<tr>
<td>HISTORY</td>
<td></td>
</tr>
<tr>
<td>SAVESTATS</td>
<td>Y</td>
</tr>
<tr>
<td>SAMPLING</td>
<td>N</td>
</tr>
<tr>
<td>ATTRIBUTES</td>
<td></td>
</tr>
<tr>
<td>POTY</td>
<td>100</td>
</tr>
<tr>
<td>ALLOCUNIT</td>
<td>1</td>
</tr>
<tr>
<td>ROW_FORMAT</td>
<td>RRF</td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>CARD</td>
<td>130986</td>
</tr>
<tr>
<td>ROWMAxFound</td>
<td>57</td>
</tr>
<tr>
<td>FULL</td>
<td>26</td>
</tr>
<tr>
<td>PCTPAGES</td>
<td>49</td>
</tr>
<tr>
<td>PCTUSED</td>
<td>100</td>
</tr>
<tr>
<td>PAGESAVE</td>
<td>0</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td></td>
</tr>
<tr>
<td>IPREFIX</td>
<td>I</td>
</tr>
<tr>
<td>EXTENTS</td>
<td>24</td>
</tr>
<tr>
<td>VOLUME</td>
<td>21855</td>
</tr>
<tr>
<td>SPACE KB</td>
<td>1049040</td>
</tr>
<tr>
<td>NUMDATASETS</td>
<td>1</td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>ROW_FORMAT</td>
<td>RRF</td>
</tr>
<tr>
<td>RBA_FORMAT</td>
<td>BASIC</td>
</tr>
<tr>
<td>VCATNAME</td>
<td>DEFFCAT</td>
</tr>
<tr>
<td>STORNAME</td>
<td>ARULARSG</td>
</tr>
<tr>
<td>PQTY</td>
<td>100</td>
</tr>
<tr>
<td>SQTY</td>
<td>100</td>
</tr>
<tr>
<td>PCTFREE</td>
<td>50</td>
</tr>
<tr>
<td>FREEPAGE</td>
<td>50</td>
</tr>
<tr>
<td>TRACKMOD</td>
<td>50</td>
</tr>
<tr>
<td>FREEPAGE</td>
<td>50</td>
</tr>
<tr>
<td>TRACKMOD</td>
<td>50</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td></td>
</tr>
<tr>
<td>PAGESAVE</td>
<td>0</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td></td>
</tr>
<tr>
<td>IPREFIX</td>
<td>I</td>
</tr>
<tr>
<td>EXTENTS</td>
<td>30</td>
</tr>
<tr>
<td>VOLUME</td>
<td>21855</td>
</tr>
<tr>
<td>SPACE KB</td>
<td>1049040</td>
</tr>
<tr>
<td>NUMDATASETS</td>
<td>1</td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>ROW_FORMAT</td>
<td>RRF</td>
</tr>
<tr>
<td>RBA_FORMAT</td>
<td>BASIC</td>
</tr>
<tr>
<td>VCATNAME</td>
<td>DEFFCAT</td>
</tr>
<tr>
<td>STORNAME</td>
<td>ARULARSG</td>
</tr>
<tr>
<td>PQTY</td>
<td>100</td>
</tr>
<tr>
<td>SQTY</td>
<td>100</td>
</tr>
<tr>
<td>PCTFREE</td>
<td>50</td>
</tr>
<tr>
<td>FREEPAGE</td>
<td>50</td>
</tr>
<tr>
<td>TRACKMOD</td>
<td>50</td>
</tr>
<tr>
<td>FREEPAGE</td>
<td>50</td>
</tr>
<tr>
<td>TRACKMOD</td>
<td>50</td>
</tr>
</tbody>
</table>

**Chapter 5**  Examples of REORG PLUS jobs 527
Example 11: Partition-by-growth table space

Figure 42  SYSPRINT for example 11 (part 1 of 12)

| PCTPAGES | PCTUSED | PAGESAVE | ALLOCATION | EXTENTS | TABLEPART | HISTORY | SAVESTATS | UPDATEDB2 | LOCATION | UTILCODE | BMCSTATS | V11.01.00 REPORT FOR DEFF V101 | TIME 2013-10-14-17.01.32.681266 | UTILCODE | HISTORY | SAVESTATS | UPDATEDB2 | LOCATION | UTILCODE | BMCSTATS | V11.01.00 REPORT FOR DEFF V101 | TIME 2013-10-14-17.01.32.681266 |
|----------|---------|----------|------------|----------|-----------|---------|-----------|-----------|----------|----------|-----------|---------------------------|-----------------------------|----------|---------|-----------|-----------|----------|-----------|---------------------------|-----------------------------|
| 50       | 100     | 0        | 1 SPACE    | 18 Volcount | RDAJRG03.LARSZABA PART 006 | N        | Y         | Y         | (NULL)   | R         | BMCSTATS V11.01.00 REPORT FOR DEFF V101 | TIME 2013-10-14-17.01.32.681266 | R        | N        | Y         | Y         | (NULL)   | R         | BMCSTATS V11.01.00 REPORT FOR DEFF V101 | TIME 2013-10-14-17.01.32.681266 |
|          | -       | -        |            | -        | RDAJRG03.LARSZABA PART 007 | N        | Y         | Y         | (NULL)   | R         |                     |                                           | R        | N        | Y         | Y         | (NULL)   | R         |                     |                                           |
|          | -       | -        |            | -        | RDAJRG03.LARSZABA PART 008 | N        | Y         | Y         | (NULL)   | R         |                     |                                           | R        | N        | Y         | Y         | (NULL)   | R         |                     |                                           |
|          | -       | -        |            | -        | RDAJRG03.LARSZABA PART 009 | N        | Y         | Y         | (NULL)   | R         |                     |                                           | R        | N        | Y         | Y         | (NULL)   | R         |                     |                                           |

Example 11: Partition-by-growth table space
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

In this example, REORG PLUS reorganizes a partitioned table space with three partitions. The table has one clustering index and one nonpartitioned secondary index. This example uses SHRLEVEL CHANGE to keep the objects in RW status during most of the reorganization.

In the LOGAPPLY phase, REORG PLUS determines the rate at which it is applying log records. If REORG PLUS estimates that it can finish applying the log records (LOGFINAL phase) by the time specified by the DEADLINE option, and without exceeding the maximum time specified by MAXRO, REORG PLUS moves from the LOGAPPLY phase into the LOGFINAL phase and completes the reorganization.

The DDLIN data set supplies the ALTER INDEX commands. REORG PLUS performs online partition rebalancing in the staging data sets, and performs the ALTER commands in the UTILTERM phase.

Message 50041I indicates the status of zIIP processing. For this example, an XBM subsystem ID was specified for SHRLEVEL CHANGE processing. REORG PLUS uses the same XBM subsystem ID for zIIP processing.

Table 76 on page 530 describes the key command options, installation options, and DD statements for this job.
### Table 76 Command options, installation options, and DD statements used in example 12 (part 1 of 3)

<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL; installation options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>SHRLEVEL CHANGE</td>
<td>specifies the reorganization type</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS will keep the original data sets in their original status (which can be RW) as long as possible.</td>
</tr>
<tr>
<td>UNLOAD CONTINUE</td>
<td>tells REORG PLUS to continue the reorganization with two-phase processing after the UNLOAD phase has unloaded the data</td>
</tr>
<tr>
<td>XBMID XBMA</td>
<td>specifies to use the XBM subsystem with the SSID of XBMA</td>
</tr>
<tr>
<td>COPY YES</td>
<td>creates full and incremental DB2 image copies of the table space</td>
</tr>
<tr>
<td></td>
<td>SHRLEVEL CHANGE sets REGISTER ALL, so all copies are registered in SYSIBM.SYSCOPY.</td>
</tr>
<tr>
<td>ORDER YES</td>
<td>sorts the rows by table as well as by each table’s clustering key</td>
</tr>
<tr>
<td></td>
<td>If no clustering key exists, x’00’s are used.</td>
</tr>
<tr>
<td>DEADLINE</td>
<td>continues the reorganization only if REORG PLUS estimates that it can finish LOGFINAL processing by the time and date specified in the timestamp</td>
</tr>
<tr>
<td></td>
<td>The UTILTERM phase can continue past the deadline.</td>
</tr>
<tr>
<td>MAXRO 600</td>
<td>starts the LOGFINAL phase when REORG PLUS estimates that it can apply the remaining log records within 10 minutes (600 seconds)</td>
</tr>
<tr>
<td></td>
<td>This estimate does not include the time needed to produce the incremental image copies and complete the UTILTERM phase processing.</td>
</tr>
<tr>
<td>LONGLOG CONTINUE (installation option)</td>
<td>tells REORG PLUS to continue the reorganization if the longlog condition still exists after the DELAY time expires</td>
</tr>
<tr>
<td>DELAY 300</td>
<td>specifies the number of seconds that are to elapse from the time REORG PLUS detects a longlog condition until the time it performs the action specified on the LONGLOG option</td>
</tr>
<tr>
<td></td>
<td>If the longlog condition no longer exists at the end of the time period specified by DELAY, the timer, is reset. When REORG PLUS detects the next longlog condition, it restarts the timer using the original DELAY value.</td>
</tr>
<tr>
<td>RIDMAPMEM 0 (installation option)</td>
<td>REORG PLUS automatically calculates the RIDMAPMEM value to be used to store the RID translation maps.</td>
</tr>
<tr>
<td>LOGMEM 0 (installation option)</td>
<td>REORG PLUS automatically calculates the amount of data space memory (in kilobytes) needed to hold the internal structures that REORG PLUS uses to store the log records</td>
</tr>
</tbody>
</table>
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

Table 76 Command options, installation options, and DD statements used in example 12 (part 2 of 3)

<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL; installation options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGSPILL (1024, 1024)</td>
<td>determines the primary and secondary space allocation (in kilobytes) for the spill data sets that hold the log records. Each number can be a nonzero positive integer.</td>
</tr>
<tr>
<td>SPILLDSNPAT</td>
<td>tells REORG PLUS to use a particular pattern of variables and text to create a prefix for the spill data set name. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.</td>
</tr>
<tr>
<td>SPILLUNIT</td>
<td>specifies the DASD unit where REORG PLUS can allocate spill data sets. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.</td>
</tr>
<tr>
<td>SPILLSTORCLAS</td>
<td>specifies the SMS storage class that REORG PLUS uses to allocate spill data sets. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>gathers information about the objects that you are reorganizing and provides estimated sizes for SYSREC, SYSUT1, SORTWK, and copy data sets.</td>
</tr>
<tr>
<td>SIZEPCT (150,150)</td>
<td>tells REORG PLUS to allocate 150% of the data set size that it calculated for the UNLOAD, WORK, and SORTWORK files. The increase allows room for the unknown number of updates that are being applied to the data sets.</td>
</tr>
<tr>
<td>DSNPAT</td>
<td>specifies a pattern of variables and text that REORG PLUS uses to create data set names for dynamic data set allocation, overriding the default pattern.</td>
</tr>
<tr>
<td>DDLIN</td>
<td>DDLIN data set contains the SQL ALTER INDEX and ALTER TABLESPACE statements with the new limit key values to use to rebalance partitions.</td>
</tr>
</tbody>
</table>
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

Table 76  Command options, installation options, and DD statements used in example 12 (part 3 of 3)

<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL; installation options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//BMCCPY //BMCCPZ //BMCRCY //BMCRCZ</td>
<td>default ddnames that are used for the data sets that receive a full image copy of the table space that you are reorganizing. The existence of the copy ddnames determines the number of copies made. COPY YES is set for SHRLEVEL CHANGE.</td>
</tr>
<tr>
<td>//BMCICY //BMCICZ //BMCIY //BMCIZ</td>
<td>default ddnames that are used for the data sets that receive an incremental image copy of the table space that you are reorganizing. The existence of the copy ddnames determines the number of copies made. COPY YES is set for SHRLEVEL CHANGE.</td>
</tr>
</tbody>
</table>

Figure 43 shows the JCL for example 12.

Figure 43  JCL for example 12 (part 1 of 2)

```
// JOB
/*@  PARTITIONED TABLESPACE, CLUSTERING INDEX AND NONPARTITIONED *
/*@  SECONDARY INDEX.                                          *
/*@  REORG TABLESPACE WITH SHRLEVEL CHANGE OPTION.            *
/*@  ALTER PARTITIONING INDEX KEYS USING DDLIN OPTION.        *
/*@  LARREORG  EXEC  PGM=ARUUMAIN,COND=(7,LT),                 *
/*@    PARM='DEFF,LARCOPB,LARBOXAB,NEW,,MSGLEVEL(1),ARU$OPTS' *
//STEPLIB  DD  DISP=SHR,DSN=ProductLibraries
// DD  DISP=SHR,DSN=DB2.DSNEXIT
// DD  DISP=SHR,DSN=DB2.DSNLOAD
/*@  SYSPRINT  DD  SYSOUT=*                                   *
/*@  SYSOUT  DD  SYSOUT=*                                    *
/*@  UTPRINT  DD  SYSOUT=*                                   *
/*@  SYSTEM  DD  SYSOUT=*                                    *
/*@  BMCCPY  DD  DSN=ARU.EXMPL12.BMCCPY,                     *
/*@    DISP=(,CATLG),                                         *
/*@  SPACE=(CYL,(50,20),RLSE),UNIT=WORK                       *
//BMCCPZ  DD  DSN=ARU.EXMPL12.BMCCPZ,                       *
/*@    DISP=(,CATLG),                                         *
/*@  SPACE=(CYL,(50,20),RLSE),UNIT=WORK                       *
//BMCRCY  DD  DSN=ARU.EXMPL12.BMCRCY,                       *
/*@    DISP=(,CATLG),                                         *
```
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

Figure 43  JCL for example 12 (part 2 of 2)

```
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCRCZ    DD DSN=ARU.EXMPL12.BMCRCZ,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCICY    DD DSN=ARU.EXMPL12.BMCICY,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCICZ    DD DSN=ARU.EXMPL12.BMCICZ,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCIRY    DD DSN=ARU.EXMPL12.BMCIRY,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCIRZ    DD DSN=ARU.EXMPL12.BMCIRZ,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//SYSIN     DD *
REORG TABLESPACE LARDBXAB.LARSXABA
  SHRLEVEL CHANGE
  UNLOAD CONTINUE
  XBMID XBMB
  FASTSWITCH YES
  COPY YES
  ORDER YES
  DEADLINE 2013-10-15-19.30.00.00
  MAXRO 600
  DELAY 300
  LOGSPILL (1024,1024)
  SPILLDSNPAT 'ARU.LARWORK.DEFF'
  SPILLUNIT NONE
  SPILLSTORCLAS COPYCLAS
  ANALYZE
  DDTYPE UNLOAD  UNIT (WORK) SIZEPCT (150,150)
  DSNPAT 'ARU.EXMPL12.&DB.&TSIX.&DDNAME'
  DDTYPE WORK  UNIT (WORK) SIZEPCT (150,150)
  DSNPAT 'ARU.EXMPL12.&DB.&TSIX.&DDNAME'
  DDTYPE LOCPFCPY ACTIVE NO
//DDLIN DD *
ALTER INDEX LARDBXAB.LARX001A
  PART 1 VALUES ('2800-12-31-00.00.00.000000'),
  PART 2 VALUES ('3800-12-31-00.00.00.000000'),
  PART 3 VALUES ('4500-12-31-00.00.00.000000');
/*
```

Figure 44 shows the SYSPRINT output for example 12.
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

Figure 44  SYSPRINT for example 12 (part 1 of 8)
### Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

**Figure 44** SYSPRINT for example 12 (part 2 of 8)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC504701 DTYPE</td>
<td>DTYPE</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 ACTIVE</td>
<td>ACTIVE</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>IFALLOC</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 ALLOC</td>
<td>ALLOC</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 SMS</td>
<td>SMS</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMSUNIT</td>
<td>SMSUNIT</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SIZEPCT</td>
<td>SIZEPCT</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC504701 VOLCNT</td>
<td>VOLCNT</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC504701 AVGVOLSP</td>
<td>AVGVOLSP</td>
<td>((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC504701 DSNTYPE</td>
<td>DSNTYPE</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC504701 DATACLAS</td>
<td>DATACLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC504701 STORCLAS</td>
<td>STORCLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC504701 DSNTYPE</td>
<td>DSNTYPE</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC504701 AVGVOLSP</td>
<td>AVGVOLSP</td>
<td>((30000, TRK), (30000, TRK))</td>
</tr>
<tr>
<td>BMC504701 DSNTYPE</td>
<td>DSNTYPE</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC504701 DATACLAS</td>
<td>DATACLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC504701 STORCLAS</td>
<td>STORCLAS</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC504701 THRESHOLD</td>
<td>THRESHOLD</td>
<td>(0, K, 0, K)</td>
</tr>
<tr>
<td>BMC504701 MAXEXTSZ</td>
<td>MAXEXTSZ</td>
<td>(0, K, 0, K)</td>
</tr>
<tr>
<td>BMC504701 EXPOT</td>
<td>EXPOT</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 RETPD</td>
<td>RETPD</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>GDGEMPTY</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 GDGSCRT</td>
<td>GDGSCRT</td>
<td>NO</td>
</tr>
</tbody>
</table>

---

**Chapter 5 Examples of REORG PLUS jobs**

535
### Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

#### Figure 44  SYSPRINT for example 12 (part 3 of 8)

<table>
<thead>
<tr>
<th>BMC504701 GDDLIMIT</th>
<th>5</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC504701 GDEMPTY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 GDSCRAT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 DDTYPE</td>
<td>SYSBUNCH</td>
<td></td>
</tr>
<tr>
<td>BMC504701 ACTIVE</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
<td></td>
</tr>
<tr>
<td>BMC504701 ALLOC</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>BMC504701 SMS</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>BMC504701 SMSUNIT</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>BMC504701 GDGSCRAT</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>BMC504701 UNLOAD</td>
<td>DSNPAT=&amp;UID..BMC.&amp;TSIX.&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC504701 WORK</td>
<td>DSNPAT=&amp;UID..BMC.&amp;TSIX.&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC504701 SORTWORK</td>
<td>DSNPAT=&amp;UID..BMC.&amp;TSIX.&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC504701 ARCHIVE</td>
<td>DSNPAT=&amp;UID..BMC.&amp;TSIX.&amp;DDNAME</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BMC504701 DDTYPE</th>
<th>REMPLCPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC504701 ACTIVE</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 ALLOC</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC504701 SMS</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 SMSUNIT</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 GDGSCRAT</td>
<td>NO</td>
</tr>
</tbody>
</table>

---

536  REORG PLUS for DB2 Reference Manual
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

Figure 44  SYSPRINT for example 12 (part 4 of 8)
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

Figure 44  SYSPRINT for example 12 (part 5 of 8)
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

Figure 44  SYSPRINT for example 12 (part 6 of 8)
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

**Figure 44** SYSPRINT for example 12 (part 7 of 8)

<table>
<thead>
<tr>
<th>BMC50749I 102:</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50716I 102: Buffer Manager Statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50717I 102: DATASET=DEFCCAT.DSNDBC.LARDBXAB.LARXABA.J0001.A001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50717I 102: DATASET=DEFCCAT.DSNDBC.LARDBXAB.LARXABA.J0001.A002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50717I 102: DATASET=DEFCCAT.DSNDBC.LARDBXAB.LARXABA.J0001.A003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 102: Page Requests 6</td>
<td>Page Hits 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 102: Page Reads 6</td>
<td>Page Writes 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 102: Write I/O Requests 3</td>
<td>New Pages 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50716I 102: Write I/O time 0:00:00</td>
<td>Read I/O time 0:00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50786I 102: LOG APPLY TASK ENDED AT 10/14/2013 17:13:54, ELAPSED TIME = 00:00:10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50786I 102: LOG RECORD SORT TASK ENDED AT 10/14/2013 17:13:54, ELAPSED TIME = 00:00:16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50786I 102: INLINE IMAGE COPY APPEND TASK ENDED AT 10/14/2013 17:13:54, ELAPSED TIME = 00:00:16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50794I 102: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXAB.LARX001B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 106: PAGE READS : 2 KEY INSERTS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 106: RID INSERTS : 0 RID DELETES (PS) : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 106: ROOT SPLITS : 0 NON-LEAF SPLITS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 106: LEAF SPLITS : 0 PAGE FREES : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 106: NEW PAGES : 0 BUFF PAGE READS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50716I 106: Buffer Manager Statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50717I 106: DATASET=DEFCCAT.DSNDBC.LARDBXAB.LARX001B.J0001.A001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 106: Page Requests 6</td>
<td>Page Hits 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 106: Page Reads 5</td>
<td>Page Writes 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 106: Write I/O Requests 1</td>
<td>New Pages 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50716I 106: Write I/O time 0:00:00</td>
<td>Read I/O time 0:00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50786I 106: INDEX APPLY TASK ENDED AT 10/14/2013 17:13:56, ELAPSED TIME = 00:00:15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50794I 103: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXAB.LARX001A DSNUM 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 103: PAGE READS : 2 KEY INSERTS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 103: RID INSERTS : 0 RID DELETES (PS) : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 103: ROOT SPLITS : 0 NON-LEAF SPLITS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 103: LEAF SPLITS : 0 PAGE FREES : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 103: NEW PAGES : 0 BUFF PAGE READS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50716I 103: Buffer Manager Statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50717I 103: DATASET=DEFCCAT.DSNDBC.LARDBXAB.LARX001A.J0001.A003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 103: Page Requests 5</td>
<td>Page Hits 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 103: Page Reads 4</td>
<td>Page Writes 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 103: Write I/O Requests 1</td>
<td>New Pages 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50716I 103: Write I/O time 0:00:00</td>
<td>Read I/O time 0:00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50786I 103: INDEX APPLY TASK ENDED AT 10/14/2013 17:13:57, ELAPSED TIME = 00:00:15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50794I 103: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXAB.LARX001A DSNUM 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 104: PAGE READS : 2 KEY INSERTS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 104: RID INSERTS : 0 RID DELETES (PS) : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 104: ROOT SPLITS : 0 NON-LEAF SPLITS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 104: LEAF SPLITS : 0 PAGE FREES : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 104: NEW PAGES : 0 BUFF PAGE READS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50716I 104: Buffer Manager Statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50717I 104: DATASET=DEFCCAT.DSNDBC.LARDBXAB.LARX001A.J0001.A002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 104: Page Requests 5</td>
<td>Page Hits 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 104: Page Reads 4</td>
<td>Page Writes 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 104: Write I/O Requests 1</td>
<td>New Pages 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50716I 104: Write I/O time 0:00:00</td>
<td>Read I/O time 0:00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50786I 104: INDEX APPLY TASK ENDED AT 10/14/2013 17:13:59, ELAPSED TIME = 00:00:15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50794I 104: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXAB.LARX001A DSNUM 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 105: PAGE READS : 2 KEY INSERTS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 105: RID INSERTS : 0 RID DELETES (PS) : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 105: ROOT SPLITS : 0 NON-LEAF SPLITS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 105: LEAF SPLITS : 0 PAGE FREES : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50791I 105: NEW PAGES : 0 BUFF PAGE READS : 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50716I 105: Buffer Manager Statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50717I 105: DATASET=DEFCCAT.DSNDBC.LARDBXAB.LARX001A.J0001.A001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 105: Page Requests 5</td>
<td>Page Hits 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 105: Page Reads 4</td>
<td>Page Writes 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50718I 105: Write I/O Requests 1</td>
<td>New Pages 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50716I 105: Write I/O time 0:00:00</td>
<td>Read I/O time 0:00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50786I 105: INDEX APPLY TASK ENDED AT 10/14/2013 17:13:59, ELAPSED TIME = 00:00:15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50868I 105: LOG RECORD STORE SPILL REQUESTS = 0, HIGH SPILL PAGE = 0, SPILL DATASETS CREATED = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50868I 105: LOG RECORD STORE WRITE REQUESTS = 0, WRITE WAIT TIME = 0.000, READ REQUESTS = 0 READ WAIT TIME = 0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50786I 108: LOG RECORD SPILL TASK ENDED AT 10/14/2013 17:13:59, ELAPSED TIME = 00:00:21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50476I DNAME = BMCCPY, I/Os = 181, I/O WAITS = 145, RDB LOCK WAITS = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50476I DNAME = BMCCPY, I/Os = 181, I/O WAITS = 145, RDB LOCK WAITS = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50476I DNAME = BMCCPY, I/Os = 181, I/O WAITS = 145, RDB LOCK WAITS = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50476I DNAME = BMCCPY, I/Os = 181, I/O WAITS = 145, RDB LOCK WAITS = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50476I DNAME = BMCCPY, I/Os = 181, I/O WAITS = 145, RDB LOCK WAITS = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50376I 2132 PAGES COPIED TO DATASET = 'ARU.EXMPL12.BMCCPY'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50376I 2132 PAGES COPIED TO DATASET = 'ARU.EXMPL12.BMCCPY'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50376I 2132 PAGES COPIED TO DATASET = 'ARU.EXMPL12.BMCCPY'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50376I 2132 PAGES COPIED TO DATASET = 'ARU.EXMPL12.BMCCPY'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50465I TOTAL RECORDS INSERTED INTO LOG RECORD STORE, DATA = 0, INDEX = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 44  SYSPRINT for example 12 (part 8 of 8)

| BMC50866I | TOTAL TABLESPACE STRINGS PROCESSED = 0, WHEN NOT FULL = 0, FROM SPILL = 0 |
| BMC50866I | TOTAL INDEXSPACE STRINGS PROCESSED = 0, WHEN NOT FULL = 0, FROM SPILL = 0 |
| BMC50866I | TOTAL INDEXSPACE QUEUE STRINGS PROCESSED = 0, WHEN NOT FULL = 0, FROM SPILL = 0 |
| BMC50859I | LOG RECORD STORE STATISTICS: MEMORY AVAILABLE = 25600K, MEMORY USED = 6120K |
| BMC50860I | TOTAL WAIT TIME FOR LOG RECORD STORE MEMORY = 0.000 DATA = 0.000 INDEX = 0.000 |
| BMC50860I | RIDMAP: PART 0001: ROWS=40592, PAIRS=40592, STORAGE=344K, MEM WAITS=0, WAIT TIME=0 |
| BMC50860I | RIDMAP: PART 0001: TRANS=0, ADDS=0, DELS=0, SPILL READS=0, WAIT TIME=0 |
| BMC50860I | RIDMAP: PART 0002: ROWS=40120, PAIRS=40120, STORAGE=336K, MEM WAITS=0, WAIT TIME=0 |
| BMC50860I | RIDMAP: PART 0002: TRANS=0, ADDS=0, DELS=0, SPILL READS=0, WAIT TIME=0 |
| BMC50860I | RIDMAP: PART 0003: ROWS=40120, PAIRS=40120, STORAGE=336K, MEM WAITS=0, WAIT TIME=0 |
| BMC50860I | RIDMAP: PART 0003: TRANS=0, ADDS=0, DELS=0, SPILL READS=0, WAIT TIME=0 |
| BMC50860I | RIDMAP: SUMMARY: ROWS=120832, PAIRS=120832, STORAGE=1016K, MEM WAITS=0, WAIT TIME=0 |
| BMC50860I | RIDMAP: SUMMARY: TRANS=0, ADDS=0, DELS=0, SPILL READS=0, WAIT TIME=0 |
| BMC50860I | RIDMAP: SUMMARY: MAX DATASPACES=1, USED DATASPACES=1, RIDMAPMEM=0K, USED MEM=2044K |
| BMC50860I | RIDMAP WRITE REQUESTS = 0, WRITE WAIT TIME = 0.000, READ REQUESTS = 0 READ WAIT TIME = 0.000 |
| BMC50868I | LOGFINAL PHASE COMPLETE. ELAPSED TIME = 00:00:06 |
| BMC50859I | LOG PROCESSING COMPLETED, RC = 0, AT 10/14/2013 17:13:59, ELAPSED TIME = 00:00:22 |
| BMC50943I | 0: ZTIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMB |
| BMC50515I | TABLESPACE LARDBXAB.LARSXABA PARTS = 3 TABLES = 1 SEGSIZE = 0 DSSIZE = 0G PAGESIZE = 4K |
| BMC50516I | PART NACTIVE CARD EXTENTS OBCARD PCOMP KSAVED PSAVED |
| BMC50517I | 1 7410 42373 6 0 0 0 0 |
| BMC50517I | 2 8251 47200 9 0 0 0 0 |
| BMC50517I | 3 5465 31259 13 0 0 0 0 |
| BMC50518I | TABLE LARDBXAB.LART001 ROWAVG NPAGES CARD PCTPAGES |
| BMC50519I | 4 20116 47200 95 |
| BMC50520I | TABLE LARDBXAB.LARX001 FIRST KEY COLUMN = CORP_TIMESTAMP|
| BMC50521I | CLUSTER = N UNIQUE = U COMPRESS = N PAGESIZE = 4K KEYLEN = 10 COCCOUNT = 1 |
| BMC50522I | PART NACTIVE CARD EXTENTS LEVELS |
| BMC50523I | 1 268 42373 1 2 |
| BMC50523I | 2 297 47200 1 2 |
| BMC50523I | 3 198 31259 2 2 |
| BMC50524I | INDEX LARDBXAB.LARX01B FIRST KEY COLUMN = COIL_INTEGER |
| BMC50525I | CLUSTER = N UNIQUE = D COMPRESS = N PAGESIZE = 4K KEYLEN = 5 COCCOUNT = 1 |
| BMC50526I | PART NACTIVE CARD EXTENTS LEVELS |
| BMC50527I | 0 439 120832 1 3 |
| BMC50528I | 02 REAL-TIME-STATISTICS - RESET STATS: FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS |
| BMC50529I | UTILITY EXECUTION COMPLETE, RETURN CODE = 0 |
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

In this example, REORG PLUS performs a SHRLEVEL CHANGE reorganization on a partitioned table space with three partitions. The table has one clustering index and one nonpartitioned secondary index. During the LOGAPPLY phase, if REORG PLUS determines that a longlog condition still exists after the time specified by DELAY has expired, REORG PLUS terminates. Otherwise, the reorganization completes normally.

This example uses the default AUTO for the ICTYPE option. Because there is an individual full copy data set on DASD for each partition, REORG PLUS changes ICTYPE to UPDATE and updates the full copy data sets rather than creating incremental copy data sets.

Message 50041I indicates the status of zIIP processing. For this example, an XBM subsystem ID was specified for SHRLEVEL CHANGE processing. REORG PLUS uses the same XBM subsystem ID for zIIP processing.

Table 77 describes the key command options for this job.

Table 77 Command options used in example 13 (part 1 of 2)

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>SHRLEVEL CHANGE</td>
<td>specifies the reorganization type</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS keeps the original data sets in their original status (which can be RW) through most of the reorganization.</td>
</tr>
<tr>
<td>XBMID XBMA</td>
<td>tells REORG PLUS to use XBM subsystem with the SSID of XBMA</td>
</tr>
<tr>
<td>FASTSWITCH YES</td>
<td>tells REORG PLUS to bypass the VSAM rename process and point the DB2 catalog to the staging data sets</td>
</tr>
<tr>
<td>COPY YES</td>
<td>creates a full and incremental DB2 image of the table space</td>
</tr>
<tr>
<td></td>
<td>SHRLEVEL CHANGE sets REGISTER ALL, so all copies are registered in SYSIBM.SYSCOPY.</td>
</tr>
<tr>
<td>ORDER YES</td>
<td>sorts the data rows</td>
</tr>
<tr>
<td>MAXRO DEFER</td>
<td>tells REORG PLUS to continue applying log records indefinitely, regardless of the value set in any other SHRLEVEL CHANGE option except DEADLINE, which is still checked</td>
</tr>
<tr>
<td></td>
<td>The LOGFINAL phase will not begin until you change this value. If DEADLINE is reached first, the job terminates after issuing message BMC50784E.</td>
</tr>
</tbody>
</table>
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

Chapter 5 Examples of REORG PLUS jobs 543

Longlog term tells REORG PLUS to end the reorganization if the longlog condition still exists after the delay time expires.

Delay 300 tells REORG PLUS to wait for 5 minutes (300 seconds) and test again if it detects a longlog condition. If the longlog condition still exists after the test, REORG PLUS terminates the reorganization.

RIDMapMem specifies the amount of data space memory (in kilobytes) that REORG PLUS can use to store the RID translation maps.

Logmem specifies the amount of data space memory (in kilobytes) needed to hold the internal structures that REORG PLUS uses to store the log records.

Logspill Logspill (1024, 1024) determines the primary and secondary space allocation (in kilobytes) for the spill data sets that hold the log records. Each number can be a nonzero positive integer.

SpillDsnPat tells REORG PLUS to use a particular pattern of variables and text to create a prefix for the spill data set names.

The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.

Spillunit specifies the DASD unit where REORG PLUS can allocate spill data sets.

The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.

Spillstorclas specifies the SMS storage class that REORG PLUS uses to allocate spill data sets.

The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.

Analyze gathers information about the objects that you are reorganizing and provides estimated sizes for SYSREC, SYSUT1, SORTWK, and copy data sets.

Sizepct (150, 150) tells REORG PLUS to allocate 150% of the data set size that it calculated for the UNLOAD, WORK, and SORTWORK files. The increase allows room for the unknown number of updates that are being applied to the data sets.

Dsnpat specifies a pattern of variables and text that REORG PLUS uses to create data set names for dynamic data set allocation, overriding the default pattern.

Table 77 Command options used in example 13 (part 2 of 2)

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longlog term</td>
<td>tells REORG PLUS to end the reorganization if the longlog condition still exists after the delay time expires.</td>
</tr>
<tr>
<td>Delay 300</td>
<td>tells REORG PLUS to wait for 5 minutes (300 seconds) and test again if it detects a longlog condition. If the longlog condition still exists after the test, REORG PLUS terminates the reorganization.</td>
</tr>
<tr>
<td>RIDMapMem</td>
<td>specifies the amount of data space memory (in kilobytes) that REORG PLUS can use to store the RID translation maps.</td>
</tr>
<tr>
<td>Logmem</td>
<td>specifies the amount of data space memory (in kilobytes) needed to hold the internal structures that REORG PLUS uses to store the log records.</td>
</tr>
<tr>
<td>Logspill</td>
<td>Logspill (1024, 1024) determines the primary and secondary space allocation (in kilobytes) for the spill data sets that hold the log records. Each number can be a nonzero positive integer.</td>
</tr>
<tr>
<td>SpillDsnPat</td>
<td>tells REORG PLUS to use a particular pattern of variables and text to create a prefix for the spill data set names. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.</td>
</tr>
<tr>
<td>Spillunit</td>
<td>specifies the DASD unit where REORG PLUS can allocate spill data sets. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.</td>
</tr>
<tr>
<td>Spillstorclas</td>
<td>specifies the SMS storage class that REORG PLUS uses to allocate spill data sets. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.</td>
</tr>
<tr>
<td>Analyze</td>
<td>gathers information about the objects that you are reorganizing and provides estimated sizes for SYSREC, SYSUT1, SORTWK, and copy data sets.</td>
</tr>
<tr>
<td>Sizepct (150, 150)</td>
<td>tells REORG PLUS to allocate 150% of the data set size that it calculated for the UNLOAD, WORK, and SORTWORK files. The increase allows room for the unknown number of updates that are being applied to the data sets.</td>
</tr>
<tr>
<td>Dsnpat</td>
<td>specifies a pattern of variables and text that REORG PLUS uses to create data set names for dynamic data set allocation, overriding the default pattern.</td>
</tr>
</tbody>
</table>
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

Figure 45 shows the JCL for example 13.

**Figure 45  JCL for example 13**

```plaintext
//
// JOB
// *
// ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// ** PARTITIONED TABLESPACE, CLUSTERING INDEX AND NONPARTITIONED  *
// ** SECONDARY INDEX.                                         *
// ** REORG TABLESPACE WITH SHRLEVEL CHANGE OPTION.            *
// ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// LARREORG EXEC PGM=ARUUMAIN,COND=(7,LT),
// PARM='DEFF,LARCOPB,LARDBXBB,NEW,,MSGLEVEL(1),ARU$OPTS'
// STEPLIB DD DISP=SHR,DSN=ProductLibraries
//       DD DISP=SHR,DSN=DB2.DSNEXIT
//       DD DISP=SHR,DSN=DB2.DSNLOAD
// *
// SYSPRINT DD SYSOUT=*  
// SYSPUT DD SYSOUT=*  
// UTPRINT DD SYSOUT=*  
// SYSTERM DD SYSOUT=* 
// *
// SYSIN DD *
REORG TABLESPACE LARDBXBB.LARSXBBA
  SHRLEVEL CHANGE
  XBMID XBMID
  FASTSWITCH YES
  COPY YES
  ORDER YES
  MAXRO DEFER
  LONGLOG TERM
  DELAY 300
  RIDMAPMEM 20480
  LOGMEM 30720
  LOGSPILL (1024,1024)
  SPILLSNPAT 'ARU.LARWORK.EX13'
  SPILLUNIT NONE
  SPILLSTORCLAS COPYCLAS
  ANALYZE
  DDTYPE UNLOAD   UNIT (WORK) SIZEPCT (150,150)
  DSNPAT 'ARU.EXMPL13.&DB.&TSIX.&DDNAME'
  DDTYPE WORK  UNIT (WORK) SIZEPCT (150,150)
  DSNPAT 'ARU.EXMPL13.&DB.&TSIX.&DDNAME'
  DDTYPE LOCPFCPY UNIT (WORK) SIZEPCT (150,150)
  DSNPAT 'ARU.EXMPL13.&DB.&TSIX.&DDNAME'
/*
```
Figure 46 shows the SYSPRINT output for example 13.

**Figure 46  SYSPRINT for example 13 (part 1 of 9)**

<table>
<thead>
<tr>
<th>BMC50471</th>
<th>REORG PLUS for DB2 -- V11R1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762</td>
<td></td>
</tr>
<tr>
<td>(C) COPYRIGHT 1988 - 2013 BMC SOFTWARE, INC.</td>
<td></td>
</tr>
</tbody>
</table>

Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

**Figure 46 SYSPRINT for example 13 (part 2 of 9)**

<table>
<thead>
<tr>
<th>BMC04701 VOLLMT</th>
<th>(25,25)</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC04701 AVGVOLSP</td>
<td>(30000,TRK),(30000,TRK)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 DSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC04701 DATACLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC04701 GMTCALAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC04701 STORCLAS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC04701 THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC04701 MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 EXPT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 RETPD</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 GDGUNIT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 GDEMPFY</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 GDGSCRAT</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 DOTYPE</td>
<td>ARCHIVE</td>
<td>LDCPFCPY</td>
</tr>
<tr>
<td>BMC04701 ACTIVE</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>BMC04701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC04701 ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC04701 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC04701 thresHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC04701 MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMC04701 EXPOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC04701 RETPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC04701 GDGUNIT</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMC04701 GDEMPFY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 dotYPE</td>
<td>LDCCFCPY</td>
<td>LDCCFCPY</td>
</tr>
<tr>
<td>BMC04701 ACTIVE</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC04701 ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC04701 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC04701 thresHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC04701 MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMC04701 EXPOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC04701 RETPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC04701 GDGUNIT</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMC04701 GDEMPFY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 dotYPE</td>
<td>REWFCP</td>
<td>REWFCP</td>
</tr>
<tr>
<td>BMC04701 ACTIVE</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC04701 ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC04701 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC04701 thresHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC04701 MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
<tr>
<td>BMC04701 EXPOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC04701 RETPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC04701 GDGUNIT</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMC04701 GDEMPFY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 dotYPE</td>
<td>REWFCP</td>
<td>REWFCP</td>
</tr>
<tr>
<td>BMC04701 ACTIVE</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC04701 ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC04701 SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC04701 UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC04701 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC04701 thresHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC04701 MAXEXTSZ</td>
<td>(0,K),(0,K)</td>
<td>(0,K),(0,K)</td>
</tr>
</tbody>
</table>

546 REORG PLUS for DB2 Reference Manual
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

Chapter 5 Examples of REORG PLUS jobs

BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I DDTYPE = SYSPUNCH
BMC50470I ACTIVE = YES
BMC50470I IFALLOC = USE
BMC50470I ALLOC = N/A
BMC50470I SMS = NO
BMC50470I SMSUNIT = NO
BMC50470I SIZEPCT = (100,100)
BMC50470I UNIT = (SYSALLDA,SYSCALLDA)
BMC50470I UNITCNT = (0,0)
BMC50470I VOLCNT = (25,25)
BMC50470I AVGVOLSZ = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I DDTYPE = LOCP punch
BMC50470I ACTIVE = NO
BMC50470I IFALLOC = USE
BMC50470I ALLOC = N/A
BMC50470I SMS = NO
BMC50470I SMSUNIT = NO
BMC50470I SIZEPCT = (100,100)
BMC50470I UNIT = (SYSALLDA,SYSCALLDA)
BMC50470I UNITCNT = (0,0)
BMC50470I VOLCNT = (25,25)
BMC50470I AVGVOLSZ = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I DDTYPE = REPL punch
BMC50470I ACTIVE = NO
BMC50470I IFALLOC = USE
BMC50470I ALLOC = N/A
BMC50470I SMS = NO
BMC50470I SMSUNIT = NO
BMC50470I SIZEPCT = (100,100)
BMC50470I UNIT = (SYSALLDA,SYSCALLDA)
BMC50470I UNITCNT = (0,0)
BMC50470I VOLCNT = (25,25)
BMC50470I AVGVOLSZ = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT =
BMC50470I RETPD =
BMC50470I DDTYPE = OSNPAT=UID..BMC.&TSIX.&DDNAME
BMC50483I UNLOAD

Figure 46 SYSPRINT for example 13 (part 3 of 9)
### Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

**Example 46**  
**SYSPRINT for example 13 (part 4 of 9)**

<table>
<thead>
<tr>
<th>BMC50483I</th>
<th>WORK</th>
<th>DSNPAT='UID..BMC..AT5IX..ADDNAME'</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50483I</td>
<td>SORTWORK</td>
<td>DSNPAT='UID..UTILPFX..ADDNAME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>ARCHIVE</td>
<td>DSNPAT='UID..UTILPFX..ADDNAME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>SYSPUNCH</td>
<td>DSNPAT='UID..UTILPFX..ADDNAME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCFCOPY</td>
<td>DSNPAT='UID..ADDNAME..AT5IX..F&amp;PART..TIME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCFCOPY</td>
<td>DSNPAT='UID..ADDNAME..AT5IX..F&amp;PART..TIME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCICOPY</td>
<td>DSNPAT='UID..ADDNAME..AT5IX..F&amp;PART..TIME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCICOPY</td>
<td>DSNPAT='UID..ADDNAME..AT5IX..F&amp;PART..TIME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCPCOPY</td>
<td>DSNPAT='UID..ADDNAME..AT5IX..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCBLCPY</td>
<td>DSNPAT='UID..ADDNAME..AT5IX..T&amp;TIME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMFCOPY</td>
<td>DSNPAT='UID..ADDNAME..AT5IX..F&amp;PART..TIME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMICOPY</td>
<td>DSNPAT='UID..ADDNAME..AT5IX..F&amp;PART..TIME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMICOPY</td>
<td>DSNPAT='UID..ADDNAME..AT5IX..F&amp;PART..TIME'</td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMICOPY</td>
<td>DSNPAT='UID..ADDNAME..AT5IX..F&amp;PART..TIME'</td>
</tr>
</tbody>
</table>

**DB2 DSNDEEP MODULE SETTINGS:**

<table>
<thead>
<tr>
<th>BMC50471I</th>
<th>VERSION</th>
<th>1010</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471I</td>
<td>SUBSYSTEM DEFAULT</td>
<td>DEFF</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>CHARACTER SET</td>
<td>ALPHANUM</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DATE FORMAT</td>
<td>USA</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>TIME FORMAT</td>
<td>USA</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LOCAL DATE LENGTH</td>
<td>0</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LOCAL TIME LENGTH</td>
<td>0</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DECIMAL POINT</td>
<td>PERIOD</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DECIMAL ARITHMETIC</td>
<td>15</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DELIMITER</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>SQL DELIMITER</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ENCODING SCHEME</td>
<td>EBCDIC</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>APPL. ENCODING SCHEME</td>
<td>EBCDIC</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>MIXED</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>EBCDIC CCSID</td>
<td>(37,65534,65534)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ASCII CCSID</td>
<td>(819,65534,65534)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>UNICODE CCSID</td>
<td>(367,1208,1200)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>IMPLICIT TIME ZONE</td>
<td>CURRENT (-05:00)</td>
</tr>
</tbody>
</table>

**REORG TABLESPACE LARDBXBB.LARSXBBA**

<table>
<thead>
<tr>
<th>BMC50471I</th>
<th>SHRLLEVEL CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471I</td>
<td>XBMID XBMB</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>FASTSWITCH YES</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>COPY YES</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ORDER YES</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>MAXRO DEFER</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LONGLOG TERM</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DELAY 300</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>RIDMAPMEM 20480</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LOGMEM 30720</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LOGSPILL (1024,1024)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>SPILLSPAT 'ARU.LARWORK.EX13'</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>SPILLUNIT NONE</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>SPILLSRCCLASS COPYCLAS</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ANALYZE</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ODTYPE UNLOAD UNIT (WORK) SIZEPCT (150,150)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>ODTYPE UNDOUNIT UNIT (WORK) SIZEPCT (150,150)</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSNPAT 'ARU.EXMPL13.AOB..AT5IX..ADDNAME'</td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DSNPAT 'ARU.EXMPL13.AOB..AT5IX..ADDNAME'</td>
</tr>
</tbody>
</table>

Figure 46  SYSPRINT for example 13 (part 4 of 9)
Figure 46  SYSPRINT for example 13 (part 6 of 9)
## Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

### Figure 46  SYSPRINT for example 13 (part 7 of 9)

| BMC50718I | 103: | Page Reads | 2561 | Page Writes | 55 |
| BMC50718I | 103: | Write 1/0 Requests | 56 | New Pages | 4366 |
| BMC50716I | 103: | Write 1/0 time | 0:00:00 | Read 1/0 time | 0:00:02 |
| BMC50766I | 103: | LOG APPLY TASK ENDED AT 10/14/2013 17:34:41, ELAPSED TIME = 00:01:28 |
| BMC50716I | 104: | Buffer Manager Statistics |
| BMC50717I | 104: | DATASET=DEFCAT.DSNDBC.LARDBXBB.LARXBBB.J0001.A003 |
| BMC50718I | 104: | Page Requests | 159833 | Page Hits | 152928 |
| BMC50718I | 104: | Page Reads | 2609 | Page Writes | 54 |
| BMC50718I | 104: | Write 1/0 Requests | 55 | New Pages | 4296 |
| BMC50716I | 104: | Write 1/0 time | 0:00:00 | Read 1/0 time | 0:00:02 |
| BMC50786I | 104: | LOG APPLY TASK ENDED AT 10/14/2013 17:34:41, ELAPSED TIME = 00:01:28 |
| BMC50770I | 102: | 792 NEW LOG APPLY BUFFERS, 13398 REUSED, 5415 INITIAL MAXIMUM, 0 WAITS, BUFFER SIZE=8380 |
| BMC50743I | 102: | LOGAPPLY STATISTICS |
| BMC50742I | 102: | PROCESSED TRANSACTIONS |
| BMC50741I | 102: | INSERTS: 8256 INSERTS(COMPENSATION): 0 |
| BMC50745I | 102: | UPDATES: 4300 UPDATES(COMPENSATION): 0 |
| BMC50746I | 102: | DELETE: 860 DELETE(COMPENSATION): 0 |
| BMC50743I | 102: | CONSOLIDATED TRANSACTIONS |
| BMC50744I | 102: | INSERTS: 8256 INSERTS(COMPENSATION): 0 |
| BMC50745I | 102: | UPDATES: 2724 UPDATES(COMPENSATION): 0 |
| BMC50746I | 102: | DELETES: 860 DELETE(COMPENSATION): 0 |
| BMC50747I | 102: | PAGES UPDATED COUNTERS FOR LARDBXBB.LARXBBB |
| BMC50748I | 102: | PART TOTAL PAGES DATA PAGES SPACE MAPS HEADER PAGES |
| BMC50749I | 102: | 1 6954 6951 2 1 |
| BMC50716I | 102: | Buffer Manager Statistics |
| BMC50717I | 102: | DATASET=DEFCAT.DSNDBC.LARDBXBB.LARXBBB.J0001.A001 |
| BMC50718I | 102: | Page Requests | 149338 | Page Hits | 142284 |
| BMC50718I | 102: | Page Reads | 2642 | Page Writes | 55 |
| BMC50718I | 102: | Write 1/0 Requests | 56 | New Pages | 4312 |
| BMC50716I | 102: | Write 1/0 time | 0:00:00 | Read 1/0 time | 0:00:02 |
| BMC50786I | 102: | LOG APPLY TASK ENDED AT 10/14/2013 17:34:42, ELAPSED TIME = 00:01:30 |
| BMC50786I | 111: | INLINE IMAGE COPY APPEND TASK ENDED AT 10/14/2013 17:34:42, ELAPSED TIME = 00:01:33 |
| BMC50786I | 109: | LOG RECORD SORT TASK ENDED AT 10/14/2013 17:34:42, ELAPSED TIME = 00:01:33 |
| BMC50794I | 106: | INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXBB.LARX001A DSNUM 2 |
| BMC50794I | 106: | PAGE READS: 27537 KEY INSERTS: 8160 |
| BMC50791I | 106: | RID INSERTS: 0 RID DELETES(P): 850 |
| BMC50791I | 106: | ROOT SPLITS: 0 NON-LEAF SPLITS: 0 |
| BMC50791I | 106: | LEAF SPLITS: 84 PAGE FREES: 0 |
| BMC50791I | 106: | NEW PAGES: 84 BUFF PAGE READS: 84 |
| BMC50716I | 106: | Buffer Manager Statistics |
| BMC50717I | 106: | DATASET=DEFCAT.DSNDBC.LARDBXBB.LARX001A.J0001.A003 |
| BMC50718I | 106: | Page Requests | 35868 | Page Hits | 35550 |
| BMC50718I | 106: | Page Reads | 256 | Page Writes | 3 |
| BMC50718I | 106: | Write 1/0 Requests | 4 New Pages | 62 |
| BMC50716I | 106: | Write 1/0 time | 0:00:00 Read 1/0 time | 0:00:00 |
| BMC50786I | 106: | INDEX APPLY TASK ENDED AT 10/14/2013 17:34:43, ELAPSED TIME = 00:01:30 |
| BMC50744I | 106: | INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXBB.LARX001A DSNUM 3 |
| BMC50794I | 107: | PAGE READS: 27537 KEY INSERTS: 8160 |
| BMC50791I | 107: | RID INSERTS: 0 RID DELETES(P): 850 |
| BMC50791I | 107: | ROOT SPLITS: 0 NON-LEAF SPLITS: 0 |
| BMC50791I | 107: | LEAF SPLITS: 84 PAGE FREES: 0 |
| BMC50791I | 107: | NEW PAGES: 84 BUFF PAGE READS: 84 |
| BMC50716I | 107: | Buffer Manager Statistics |
| BMC50717I | 107: | DATASET=DEFCAT.DSNDBC.LARDBXBB.LARX001A.J0001.A002 |
| BMC50718I | 107: | Page Requests | 35868 | Page Hits | 35550 |
| BMC50718I | 107: | Page Reads | 256 | Page Writes | 3 |
| BMC50718I | 107: | Write 1/0 Requests | 4 New Pages | 62 |
| BMC50716I | 107: | Write 1/0 time | 0:00:00 Read 1/0 time | 0:00:00 |
| BMC50786I | 107: | INDEX APPLY TASK ENDED AT 10/14/2013 17:34:43, ELAPSED TIME = 00:01:30 |
| BMC50794I | 107: | INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXBB.LARX001B |
| BMC50794I | 108: | PAGE READS: 130527 KEY INSERTS: 0 |
| BMC50791I | 108: | RID INSERTS: 26725 RID DELETES(P): 5120 |
| BMC50791I | 108: | ROOT SPLITS: 0 NON-LEAF SPLITS: 1 |
| BMC50791I | 108: | LEAF SPLITS: 131 PAGE FREES: 0 |
| BMC50791I | 108: | NEW PAGES: 132 BUFF PAGE READS: 263 |
| BMC50716I | 108: | Buffer Manager Statistics |
| BMC50717I | 108: | DATASET=DEFCAT.DSNDBC.LARDBXBB.LARX001B.J0001.A001 |
| BMC50718I | 108: | Page Requests | 158097 | Page Hits | 157561 |
| BMC50718I | 108: | Page Reads | 443 | Page Writes | 5 |
| BMC50718I | 108: | Write 1/0 Requests | 6 New Pages | 93 |
| BMC50716I | 108: | Write 1/0 time | 0:00:00 Read 1/0 time | 0:00:00 |
| BMC50786I | 108: | INDEX APPLY TASK ENDED AT 10/14/2013 17:34:45, ELAPSED TIME = 00:01:30 |
| BMC50794I | 108: | INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXBB.LARX001A DSNUM 1 |
| BMC50794I | 108: | PAGE READS: 27861 KEY INSERTS: 8256 |
| BMC50791I | 108: | RID INSERTS: 0 RID DELETES(P): 860 |
| BMC50791I | 108: | ROOT SPLITS: 0 NON-LEAF SPLITS: 0 |

Chapter 5  Examples of REORG PLUS jobs  551
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

Figure 46  SYSPRINT for example 13 (part 8 of 9)
EXAMPLE 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

In this example, REORG PLUS performs a SHRLEVEL CHANGE reorganization of a partitioned table space that uses table-controlled partitioning and has data partitioned secondary (DPSI) and nonpartitioned secondary (NPSI) indexes defined.

The REBALANCE command option tells REORG PLUS to define new partition boundaries and evenly redistribute rows across the reorganized partitions. The JCL also includes a DD statement for a DDLOUT data set. This DD statement tells REORG PLUS to write the ALTER statements that REORG PLUS used for rebalancing to this data set. Figure 49 on page 564 shows the contents of the DDLOUT data set for this example.

Message 50041I indicates the status of zIIP processing. For this example, an XBM subsystem ID was specified for SHRLEVEL CHANGE processing. REORG PLUS uses the same XBM subsystem ID for zIIP processing.

Table 78 describes the key command options for this job.

---

**Table 78  Key command options used in example 14  (part 1 of 2)**

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG TABLESPACE</td>
<td>specifies that the table space named in the statement is to be reorganized</td>
</tr>
<tr>
<td>ORDER YES</td>
<td>sorts the data rows</td>
</tr>
<tr>
<td>REBALANCE</td>
<td>defines new partition boundaries and evenly redistributes rows across the reorganized partitions</td>
</tr>
<tr>
<td>COPY YES</td>
<td>creates a DB2 image copy of the table space</td>
</tr>
<tr>
<td>ICTYPE INCREMENTAL</td>
<td>tells REORG PLUS to create an incremental image copy instead of updating the full image copy</td>
</tr>
</tbody>
</table>

---

Because the INLINECP installation option is YES, REORG PLUS would normally create an inline image copy. However, in this example, REORG PLUS changes INLINECP=YES to INLINECP=NO because the ICTYPE is INCREMENTAL.
Table 78  Key command options used in example 14 (part 2 of 2)

<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE SAMPLE</td>
<td>gathers information about the objects that you are reorganizing and provides estimated sizes for SYSREC, SYSUT1, SORTWK, and copy data sets. SAMPLE tells REORG PLUS to read the minimum number of pages needed to determine a reasonable estimate for the cardinality.</td>
</tr>
<tr>
<td>PREFORMAT YES</td>
<td>REORG PLUS preformats the unused pages of the data set, overriding the default. After REORG PLUS reorganizes the data and indexes, it writes pages that have been initialized with zeros, up to the high-allocated relative byte address (RBA) of the table space and index spaces that were reorganized.</td>
</tr>
<tr>
<td>BMCSTATS YES</td>
<td>tells REORG PLUS to update statistics in the DASD MANAGER PLUS database statistics tables and the DB2 catalog.</td>
</tr>
<tr>
<td>UPDATEDB2STATS YES</td>
<td>The JCL for this example also includes a DD statement for the ASUSRPRT data set. This DD statement tells REORG PLUS not to print the statistics from the Common Statistics component to SYSPRINT, but to send them to a separate data set.</td>
</tr>
<tr>
<td>SIZEPCT (150,150)</td>
<td>tells REORG PLUS to allocate 150% of the data set size that it calculated for the local primary full copy data set.</td>
</tr>
<tr>
<td>DSNPAT</td>
<td>specifies a pattern of variables and text that REORG PLUS uses to create data set names for dynamic data set allocation, overriding the default pattern.</td>
</tr>
</tbody>
</table>

Figure 47 shows the JCL for example 14.

---

**Figure 47**  JCL for example 14 (part 1 of 2)

```plaintext
// ** JOB  
// ** PARTITIONED TABLESPACE THAT USES TABLE-BASED PARTITIONING.  
// ** DATA-PARTITIONED SECONDARY INDEX (DPSI) AND NON-PARTITIONED  
// ** SECONDARY (NPSI) INDEXES DEFINED.  
// LARREORG EXEC PGM=ARUUMAIN,REGION=0M,COND=(7,LT),  
// PARM='DEFF,ARUEXP14.RG1,NEW,,MSGLEVEL(1),ARU$OPTS'  
// STEPLIB DD DISP=SHR,DSN=ProductLibraries  
// DD DISP=SHR,DSN=DB2.DSNEXIT  
// DD DISP=SHR,DSN=DB2.DSNLOAD  
// SYSPRINT DD SYSOUT=*  
// SYSSOUT DD SYSOUT=*  
// UTPRINT DD SYSOUT=*  
// SYSTERM DD SYSOUT=*  
// DDLOUT DD SYSOUT=*  
```
Figure 47  JCL for example 14 (part 2 of 2)

```plaintext
//ASUSRPRRT DD SYSOUT=*  
//SYSSIN DD *  
REORG TABLESPACE ARUEXP14.LARS$XBA  
SHRLEVEL CHANGE  
XBMID XBMB  
ORDER YES  
REBALANCE  
COPY YES  
ICTYPE INCREMENTAL  
LOGTHRESHLD 1000  
DELAY 300  
LOGSPILL (20480,10240)  
SPILLDSPAT 'ARU.LARWORK.DEFF'  
SPILLUNIT NONE  
SPILLSTORCLASS COPYCLAS  
ANALYZE SAMPLE  
PREFORMAT YES  
UPDATEDB2STATS YES  
BMCSTATS YES  
FASTSWITCH YES  
DDTYPE WORK UNIT (WORK) SIZEPCT (150,150)  
DSNPAT 'ARU.EXMPL14.&DB.&TSIX.&DDNAME'  
DDTYPE LOCPFCPY UNIT (WORK) SIZEPCT (150,150)  
DSNPAT 'ARU.EXMPL14.&DB.&TSIX.&DDNAME'  
DDTYPE LOCPICPY UNIT (WORK)  
DSNPAT 'ARU.EXMPL14.&DB.&TSIX.&DDNAME'  
/*
```

Figure 48 shows the SYSPRINT output for example 14.

Figure 48  SYSPRINT for example 14 (part 1 of 10)
### Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

**SYSPRINT for example 14 (part 2 of 10)**

<table>
<thead>
<tr>
<th>BMC50471</th>
<th>ARCHFORMAT=HMC</th>
<th>INLINEP=YES</th>
<th>SHORTMEMORY=CONTINUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471</td>
<td>AVAILPAGEPCT=100</td>
<td>IXONEX=NO</td>
<td>SIXSNAP=NO</td>
</tr>
<tr>
<td>BMC50471</td>
<td>BLIDMAX=3000</td>
<td>IXRANDOM=NO</td>
<td>SMAX=0</td>
</tr>
<tr>
<td>BMC50471</td>
<td>BMXIST=YES</td>
<td>KEPTDICTIONARY=NO</td>
<td>SMCORE=(OK,OK)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>CBUFSIZE=30</td>
<td>LEAFDSMN=200</td>
<td>SORTDEVT=(SYSALLDA)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>CONDEXEC=NO</td>
<td>LBDB=YES</td>
<td>SORTNO=32</td>
</tr>
<tr>
<td>BMC50471</td>
<td>COPYDOM=(BCMCY,BMCCL2)</td>
<td>LOCKRNM=YES</td>
<td>SPILDSNP=40</td>
</tr>
<tr>
<td>BMC50471</td>
<td>COPYL=PART</td>
<td>LOGDFINAL=NONE</td>
<td>SPILSCLS=NONE</td>
</tr>
<tr>
<td>BMC50471</td>
<td>COPYMAX=1000</td>
<td>LOGMEM=B</td>
<td>SPUNIT=WORK</td>
</tr>
<tr>
<td>BMC50471</td>
<td>COPYSUBSET=NO</td>
<td>LOGSPIL=(20000,10000)</td>
<td>SPOOL=3</td>
</tr>
<tr>
<td>BMC50471</td>
<td>CPYFAIL=TERM</td>
<td>LOGTHRSH=0</td>
<td>SORTRETRY=100</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DATACAP=NO</td>
<td>LONGBLOCK=CONTINUE</td>
<td>STAGESN=NONE</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DOLDDOM=DOLIN</td>
<td>LONGMAPTRUNC=MIDDLE</td>
<td>STPROCMT=YES</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DEADLINE=NONE</td>
<td>MAXEXIT=(NONE,REXX)</td>
<td>STOPDELAY=1</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DELFILE=YES</td>
<td>MAXREP=300</td>
<td>STOPTRY=300</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DESCODE=(3,7)</td>
<td>MAXSORTMEMORY=O</td>
<td>TASKMAX=1000</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DRATYPE=ALL</td>
<td>MAXTAPE=3</td>
<td>TERMEXIT=(NONE,REXX)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DONTDELAY=1</td>
<td>MGEXTENT=CONTINUE</td>
<td>TIMEOUT=TERM</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DONTRETRY=255</td>
<td>MINSORTMEMORY=O</td>
<td>TOTALPAGEPCT=0</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DONTWAIT=NONE</td>
<td>MSLEVEL=1</td>
<td>TSAMPLEPCT=100</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DONTEXIT=(NONE,ASM)</td>
<td>OFFPAGE=10</td>
<td>UBBFST=20</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DSNUTILB=YES</td>
<td>OMPDB2ID=YES</td>
<td>UNLD=SYSSREC</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DSNDB=DEBFAIL</td>
<td>ORIGDISP=DELETE</td>
<td>UNLJMK=3000</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DSREEXIT=(NONE,REXX)</td>
<td>PENDOM=YES</td>
<td>UNLOAD=RELOAD</td>
</tr>
<tr>
<td>BMC50471</td>
<td>EXCLNUM=(87,82,106)</td>
<td>PREFORMAT=NO</td>
<td>UTLB_COLCONS=UTLB</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FASTMATCH=NO</td>
<td>RCVCMD=(BMC1R,BMC1R2)</td>
<td>UTILB_NULLIX=UTLB</td>
</tr>
</tbody>
</table>
| BMC50471 | FILE CHK=WRN | RCVYDOM=(BMC1RY,BMC1RZ) | UXTAGE=1 

### BMC50471 I PLAIN-ARUQA

<table>
<thead>
<tr>
<th>BMC50471</th>
<th>DTYPE = UNLOAD</th>
<th>WORK</th>
<th>SORTWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471</td>
<td>ACTIVE = YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>BMC50471</td>
<td>IFALOC = USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50471</td>
<td>ALLOC = N/A</td>
<td>N/A</td>
<td>ANY</td>
</tr>
<tr>
<td>BMC50471</td>
<td>SMS = NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50471</td>
<td>SMUNIT = NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50471</td>
<td>SIZEPCT = (100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNIT = (SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNITCNT = (0,0)</td>
<td>(0,0)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>VOLTENT = (25,25)</td>
<td>(25,25)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGVOLSP = ((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DSNTYPE = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DATACLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MGMTCLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>STORCLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>THRESHLD = 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MAXEXTSZ = ((0,K),(0,K))</td>
<td>((0,K),(0,K))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>EXPDT = N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MAXREXEC=0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MAXREPT=0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MAXREDEF=0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MAXREUSE=0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MAXREUSE=0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MAXREUSE=0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### BMC50471 I DTYPE = ARCHIVE

<table>
<thead>
<tr>
<th>BMC50471</th>
<th>DTYPE = ARCHIVE</th>
<th>LDPFCPY</th>
<th>LDPFCPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471</td>
<td>ACTIVE = NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>BMC50471</td>
<td>IFALOC = USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50471</td>
<td>ALLOC = N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>SMS = NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50471</td>
<td>SMUNIT = NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50471</td>
<td>SIZEPCT = (100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNIT = (SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>UNITCNT = (0,0)</td>
<td>(0,0)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>VOLTENT = (25,25)</td>
<td>(25,25)</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>AVGVOLSP = ((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DSNTYPE = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>DATACLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>MGMTCLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>STORCLAS = (NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>THRESHLD = 0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 48  SYSPRINT for example 14 (part 3 of 10)

<table>
<thead>
<tr>
<th>BMC504701 MAXEXTSZ</th>
<th>((0,K),(0,K))</th>
<th>((0,K),(0,K))</th>
<th>((0,K),(0,K))</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC504701 EXPOT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC504701 RETPD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC504701 DDTYPE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC504701 DDTYPE</td>
<td>LOCBFCPY</td>
<td>LOCBFCPY</td>
<td>REMBFCPY</td>
</tr>
<tr>
<td>BMC504701 ACTIVE</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 IFALLOC</td>
<td>USE</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC504701 GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 GDGLIMIT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC504701 RETPD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC504701 GDGLIMIT</td>
<td>(5,5)</td>
<td>(5,5)</td>
<td>(5,5)</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(sysALLDA,sysALLDA)</td>
<td>(sysALLDA,sysALLDA)</td>
<td>(sysALLDA,sysALLDA)</td>
</tr>
<tr>
<td>BMC504701 AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC504701 SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC504701 UNIT</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
<td>(SYSLDLA,SYSLDLA)</td>
</tr>
<tr>
<td>BMC504701 UNICTNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
</tbody>
</table>

Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

Chapter 5 Examples of REORG PLUS jobs 557
### Figure 48 SYSPRINT for example 14 (part 4 of 10)

<table>
<thead>
<tr>
<th>BMC50470I</th>
<th>ACTIVE</th>
<th>NO</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I</td>
<td>IFALLOC</td>
<td>USE</td>
<td>USE</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>ALLOC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SMS</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SMSUNIT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNIT</td>
<td>(SYSALLDA, SYSALLDA)</td>
<td>(SYSALLDA, SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>VOLCNT</td>
<td>(25,25)</td>
<td>(25,25)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>AVGVOLSP</td>
<td>(((30000,TRK), (30000,TRK)), ((30000,TRK), (30000,TRK))</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DSNDOTYPE</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DATACLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MHGMTCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>STORCLAS</td>
<td>(NONE, NONE)</td>
<td>(NONE, NONE)</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>THRESHLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>MAXEXTSZ</td>
<td>((0,K), (0,K))</td>
<td>((0,K), (0,K))</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>EXPDT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>RETPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>GDGLIMIT</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>GDGEMPTY</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>GDGSCRAT</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>DDTYPE</td>
<td>REMPLCPY</td>
<td>REMBLCPY</td>
</tr>
<tr>
<td>BMC50470I</td>
<td>UNLOAD</td>
<td>DSNPAT= &amp;UID..BMC..&amp;TSIX..&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50470I</td>
<td>WORK</td>
<td>DSNPAT= &amp;UID..BMC..&amp;TSIX..&amp;DDNAME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCFCOPY</td>
<td>DSNPAT= &amp;UID..AT66..FAPART..&amp;TTIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCICOPY</td>
<td>DSNPAT= &amp;UID..AT66..FAPART..&amp;TTIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCFCOPY</td>
<td>DSNPAT= &amp;UID..AT66..FAPART..&amp;TTIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCICOPY</td>
<td>DSNPAT= &amp;UID..AT66..FAPART..&amp;TTIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>LOCLCFC</td>
<td>DSNPAT= &amp;UID..AT66..FAPART..&amp;TTIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMFPCPY</td>
<td>DSNPAT= &amp;UID..AT66..FAPART..&amp;TTIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMIPCPY</td>
<td>DSNPAT= &amp;UID..AT66..FAPART..&amp;TTIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMFCOPY</td>
<td>DSNPAT= &amp;UID..AT66..FAPART..&amp;TTIME</td>
<td></td>
</tr>
<tr>
<td>BMC50483I</td>
<td>REMIPCPY</td>
<td>DSNPAT= &amp;UID..AT66..FAPART..&amp;TTIME</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DB2 DSNHDECP MODULE SETTINGS:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>VERSION</td>
<td>1010</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>CHARACT SET</td>
<td>ALPHANUM</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DATE FORMAT</td>
<td>USA</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>TIME FORMAT</td>
<td>USA</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LOCAL DATE LENGTH</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>LOCAL TIME LENGTH</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DECIMAL POINT</td>
<td>PERIOD</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DECIMAL ARITHMETIC</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>DELIMITER</td>
<td>DEFAULT</td>
<td></td>
</tr>
<tr>
<td>BMC50471I</td>
<td>SQL DELIMITER</td>
<td>DEFAULT</td>
<td></td>
</tr>
</tbody>
</table>
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

Chapter 5 Examples of REORG PLUS jobs 559
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

Figure 48 SYSPRINT for example 14 (part 6 of 10)
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

Chapter 5 Examples of REORG PLUS jobs 561
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

**Figure 48** SYSPRINT for example 14 (part 8 of 10)

<table>
<thead>
<tr>
<th>BMC50744I</th>
<th>102</th>
<th>INSERTS</th>
<th>0</th>
<th>INSERTS(COMPENSATION):</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50745I</td>
<td>102</td>
<td>UPDATES</td>
<td>0</td>
<td>UPDATES(COMPENSATION):</td>
<td>0</td>
</tr>
<tr>
<td>BMC50746I</td>
<td>102</td>
<td>DELETES</td>
<td>0</td>
<td>DELETES(COMPENSATION):</td>
<td>0</td>
</tr>
<tr>
<td>BMC50747I</td>
<td>102</td>
<td>PAGE UPDATES COUNTERS FOR ARUEXP14.LARS$XBA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50748I</td>
<td>102</td>
<td>PART TOTAL PAGES</td>
<td>1</td>
<td>SPACE MAPS</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>HEADER PAGES</td>
<td>1</td>
</tr>
<tr>
<td>BMC50749I</td>
<td>102</td>
<td>Page Requests</td>
<td>9</td>
<td>Page Hits</td>
<td>0</td>
</tr>
<tr>
<td>BMC50750I</td>
<td>102</td>
<td>Write I/O Requests</td>
<td>0</td>
<td>New Pages</td>
<td>0</td>
</tr>
<tr>
<td>BMC50752I</td>
<td>102</td>
<td>Write I/O time</td>
<td>0:00:00</td>
<td>Read I/O time</td>
<td>0:00:00</td>
</tr>
<tr>
<td>BMC50754I</td>
<td>102</td>
<td>LOG APPL TASK ENDED AT</td>
<td>10/14/2013 17:42:05,</td>
<td>ELAPSED TIME = 00:00:15</td>
<td></td>
</tr>
<tr>
<td>BMC50756I</td>
<td>102</td>
<td>BUFFER MANAGER STATISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50757I</td>
<td>104</td>
<td>PAGE READS</td>
<td>0</td>
<td>BUFF PAGE READS</td>
<td>0</td>
</tr>
<tr>
<td>BMC50759I</td>
<td>104</td>
<td>RID INSERTS</td>
<td>0</td>
<td>RID DELETES (PS):</td>
<td>0</td>
</tr>
<tr>
<td>BMC50760I</td>
<td>104</td>
<td>ROOT SPLITS</td>
<td>0</td>
<td>NON-LEAF SPLITS:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50761I</td>
<td>104</td>
<td>LEAF SPLITS</td>
<td>0</td>
<td>PAGE FREES:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50762I</td>
<td>104</td>
<td>NEW PAGES</td>
<td>0</td>
<td>BUFFER PAGE READS:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50764I</td>
<td>104</td>
<td>INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX03A DSNUM 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50765I</td>
<td>104</td>
<td>PAGE READS</td>
<td>0</td>
<td>KEY INSERTS:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50766I</td>
<td>104</td>
<td>RID INSERTS</td>
<td>0</td>
<td>RID DELETES (PS):</td>
<td>0</td>
</tr>
<tr>
<td>BMC50767I</td>
<td>104</td>
<td>ROOT SPLITS</td>
<td>0</td>
<td>PAGE FREES:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50769I</td>
<td>104</td>
<td>NEW PAGES</td>
<td>0</td>
<td>BUFFER PAGE READS:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50771I</td>
<td>104</td>
<td>INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX03E DSNUM 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50772I</td>
<td>104</td>
<td>PAGE READS</td>
<td>0</td>
<td>KEY INSERTS:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50773I</td>
<td>104</td>
<td>RID INSERTS</td>
<td>0</td>
<td>RID DELETES (PS):</td>
<td>0</td>
</tr>
<tr>
<td>BMC50774I</td>
<td>104</td>
<td>ROOT SPLITS</td>
<td>0</td>
<td>PAGE FREES:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50776I</td>
<td>104</td>
<td>NEW PAGES</td>
<td>0</td>
<td>BUFFER PAGE READS:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50778I</td>
<td>104</td>
<td>INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003J DSNUM 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50779I</td>
<td>104</td>
<td>PAGE READS</td>
<td>0</td>
<td>KEY INSERTS:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50780I</td>
<td>104</td>
<td>RID INSERTS</td>
<td>0</td>
<td>RID DELETES (PS):</td>
<td>0</td>
</tr>
<tr>
<td>BMC50781I</td>
<td>104</td>
<td>ROOT SPLITS</td>
<td>0</td>
<td>PAGE FREES:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50783I</td>
<td>104</td>
<td>NEW PAGES</td>
<td>0</td>
<td>BUFFER PAGE READS:</td>
<td>0</td>
</tr>
<tr>
<td>BMC50784I</td>
<td>104</td>
<td>BUFFER MANAGER STATISTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

562 REORG PLUS for DB2 Reference Manual
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

Figure 48  SYSPRINT for example 14 (part 9 of 10)

| BMC50791I | 106: ROOT SPLITS : 0  NON-LEAF SPLITS : 0 |
| BMC50791I | 106: LEAF SPLITS : 0  PAGE FREES : 0 |
| BMC50791I | 106: NEW PAGES : 0  BUFF PAGE READS : 0 |
| BMC50716I | 106: Buffer Manager Statistics |
| BMC50717I | 106: DATASET=DEFFCAT.DSNDBC.ARUEXP14.LARX003A.J0001.A003 |
| BMC50717I | 106: DATASET=DEFFCAT.DSNDBC.ARUEXP14.LARX003B.J0001.A001 |
| BMC50718I | 106: Page Requests 12 | Page Hits 2 |
| BMC50718I | 106: Page Reads 10 | Page Writes 1 |
| BMC50718I | 106: Write 1/O Requests 2 | New Pages 0 |
| BMC50716I | 106: Write 1/O time 0:00:00 | Read 1/O time 0:00:00 |
| BMC50766I | 106: INDEX APPLY TASK ENDED AT 10/14/2013 17:42:07, ELAPSED TIME = 00:00:15 |
| BMC50794I | 107: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003B |
| BMC50791I | 107: PAGE READS : 2  KEY INSERTS : 0 |
| BMC50791I | 107: RID INSERTS : 0  RID DELETES (PS) : 0 |
| BMC50791I | 107: ROOT SPLITS : 0  NON-LEAF SPLITS : 0 |
| BMC50791I | 107: LEAF SPLITS : 0  PAGE FREES : 0 |
| BMC50791I | 107: NEW PAGES : 0  BUFF PAGE READS : 0 |
| BMC50716I | 107: Buffer Manager Statistics |
| BMC50717I | 107: DATASET=DEFFCAT.DSNDBC.ARUEXP14.LARX003C.J0001.A001 |
| BMC50717I | 107: DATASET=DEFFCAT.DSNDBC.ARUEXP14.LARX003B.J0001.A001 |
| BMC50718I | 107: Page Requests 12 | Page Hits 2 |
| BMC50718I | 107: Page Reads 10 | Page Writes 1 |
| BMC50718I | 107: Write 1/O Requests 2 | New Pages 0 |
| BMC50716I | 107: Write 1/O time 0:00:00 | Read 1/O time 0:00:00 |
| BMC50766I | 107: INDEX APPLY TASK ENDED AT 10/14/2013 17:42:08, ELAPSED TIME = 00:00:15 |
| BMC50794I | 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003C DSNUM 3 |
| BMC50791I | 103: PAGE READS : 2  KEY INSERTS : 0 |
| BMC50791I | 103: RID INSERTS : 0  RID DELETES (PS) : 0 |
| BMC50791I | 103: ROOT SPLITS : 0  NON-LEAF SPLITS : 0 |
| BMC50791I | 103: LEAF SPLITS : 0  PAGE FREES : 0 |
| BMC50791I | 103: NEW PAGES : 0  BUFF PAGE READS : 0 |
| BMC50716I | 103: Buffer Manager Statistics |
| BMC50717I | 103: DATASET=DEFFCAT.DSNDBC.ARUEXP14.LARX003C.J0001.A003 |
| BMC50717I | 103: DATASET=DEFFCAT.DSNDBC.ARUEXP14.LARX003A.J0001.A001 |
| BMC50718I | 103: Page Requests 12 | Page Hits 2 |
| BMC50718I | 103: Page Reads 10 | Page Writes 1 |
| BMC50718I | 103: Write 1/O Requests 2 | New Pages 0 |
| BMC50716I | 103: Write 1/O time 0:00:00 | Read 1/O time 0:00:00 |
| BMC50766I | 103: INDEX APPLY TASK ENDED AT 10/14/2013 17:42:10, ELAPSED TIME = 00:00:39 |
| BMC50716I | 103: Buffer Manager Statistics |
| BMC50717I | 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003A DSNUM 1 |
| BMC50791I | 103: PAGE READS : 2  KEY INSERTS : 0 |
| BMC50791I | 103: RID INSERTS : 0  RID DELETES (PS) : 0 |
| BMC50791I | 103: ROOT SPLITS : 0  NON-LEAF SPLITS : 0 |
| BMC50791I | 103: LEAF SPLITS : 0  PAGE FREES : 0 |
| BMC50791I | 103: NEW PAGES : 0  BUFF PAGE READS : 0 |
| BMC50716I | 103: Buffer Manager Statistics |
| BMC50717I | 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003C DSNUM 1 |
| BMC50791I | 103: PAGE READS : 2  KEY INSERTS : 0 |
| BMC50791I | 103: RID INSERTS : 0  RID DELETES (PS) : 0 |
| BMC50791I | 103: ROOT SPLITS : 0  NON-LEAF SPLITS : 0 |
| BMC50791I | 103: LEAF SPLITS : 0  PAGE FREES : 0 |
| BMC50791I | 103: NEW PAGES : 0  BUFF PAGE READS : 0 |
| BMC50716I | 103: Buffer Manager Statistics |
| BMC50717I | 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003B DSNUM 3 |
| BMC50791I | 103: PAGE READS : 2  KEY INSERTS : 0 |
| BMC50791I | 103: RID INSERTS : 0  RID DELETES (PS) : 0 |
| BMC50791I | 103: ROOT SPLITS : 0  NON-LEAF SPLITS : 0 |
| BMC50791I | 103: LEAF SPLITS : 0  PAGE FREES : 0 |
| BMC50791I | 103: NEW PAGES : 0  BUFF PAGE READS : 0 |
| BMC50716I | 103: Buffer Manager Statistics |
| BMC50717I | 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003C DSNUM 2 |
| BMC50791I | 103: PAGE READS : 2  KEY INSERTS : 0 |
| BMC50791I | 103: RID INSERTS : 0  RID DELETES (PS) : 0 |
| BMC50791I | 103: ROOT SPLITS : 0  NON-LEAF SPLITS : 0 |
| BMC50791I | 103: LEAF SPLITS : 0  PAGE FREES : 0 |
| BMC50791I | 103: NEW PAGES : 0  BUFF PAGE READS : 0 |
| BMC50716I | 103: Buffer Manager Statistics |
| BMC50717I | 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003B DSNUM 1 |
| BMC50791I | 103: PAGE READS : 2  KEY INSERTS : 0 |
| BMC50791I | 103: RID INSERTS : 0  RID DELETES (PS) : 0 |
| BMC50791I | 103: ROOT SPLITS : 0  NON-LEAF SPLITS : 0 |
| BMC50791I | 103: LEAF SPLITS : 0  PAGE FREES : 0 |
| BMC50791I | 103: NEW PAGES : 0  BUFF PAGE READS : 0 |
| BMC50716I | 103: Buffer Manager Statistics |
| BMC50717I | 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003C DSNUM 3 |
| BMC50791I | 103: PAGE READS : 2  KEY INSERTS : 0 |
| BMC50791I | 103: RID INSERTS : 0  RID DELETES (PS) : 0 |
| BMC50791I | 103: ROOT SPLITS : 0  NON-LEAF SPLITS : 0 |
| BMC50791I | 103: LEAF SPLITS : 0  PAGE FREES : 0 |
| BMC50791I | 103: NEW PAGES : 0  BUFF PAGE READS : 0 |
| BMC50716I | 103: Buffer Manager Statistics |
| BMC50717I | 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003C DSNUM 1 |
| BMC50791I | 103: PAGE READS : 2  key INSERTS : 0 |
| BMC50791I | 103: RID INSERTS : 0  RID DELETES (PS) : 0 |
| BMC50791I | 103: ROOT SPLITS : 0  NON-LEAF SPLITS : 0 |
| BMC50791I | 103: LEAF SPLITS : 0  PAGE FREES : 0 |
| BMC50791I | 103: NEW PAGES : 0  BUFF PAGE READS : 0 |
| BMC50716I | 103: Buffer Manager Statistics |

Chapter 5  Examples of REORG PLUS jobs 563
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

Figure 49  DDLOUT for example 14

```
ALTER TABLE "ARUEXP14"."LART003" ALTER PARTITION 0001 ENDING AT ('2760-02-28-15.00.00.000000',1000);
ALTER TABLE "ARUEXP14"."LART003" ALTER PARTITION 0002 ENDING AT ('3610-04-28-07.00.00.000000',7400);
```

Figure 49 shows the DDLOUT output for example 14.
Chapter 6 Tuning REORG PLUS jobs

This chapter presents the following topics:

Tuning for performance ................................................................. 566
  Setting installation options for optimal performance .................. 566
  Setting SHRLEVEL CHANGE installation options for optimal performance .. 567
  Specifying command options for optimal performance ............... 568
  Specifying SHRLEVEL CHANGE command options for optimal performance 568
  Additional performance information for installation and command options . 569
  Enabling multitasking for performance ........................................ 584
  Additional performance tuning recommendations ....................... 589
  Additional performance information for SHRLEVEL CHANGE options ..... 590
  Performance tuning for specific scenarios (any SHRLEVEL) ........... 590
  Performance tuning for specific scenarios (SHRLEVEL CHANGE only) ... 591
  Interpreting performance-related messages ............................... 592
Tuning to improve memory use .................................................... 597
  Memory requirements of the ORDER command option .................. 597
  Tuning sort processing ............................................................. 598
  Using multitasking to improve memory use ............................... 598
  Tuning copy processing ........................................................... 598
Tuning for availability ............................................................... 599
  Specifying SHRLEVEL ............................................................. 599
  Renaming or switching staging data sets .................................... 599
  Granting data set authority ..................................................... 599
REORG PLUS processing phases .................................................. 600
  REORG PLUS architecture ....................................................... 600
  ANALYZE phase .................................................................... 601
  UNLOAD phase .................................................................... 602
  RELOAD phase .................................................................... 605
  REORG phase for a single-phase reorganization ......................... 609
Tuning for performance

REORG PLUS provides several options that you can use to influence its performance. Some options affect the performance of the entire product; others influence processing only for specific functions of REORG PLUS. This section explains the effect that these options have on the performance of REORG PLUS.

Setting installation options for optimal performance

In general, you should use the values that were shipped with REORG PLUS for the installation options. These values usually enable optimal performance. However, for certain environments, BMC recommends that you make adjustments to these values. Table 79 describes these recommendations. For a complete description of each installation option, see Appendix A, “REORG PLUS installation options.”

For additional installation option changes for SHRLEVEL CHANGE, see page 567.

**NOTE**

If you modify these options after installation, you must resubmit the installation job for the modifications to take effect.

<table>
<thead>
<tr>
<th>Installation option</th>
<th>Recommended value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCHIST</td>
<td>NO</td>
<td>Set this value if you do not need to update the BMCHIST table.</td>
</tr>
<tr>
<td>FSTHRESHOLD</td>
<td>50</td>
<td>Set this value to improve performance for SHRLEVEL REFERENCE and SHRLEVEL CHANGE jobs that use a small number of staging data sets.</td>
</tr>
<tr>
<td>KEEPDICITIONARY</td>
<td>YES</td>
<td>Consider this value to avoid building a new dictionary if one already exists.</td>
</tr>
<tr>
<td>REDEFINE</td>
<td>NO</td>
<td>To improve performance, BMC recommends that you specify this value. A value of NO for this option is also useful when there are table spaces with many tables and indexes (for example, in an ERP environment) if the purpose of the reorganization is solely to reorganize the data, not to redefine the data sets. In a production environment, however, use a value of YES for this option to redefine the data sets and reorganize the data. You can also specify the DSRSEEXIT REXX exit on the REORG command to delete and define only objects that require it.</td>
</tr>
<tr>
<td>STAGEDSN</td>
<td>DSN</td>
<td>For SHRLEVEL CHANGE and SHRLEVEL REFERENCE, specify this value to avoid redundant data set rules.</td>
</tr>
</tbody>
</table>
Setting SHRLEVEL CHANGE installation options for optimal performance

The recommendations in Table 79 on page 566 apply to all reorganizations. BMC recommends the additional changes in Table 80 to enhance performance of SHRLEVEL CHANGE jobs. You can change most of these options on the REORG command. For more information, see “Specifying command options for optimal performance” on page 568. For more information about using SHRLEVEL CHANGE jobs, see Chapter 7, “Online reorganization.”

### Table 79  Installation default option changes for performance (part 2 of 2)

<table>
<thead>
<tr>
<th>Installation option</th>
<th>Recommended value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNITCNT</td>
<td></td>
<td>For optimal performance, specify only the number of devices that you need.</td>
</tr>
<tr>
<td>VOLCNT</td>
<td>AUTO</td>
<td>For optimal performance, BMC recommends that you specify AUTO to have REORG PLUS compute the volume count based on the amount of data, adjusted for the estimated space required.</td>
</tr>
</tbody>
</table>

### Setting SHRLEVEL CHANGE installation options for optimal performance

The recommendations in Table 79 on page 566 apply to all reorganizations. BMC recommends the additional changes in Table 80 to enhance performance of SHRLEVEL CHANGE jobs. You can change most of these options on the REORG command. For more information, see “Specifying command options for optimal performance” on page 568. For more information about using SHRLEVEL CHANGE jobs, see Chapter 7, “Online reorganization.”

### Table 80  SHRLEVEL CHANGE installation default option changes for performance

<table>
<thead>
<tr>
<th>Installation option</th>
<th>Value and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGSPIL</td>
<td>Specify the amount of DASD space, in kilobytes, for REORG PLUS to use when the log records spill (that is, when LOGMEM space becomes full). If you are processing an object that contains a large amount of data and you have many updates to the data, consider increasing the values for this option.</td>
</tr>
</tbody>
</table>
| MAXRO               | The MAXRO option specifies the maximum number of seconds that you want REORG PLUS to spend applying log records in the LOGFINAL phase.  

**Note:** The value for MAXRO applies only to the time that REORG PLUS spends applying log records in the LOGFINAL phase, not to the read-only time spent producing the incremental image copies and running the UTILTERM phase. |
| SIXSNAP             | If your environment contains the appropriate hardware, specify AUTO to ensure that REORG PLUS uses the hardware if it is available and falls back to a regular copy if the hardware is not available. Specifying AUTO can improve performance. However, if the object that you are reorganizing has a large number of indexes or multiple data set indexes, SQLCODE –911 timeouts might occur because of the serialization of the Instant Snapshot requests, which are performed on a per-index basis. In this scenario, change SIXSNAP to NO. |
| SPILDSNP            | Specify the data set name pattern to use when allocating the LOGSPIL data set. The default value is &&UID. To prevent duplicate names when the same user runs two jobs concurrently, change the default value. |
| SPILECL            | Specify the SMS storage class, if any, from which you want to obtain the LOGSPIL data set volume. |
| SPILEUNIT           | Specify the device type or generic unit type to which you want the LOGSPIL data set allocated. |
Specifying command options for optimal performance

Table 81 lists REORG command options that correspond to installation options that are listed in Table 79 on page 566. BMC recommends that you specify the REORG command options that are listed in Table 81 if they are not already specified in the installation options.

If you are using SHRLEVEL CHANGE, also see Table 83 on page 569 for additional command options.

### Table 81  REORG PLUS command options that correspond to installation options

<table>
<thead>
<tr>
<th>Command option</th>
<th>Corresponding installation option</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCHIST NO</td>
<td>BMCHIST=NO</td>
</tr>
<tr>
<td>FSTHRESHOLD 50</td>
<td>FSTHRESHOLD=50</td>
</tr>
<tr>
<td>KEEPDICITIONARY YES</td>
<td>KEEPDICITIONARY=YES</td>
</tr>
<tr>
<td>REDEFINE NO</td>
<td>REDEFINE=NO</td>
</tr>
<tr>
<td>UNITCNT (n)</td>
<td>UNITCNT=(n) where (n) is only the number of devices that you need</td>
</tr>
<tr>
<td>VOLCNT AUTO</td>
<td>VOLCNT=AUTO</td>
</tr>
</tbody>
</table>

Table 82 describes additional recommended command options. (No installation options correspond to these command options.)

### Table 82  Additional REORG PLUS command syntax options

<table>
<thead>
<tr>
<th>Command option</th>
<th>Conditions and comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON FAILURE phase TERMINATE UTILITY RETCODE integer</td>
<td>Specify this option to provide a clean termination if you plan to rerun (in other words, start over from the beginning) rather than restart the job.</td>
</tr>
<tr>
<td>ORDER NO</td>
<td>Specify this option if you do not need sorted data.</td>
</tr>
</tbody>
</table>

Specifying SHRLEVEL CHANGE command options for optimal performance

Table 83 on page 569 shows the command option equivalents for the installation options that apply to SHRLEVEL CHANGE. BMC recommends that you specify these REORG command options for a SHRLEVEL CHANGE job if they are not already specified in the installation options.
Table 83  SHRLEVEL CHANGE command options that correspond to installation options

<table>
<thead>
<tr>
<th>Command option</th>
<th>Corresponding installation option</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEADLINE</td>
<td>DEADLINE</td>
</tr>
<tr>
<td>DELAY</td>
<td>DELAY</td>
</tr>
<tr>
<td>LOGFINAL</td>
<td>LOGFINAL</td>
</tr>
<tr>
<td>LOGMEM</td>
<td>LOGMEM</td>
</tr>
<tr>
<td>LOGSPILL</td>
<td>LOGSPIL</td>
</tr>
<tr>
<td>LOGTHRESHLD</td>
<td>LOGTHRESH</td>
</tr>
<tr>
<td>LONGBLOB</td>
<td>LONGBLOB</td>
</tr>
<tr>
<td>MAXRO</td>
<td>MAXRO</td>
</tr>
<tr>
<td>RIMAPMEM</td>
<td>RIMAPMEM</td>
</tr>
<tr>
<td>Not available</td>
<td>RIMAXD</td>
</tr>
<tr>
<td>Not available</td>
<td>RIMAXD</td>
</tr>
<tr>
<td>SIXSNAP</td>
<td>SIXSNAP</td>
</tr>
<tr>
<td>SPILDSNPAT</td>
<td>SPILDSNP</td>
</tr>
<tr>
<td>SPILLSTORCLAS</td>
<td>SPILSCLS</td>
</tr>
<tr>
<td>SPILLUNIT</td>
<td>SPILUNIT</td>
</tr>
</tbody>
</table>

Additional performance information for installation and command options

The following sections provide additional information for some installation and command options. This information can help you make decisions about these options that can affect the performance of your reorganization jobs. Use Table 84 to find information about a specific option.

Table 84  Additional performance information about options (part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE</td>
<td>provides information to help you determine data set sizes and the optimal number of tasks</td>
<td>570</td>
</tr>
<tr>
<td>AVAILPAGEPCT</td>
<td>controls the amount of system memory that REORG PLUS can allocate to BMCSORT for sort processing</td>
<td>580</td>
</tr>
<tr>
<td>CBUFFS</td>
<td>controls buffers for writing copy data sets</td>
<td>575</td>
</tr>
<tr>
<td>DDTYPE</td>
<td>controls dynamic allocation</td>
<td>579</td>
</tr>
<tr>
<td>MAXSORTS</td>
<td>controls number of sort processes</td>
<td>580</td>
</tr>
<tr>
<td>ORDER</td>
<td>controls whether REORG PLUS sorts the data</td>
<td>580</td>
</tr>
<tr>
<td>MAXSORTMEMORY</td>
<td>maximum amount of memory that REORG PLUS can allocate to each sort task</td>
<td>580</td>
</tr>
</tbody>
</table>
Table 84  Additional performance information about options (part 2 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINSORTMEMORY</td>
<td>minimum amount of memory that REORG PLUS should allocate to each sort task</td>
<td>580</td>
</tr>
<tr>
<td>SHORTMEMORY</td>
<td>controls the action that REORG PLUS takes when a memory shortage exists that affects sort processing</td>
<td>580</td>
</tr>
<tr>
<td>SMAX</td>
<td>controls number of sort processes</td>
<td>580</td>
</tr>
<tr>
<td>SMCORE</td>
<td>controls sort memory usage</td>
<td>580</td>
</tr>
<tr>
<td>UBUFFS</td>
<td>controls buffers for reading and writing SYSREC data sets</td>
<td>575</td>
</tr>
<tr>
<td>UXSTATE</td>
<td>controls DB2 user exit processing</td>
<td>584</td>
</tr>
<tr>
<td>WBUFFS</td>
<td>controls buffers for reading and writing SYSUT1 data sets</td>
<td>575</td>
</tr>
</tbody>
</table>

ANALYZE command options

The ANALYZE options limit abends caused by inadequate size allocations for data sets. These options provide input to dynamic allocation processing or provide information that allows you to allocate space more accurately. REORG PLUS uses the results of the ANALYZE phase to verify work data set sizes and optimize tasks. For details about the ANALYZE command options, including restrictions on each value, see “ANALYZE” on page 199.

The SCAN, SAMPLE, HURBA, and BMCSTATS options determine how REORG PLUS gathers the information that it needs for analysis. Table 85 shows how these options affect the elapsed time of the reorganization and the accuracy of the data set sizings. The needs of your organization will determine which option you should select.

Table 85  Time versus accuracy for the ANALYZE options

<table>
<thead>
<tr>
<th>Option</th>
<th>Time required</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCAN</td>
<td>most</td>
<td>most</td>
</tr>
<tr>
<td>SAMPLE</td>
<td>more</td>
<td>less</td>
</tr>
<tr>
<td>HURBA</td>
<td>less</td>
<td>least</td>
</tr>
<tr>
<td>BMCSTATS(^a)</td>
<td>least</td>
<td>most</td>
</tr>
</tbody>
</table>

\(^a\) BMCSTATS is available only if you are using the Database Performance solution.
ANALYZE

For both table space and index space reorganizations, REORG PLUS decides whether to sample or scan for cardinality if you specify one of the following options:

- ANALYZE with no other ANALYZE options
- ANALYZE ONLY or ANALYZE PAUSE with no other ANALYZE options

For a table space reorganization, unless you specify ANALYZE BMCSTATS, REORG PLUS always samples the table space for the average row length.

ANALYZE SAMPLE and SCAN

Table 86 describes the actions that REORG PLUS takes based on the object that you are reorganizing and the ANALYZE option that you specify.

<table>
<thead>
<tr>
<th>Reorganization type</th>
<th>ANALYZE option</th>
<th>REORG PLUS actions</th>
</tr>
</thead>
</table>
| table space         | SAMPLE         | Samples the table space to  
|                     |                | - estimate the cardinality  
|                     |                | - determine the average row length  
|                     | SCAN           | To determine the cardinality:  
|                     |                | - scans one index on each table to find the exact cardinality.  
|                     |                | - samples each table with no index to estimate the cardinality for that table.  
|                     |                | Samples the table space to determine the average row length. |
| index               | SAMPLE         | Samples the index to estimate the cardinality. |
|                     | SCAN           | Scans each leaf page of the index to determine the exact cardinality. |

ANALYZE HURBA

If you specify ANALYZE HURBA for either a table space or an index reorganization, REORG PLUS does not read the table space or the index during the ANALYZE phase. Instead, it uses the high-used relative byte address (HURBA) and information about column lengths from the DB2 catalog to estimate the cardinality and average row length.
**ANALYZE BMCSTATS**

The BMCSTATS option is available only if you are using REORG PLUS as a component of the Database Performance for DB2 solution. Otherwise, REORG PLUS changes BMCSTATS to SAMPLE and continues the reorganization.

If you specify ANALYZE BMCSTATS for either a table space or an index reorganization, REORG PLUS uses the information that already exists in the DASD MANAGER PLUS statistics tables to perform the analysis. If the information in the DASD MANAGER PLUS statistics tables is current, using ANALYZE BMCSTATS is as accurate as using ANALYZE SCAN, and is the fastest method available because REORG PLUS does not have to gather the data.

If the DASD MANAGER PLUS statistics tables contain incomplete information for the object that you are reorganizing, REORG PLUS changes BMCSTATS to SAMPLE and continues the reorganization.

**ANALYZE PAUSE and ONLY**

The ANALYZE PAUSE and ANALYZE ONLY options provide estimates of the space needed for the following data sets:

- unload (SYSREC)
- work (SYSUT1)
- sort (SORTWK)
- full image copy (BCPY, BCPZ, BRCY, and BRCZ)
- incremental image copy (BICY, BICZ, BIRY, and BIRZ)

REORG PLUS writes the statistics to SYSPRINT. You must specify PAUSE or ONLY to get this statistics report.

If you specify ANALYZE ONLY and use the information to allocate your data sets, you can improve performance by changing the REORG command option to ANALYZE HURBA when you rerun the job. Specifying ANALYZE HURBA bypasses the ANALYZE phase. For exceptions when specifying HURBA, see the restrictions listed in “ANALYZE HURBA” on page 571.

**ANALYZE messages**

When you specify ANALYZE PAUSE or ANALYZE ONLY, REORG PLUS issues the messages listed in this section. For more information about using the information from the ANALYZE phase for allocating your data sets, see “ANALYZE option for estimating data set allocation” on page 386.
If you do not want to specify ANALYZE PAUSE or ANALYZE ONLY, you must either have REORG PLUS dynamically allocate your data sets (see “Dynamic allocation options” on page 293), or calculate the data set sizes yourself.

**Message overview**

The ANALYZE messages provide information about data set size allocation in a table format:

- Messages BMC51260I through BMC51262I provide heading information.
- Multiple BMC51263I messages provide the estimated values.
- Messages BMC51264I through BMC51266I provide estimates that you can use to gauge the elapsed time of the reorganization.

**Message descriptions**

This section lists and describes the messages that REORG PLUS issues when you specify ANALYZE PAUSE or ANALYZE ONLY.

<table>
<thead>
<tr>
<th>BMC51260I</th>
<th>REORG PLUS DASD REQUIREMENT ESTIMATES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC51261I</td>
<td>3380 CYLS 3390 CYLS</td>
</tr>
<tr>
<td>BMC51262I</td>
<td>DDNAME KBYTES PRI SEC PRI SEC INDEX</td>
</tr>
<tr>
<td>BMC51263I</td>
<td>ddname k p s p s creator.indexName</td>
</tr>
</tbody>
</table>

Explanation: After providing heading information in messages BMC51260I through BMC51262I, REORG PLUS issues a separate BMC51263I message for each data set to provide the following information:

- data set name
- number of kilobytes
- primary and secondary 3380 cylinder quantities
- primary and secondary 3390 cylinder quantities
- index name, where applicable

User Response: No action is required.

<table>
<thead>
<tr>
<th>BMC51264I</th>
<th>UNLOAD WILL READ n DATA PAGES FROM SPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>'databaseName.tableSpaceName'</td>
</tr>
</tbody>
</table>

Explanation: This message provides the count of data pages that REORG PLUS will read during the unload process.

User Response: Because the reorganization process is I/O bound, you can use this number, along with the cardinality, to gauge the elapsed time of the reorganization, based on the performance of previous runs of the utility.
**BMC51265I**

**ESTIMATED CARDINALITY OF SPACE = n**
**ESTIMATED CARDINALITY OF SPACE = n**
**AVG SORTWK ROW LENGTH = n**
**AVG UNLOAD ROW LENGTH = n**

*Explanation:* This message provides an estimate of the cardinality of the table space or index space. For table space reorganizations, the message also provides an estimate of the average row length for the SORTWK and SYSREC data sets.

*User Response:* You can compare these estimates with estimates provided by other messages and previous runs of the utility to gauge the elapsed time of the reorganization.

**BMC51266I**

**CARDINALITY OF SPACE = n**
**CARDINALITY OF SPACE = n**
**AVG SORTWK ROW LENGTH = n**
**AVG UNLOAD ROW LENGTH = n**

*Explanation:* This message provides the exact cardinality of the table space or index space. For table space reorganizations, the message also provides an estimate of the average row length for the SORTWK and SYSREC data sets.

*User Response:* You can compare these values with estimates provided by other messages and previous runs of the utility to gauge the elapsed time of the reorganization.

**BMC50484I**

**ESTIMATED CARDINALITY OF objectType = n**
**AVG SORTWK ROW LENGTH = n**
**AVG UNLOAD ROW LENGTH = n**

*Explanation:* This message provides either an estimate of or the exact cardinality of each table in a nonpartitioned table space, or each partition in a partitioned table space. The message also provides an estimate of the average row length for the SORTWK and SYSREC data sets.

*User Response:* Use this message in conjunction with message BMC51265I or BMC51266I.

**BMC50485I**

**ESTIMATED CARDINALITY OF PART partitionNumber = n**
**ESTIMATED CARDINALITY OF TABLE tableName = n**

*Explanation:* REORG PLUS issues this message for an index-only reorganization. This message provides either an estimate of or the exact cardinality of each partition in a partitioned index space.

*User Response:* Use this message in conjunction with message BMC51265I or BMC51266I.
Buffer installation options

You can use installation options to control the amount of buffer space that the data sets use during REORG PLUS processing. For each option, REORG PLUS multiplies the number that you specify by 32 KB to determine the amount of virtual storage needed for the buffer pool. REORG PLUS then allocates as many buffers as will fit into that space.

REORG PLUS attempts to allocate all buffers above the 16-megabyte (MB) line.

SYSREC data set

REORG PLUS does not use the SYSREC data set when you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE for a single-phase table space reorganization. References to single-phase reorganizations in this section are for SHRLEVEL NONE only.

The SYSREC data set contains the unloaded rows from the table space that you are reorganizing. For a two-phase reorganization, REORG PLUS writes this data set during the UNLOAD phase and reads it during the RELOAD phase. For a single-phase reorganization, multiple SYSRECnn data sets (one per partition) are optional. When you specify them, REORG PLUS writes the data sets during unload processing in the REORG phase for restart purposes only. For detailed specification and allocation guidelines for the SYSREC data set, see “SYSREC data sets” on page 379. For information about having REORG PLUS dynamically allocate the SYSREC data set, see “Dynamic allocation options” on page 293.

For a partitioned table space, you can improve I/O performance by specifying one SYSRECnn data set for each partition. Using multiple data sets allows REORG PLUS to concurrently unload the table space information from each partition during unload processing and, for a single-phase reorganization, concurrently reload the information during reload processing.

For a single-phase table space reorganization, you can improve I/O performance by not specifying any SYSRECnn data set in your JCL and not having REORG PLUS dynamically allocate one. Not having a SYSREC data set eliminates writing to the data set. However, not having the SYSREC data set also means that the job might not be restartable.

You can still achieve a performance gain when you specify a SYSREC data set or use dynamic data set allocation for a single-phase reorganization because the table space information is written to the SYSREC data set for restart purposes only. REORG PLUS does not reload the table space from this data set, thus eliminating reading from the SYSREC data set.
REORG PLUS uses the UBUFFS installation option to determine the amount of buffer pool storage that is allocated for reading and writing the SYSREC data sets. The multiplier that REORG PLUS applies to the UBUFFS option value is 32 KB. For the UBUFFS option, BMC recommends a value of 20 (the value that is shipped with REORG PLUS). REORG PLUS determines the optimal block size of the SYSREC data set, depending on the device type containing the data set.

**SYSUT1 data set**

For a table space reorganization, REORG PLUS requires one or more SYSUT1 data sets. For a single-phase index reorganization, REORG PLUS does not use the SYSUT1 data set when you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE. References to single-phase reorganizations in this section are for SHRLEVEL NONE only.

The SYSUT1 data set contains the information that REORG PLUS needs to build all indexes (when you specify ORDER NO) or only the non-data-sorting indexes (when you specify ORDER YES). For a two-phase reorganization, REORG PLUS writes this data set during the UNLOAD phase and reads it during the RELOAD phase. For a single-phase table space reorganization, REORG PLUS both writes and reads the data set during the REORG phase. For a single-phase index reorganization, the SYSUT1 data set is optional. When you specify it, REORG PLUS writes this data set during the REORG phase for restart purposes only. For specification and allocation guidelines for the SYSUT1 data set, see “SYSUT1 data sets” on page 382. For information about having REORG PLUS dynamically allocate the SYSUT1 data set, see “Dynamic allocation options” on page 293.

When more than one non-data-sorting index exists, you can improve I/O performance by using multiple SYSUT1 data sets. Using multiple data sets allows I/O operations to overlap.

For a single-phase index reorganization, you can improve I/O performance by not specifying a SYSUT1 data set in your JCL and not having REORG PLUS dynamically allocate one. Not having a SYSUT1 data set eliminates writing to the data set. However, not having the SYSUT1 data set also means that the job might not be restartable.

You can still achieve a performance gain when you specify a SYSUT1 data set or use dynamic data set allocation for a single-phase reorganization because REORG PLUS writes the index information to SYSUT1 for restart purposes only. REORG PLUS does not reload the index from this data set, thus eliminating reading from the SYSUT1 data set.

A single SYSUT1 data set for all indexes requires a record length long enough to hold information for the longest key. If shorter keys exist, REORG PLUS pads them so that they are as long as the longest key. With multiple SYSUT1 data sets, REORG PLUS writes information for each index to its own SYSUT1 data set, and does not pad the keys.
REORG PLUS uses the WBUFFS installation option to determine the amount of buffer pool storage that is allocated for reading and writing the SYSUT1 data sets. The multiplier that REORG PLUS applies to the WBUFFS option value is 32 KB. The WBUFFS option has two subparameter values:

- The first subparameter specifies the value to use when you use a single SYSUT1 data set.

- The second subparameter specifies the value to use for each data set when you use multiple SYSUT1 data sets.

For this option, BMC recommends a value of 20 for a single SYSUT1 data set and 10 for multiple SYSUT1 data sets.

REORG PLUS determines the optimal block size of the SYSUT1 data set, depending on the device type containing the data set.

**Copy data sets**

The copy data sets contain identical copies of the reorganized table space or partitions of the table space after it is reorganized. For a two-phase table space reorganization, REORG PLUS writes these data sets during the RELOAD phase. For a single-phase table space reorganization, REORG PLUS writes these data sets during the REORG phase.

If the copies are registered as the DB2 local and remote copies, they can provide input to a DB2 recovery utility, such as the BMC Software RECOVER PLUS product or the IBM DB2 RECOVER utility. For guidelines about specifying and allocating the copy data sets, see “Copy data sets” on page 352.

REORG PLUS uses the CBUFFS installation option to determine the amount of buffer pool storage that is allocated for writing the copy data sets. The multiplier that REORG PLUS applies to the CBUFFS option value is 32 KB. For the CBUFFS option, BMC recommends a value of 30. When copying multiple partitions of the table space to separate data sets, REORG PLUS uses only one group of buffers.

REORG PLUS normally determines the optimal block size of the copy data sets, depending on the device type that contains the data sets. However, you can control the block size through the execution JCL by allocating the copy data sets with a block size that is greater than 0 and is an even multiple of four kilobytes. If you are making more than one copy, REORG PLUS uses the same block size as you specify for BCPY for all subsequent copies, regardless of the block size that you specify in the JCL for any additional copies.

As an alternative to specifying the copy data sets in a DD statement, you can have REORG PLUS dynamically allocate the data sets.
For additional performance considerations for copy data sets, see “Copy options” on page 579.

**Incremental copy data sets**

The incremental copy data sets (shipped with the default names of BICY, BICZ, BIRY, and BIRZ) contain identical copies of the pages that have changed since the full copies were made of the reorganized table space or partitions of the table space. When performing a SHRLEVEL CHANGE reorganization, you might need one of these data sets for each full copy data set that you specify in your JCL, depending on the value of the ICTYPE option.

REORG PLUS writes the incremental data sets during the LOGFINAL phase. The copies are registered as the DB2 local and remote copies and can provide input to a DB2 recovery utility, such as the BMC Software RECOVER PLUS product or the IBM DB2 RECOVER utility. For guidelines about specifying and allocating the incremental copy data sets, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 653.

REORG PLUS uses the CBUFFS installation option to determine the amount of buffer pool storage that is allocated for writing the copy data sets. The multiplier that REORG PLUS applies to the CBUFFS option value is 32 KB. For the CBUFFS option, BMC recommends a value of 30. When copying multiple partitions of the table space to separate data sets, REORG PLUS uses only one group of buffers.

REORG PLUS normally determines the optimal block size of the incremental copy data sets, depending on the device type containing the data sets. However, you can control the block size through the execution JCL by allocating the incremental copy data sets with a block size that is greater than 0 and is an even multiple of four kilobytes. If you are making more than one copy, REORG PLUS uses the same block size as you specify for BMCICY for all subsequent copies, regardless of the block size you specify in the JCL for any additional copies.

As an alternative to specifying the incremental copy data sets in a DD statement, you can have REORG PLUS dynamically allocate the data sets.

For additional performance considerations for incremental copy data sets, see “Improving performance” on page 656.

**DB2 data sets**

The DB2 data sets contain the table space that you are reorganizing and its associated indexes. For a two-phase reorganization, REORG PLUS reads the data sets during the UNLOAD phase and writes the data sets during the RELOAD phase. For a single-phase reorganization, REORG PLUS reads and writes the data sets during the REORG phase.
REORG PLUS assigns buffers for these data sets based on available virtual storage and the number of partitions processed concurrently. Providing as much virtual storage as possible allows processing of several partitions and indexes concurrently. For more information, see “Providing maximum virtual storage” on page 589.

### Copy options

Consider the following recommendations when making copies during the reorganization job:

- Specify the INLINECP installation option or the COPY YES INLINE YES command option to make inline image copies during the reorganization, particularly if you are reorganizing a partitioned object.

  The difference between inline image copies and standard image copies is that REORG PLUS creates inline copies as it reloads the table space, rather than after it reloads the table space. This process can reduce the elapsed time of the reorganization. For details about how to create inline image copies, see “Copy options for REORG TABLESPACE” on page 264.

- Specify a value of PART for the COPYLVL installation or command option if you are reorganizing a partitioned object.

  COPYLVL=PART facilitates multitasking by providing a unique full copy data set for each partition that you are reorganizing and ensures that each copy data set resides on DASD.

### Dynamic allocation options

Although dynamically allocating each data set requires some processing time, enabling dynamic data set allocation can improve performance in many cases. If you are reorganizing very small quantities of data, the extra processing time might be a nontrivial percentage of the elapsed time. If you are reorganizing large quantities of data, especially in a partitioned data set, using dynamic data set allocation might improve performance; dynamic data set allocation always uses the optimal number of data sets, which enables REORG PLUS to use DASD more efficiently.

In general, BMC recommends that you dynamically allocate the following data sets with the DDTYPE options:

- SYSREC
- SYSUT1
- full copy (BCPY, BCPZ, BRCY, and BRCZ)
- incremental copy (BICY, BICZ, BIRY, and BIRZ)
For more information, see “Dynamic data set allocation” on page 90 and “Dynamic allocation options” on page 293.

### ORDER command option

The ORDER command option controls whether the unloaded rows are sorted. In determining how to optimize the performance of REORG PLUS, you must balance your need for fast execution and data availability with your need to have data sorted when it is reorganized. The following discussion can help you determine how to specify the ORDER command option to meet your specific performance needs.

**ORDER YES**

When you specify ORDER YES, REORG PLUS sorts the unloaded rows during unload processing. No data-sorting index work information is required in the SYSUT1 data set, thus reducing I/O operation and DASD space for that data set. Instead, REORG PLUS takes the index information from the data rows during reload processing.

**ORDER NO**

When you specify ORDER NO, REORG PLUS does not sort the data, thus decreasing processing time. When your tables have clustering or partitioning indexes, this option removes indirection and deleted table rows, and restores the PCTFREE and FREEPAGE values. This process avoids the overhead of sorting the unloaded rows if your tables have clustering or partitioning indexes. REORG PLUS writes the data-sorting index work information to the SYSUT1 data sets.

### Sort processing options

The BMC BMCSORT technology provides REORG PLUS with more control of the sort process than external sort routines provide. This added control helps prevent memory-related problems during the sort process. REORG PLUS allocates the amount of resources to each sort process based on the amount of work that REORG PLUS determines the sort process will perform. Depending on the table space that you are reorganizing and on its index characteristics, REORG PLUS invokes BMCSORT one or more times.

---

**NOTE**

Ensure that the data set name pattern (DSNPAT) for each data set type generates unique data set names. BMC recommends that you use the database and table space names in the DSNPAT option to prevent duplicate data set names. If you are reorganizing partitioned table spaces, include the partition number in the pattern name. Using the date and time in work data sets is also useful to avoid duplicate data set names.
REORG PLUS does not require sort work data sets when any of the following conditions exists. In each of these cases, REORG PLUS does not perform a sort.

- You are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect.
- Either of the following conditions exists for a two-phase reorganization, or for a single-phase reorganization with SHRLEVEL REFERENCE or SHRLEVEL CHANGE specified:
  - You are performing an index reorganization with ORDER NO specified.
  - You are performing one of the following table space reorganizations:
    - a single-table table space (simple or segmented) with no indexes defined
    - a multitable simple table space with no indexes defined and ORDER NO specified

All other types of reorganizations perform a sort.

To enhance the performance of REORG PLUS and other applications running on your system, you can modify options that control the BMCSORT technology and sort optimization. In addition, the SMAX or MAXSORTS option controls the number of sort processes. The following sections provide information that can help you improve the performance of your sort processing.

**Controlling memory usage**

REORG PLUS provides several installation and command options to give you control, when necessary, over the amount of memory that BMCSORT uses during a reorganization job.

The SMCORE option contains two parameters: total memory and below-the-line memory. BMC strongly recommends that you use the values 0K and 0K for the SMCORE option. With these values, you generally obtain the highest sort performance for your REORG PLUS job. Using 0K for total memory and below-the-line memory indicates that REORG PLUS is to determine the appropriate amount of memory to use for each sort based on the following criteria:

- amount of data to be sorted
- number of sorts to process, as discussed in “Controlling the number of sort processes” on page 583
- value that you specify for REGION in either your JCL or system exits
- amount of memory that is available during optimization
percentage of available or total 4-KB system pages that you specify with the AVAILPAGEPCT and TOTALPAGEPCT options

---

**NOTE**
The SHORTMEMORY installation or command option controls the action that REORG PLUS takes when a memory shortage exists. For more information about SHORTMEMORY, see page 195 or page 721.

---

minimum and maximum amounts of memory per sort that you specify with the following options:

- MINSORTMEMORY installation (page 712) or command option (page 197)
- MAXSORTMEMORY installation (page 711) or command option (page 197)

values that you specify for the multitasking installation options described on page 584

**Total memory**

The first parameter value of the SMCORE option tells REORG PLUS how much total memory, both above and below the 16-MB line, that you want BMCSORT to use during a single invocation. BMC strongly recommends that you specify a value of 0K. However, other valid values are 4096K through 65536K. You can also specify the SMCORE value in megabytes (0M or 4M through 64M).

The region size available for your reorganization job in conjunction with the value that you specify for total memory can constrain the number of sort processes that REORG PLUS starts. Because the region size must include space for buffers and other required structures, the entire region size is not available for sort processing. You can avoid this constraint by using values of 0K and 0K, allowing REORG PLUS to determine the optimal amount of total memory to use.

**Additional considerations**
The following additional considerations apply when you use the sort optimization options and also specify a value greater than 0 for the SMCORE total memory parameter:

- REORG PLUS uses the SMCORE value that you specify, regardless of other memory factors, including values specified for the MINSORTMEMORY installation (page 712) or command option (page 197) and MAXSORTMEMORY installation (page 711) or command option (page 197).

That is, if you specify an SMCORE total memory value that is lower than the MINSORTMEMORY value or greater than the MAXSORTMEMORY value, REORG PLUS still uses your specified value.
If you specify a value greater than 0 for the total memory parameter, REORG PLUS honors your specified total memory value.

**Below-the-line memory**

The second parameter value of the SMCORE option indicates how much memory BMCSORT should use below the 16-MB line during a single invocation. Unless otherwise noted, BMC recommends that you specify a value of 0K, which allows REORG PLUS to determine the optimal amount of below-the-line memory to use. In addition to 0K, valid values are 256K through 4096K. You can also specify this value in megabytes (either 0M or 1M through 4M). If you specify a value of 0 for the below-the-line memory parameter, REORG PLUS computes below-the-line memory.

When you specify a value of 0 for the below-the-line memory parameter, the amount of memory that BMCSORT uses is controlled by the value of the SORTNUM option (page 188 or page 725), as follows:

<table>
<thead>
<tr>
<th>SORTNUM value</th>
<th>Minimum amount of memory used for all sorts</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 or less</td>
<td>256 KB</td>
</tr>
<tr>
<td>33-99</td>
<td>384 KB</td>
</tr>
<tr>
<td>100-255</td>
<td>1024 KB</td>
</tr>
</tbody>
</table>

When you specify a value greater than 0 for the below-the-line memory parameter, REORG PLUS honors the other sort optimization options (which apply to total memory only) and also uses your specified SMCORE value.

The following example illustrate the use of SMCORE and SORTNUM to control the amount of memory that REORG PLUS uses for sort processing:

You specify SMCORE=(0K,700K) and SORTNUM=32.

REORG PLUS uses all of the sort optimization installation and command options to calculate the total memory that is required for sort processing. REORG PLUS uses 700 KB of memory below the line.

**Controlling the number of sort processes**

REORG PLUS determines the optimal number of sort processes that can execute concurrently, depending on available resources. Under normal circumstances, you should allow REORG PLUS to control the number of sorts processed concurrently.

You can specify the maximum number of concurrent sort processes by using the SMAX installation option. BMC recommends that you use the default value of 0 for the SMAX installation option.
You can also specify the MAXSORTS option of the REORG command to set the maximum number of concurrent sort processes. For more information, see the description of "MAXSORTS" on page 189 and Table 90 on page 586, which shows MAXSORTS combined with the other sort options.

If the value of the MAXSORTS command option or SMAX installation option is nonzero, REORG PLUS starts at most one task per CPU. You can improve performance by starting more than one task per CPU. To do so, set SMAX=0 (if you specify MAXSORTS in the command, use 0 for the MAXSORTS value), and use the multitasking installation options (TASKMAX, RORGMAX, UNLDMAX, and BILDMAX). For information about using these options, see “Multitasking installation options.” For information about specifying the installation options, see Appendix A, “REORG PLUS installation options.”

**UXSTATE installation option**

By default, during processing REORG PLUS invokes EDITPROC DB2 user exits in supervisor state and program status word (PSW) key=7. You can specify UXSTATE=PROB to tell REORG PLUS to call EDITPROCs in problem state and PSW key=7. The requirements of the exits dictate the UXSTATE setting. Check with the exit author (or vendor) before changing the value of UXSTATE to PROB.

**Enabling multitasking for performance**

During execution, REORG PLUS determines the most effective arrangement of tasks when running in a multiprocessor environment. Although REORG PLUS runs very efficiently on single-processor computers, it performs most efficiently on large multiprocessor systems.

REORG PLUS considers the following factors when it determines the maximum number of tasks to start:

- available memory above and below the 16-MB line
- number of SYSREC and SYSUT1 data sets that you specify
- governing limitations that you specify through the multitasking options
- number and size of the preallocated sort work data sets
- number of DB2 objects on which REORG PLUS must operate for each phase or process

**Multitasking installation options**

Table 87 on page 585 lists the REORG PLUS installation options that allow you to control the level of multitasking for various phases of the reorganization and for processes within those phases.
Enabling multitasking for performance

Table 87 Multitasking options

<table>
<thead>
<tr>
<th>Multitasking option</th>
<th>Phase or process affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASKMAX</td>
<td>global</td>
</tr>
<tr>
<td>ANALMAX</td>
<td>ANALYZE phase</td>
</tr>
<tr>
<td>RORGMAX(^a)</td>
<td>REORG phase</td>
</tr>
<tr>
<td>UNLDMAX(^a)</td>
<td>UNLOAD phase</td>
</tr>
<tr>
<td>BILDMAX(^a)</td>
<td>index build process</td>
</tr>
<tr>
<td>COPYMAX</td>
<td>image copy process</td>
</tr>
<tr>
<td>RENMMAX</td>
<td>rename and delete process for SHRLEVEL REFERENCE and SHRLEVEL CHANGE</td>
</tr>
<tr>
<td>SCPYMAX</td>
<td>nonpartitioned index copy process for SHRLEVEL REFERENCE and SHRLEVEL CHANGE partial reorganizations</td>
</tr>
</tbody>
</table>

\(^a\) To enable this option, you must specify SMAX=0, and either omit the MAXSORTS command option or specify MAXSORTS 0.

The TASKMAX option sets the default for all of the other multitasking options in the installation options module. TASKMAX does not directly provide the number of tasks to use for any phase or process. Instead, TASKMAX provides a default for any option that refers to it. Table 88 lists the valid values for TASKMAX.

Table 88 Values for the global multitasking option

<table>
<thead>
<tr>
<th>TASKMAX value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>REORG PLUS determines the number of tasks to start.</td>
</tr>
<tr>
<td>(n)</td>
<td>The maximum number of tasks that REORG PLUS can start is (n) (where (n) is a positive integer from 1 through 32767).</td>
</tr>
<tr>
<td>(n%)</td>
<td>The maximum number of tasks that REORG PLUS can start is (n)% of the number of CPUs on the system (where (n) is a positive integer from 1 through 32768).</td>
</tr>
</tbody>
</table>

Table 89 lists the valid values for all of the other multitasking options.

Table 89 Values for the individual multitasking options

<table>
<thead>
<tr>
<th>Option value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>REORG PLUS automatically determines the number of tasks to start.</td>
</tr>
<tr>
<td>*</td>
<td>REORG PLUS uses the TASKMAX value.</td>
</tr>
<tr>
<td>(n)</td>
<td>The maximum number of tasks that REORG PLUS can start is (n) (where (n) is a positive integer from 1 through 32767).</td>
</tr>
<tr>
<td>(n%)</td>
<td>The maximum number of tasks that REORG PLUS can start is (n)% of the number of CPUs on the system (where (n) is a positive integer from 1 through 32768).</td>
</tr>
</tbody>
</table>
For example, assume that you have three CPUs and you specify TASKMAX=5, ANALMAX=*, and RORGMAX=*. REORG PLUS starts up to five tasks for the ANALYZE phase and five for the REORG phase. If you change the TASKMAX value to 500%, REORG PLUS can start as many as 15 tasks for each of those phases. The more tasks that REORG PLUS can start, the better performance it will achieve.

RORGMAX, UNLDMAX, and BILDMAX are enabled only when all of the following conditions exist:

- The value of the SMAX installation option is 0.
- You have not specified the MAXSORTS command option or its value is 0.

Table 90 provides examples of the option combinations. For more information about specifying the RORGMAX, UNLDMAX, and BILDMAX options, see “Multitasking processes that invoke BMCSORT” on page 587.

### Table 90  Sort task options hierarchy

<table>
<thead>
<tr>
<th>When SMAX value</th>
<th>And MAXSORTS value</th>
<th>REORG PLUS uses up to</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>5</td>
<td>5 (MAXSORTS overrides SMAX)</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>16 (SMAX value)</td>
</tr>
<tr>
<td>0</td>
<td>16</td>
<td>16 (MAXSORTS value)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>RORGMAX, UNLDMAX, and BILDMAX values</td>
</tr>
</tbody>
</table>

REORG PLUS always uses the ANALMAX, COPYMAX, SCPYMAX, and RENMMAX multitasking options. For information about determining the values for ANALMAX, COPYMAX, and SCPYMAX, see “Multitasking I/O bound processes.” For a description of the installation options, see Appendix A, “REORG PLUS installation options.”

### Multitasking I/O bound processes

The following multitasked processes are I/O bound:

- ANALYZE phase
- table space copy process
- nonpartitoned index copy process for SHRLEVEL REFERENCE and SHRLEVEL CHANGE partial reorganizations that are not using Instant Snapshot technology

You can control the multitasking for these processes with the ANALMAX, COPYMAX, and SCPYMAX installation options, respectively. Because these processes are I/O bound, the number of tasks does not need to be limited to the number of CPUs. In fact, in many cases, the processes are so I/O bound that they require hardly any CPU service at all. Therefore, you can start many more tasks for these operations than the number of CPUs without adversely affecting performance.
By specifying a percentage rather than a fixed maximum number of tasks, you might get similar performance improvements regardless of the number of CPUs on which REORG PLUS is running. REORG PLUS never starts more tasks than it can effectively use. So, for example, you will see a larger number of tasks for a partitioned table space with 254 partitions than for a simple table space with no indexes.

### Multitasking processes that invoke BMCSORT

In addition to the processes mentioned in “Multitasking I/O bound processes” on page 586, the following processes are also typically I/O bound:

- single-phase REORG phase
- two-phase UNLOAD phase
- the index build process

You can control the multitasking for these processes with the RORGMAX, UNLDMAX, and BILDMAX installation options, respectively. However, for certain architectural considerations, REORG PLUS starts a maximum of 16 tasks for the REORG phase, UNLOAD phase even if you specify a value greater than 16 on the RORGMAX, UNLDMAX, or BILDMAX option.

**NOTE**

For REORG PLUS to honor the RORGMAX, UNLDMAX, and BILDMAX options, you must specify SMAX=0 in the installation options module, and either not specify the MAXSORTS command option or specify MAXSORTS 0.

To achieve the most flexibility and highest level of multitasking, observe the following guidelines:

- Do not include any SORTWK\textit{nn} DD statements in your JCL.
- Specify the following:
  - DDTYPE SORTWORK ACTIVE NO on your REORG command, or disable dynamic allocation for the SORTWK\textit{nn} DDs in your options module
  - the SORTDEVT installation or command option
  - one SYSREC\textit{nn} DD per partition
  - one SYSUT1\textit{nn} DD per index
  - SORTNUM 32

For more information about setting the environment for BMCSORT, see “Sort processing options” on page 580.
Enabling multitasking for performance

### Using multiple SYSREC data sets

When you are reorganizing a partitioned table space, you can improve performance by specifying one SYSRECnn data set for each partition. Using multiple data sets accomplishes the following goals:

- reduces CPU and elapsed time
  
  REORG PLUS can concurrently unload the table space information from each partition during unload processing.

- facilitates faster reloading of the table space that you are reorganizing

For more information about specifying multiple SYSRECnn data sets, see page 575.

### Using multiple SYSUT1 data sets

Using multiple SYSUT1 data sets provides the following performance advantages:

- I/O processing to each SYSUT1 data set is overlapped with other I/O processing and with CPU processing.

- REORG PLUS writes any non-unique index information when the index process receives it and does not pass the information to BMCSORT, thus reducing the amount of data that is sorted. This process also reduces the amount of DASD space that is required for all index information.

- With a single SYSUT1 data set, REORG PLUS pads all keys to the length of the longest key being processed. Padded keys require more DASD space, and more I/O operations are required to process the index information. With multiple SYSUT1 data sets, less padding of keys is needed.

- REORG PLUS attempts to build indexes concurrently. REORG PLUS determines resource utilization in the same way as for the UNLOAD phase (see “Resource allocation in the UNLOAD phase” on page 603) with one exception: instead of checking the number of partitions in the table space, REORG PLUS checks the number of data sets that you specified. The index build phase always runs concurrently with reloading the rows into the table space.

If you are using multiple index data sets, specify one SYSUT1 data set for each participating index.

**NOTE**

If you are reorganizing a table space with a large number of indexes, BMC recommends that you specify a single SYSUT1 data set to avoid data set allocation limitations of the operating system.
When you specify multiple SYSUT1 data sets, the DD statement specification is SYSUT1nn, where nn is a unique suffix for each DD statement. The suffix is not used to identify which index is assigned to a specific SYSUT1 data set.

**Additional performance tuning recommendations**

The following sections describe additional steps that you can take to improve the performance of your reorganization jobs.

**Tuning I/O**

The single most important factor affecting performance in REORG PLUS is I/O processing. During typical reorganization processing, REORG PLUS reads and writes large amounts of data. To maximize I/O performance, REORG PLUS handles all of its own buffering and performs I/O operations at the lowest level possible. Doing so allows REORG PLUS to read or write several blocks of data with each I/O operation and permits REORG PLUS to prefetch subsequent data.

To avoid I/O queueing, allocate REORG PLUS data sets on separate channels and drives. If you do not have sufficient channels available, use separate drives and control units.

Because REORG PLUS I/O processing is primarily sequential, DASD caching provides no benefit. Avoid DASD caching because the overhead might slightly increase I/O processing time.

For more information that can help you tune your I/O processing, see “Buffer installation options” on page 575 and “Sort processing options” on page 580.

**Providing maximum virtual storage**

Because each task requires virtual storage for processing, REORG PLUS balances the multiprocessing performed with the amount of virtual storage that is available. The primary use of virtual storage is for I/O buffers by either REORG PLUS itself or by BMCSORT.

REORG PLUS uses as much virtual storage as needed for each task. For the best performance, BMC recommends that you specify REGION=0M in the JOB or EXEC statement of the execution JCL. If your data center does not permit you to specify REGION=0M, specify the amount that allows the most virtual storage, both above and below the 16-MB line.
Using a region size that is less than optimal risks the following potential issues:

- running less efficiently, which could result in additional CPU and elapsed time
- encountering memory failures or jobs that fail when new versions implement changes that require additional memory

If you specify a value for REGION other than 0M, ensure that you have an appropriate value set for the MEMLIMIT parameter, either as your site’s default SMF option or on your JOB statement or EXEC statement. BMC makes the following recommendations for the MEMLIMIT option:

- Specify NOLIMIT to allow unlimited above-the-bar memory.
- If you are unable to specify NOLIMIT, specify at least 4 GB; if you are reorganizing LOB or XML data, specify at least 32 GB.

Selectively redefining VSAM data sets

You can use the DSRSEXIT user exit to selectively redefine each object. By not performing the VSAM DELETE/DEFINE process for an object, you can significantly reduce elapsed and CPU time. For information about the DSRSEXIT user exit, see “Using DSRSEXIT to manage VSAM data set redefinition” on page 834.

Additional performance information for SHRLEVEL CHANGE options

If INLINECP is not already specified in the installation options, BMC recommends that you specify COPY YES INLINE YES. If you do not want to make inline copies, specify the ICTYPE AUTO option. When you specify ICTYPE AUTO, REORG PLUS determines whether it can update the full copy data sets or will need to create an incremental image copy.

Specifying AUTO requires that you provide incremental copy data sets if REORG PLUS determines that it needs to create an incremental image copy. BMC recommends that you also specify DDTYPE LOCPICPY ACTIVE YES on the command to enable REORG PLUS to dynamically allocate the incremental copy data sets when they are needed.

Performance tuning for specific scenarios (any SHRLEVEL)

In addition to the general recommendations for REORG PLUS, use the following guidelines based on the needs of your site. The guidelines describe steps that you can take to tune specific types of reorganization jobs to improve performance.
Multiple indexes

Unless you are reorganizing an object that has many indexes but not much data, allocate one SYSUT1 data set per index. If you are reorganizing an object that has many indexes but not much data (such as an SAP object with hundreds of indexes), BMC recommends that you allocate a single SYSUT1 data set instead of one per index.

Many tables and indexes

Specify REDEFINE NO if both of the following conditions exist:

- You are reorganizing an object that has many tables and indexes.
- The purpose of the reorganization is solely to reorganize the data, not to redefine the data sets.

High availability

Specify the appropriate SHRLEVEL option for the performance goal. Note the following considerations when running SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganizations:

- If you are running in a non-RACF® site and you establish authority at a node lower than the highest node, see Chapter 2, “Operational considerations,” for information about setting authorizations for the data sets that are involved in the renaming process.

- Specify a value of DSN for the STAGEDSN installation option to avoid redundant data set rules.

Performance tuning for specific scenarios (SHRLEVEL CHANGE only)

The following sections describe steps that you can take to tune specific types of SHRLEVEL CHANGE reorganization jobs to improve performance.
Reorganization must complete

In an environment in which the reorganization must complete and you have transactions that will retry after a timeout, consider specifying the following values for the installation options:

- DRNWAIT=UTIL
- DRNRETRY=10
- DRNDELAY=3
- DSPLOCKS=RETRY

Transactions and reorganization are of equal priority

In an environment in which you want REORG PLUS to work like any other transaction, consider specifying the following values for the installation options:

- DRNWAIT=SQL
- DRNRETRY=15
- DRNDELAY=3
- DSPLOCKS=RETRY

Interpreting performance-related messages

REORG PLUS issues performance-related messages if you specify MSGLEVEL(1) on your EXEC statement or in your installation options. For more information about specifying this utility parameter, see “Utility parameters on the EXEC statement” on page 343.

Use the information provided in these messages to monitor REORG PLUS performance and to fine-tune future runs. For message explanations, access the BMC Documentation Center from the BMC Support Central site (http://www.bmc.com/support).

BMC50364I SORT PROCESSES SEVERELY CONSTRAINED BY MEMORY RESOURCES. SORTING CONTINUES, BUT PERFORMANCE MAY BE IMPACTED

This message indicates that BMCSORT found that the memory resources available are insufficient to provide the most efficient sorting. To allow the utility job to complete, BMCSORT might choose a less efficient sorting algorithm, which can increase the elapsed time of the job.
Also see message BMC50474I, which reports the memory that is available to the utility at the time of optimization. If the value for ABOVE as indicated in BMC50474I is substantially less than the value for your REGION parameter, your system might have a memory-limiting exit active.

One of the following actions might improve the sort efficiency for future jobs:

- Increase your region size, if possible.
- Limit the scope of the job that you are running to reduce the amount of work that the utility needs to do.
- Contact your systems programmer to increase the memory limit, if possible.

**BMC50397I**  
**phase PROCESSING CONSTRAINED BY REGION SIZE**

This message indicates that the reorganization is constrained because the amount of virtual memory available is insufficient for optimal performance. For most jobs, REORG PLUS continues processing. If you also receive message BMC50399E, however, the amount of virtual memory available to REORG PLUS is insufficient to continue.

BMC recommends that you specify REGION=0M on the JOB or EXEC statement of your execution JCL to tell the system to allocate the optimal amount of available virtual storage to the REORG PLUS job. However, if you are processing a large number of partitions, processing might be constrained, even if you specified REGION=0M. In this case, consider reorganizing fewer partitions in a single job.

**BMC50398I**  
**phase PROCESSING CONSTRAINED BY SORT WORK FILES**

This message indicates that the reorganization is constrained because the number of sort work files or the total amount of space for the sort work files is insufficient for optimal performance. For most jobs, REORG PLUS continues processing. If you also receive message BMC50399E, however, the number or size of the sort work files is insufficient for REORG PLUS to continue.

No action is required. However, consider specifying larger sort work files or more sort work files to improve performance.
Interpreting performance-related messages

**BMC50399E** *phase* PROCESSING UNABLE TO CONTINUE DUE TO CONSTRAINED RESOURCES

The reorganization is unable to continue because of constrained resources. Message BMC50397I, message BMC50398I, or both are issued before this message.

Make one or more of the following adjustments:

- Specify a larger region size.
- Specify larger sort work files or more sort work files.
- Adjust the installation options to improve performance.

**BMC50400I** *phase* PROCESSING CONSTRAINED BY INDEX WORK FILES

This message indicates that the reorganization is constrained because insufficient index work files are available, but the reorganization continues.

Specify additional index work data sets (SYSUT1nn).

**BMC50471I** *environment_information*

This message displays current values for each option in the installation options module. Use this information to verify that REORG PLUS is using the option value that you want to use.

**BMC50474I** BELOW 16M = nK, ABOVE 16M = nK, CPUS = n

This message displays the following information about virtual storage and CPU usage:

- amount of virtual storage available below the 16-MB line
- amount of virtual storage available above the 16-MB line
- number of physical CPUs available in the processor

Use this information to ensure that adequate virtual storage is available for REORG PLUS to use. For more information, see “Providing maximum virtual storage” on page 589 and messages BMC50475I and BMC50479I.

**BMC50476I** DDNAME = ddname, I/OS = n, I/O WAITS = w, RDB LOCK WAITS = r

This message displays the following performance information about I/O operations to sequential data sets:

- ddname associated with the I/O operations
- number of I/Os (blocks) written to or read from the data set
- number of waits issued for I/O completion
- number of waits for serialization of the data set
A wait count that is greater than 10 percent of the block count might indicate degraded performance.

Try allowing REORG PLUS additional buffer space for the associated data set (see “Buffer installation options” on page 575) or using multiple data sets. A high number of serialization waits might indicate the need to decrease the number of concurrent tasks or use multiple data sets. Also, with a single-phase reorganization, you have the option of not using SYSREC (for a table space reorganization) or SYSUT1 (for an index reorganization) data sets.

**BMC50477I**

```
  taskNumber: PARTITION = partitionNumber, ROWS/KEYS = n,
  I/O WAITS = w, DDNAME = ddname
```

This message displays the following performance information about I/O operations to VSAM data sets:

- processing task number
- table space partition number
- number of rows or keys in the partition
- number of waits issued for I/O completion
- ddname associated with the I/O operations

Use the ddname to find the actual number of I/Os that were issued to the associated data set. A wait count that is greater than 20 percent of the actual I/Os might indicate degraded performance. Try allowing REORG PLUS more virtual storage if REORG PLUS also issues message BMC50397I. Provide more sort work file space if REORG PLUS also issues message BMC50398I. Changing these values enables REORG PLUS to start more concurrent tasks.

**BMC50479I**

```
  TOTAL PAGES: availablePages, ALLOWED: allowedAvailablePages;
  AVAILABLE PAGES: freePages, ALLOWED: allowedFreePages
```

This message displays information that REORG PLUS obtains from the operating system (not the region). The `allowedAvailablePages` value is the value that you specified for the TOTALPAGEPCT option, and `allowedFreePages` is the value that you specified for the AVAILPAGEPCT option.

REORG PLUS considers the percentage of available and free pages when determining the maximum amount of memory that is allowed for sort processing. If insufficient memory exists to perform a minimum number of optimal sort processes, the job terminates with a constrained memory message.

No action is required. However, if you experience frequent system memory shortages, if possible, schedule your REORG PLUS jobs during periods of lower system activity.
**Interpreting performance-related messages**

**BMC50486I**  
*taskNumber: BMCSORT STARTED, \( n_K \) BELOW 16M, \( n_K \) TOTAL MEMORY, \( n \) PAGES HYPERSPACE*

This message displays the maximum amount of memory and 4-KB pages of hyperspace that the utility allows for each sort task. No action is required.

**BMC50720I**  
*taskNumber: \( n \) NEW LOG APPLY BUFFERS, \( n \) REUSED, \( n \) INITIAL MAXIMUM, \( n \) WAITS, BUFFER SIZE=\( bufferSize \)*

This message displays buffer information for the log apply process. The initial maximum value is the number of buffers that the utility determined that it needs for each log apply task. No action is required. However, the information displayed in this message can help you tune performance of the utility.

**BMC51302I**  
*MAX TASKS = \( t \), MAX PARTITIONS PER TASK = \( p \), SORTWKS PER TASK = \( s \), MAX OPEN PARTITIONS PER TASK = \( o \)*

This message displays the following results of task optimization for unload processing:

- maximum number of concurrent tasks
- maximum number of partitions that can be processed per task
- number of sort work files assigned to each task
- maximum number of open partitions per task

The maximum number of tasks that REORG PLUS runs depends on the available system resources, such as

- virtual storage
- number of physical CPUs
- number of unload index work and sort work files
- sort work file space

REORG PLUS might be able to run more tasks concurrently if you increase the amount of virtual storage and allocate more sort work files.

Based on the size of the partition and available sort work file space, REORG PLUS determines the number of partitions to process per task in order to allow the sort to run in the fastest manner. The actual number of partitions processed per task might be less than the maximum stated in the message.
This message displays the following results of index build task optimization:

- maximum number of concurrent tasks
- number of indexes per task
- number of sort work files assigned to each task
- maximum number of open index partitions per task

The maximum number of tasks depends on the available system resources, such as

- virtual storage
- number of physical CPUs
- number of unload index work and sort work files
- sort work file space

REORG PLUS might be able to run more tasks concurrently if you increase the amount of virtual storage and allocate more sort work files.

Based on the number of indexes, work data sets, and sort work file space, REORG PLUS determines the number of indexes to process per task in order to allow the sort to run in the fastest manner.

**Tuning to improve memory use**

To help you determine how to improve memory use, consider the information in the following sections.

**Memory requirements of the ORDER command option**

Jobs that use ORDER NO require that all partitions of a table space be open at one time, and therefore have a higher memory requirement. Jobs that use ORDER YES require that only one partition per task be opened at one time.
Tuning sort processing

Specifying a value greater than 0 for the SORTNUM installation or command option tells REORG PLUS that BMCSORT will dynamically allocate sort work data sets as needed, which should eliminate sort work constraints. The amount of sort multitasking that REORG PLUS performs depends on the number of CPUs, the SMAX or MAXSORTS option value, and available memory. You might be able to improve performance by adjusting the SMAX or MAXSORTS option value.

Using multitasking to improve memory use

Multitasking can improve memory use in your reorganization jobs. For information, see “Enabling multitasking for performance” on page 584.

Tuning copy processing

The number of copies that you make during a reorganization can affect the memory that your system uses. Balance your site’s backup copy needs with your system’s memory use requirements to determine the optimal number of copy data sets to allocate.

Memory constraints and data set allocation constraints are particularly critical when you are making copies while reorganizing a large number of partitions. Consider one of the following options in this case:

- Specify COPYLVL FULL to have REORG PLUS allocate a single copy data set for the table space.

- If you are reorganizing a subset of contiguous partitions, create a single copy by specifying COPYSUBSET=YES in the installation options module and COPYLVL FULL on your REORG command.

- If you need copies by partition, specify as few copies per partition as possible or consider specifying fewer partitions per job.
Tuning for availability

This section describes recommendations for the different availability requirements of your REORG PLUS jobs.

Specifying SHRLEVEL

Specify the appropriate SHRLEVEL option for your availability requirements. Use Table 91 to help determine the appropriate option.

Table 91  REORG PLUS availability options

<table>
<thead>
<tr>
<th>Access to data needed during the reorganization process</th>
<th>Additional site or application requirements</th>
<th>SHRLEVEL option</th>
</tr>
</thead>
<tbody>
<tr>
<td>no access</td>
<td>not applicable</td>
<td>SHRLEVEL NONE</td>
</tr>
<tr>
<td>read-only access</td>
<td>limited outage required</td>
<td>SHRLEVEL REFERENCE</td>
</tr>
<tr>
<td>read/write access</td>
<td>limited or no outage required</td>
<td>SHRLEVEL CHANGE</td>
</tr>
</tbody>
</table>

Renaming or switching staging data sets

To help ensure that a SHRLEVEL REFERENCE or SHRLEVEL CHANGE job completes successfully, BMC recommends that you specify FSFALLBACK=YES in either your installation options or command options. If you request FASTSWITCH processing for your staging data sets but FASTSWITCH processing fails, this option tells REORG PLUS to use rename processing instead. For more information, see “FSFALLBACK” on page 220.

Granting data set authority

For the data sets that REORG PLUS uses as staging data sets, ensure that UPDATE and CONTROL authorization is established when all of the following conditions exist:

- You are using SHRLEVEL CHANGE or SHRLEVEL REFERENCE.
- You are running REORG PLUS in a non-RACF environment.
- You establish authority at a node lower than the highest node.

For more information, see “Data set authorization” on page 69.
REORG PLUS processing phases

This section describes the processing phases of REORG PLUS in detail. This information can be useful when tuning your reorganization jobs.

NOTE
For a DSNUTILB reorganization, REORG PLUS passes processing to DSNUTILB after the UTILINIT phase and regains control during the UTILTERM phase. Therefore, the information in this section does not apply to a DSNUTILB reorganization.

REORG PLUS exploits the technology provided by large-scale processors. During execution, REORG PLUS examines available resources and uses as much of these resources as possible to maximize performance.

REORG PLUS architecture

The architecture of REORG PLUS differs from that of the IBM DB2 REORG utility. The DB2 REORG utility performs reorganization processing in several phases (UNLOAD, RELOAD, SORT, and BUILD), which run serially. However, REORG PLUS combines these phases into either a two-phase architecture or a single-phase architecture.

Two-phase architecture

When you specify UNLOAD CONTINUE or UNLOAD PAUSE, REORG PLUS uses two phases: UNLOAD and RELOAD. When you specify UNLOAD RELOAD, REORG PLUS combines the UNLOAD and RELOAD phases into a single processing phase named REORG. All functions of the DB2 REORG utility are performed in these phases of REORG PLUS.

The UNLOAD installation option determines the phase processing that REORG PLUS uses at a global level. You can override this value for a particular reorganization job by specifying the UNLOAD command option. You can only specify UNLOAD PAUSE on the command option.

The two-phase architecture allows REORG PLUS to perform several tasks concurrently, reducing the elapsed time for a table space reorganization. For example, REORG PLUS sorts the index keys and builds the indexes while it reloads the table space rows.
**Single-phase architecture**

The REORG PLUS single-phase architecture builds on the advantages of the two-phase processing and allows for even greater reductions in CPU usage and elapsed time. Single-phase processing eliminates some read and write processes by combining the UNLOAD and RELOAD phases.

**ANALYZE phase**

The ANALYZE phase provides information about the number of rows (cardinality) and the average row size of the data to be reorganized. The following performance factors arise from the ANALYZE phase, and command options that you specify control these factors:

- amount of time that the phase requires to run
- accuracy of the information gathered during the phase

Figure 50 and Figure 51 on page 602 show the objects that the ANALYZE phase might use (except ANALYZE HURBA, which uses no objects).

**Figure 50   ANALYZE phase for a table space reorganization**

---

*a* Only if you are using the SCAN option.

*b* Only if you are using REORG PLUS as part of the Database Performance solution.
UNLOAD phase

During the UNLOAD phase, REORG PLUS prepares data to allow concurrent processing for reloading table spaces and indexes. The UNLOAD phase prepares the unloaded rows for reloading into the specified tables by the RELOAD phase.

Depending on the data and the characteristics of the tables, REORG PLUS performs one or more of the following tasks during the UNLOAD phase:

- reads the rows from the DB2 tables or indexes
- sorts the rows (except in the instances noted in “SORTWK data sets” on page 366)
- writes the row images to the SYSREC data sets
- writes the index work records to the SYSUT1 data sets
- writes discarded rows from SELECT or DELETE processing to the SYSARC data set (if specified)
- builds or keeps the compression dictionary and compresses the rows

For additional information about the UNLOAD phase for a SHRLEVEL CHANGE reorganization, see “UNLOAD phase” on page 619 in Chapter 7, “Online reorganization.”
Resource allocation in the UNLOAD phase

REORG PLUS checks the amount of available virtual storage within the region and the number of processors in the CPU to determine how much data can be unloaded concurrently. If you specify a nonzero value for SMAX or MAXSORTS, REORG PLUS uses the lowest of the following values to determine the maximum number of tasks that can run concurrently:

- 16
- SMAX value specified in the installation options
- MAXSORTS value specified in the REORG command statement
- number of processors

If you specify 0 for SMAX and MAXSORTS, REORG PLUS uses the lower of the following values to determine the maximum number of tasks that can run concurrently:

- 16
- value of the multitasking options RORGMAX, UNLDMAX, or BILDMAX

In addition to using the various command and installation options, REORG PLUS considers the following factors when calculating the maximum number of tasks that can run concurrently:

- number of partitions in the table space and index space
- number of sort work data sets divided by two

After determining the maximum number of tasks, REORG PLUS calculates the number of sort work files per task by dividing the number of sort work files specified in the JCL by the number of tasks. REORG PLUS then calculates the amount of sort work space per task by multiplying the number of sort work files per task by the size of the sort work files.

Figure 52 on page 604 and Figure 53 on page 604 display the objects that REORG PLUS might use in the UNLOAD phase.
If the sort work space per task is sufficient to sort the largest partition, unload processing continues. If insufficient space is available to sort the largest partition, REORG PLUS reduces the number of tasks by one and recalculates the sort work space per task. The recalculations continue until either enough sort work space is available to sort the largest partition in a single task, or the number of tasks is reduced to less than one. When the number of tasks is reduced to less than one, REORG PLUS processing terminates with message BMC50399E, indicating constrained resources.
After REORG PLUS determines the number of sort tasks and amount of space, unload processing begins. REORG PLUS assigns partitions to each task as the task starts. The number of partitions unloaded per task varies with the size of the partition and the amount of sort work space available per task. The unload process determines the optimal number of partitions for each task as the task starts.

This balancing of tasks optimizes the reorganization process when the resources are available and allows nonoptimized processing to continue when only minimal resources are available.

**RELOAD phase**

REORG PLUS performs the following functions during the RELOAD phase:

- if required, adds partitions to a partition-by-growth table space (SHRLEVEL NONE and SHRLEVEL REFERENCE only)
- redefines the VSAM data sets (including the staging data sets for SHRLEVEL REFERENCE or SHRLEVEL CHANGE) when the value of the REDEFINE command or installation option is YES
- sorts indexes as required before index build
- rebuilds any indexes
- reloads the data into the table space or index spaces
- collects statistics
- produces image copies either as data is reloaded or after the data is reloaded
- for SHRLEVEL NONE, registers the copies in the DB2 catalog

For more information about the RELOAD phase for a SHRLEVEL CHANGE reorganization, see “RELOAD phase” on page 620 in Chapter 7, “Online reorganization.”
Resource allocation in the RELOAD phase

REORG PLUS checks the amount of available virtual storage within the region to determine how many indexes it can build concurrently. If you specify a nonzero value for SMAX or MAXSORTS, REORG PLUS uses the lowest of the following values to determine the maximum number of tasks that can run concurrently:

- SMAX value specified in the installation options
- MAXSORTS value specified in the REORG command statement
- number of processors

If you specify 0 for SMAX and MAXSORTS, REORG PLUS uses the value of the multitasking options RORGMAX, UNLDMAX, or BILDMAX to determine the maximum number of tasks that can run concurrently.

In addition to using the various command and installation options, REORG PLUS considers the following factors when calculating the maximum number of tasks that can run concurrently:

- number of indexes to be built
- number of SYSUT1 data sets
- number of sort work data sets divided by two

After REORG PLUS determines the maximum number of tasks that can run, it uses a recursive algorithm to balance the available virtual storage and sort work files. (Sort work files are not a consideration if you do not specify them in your JCL and you use the SORTNUM installation or command option.) As long as enough virtual storage space and sort work file space is available, processing continues. If either resource appears to be constrained, REORG PLUS adjusts downward the number of indexes built concurrently and repeats the checks.

If the maximum number of indexes to be built decreases to less than one, the reorganization terminates because of constrained resources. This balancing algorithm allows optimized reorganization when enough resources are available. It also allows processing to continue in a nonoptimized manner if only minimal resources are available.

Figure 54 on page 607 through Figure 57 on page 608 show the objects that REORG PLUS might use in the RELOAD phase.
During the UTILTERM phase, REORG PLUS performs rename and delete operations that result in the staging data sets replacing the original data sets.
Figure 56  RELOAD phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY index reorganization

Figure 57  RELOAD phase: SHRLEVEL REFERENCE index reorganization

During the UTILTERM phase, REORG PLUS performs rename and delete operations that result in the staging data sets replacing the original data sets.
**REORG phase for a single-phase reorganization**

During the REORG phase, REORG PLUS performs almost all of the same processes as in the UNLOAD and RELOAD phases. Depending on your command specifications, REORG PLUS performs one or more of the following tasks:

- reads the rows from the DB2 tables or indexes
- sorts the rows (except in the instances noted in “SORTWK data sets” on page 366)
- writes the row images to the SYSREC data sets
- writes the index work records to the SYSUT1 data sets
- writes discarded rows from SELECT or DELETE processing to the SYSARC data set (if specified)
- builds or keeps the compression dictionary and compresses the rows
- if required, adds partitions to a partition-by-growth table space (SHRLEVEL NONE and SHRLEVEL REFERENCE only)
- redefines the VSAM data sets (including the staging data sets for SHRLEVEL REFERENCE or SHRLEVEL CHANGE) when the value of the REDEFINE command or installation option is YES
- checks for duplicate key values in a unique index
- concurrently loads multiple partitions
- concurrently loads the table space and builds data-sorting indexes
- concurrently sorts and builds non-data-sorting indexes
- concurrently builds multiple, non-data-sorting indexes
- collects statistics while loading table spaces and building indexes
- produces image copies either while reloading data or after reloading rows

If you have a large number of nonpartitioned indexes, consider using a two-phase reorganization to have REORG PLUS concurrently reload the table space and build the nonpartitioned indexes. Otherwise, a single-phase reorganization provides significant CPU and elapsed time savings over a two-phase reorganization because REORG PLUS does not have to read from the SYSREC data set (for a table space reorganization) or the SYSUT1 data set (for an index reorganization).
For a single-phase reorganization, the considerations in Table 92 apply to the SYSREC data set for a table space reorganization or the SYSUT1 data set for an index reorganization, and depend on the SHRLEVEL that you specify.

Table 92  Single-phase reorganization considerations for SYSREC or SYSUT1 data set

<table>
<thead>
<tr>
<th>Specified SHRLEVEL</th>
<th>Type of reorganization</th>
<th>Effect on restartabilitya</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Table space</td>
<td>Index</td>
</tr>
<tr>
<td>SHRLEVEL NONE</td>
<td>You can omit SYSREC</td>
<td>You can omit SYSUT1</td>
</tr>
<tr>
<td>(the default)</td>
<td>data set for an</td>
<td>data set for an</td>
</tr>
<tr>
<td></td>
<td>additional performance</td>
<td>additional performance</td>
</tr>
<tr>
<td></td>
<td>gain.</td>
<td>gain.</td>
</tr>
<tr>
<td>SHRLEVEL REFERENCE</td>
<td>REORG PLUS does not</td>
<td>REORG PLUS does not</td>
</tr>
<tr>
<td></td>
<td>use SYSREC data set,</td>
<td>use SYSUT1 data set,</td>
</tr>
<tr>
<td></td>
<td>even if you specify it.</td>
<td>even if you specify it.</td>
</tr>
<tr>
<td>SHRLEVEL CHANGE</td>
<td>REORG PLUS does not</td>
<td>REORG PLUS does not</td>
</tr>
<tr>
<td></td>
<td>use SYSREC data set,</td>
<td>use SYSUT1 data set,</td>
</tr>
<tr>
<td></td>
<td>even if you specify it.</td>
<td>even if you specify it.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a  For more information about restarting the job during a single-phase reorganization, see “Restarting REORG PLUS” on page 391.

Except for the option of omitting the SYSREC or SYSUT1 data set, all of the performance considerations for the REORG phase are the same as those in the UNLOAD phase (page 602) and the RELOAD phase (page 605).

Figure 58 on page 611 through Figure 61 on page 613 show the objects that REORG PLUS uses in the REORG phase.
Figure 58  REORG phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY table space reorganization

The SYSRECnn data set is optional for a single-phase table space reorganization but is required for restarting.
REORG phase for a single-phase reorganization

Figure 59  REORG phase: SHRLEVEL REFERENCE table space reorganization

Figure 60  REORG phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY index reorganization
During the UTILTERM phase, REORG PLUS performs rename and delete operations that result in the staging data sets replacing the original data sets.
Chapter 7

Online reorganization

This chapter presents the following topics:

- Introduction to online reorganizations .................................................. 616
  - Overview of SHRLEVEL CHANGE .................................................. 616
  - How SHRLEVEL CHANGE works .................................................. 618
- How SHRLEVEL CHANGE differs in REORG PLUS and IBM DB2 REORG .... 629
  - SHRLEVEL CHANGE processing differences from DB2 REORG .......... 629
  - SHRLEVEL CHANGE administrative differences from DB2 REORG ..... 630
- Running a SHRLEVEL CHANGE reorganization .................................... 631
- Control of the log apply process ....................................................... 633
  - Log apply control options ....................................................... 633
  - Hierarchy of log apply control options ..................................... 636
  - Using XBM to view and dynamically control the log apply process .... 639
  - Log apply control option scenarios ........................................... 644
- Serialization and concurrency for SHRLEVEL CHANGE ....................... 646
  - Concurrency with other applications ....................................... 646
  - Object status for SHRLEVEL CHANGE ...................................... 647
- Operational considerations for online reorganizations .......................... 647
  - Interacting with applications .................................................. 648
  - Incompatible REORG PLUS options ........................................... 648
  - Allocation of spill data sets .................................................... 648
  - SHRLEVEL CHANGE considerations for using SELECT or DELETE .... 649
  - SHRLEVEL CHANGE statistics considerations ............................. 649
  - Support for APPLICATION RESTART CONTROL (AR/CTL) .......... 650
- Recoverability of the reorganized table space ..................................... 651
  - Copy data sets for SHRLEVEL CHANGE ..................................... 652
  - Incremental copy data sets for SHRLEVEL CHANGE ..................... 653
  - Copy registration failure during a SHRLEVEL CHANGE reorganization 656
- Restart considerations for a SHRLEVEL CHANGE reorganization ............. 657
  - Restarting before the UTILTERM phase ....................................... 657
  - Not completing the reorganization before the UTILTERM phase ....... 657
  - Restarting in the UTILTERM phase ........................................... 658
  - Not completing the reorganization in the UTILTERM phase .......... 658
- Performance considerations .............................................................. 659
  - Sizing memory for the RID translation map .................................... 660
  - Sizing memory for log records ................................................. 662
Introduction to online reorganizations

Performing an online reorganization by specifying SHRLEVEL CHANGE on the REORG command offers the following benefits:

- **allows full access** to DB2® data during most of the reorganization
- **delivers improved data availability** by greatly reducing the outage for the DB2 objects, thus helping to meet growing 24 x 7 requirements
- **operates in a nondestructive manner**, allowing you to make the objects available without having to recover in the event of a failure
- **optionally provides support for batch applications** that use the BMC Software APPLICATION RESTART CONTROL (AR/CTL) product

Online reorganizations use the AR/CTL suspend-and-resume interface, which eliminates the outage for DB2 objects.

To use the SHRLEVEL CHANGE option, you must have installed the BMC Software EXTENDED BUFFER MANAGER (XBM) product or SNAPSHOT UPGRADE FEATURE (SUF) component of XBM.

This chapter discusses online reorganizations in depth, concentrating on the aspects of reorganization processing that are unique to SHRLEVEL CHANGE. Aspects that are the same as those for other types of reorganizations are covered in other chapters and appendixes.

Overview of SHRLEVEL CHANGE

To invoke an online reorganization, REORG PLUS provides a SHRLEVEL CHANGE keyword. The keyword allows read-write (RW) application access to DB2 table space and index space objects during the reorganization process.
When you specify the SHRLEVEL CHANGE keyword, REORG PLUS performs an online reorganization, which includes the following processing steps after the utility initializes:

1. *(optional)* analyzes the original DB2 table space and index space objects

2. externalizes changed pages for the original table space and index space objects and starts capturing log records

3. unloads table space or index space data from the original DB2 objects by using the XBM product or its SUF technology

4. allocates staging objects that mirror the original table space or index space objects, or uses predefined staging objects

5. reloads data into the staging table space objects, and builds or updates the staging index space objects

6. creates a row ID (RID) translation map for a table space reorganization

7. applies log records to the staging objects to reflect all update activity since XBM was initialized

   REORG PLUS uses the RID translation map to reconcile the log records for a table space reorganization.

8. prevents any access to the original objects and switches the data sets, which causes the staging objects to replace the original objects

9. registers full and incremental copies, ensuring recoverability of objects

10. allows full access to the objects to resume

11. optionally updates the DB2 catalog and DASD MANAGER PLUS statistics tables

Figure 62 on page 618 illustrates the processing flow for a SHRLEVEL CHANGE reorganization. For details and information about the objects’ status during the reorganization, see “How SHRLEVEL CHANGE works” on page 618.
How SHRLEVEL CHANGE works

Processing for SHRLEVEL CHANGE is similar to SHRLEVEL REFERENCE. However, some additional tasks are performed in the UNLOAD and RELOAD phases, and two additional phases occur, LOGAPPLY and LOGFINAL. This section describes the key tasks that are performed during each execution phase. Figure 63 on page 623 through Figure 66 on page 628 illustrate the data sets that REORG PLUS uses during each phase.

**UTILINIT and ANALYZE phases**

In the initial phases, a SHRLEVEL CHANGE reorganization is the same as SHRLEVEL REFERENCE. The UTILINIT phase initializes the job, verifies the REORG command and IDCAMS control statements, and performs DB2 catalog lookups. The ANALYZE phase analyzes the objects and optionally produces statistics.
UNLOAD phase

At the beginning of the UNLOAD phase (for a two-phase reorganization), REORG PLUS initializes the XBM interface and, during initialization, prevents update access to all of the tables that are involved in the reorganization. However, for the remainder of the UNLOAD and RELOAD phases, REORG PLUS allows updates to all of the tables. REORG PLUS performs the actual reorganization and rebuilding of indexes on the staging data sets.

At the beginning of the UNLOAD phase, REORG PLUS initializes the interface to the XBM Utility Monitor function. Under certain limited conditions, REORG PLUS turns on DB2 Data Capture Changes, telling DB2 to log the entire row rather than only the changed portion of the row.

REORG PLUS also initializes the log-control tasks, which are specific to SHRLEVEL CHANGE. These tasks capture any changes that the user makes to the table space or the indexes after XBM initialization. REORG PLUS obtains the log record that reflects each change and stores that record in an internal structure in memory. If necessary, this structure spills to disk. You can allocate the amount of memory and disk space by using the options for SHRLEVEL CHANGE.

For a table space reorganization, REORG PLUS creates a RID translation map, which is used in the LOGAPPLY phase to match DB2 log records with the appropriate rows in the newly reorganized object. The RID maps are also stored in data spaces and spill to disk if necessary. You can allocate the amount of RID map memory by using the options for SHRLEVEL CHANGE.

REORG PLUS unloads the table space and index space data. REORG PLUS uses a snapshot of the object that XBM provides for both table space and index space data. Using the snapshot ensures that REORG PLUS reads the original pages (those present when XBM was initialized) rather than pages that contain updates.

For a partial table space reorganization of a partitioned table space where nonpartitioned indexes are defined, REORG PLUS defines the staging data sets and uses XBM to copy the nonpartitioned indexes from the original data sets to the staging data sets in this phase.

For more information about topics discussed in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>when REORG PLUS turns on Data Capture Changes</td>
<td>DATACAP installation option on page 683</td>
</tr>
<tr>
<td>spill data sets</td>
<td>“Allocation of spill data sets” on page 648</td>
</tr>
<tr>
<td>SHRLEVEL CHANGE options</td>
<td>“SHRLEVEL CHANGE options” on page 326</td>
</tr>
<tr>
<td>performance considerations for the UNLOAD phase</td>
<td>“UNLOAD phase” on page 602</td>
</tr>
</tbody>
</table>
How SHRLEVEL CHANGE works

RELOAD phase

In the RELOAD phase, REORG PLUS performs the same tasks as for SHRLEVEL REFERENCE, including reloading the table space and index space data. REORG PLUS allocates the staging data sets and writes the reorganized data to them, as follows:

<table>
<thead>
<tr>
<th>Reorganization type</th>
<th>REORG PLUS performs this action</th>
</tr>
</thead>
<tbody>
<tr>
<td>full or partial table space</td>
<td>loads the table space data into the table space staging data sets</td>
</tr>
<tr>
<td>full table space</td>
<td>loads the index data into the index staging data sets</td>
</tr>
<tr>
<td>partial table space</td>
<td>updates the nonpartitioned indexes that REORG PLUS previously copied to the staging data sets</td>
</tr>
<tr>
<td>index-only</td>
<td>loads the index data into the index staging data sets</td>
</tr>
</tbody>
</table>

During this phase, REORG PLUS also sorts the indexes and collects statistics. For a table space reorganization, REORG PLUS makes a full image copy of the table space. However, REORG PLUS does not register the copy in the SYSIBM.SYSCOPY table until the UTILTERM phase. At the end of the RELOAD phase, the staging data sets contain a reorganized version of the original data sets as they appeared when XBM took the snapshot.

For information about performance considerations for the RELOAD phase, see “RELOAD phase” on page 605.

REORG phase

For SHRLEVEL CHANGE, REORG PLUS combines the operations of the UNLOAD and RELOAD phases of a two-phase reorganization into the REORG processing phase for a single-phase reorganization.

LOGAPPLY phase

When the RELOAD or REORG phase is complete, the LOGAPPLY phase begins. This phase applies the stored log records to the reorganized staging data sets. REORG PLUS continuously monitors parameters and events until it reaches one of the criteria that triggers the LOGFINAL phase, such as reaching the log threshold (as specified with the LOGTHRESHLD option). The criteria are based on the log apply control options that you set in the options module or with the REORG command. For more information, see “Control of the log apply process” on page 633.
REORG PLUS also provides statistics regarding the progress of the LOGAPPLY phase and the overall reorganization. This information is available through the XBM Utility Monitor interface or the MVS™ console. In addition to viewing the status of the job, you can dynamically change the settings of any of the log apply control options by using the XBM Utility Monitor interface or the MVS console. For more information, see “Using XBM to view and dynamically control the log apply process” on page 639.

Throughout this phase, REORG PLUS continues to monitor the DB2 log for records that pertain to the objects that you are reorganizing. If the value for ICTYPE is UPDATE, REORG PLUS updates the full copy data sets throughout the LOGAPPLY phase.

**LOGFINAL phase**

During the LOGFINAL phase, REORG PLUS controls access to the original table space and index space objects based on the value that you specify for the DRAIN command option or DRAINTYP installation option, as follows:

- Prevents only updates if you specify WRITERS
- Prevents all access if you specify ALL

Controlling access stops the arrival of new log records and allows REORG PLUS to finish applying the last of the existing log records to the staging data sets. If required and the table space is partition-by-growth, REORG PLUS adds partitions if the table space is partition-by-growth. Then, depending on the value of ICTYPE, REORG PLUS performs one of the following actions:

- Creates incremental copy data sets to record changes made since REORG PLUS created the full copy data sets
- Updates the full copy data sets

However, REORG PLUS does not register any copies in SYSIBM.SYSCOPY until the UTILTERM phase.

If Data Capture Changes was turned on in the UNLOAD phase, REORG PLUS turns it off in this phase. In some error conditions, REORG PLUS might not be able to turn off Data Capture Changes. For more information, see “Not completing the reorganization before the UTILTERM phase” on page 657.

Before UTILTERM, REORG PLUS makes changes only to the staging data sets. If a failure occurs, no restart is needed. You can continue using the original data sets as if no reorganization attempt was made. For more information, see “Not completing the reorganization in the UTILTERM phase” on page 400.
How SHRLEVEL CHANGE works

**UTILTERM phase**

After the LOGFINAL phase completes and the UTILTERM phase begins, REORG PLUS prevents any access to the original table space or index space objects and performs either the rename or FASTSWITCH process, as shown in the following table. For more information about the rename and FASTSWITCH processes, see “Staging data sets” on page 101.

<table>
<thead>
<tr>
<th>Process used</th>
<th>REORG PLUS action</th>
</tr>
</thead>
</table>
| rename       | - renames the original data sets to a backup name  
- renames the staging data sets to the original data set names  
This action causes the staging data sets to replace the original data sets. |
| FASTSWITCH   | bypasses the rename process, and changes the DB2 catalog to point to the staging data sets |

If REORG PLUS fails during the rename or FASTSWITCH process, BMC recommends that you restart the job. If you are unable to restart the job, see “Not completing the reorganization in the UTILTERM phase” on page 400 for information about the actions to take next.

**NOTE**

REORG PLUS automatically backs out the reorganization when a failure occurs during rename processing and you are reorganizing a catalog object.

If you are using the DDLIN data set to alter limit keys, REORG PLUS reorganizes the data in the partitions based on the new key values and alters the limit keys during the UTILTERM phase, thus rebalancing the data in the partitions. For more information, see “DDLIN data set considerations” on page 131.

REORG PLUS registers the full image copy data sets and incremental copy data sets (if incremental copies were created) in the SYSIBM.SYSCOPY table.

After completing the rename or FASTSWITCH process, REORG PLUS allows all access to the objects to resume. REORG PLUS optionally deletes the original data sets or renames them in preparation for a subsequent reorganization. REORG PLUS updates the BMCHIST table and optionally updates

- real-time statistics  
- statistics in the DB2 catalog  
- DASD MANAGER PLUS statistics tables

When REORG PLUS completes this phase, the reorganization is complete.
Data sets by execution phase

Figure 63 through Figure 66 on page 628 show the data sets that each execution phase uses during a SHRLEVEL CHANGE reorganization. For more information about the data sets, see “REORG PLUS data sets” on page 59 and “Staging data sets” on page 101.

Figure 63  Processing phases: SHRLEVEL CHANGE two-phase table space reorganization (part 1 of 2)
How SHRLEVEL CHANGE works

Figure 63  Processing phases: SHRLEVEL CHANGE two-phase table space reorganization (part 2 of 2)

All phases use the SYSPRINT data set and update the BMCUTIL and BMCSYNC tables.

For partial reorganizations, the nonpartitioned indexes are copied to the staging data sets before the index update operation.

All sort processing uses the UTPRINT data set.

The UNLOAD phase updates the BMCDICT table.

The UTILTERM phase updates the BMCHIST table.

LOGAPPLOY phase

LOGFINAL phase

UTILTERM phase

DB2 catalog

DASD MANAGER PLUS statistics tables

BICYnn

BICZnn

BIRYnn

BIRZnn

Staging table space

Staging indexes

During the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

Depending on the options that you specify, REORG PLUS either updates the full image copy data sets in the LOGAPPLOY and LOGFINAL phases, or creates incremental copy data sets in the LOGFINAL phase.

continued from the preceding diagram
How SHRLEVEL CHANGE works

Chapter 7 Online reorganization 625

Figure 64 Processing phases: SHRLEVEL CHANGE two-phase index reorganization

All phases use the SYSPRINT data set and update the BMCUTIL and BMCSYNC tables.

All sort processing uses the UTPRINT data set. The UTILTERM phase updates the BMCHIST table.

During the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

How SHRLEVEL CHANGE works
Figure 65  Processing phases: SHRLEVEL CHANGE single-phase table space reorganization (part 1 of 2)
Figure 65  Processing phases: SHRLEVEL CHANGE single-phase table space reorganization (part 2 of 2)

- LOGAPPLY phase
- LOGFINAL phase
- UTILTERM phase

Depending on the options that you specify, REORG PLUS either updates the full image copy data sets in the LOGAPPLY and LOGFINAL phases or creates incremental copy data sets in the LOGFINAL phase.

During the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

All phases use the SYSPRINT data set and update the BMCUTIL and BMCSYNC tables. For partial reorganizations, the nonpartitioned indexes are copied to the staging data sets before the index update operation.

All sort processing uses the UTPRINT data set. The UNLOAD phase updates the BMCDICT table. The UTILTERM phase updates the BMCHIST table.
How SHRLEVEL CHANGE works

Figure 66   Processing phases: SHRLEVEL CHANGE single-phase index reorganization

All phases use the SYSPRINT data set and update the BMCUUTIL and BMCSYNC tables.

During the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

All sort processing uses the UTPRINT data set.
The UTILTERM phase updates the BMCHIST table.
How SHRLEVEL CHANGE differs in REORG PLUS and IBM DB2 REORG

REORG PLUS has several features that distinguish it from the IBM DB2 REORG utility. These features improve performance, make a SHRLEVEL CHANGE reorganization easier to administer, and provide the least disruptive reorganization to applications that are accessing the data.

SHRLEVEL CHANGE processing differences from DB2 REORG

The SHRLEVEL CHANGE option of REORG PLUS allows DB2 table space and index space objects to be available for RW access by DB2 applications during most of the reorganization.

Allocating storage versus using DB2 resources

To be truly nondisruptive, a reorganization utility must do more than simply allow application access. If the reorganization utility competes with the application for database resources, disruption in the form of degraded performance can still occur. To avoid this situation, REORG PLUS does not use the DB2 buffer pool, buffer manager, data manager, or work databases to accomplish its tasks.

Instead, REORG PLUS allocates data sets, staging copies of the DB2 objects, and virtual storage as needed. This additional DASD and utilization of virtual storage space eliminates the use of DB2 resources to accomplish the reorganization. In addition, the RID map is not a DB2 object in REORG PLUS as it is in the IBM DB2 REORG utility. This factor prevents RID map access from interfering with DB2 processing. By eliminating contention for these resources between REORG PLUS and the application, the reorganization can proceed in a less disruptive manner.

Determining the longlog condition

REORG PLUS and DB2 REORG differ fundamentally in how they determine a longlog condition. DB2 REORG has an internal buffer that contains log records. When this buffer becomes full, DB2 REORG processes the records. DB2 REORG then repeats the process, making another pass at the next batch of log records. It uses an average of the last \( n \) passes to determine whether a longlog condition exists, or whether processing can be completed by the time that is set by the DEADLINE option or by the MAXRO value.
In contrast, REORG PLUS samples the arrival rate of the log records every 15 seconds to determine the rate at which the records are being applied. It uses a weighted average of the last 15 samples to determine whether a longlog condition exists, or whether processing can be completed by the time that is set by DEADLINE or by the MAXRO value.

By using more samples and a weighted average, REORG PLUS provides the following advantages over DB2 REORG:

- a greater level of accuracy in determining whether processing can be completed by the time that is set by DEADLINE or by the MAXRO value
- an increased probability that sudden changes in activity levels will not force unnecessary longlog conditions
- the ability to specify LOGTHRSHLD to define when the process will end

**Online partition rebalancing**

When you use a DDLIN data set with ALTER statements, REORG PLUS performs online partition rebalancing and alters limit keys for you with no outage and without your space entering REORP status. Online partition rebalancing is performed on the staging data sets and the ALTER commands are performed in the UTILTERM phase. For more information, see “DDLIN data set considerations” on page 131.

Alternatively, you can use the REBALANCE command option to rebalance partitions. By specifying REBALANCE, you can have REORG PLUS determine the limit keys for you. For more information, see “REBALANCE” on page 179.

**RID translation maps**

One of the memory structures that is required for a log apply reorganization is a RID translation map, which REORG PLUS maintains in data spaces. This object is required only during the reorganization so there is no need for it to persist after the reorganization is complete. The RID translation map is a temporary object.

SHRLEVEL CHANGE administrative differences from DB2 REORG

To reduce the administrative burden of running an online reorganization, REORG PLUS has automated and simplified some of the tasks.
REORG PLUS performs the following RID map tasks:

- allocates it for you

  You do not need to define a data set or DB2 table in advance to contain the RID map.

- assigns the name and ensures that it is unique across your systems

  There is no contention among reorganization jobs for the use of a RID map of a particular name.

**Staging data set allocation**

REORG PLUS offers several options for defining the staging data sets. You can choose the option that best suits your environment:

- Define the staging data sets through IDCAMS commands that you supply to REORG PLUS in the SYSIDCIN DD statement for VCAT-defined objects.

- Have REORG PLUS automatically define the staging data sets for storage-group-defined objects.

- Predefine the staging data sets by using a separate IDCAMS step, and specify NO on the REDEFINE command or installation option.

You can instruct REORG PLUS to complete one of the following actions after the reorganization is complete:

- delete the old data sets that previously held your DB2 table space and index space objects

- automatically rename the data sets to the staging data set names

**Running a SHRLEVEL CHANGE reorganization**

This section describes the steps to take to run a SHRLEVEL CHANGE reorganization. For examples of syntax and output, see the following sections:

- “Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning” on page 529

- “Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY” on page 542
Running a SHRLEVEL CHANGE reorganization

“Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing” on page 553

Before you begin

Ensure that you have installed a version of XBM or SUF that is supported for SHRLEVEL CHANGE in REORG PLUS as described in “Software requirements” on page 66.

Also, ensure that you have the appropriate authorizations as described in “Additional authorizations for SHRLEVEL CHANGE” on page 67 and “Additional authorizations for using XBM or SUF” on page 69.

To run a SHRLEVEL CHANGE reorganization

1 Complete the following XBM or SUF steps:

   A Create and activate a management set that contains the appropriate snapshot object definition.

   B Create and activate a configuration that contains the appropriate cache attributes.

   For more information, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

2 Specify SHRLEVEL CHANGE on the REORG command statement (see page 173).

3 (optional) Use the XBMID installation or command option to specify the XBM subsystem that you want REORG PLUS to use for snapshot processing.

   For information about the XBMID command option, see “XBMID” on page 253.

4 If you are running a reorganization that might invoke DSNUTILB, supply a mapping table name or ensure that a mapping table will be created.

   For more information, see “Enabling REORG PLUS to invoke DSNUTILB” on page 75.

5 Specify any additional SHRLEVEL CHANGE options as needed (see descriptions beginning on page 326).

6 Use the information in the remainder of this chapter to ensure that you create JCL that is appropriate for your environment and job requirements.
Control of the log apply process

You can control the log apply process with log control options:

- set defaults for the log apply control options in the installation options module
- override the default options by using the REORG command
- change the log apply control options while the reorganization is running (up to and including the LOGAPPLY phase)

To change options while the reorganization is running, you can use the XBM Utility Monitor function or the MVS console. However, after beginning the LOGFINAL phase, REORG PLUS ignores any changes to the options. For more information, see “Using XBM to view and dynamically control the log apply process” on page 639.

Depending on the values that have been set for these options, you can allow the LOGAPPLY phase to continue indefinitely, start the LOGFINAL phase, or terminate the reorganization.

Log apply control options

Table 93 lists the options that REORG PLUS provides to allow you to control the log apply process during the reorganization. You do so by changing any one of the values for the options that REORG PLUS uses in the LOGAPPLY phase. In addition to the options that change execution, REORG PLUS provides the DISPLAY option to monitor the process.

Table 93 Log apply control options (part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEADLINE</td>
<td>DEADLINE specifies the time by which the LOGFINAL phase should finish applying log records. If REORG PLUS determines that the LOGFINAL phase will not finish by the deadline, it terminates the reorganization. When calculating the estimate, REORG PLUS does not include the time required to produce incremental copy data sets (if creating them) or the time needed for the UTILTERM phase. REORG PLUS does not check the DEADLINE value until the LOGAPPLY phase begins. If a timestamp or time value is specified on DEADLINE and LOGFINAL, the calculated LOGFINAL timestamp must be earlier than the calculated DEADLINE timestamp. You can specify NONE to indicate that there is no deadline.</td>
</tr>
</tbody>
</table>

Considerations for log apply control options

The following additional considerations apply when you use the log apply control options:

- If MAXRO is DEFER, the LOGAPPLY phase continues indefinitely, and REORG PLUS ignores all other options (except DEADLINE). You must change MAXRO to a value (and possibly set other options) to complete the reorganization.
Because it uses a weighted average of samples, REORG PLUS does not detect a longlog condition during brief spurts of activity.

If REORG PLUS detects a longlog condition but ends before reaching the DELAY time, REORG PLUS resets the DELAY time to its original value.

**Criteria for entering the LOGFINAL phase**

For REORG PLUS to stop the LOGAPPLY phase and enter the LOGFINAL phase, both of the following conditions must exist:

- REORG PLUS estimates that it can complete the LOGFINAL phase by the time that is specified in DEADLINE (if DEADLINE has a value other than NONE).
- MAXRO is set to a value other than DEFER.

In addition, at least one of the following conditions must exist:

- The time specified in the LOGFINAL option has been reached.
- LOGFINAL is NONE, a longlog condition does not exist, and either the MAXRO or the LOGTHRESHLD conditions exist.
- The longlog condition exists, the time that is specified in the DELAY option has been reached, and LONGLOG is set to DRAIN.

The reorganization cannot complete if the LOGFINAL phase does not start. Also, if too many changes are made to the data sets between the time that XBM is initialized and the time that REORG PLUS applies the incremental image copies, the new data sets might be relatively disorganized. However, the new data sets should be less disorganized than the originals.

**Criteria for ending the reorganization**

If necessary, you can also end the reorganization before it completes, based on how you set the log apply control options. The reorganization ends if either of the following conditions exists:

- REORG PLUS estimates that it cannot finish processing before the time that you specify in the DEADLINE option.
- A longlog condition exists, the time that you specify in the DELAY option is reached, and you set LONGLOG to TERM.
Figure 67 illustrates how REORG PLUS uses the log apply control options to control SHRLEVEL CHANGE processing after the LOGAPPLY phase starts. The figure represents one 15-second sample interval. Processing repeats indefinitely, until either the LOGFINAL phase begins or the reorganization ends.

**Hierarchy of log apply control options**

Figure 67  How options determine log apply processing (part 1 of 3)
Figure 67  How options determine log apply processing (part 2 of 3)

- **LOGFINAL=NONE?**
  - Yes: Start LOGFINAL phase
  - No: Is it time to start LOGFINAL phase?
    - Yes: Start LOGFINAL phase
    - No:
      - Does a longlog condition exist?
        - Yes: Start LOGFINAL phase
        - No: Stop timer; reset DELAY value
      - MAXRO or LOGTHRESHLD conditions exist?
        - Yes: Start LOGFINAL phase
        - No: A

- A
- C
Figure 67  How options determine log apply processing (part 3 of 3)

- **C**:
  - Timer already started? (No)
    - End of DELAY interval reached?
      - Yes: Set timer to DELAY value and start timing
      - No: LONGLOG = CONTINUE?
    - No: LONGLOG = TERMINATE?
      - Yes: Reorganization fails
      - No: LONGLOG = DRAIN
        - Start LOGFINAL phase
  - Yes: Start LOGFINAL phase

- **A**
Using XBM to view and dynamically control the log apply process

With XBM or SUF, you can view information about the reorganization while it is running, and change the options that affect the LOGAPPLY phase. You can access REORG PLUS by using the XBM Utility Monitor function, or you can send a command from the MVS console through XBM to REORG PLUS.

Utility Monitor

The XBM ISPF interface provides a Utility Monitor. You can use this monitor to display statistics and information about the SHRLEVEL CHANGE reorganization jobs that are currently running. You can also use the Utility Monitor to change the log apply control options (as discussed in “Log apply control options” on page 633) for any of those jobs.

Monitor a job

You can use the XBM ISPF interface for a menu-driven approach to monitoring and changing a log apply reorganization. After you select the Utility Monitor function and a specific reorganization, the XBM REORG PLUS Online panel displays information about that reorganization job. The panel displays the job name, step name, start date, start time, and execution phase that is currently processing. The panel also displays the current values for the log apply control options, as well as information about the data sets that have been registered with XBM.

After REORG PLUS enters the LOGAPPLY phase, the REORG PLUS Online panel replaces the registered data set information with statistics about the log records for both the index and the table space. These statistics include the number of records that have been applied, the number that have been queued, the arrival rate, and the application rate.

You can also enter the REORG PLUS DISPLAY option from the XBM Send New command panel to get additional information about processing of the log records. For a description and syntax, see “Display information about a job” on page 640.

For detailed information about the REORG PLUS Online panel, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.
Change the log apply process

You can use the following methods to change the log apply process:

- Type over the values that are displayed for any of the log apply control options on the REORG PLUS Online panel.

  You can type over any value until the LOGFINAL phase begins. After the LOGFINAL phase begins, REORG PLUS ignores any changes that you make to these values.

- Use the Send New command from the Command menu to send a new value for an option to REORG PLUS.

  You can enter only one option and value on each Send New command. Use the format option value, as in the following example:

  MAXRO 300

Display information about a job

The REORG PLUS DISPLAY option displays information about the selected log apply job. The syntax of the DISPLAY option follows:

| DISPLAY | TERSE | VERBOSE |

You can display information about log record processing before and during the LOGAPPLY phase, or you can get detailed information about each log apply task. To do so, enter the DISPLAY option with a value of TERSE or VERBOSE on the Utility Monitor Send New Command panel:

- DISPLAY TERSE displays a summary of information for all of the log apply tasks for this job, including the table space and index space.

- DISPLAY VERBOSE displays information and status for each log apply task that is associated with this job.

**NOTE**

Alternatively, you can enter the DISPLAY option with the XBM SEND command on the MVS console. For more information, see “MVS console” on page 644.
**DISPLAY TERSE output**

**Figure 68** shows an example of the output from the DISPLAY TERSE command.

**Figure 68  DISPLAY TERSE output**

```
<table>
<thead>
<tr>
<th>View Last Command</th>
<th>Row 1 to 5 of 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job name . . : ARUAVR$E</td>
<td></td>
</tr>
<tr>
<td>Utility name : REORG PLUS Online</td>
<td></td>
</tr>
<tr>
<td>Command . . : display terse</td>
<td></td>
</tr>
</tbody>
</table>

Following response received return code : 0

BMC50803  LOG APPLY DISPLAY AT 05/30/2005 09:17:17.974617
BMC50822  CURRENT PHASE: LOGAPPLY, STARTED: 05/30/2005 08:58:35.938669
BMC50804  LOG APPLY TASKS : 1 TASKS, 1 STARTED, 0 FINISHED
BMC50805  0 RECORDS QUEUED, 9984 RECORDS APPLIED
BMC50806  INDEX APPLY TASKS: 3 TASKS, 3 STARTED, 0 FINISHED

Command ===> ____________________________________________________________
F1=Help     F2=Split    F3=Exit     F7=Bkwd     F8=Fwd      F9=Swap      F10=Actions F12=Cancel

<table>
<thead>
<tr>
<th>View Last Command</th>
<th>Row 6 to 6 of 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job name . . : ARUAVR$E</td>
<td></td>
</tr>
<tr>
<td>Utility name : REORG PLUS Online</td>
<td></td>
</tr>
<tr>
<td>Command . . : display terse</td>
<td></td>
</tr>
</tbody>
</table>

Following response received return code : 0

BMC50805  0 RECORDS QUEUED, 29952 RECORDS APPLIED
****************************************************************************** Bottom of data ******************************************************************************

Command ===> ____________________________________________________________
F1=Help     F2=Split    F3=Exit     F7=Bkwd     F8=Fwd      F9=Swap      F10=Actions F12=Cancel
```
**DISPLAY VERBOSE output**

Figure 69 shows an example of the output from the DISPLAY VERBOSE command.

**Figure 69  DISPLAY VERBOSE output (part 1 of 2)**

---

**Figure 69  DISPLAY VERBOSE output (part 2 of 2)**
Figure 69  DISPLAY VERBOSE output (part 2 of 2)
MVS console

You can use the MVS console to issue commands to an XBM subsystem, and to view and respond to write-to-operator (WTO) notifications.

Issuing commands to an XBM subsystem

Instead of using the XBM Utility Monitor menu interface, you can issue commands at the MVS console to a particular XBM subsystem, which then routes the commands to REORG PLUS for the specified job. You can also use a batch program to send the commands to the MVS console. To send a command from the MVS console, use the following format, where ssid represents the XBM subsystem ID:

\[
\text{ssid SEND jobName option value}
\]

For example, to set the MAXRO value to 300 for job ARUAVR$A that is running on XBM0, issue the following command:

XBM0 SEND ARUAVR$A MAXRO 300

The valid options and values are the same as those for the XBM Send New command, which you can issue from the Command menu option within the Utility Monitor. The only difference is that you must supply the SSID and job name, whereas the Utility Monitor appends that information for you.

WTO notifications

When REORG PLUS detects a longlog condition, it sends an action WTO notification. The ROUTCDE and DESCDEC installation options determine which console receives the WTO and how it is formatted. REORG PLUS deletes the WTO when the longlog condition ends or the LOGFINAL phase begins.

For information about the ROUTCDE and DESCDEC installation options, see Appendix A, “REORG PLUS installation options.” For information about the LONGLOG option, see “LONGLOG” on page 329.

Log apply control option scenarios

The following scenarios are typical scenarios that you might encounter when using the log apply options to control the reorganization to fit your business needs.
I do not want the reorganization to end until some other event occurs (for example, my batch job finishes or the IBM CICS® environment is brought down).

Either set the value of MAXRO to DEFER in the options module, or specify MAXRO DEFER along with SHRLEVEL CHANGE on the REORG command. After the designated event occurs, you can use the XBM Utility Monitor function to dynamically change MAXRO to a reasonable value. Alternatively, an automated program can use the MVS console to change the value of MAXRO to allow the reorganization to be completed.

Ensure that the value for DEADLINE allows enough time for REORG PLUS to complete its processing. Also, if a longlog condition occurs, set the DELAY to a reasonable interval, and set LONGLOG to DRAIN.

I want to reorganize a large table space. However, the data must be available at 8:00 A.M. when my workforce begins the day.

Use either of the following approaches:

- Set the value for DEADLINE to a time before 8:00 A.M., allowing enough time between the deadline and 8:00 for REORG PLUS to produce incremental copies (if needed) and switch the data sets.

  If REORG PLUS determines that the LOGFINAL phase will not finish by the deadline, it terminates. When calculating the estimate, REORG PLUS does not include the time that is required to produce incremental copy data sets (if needed) or the time that is needed for the UTILTERM phase.

  If REORG PLUS determines that the LOGFINAL phase will finish by the deadline, it begins the UTILTERM phase and switches the data sets.

- Use the LOGFINAL command option with a time value before 8:00 A.M. After you calculate the approximate time for REORG PLUS to apply the log records and complete the LOGFINAL and UTILTERM phases, subtract that time from 8:00 A.M. and set the LOGFINAL command option to that value.

My telephone operators finish taking orders at 6:00 P.M. and the database is not updated until the next shift begins at 8:00 P.M. I need to schedule my reorganization so that the rename or FASTSWITCH process occurs only during this window.

To minimize impact, start the reorganization early enough before the beginning of the rename or FASTSWITCH processing window so that REORG PLUS will be ready to perform the rename or FASTSWITCH process at the appropriate time (6:30 P.M. in this example). When you enter the REORG command, set the time (or equivalent timestamp) on the LOGFINAL option to 18:30:00. If the REORG (or RELOAD) phase
has completed, REORG PLUS starts the LOGFINAL phase by 6:30 P.M., ensuring that the rename or FASTSWITCH process occurs when you want it to. You can also use the DEADLINE option to ensure that the reorganization terminates if the reorganization cannot start the LOGFINAL phase by 8:00 P.M.

The log apply processing seems to be taking longer than expected and a longlog condition exists.

If you set LONGLOG to CONTINUE and set DELAY to a reasonable amount of time, you will receive a warning message in the SYSPRINT data set and on the MVS console when the DELAY interval has been reached. At that time, you can decide what to do next. You can end the reorganization (by setting LONGLOG to TERM), start the LOGFINAL phase to complete the reorganization (by setting LONGLOG to DRAIN), or allow the reorganization to continue (by doing nothing). Use either the XBM Utility Monitor function or the MVS console to change the LONGLOG value.

Serialization and concurrency for SHRLEVEL CHANGE

DB2 applications might be using the affected objects during a SHRLEVEL CHANGE reorganization. Consequently, you cannot perform certain functions, and certain restrictions apply to the status of the objects before the reorganization.

Concurrency with other applications

During a reorganization, the SHRLEVEL CHANGE option of REORG PLUS allows DB2 SQL applications to continue to use the affected objects in RW mode. This option is not intended to allow concurrency with data definition language (DDL) or other DB2 or third-party utilities. The following additional restrictions and considerations apply to concurrency with SQL statements and other applications:

- REORG PLUS terminates if you are reorganizing a versioned XML table space and an SQL UPDATE statement runs against the table space.

- BMC recommends that you do not run any of the following utilities or SQL statements during the reorganization:
  
  — any IBM, BMC, or other third-party utility (such as a load utility) that changes the table space or index space that you are reorganizing

  — any of these utilities against any table space or index space that contains a partition that you are reorganizing
— an SQL UPDATE statement for the clustering or partitioning key of a partitioned table space that you are reorganizing

— an SQL DELETE statement without a WHERE clause (mass DELETE)

### Object status for SHRLEVEL CHANGE

For information about the initial statuses that REORG PLUS requires, see “Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE” on page 86. For a discussion of the statuses that REORG PLUS changes during execution, see Table 18 on page 90.

### Operational considerations for online reorganizations

Some REORG PLUS options are not available when you perform a SHRLEVEL CHANGE reorganization. In addition to the considerations described in this section, the following sections contain additional considerations:

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>authorizations</td>
<td>“Additional authorizations for SHRLEVEL CHANGE” on page 67</td>
</tr>
<tr>
<td>SHRLEVEL CHANGE jobs that invoke DSNUTILB</td>
<td>“Reorganization jobs that invoke DSNUTILB” on page 72</td>
</tr>
<tr>
<td>status requirements</td>
<td>“Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE” on page 86 Table 18 on page 90</td>
</tr>
<tr>
<td>staging data sets</td>
<td>“Staging data sets” on page 101</td>
</tr>
<tr>
<td>partial reorganization considerations</td>
<td>“Partial reorganization” on page 121</td>
</tr>
<tr>
<td>single-phase reorganization considerations</td>
<td>“Single-phase reorganization” on page 144</td>
</tr>
<tr>
<td>performance considerations</td>
<td>“Additional performance information for SHRLEVEL CHANGE options” on page 590</td>
</tr>
<tr>
<td></td>
<td>“Performance tuning for specific scenarios (SHRLEVEL CHANGE only)” on page 591</td>
</tr>
</tbody>
</table>
Interacting with applications

You can use the WTOMSG SUFSTART command option (page 254) to write message BMC50008I to the MVS system log. The message indicates that the XBM or SUF snapshot initialization has successfully completed. You can use the text of this message to trigger the submission of jobs that you want to run concurrently with the REORG PLUS job.

Incompatible REORG PLUS options

You can specify SHRLEVEL CHANGE for a two-phase or single-phase reorganization of a table space or index space. However, when you specify SHRLEVEL CHANGE, you cannot specify the following options:

- UPDATE
- UNLOAD PAUSE
- ORDER NO for an index-only reorganization

REORG PLUS handles the following options differently for SHRLEVEL CHANGE:

- forces COPY YES REGISTER ALL, regardless of the values that you specify
- changes ANALYZE HURBA to ANALYZE
- for ON FAILURE, always functions as if you specified TERMINATE UTILITY

Allocation of spill data sets

REORG PLUS uses spill data sets to store log records and RID maps when memory overflows. Each reorganization must have its own spill data sets. If your spill data sets have insufficient space for the log records and RID maps, REORG PLUS terminates the reorganization.

You can specify the options that control spill data set allocation in your installation options module. These installation options are shipped with the following values:

- LOGSPIL=20000, 10000
- SPILUNIT=WORK
- SPILSCLS=NONE
- SPILDSNP=&&UID

For information about these options, see Appendix A, “REORG PLUS installation options.”
You can also override the values through options on the REORG PLUS command (LOGSPILL, SPILLUNIT, SPILLSTORCLAS, and SPILLDSNPAT). For the syntax of these command options, see “SHRLEVEL CHANGE options” on page 326.

The LOGMEM command or installation option tells REORG PLUS how much data space memory (in kilobytes) it can use for storing the log records. REORG PLUS stores only the log records for the DB2 rows that have been updated since XBM took the snapshot of the object. When this memory overflows, REORG PLUS begins using the spill data sets.

The RIDMAPMEM command or RMAPMEM installation option tells REORG PLUS how much data space memory (in kilobytes) it can use for storing the RID maps. REORG PLUS uses the RID maps to translate the RIDs of the original objects to the RIDs of the reorganized objects. When this memory overflows, REORG PLUS begins using the spill data sets.

The LOGSPILL command or LOGSPIL installation option tells REORG PLUS how much disk space to allocate for the log record spill data sets. You can set both primary and secondary allocations. You can use the SPILLUNIT command or SPILUNIT installation option to specify the DASD where you want REORG PLUS to allocate the spill data sets. Alternatively, you can use the SPILLSTORCLAS command or SPILSCLS installation option if you have SMS storage allocation.

The SPILLDSNPAT command or SPILDSNP installation option directs REORG PLUS to use a pattern of variables and text to create prefixes for spill data set names.

The spill data sets are VSAM files. You might need to take this fact into consideration when you specify the SPILLUNIT, SPILLSTORCLAS, or SPILLDSNPAT command options or their equivalent installation options.

**SHRLEVEL CHANGE considerations for using SELECT or DELETE**

You can use the SELECT or DELETE option on the REORG command to filter data rows so that they will be removed from the table space. These filtered rows do not exist in the staging data sets, but they still exist in the original data sets that applications are accessing. If an application issues an SQL UPDATE or DELETE to a row in the original data set that SHRLEVEL CHANGE has filtered, REORG PLUS protects the integrity of the data by terminating the SHRLEVEL CHANGE job without completing the reorganization.

**SHRLEVEL CHANGE statistics considerations**

Statistics that follow message BMC50501I in the REORG PLUS SYSPRINT do not reflect activity that occurs during the LOGAPPLY phase.
For batch applications that are running the BMC AR/CTL product, an online reorganization can use the AR/CTL suspend-and-resume interface. The SHRLEVEL CHANGE reorganization must obtain temporary exclusive access to the object that you are reorganizing. AR/CTL suspends batch DB2 processing to allow exclusive access, and resumes batch processing when exclusive access is no longer needed.

For more information about AR/CTL, see the APPLICATION RESTART CONTROL User Guide.

Requirements

The suspend-and-resume interface between a SHRLEVEL CHANGE reorganization and AR/CTL has the following requirements:

- You must have a license for AR/CTL for DB2.
- The CPU authorization password for REORG PLUS must be available to the BMC Consolidated Subsystem (BCSS).
- In the subsystem that you use to run the batch DB2 application, the BCSS must be active. The BCSS component that supports the DB2 and VSAM features of the Application Enhancement Series products must be initialized.
- The batch program must use AR/CTL checkpoint/restart services.

Interface processing

The suspend-and-resume interface with the AR/CTL product works as follows:

1. Batch job steps run under the control of AR/CTL.
2. When REORG PLUS is initialized, it tells AR/CTL which table spaces and indexes are affected by the reorganization.
3. When an application program attempts an SQL access to a table that would normally receive an SQLCODE –911 or –904 because of REORG PLUS, AR/CTL performs suspend processing for the application.
4. When REORG PLUS completes the function that would cause a SQLCODE –911 or –904, it signals AR/CTL to resume processing on the application unit of work.
The following REORG PLUS functions can cause AR/CTL to suspend an application:

- all processing during the LOGFINAL phase
- rename processing during the UTILTERM phase

**Implementing the interface**

To implement the suspend-and-resume interface with AR/CTL, perform the following steps:

1. Ensure that the AR/CTL product is installed.

2. Ensure that the REORG PLUS ARC installation option is set to YES.

3. Make the CPU authorization password for REORG PLUS available to the BCSS, using one of the following methods:
   - Include the library that contains the password module in the AESPAUTH DD statement concatenation in the BCSS startup procedure
   - Copy the password module to a library that is already in the AESPAUTH DD concatenation.

4. Set up the reorganization job step.

5. In each batch program that you want AR/CTL to control, implement AR/CTL checkpoint/restart services as described in the AR/CTL documentation.

6. Run the reorganization job.

**Recoverability of the reorganized table space**

To ensure the recoverability of reorganized objects, REORG PLUS makes and registers full image copies of the table space objects. If the value for ICTYPE is INCREMENTAL on the command or installation option, REORG PLUS also creates and registers incremental image copies of the table space objects. Updating the full copies provides the best performance. BMC recommends that you specify ICTYPE AUTO so that REORG PLUS can decide which type of copy is best.
If you specify ICTYPE AUTO, REORG PLUS determines whether to create incremental copies or update the full image copies based on certain criteria. REORG PLUS treats the option as if you specified ICTYPE UPDATE and updates the full copies when either of the following conditions exists:

- The data sets are on DASD and one full copy data set exists for each partition that you are reorganizing.
- You specify INLINE YES on the command or INLINECP=YES in the installation options module. In this case, REORG PLUS appends the updated pages to the full copy data sets, and you can put the copy data sets on tape.

**NOTE**
If INLINE NO is in effect, do not put the copy data sets on tape because REORG PLUS updates the full copy data sets in place.

If you are reorganizing multiple partitions and using one copy data set, REORG PLUS treats ICTYPE AUTO as if you specified ICTYPE INCREMENTAL. You must have previously defined or dynamically allocated the incremental copy data sets. For more information, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 653.

### Copy data sets for SHRLEVEL CHANGE

During the RELOAD and REORG phases of a SHRLEVEL CHANGE reorganization, REORG PLUS makes full copies of the table space objects, just as it does for SHRLEVEL NONE and SHRLEVEL REFERENCE. However, these copies are not registered in SYSIBM.SYSCOPY until the UTILTERM phase.

COPY YES REGISTER ALL is required for a SHRLEVEL CHANGE table space reorganization and will be set regardless of your specifications. REORG PLUS does not make image copies for an index-only reorganization. For more information, see “Copy data sets” on page 352.

**NOTE**
If an incremental copy data set was created, both it and the full copy data set are required for any type of recovery of this table space.
Incremental copy data sets for SHRLEVEL CHANGE

REORG PLUS produces incremental copy data sets when either of the following conditions exists:

- you set the ICTYPE command or installation option to INCREMENTAL
- you set ICTYPE to AUTO and REORG PLUS determines that incremental copies should be taken

During the LOGFINAL phase for SHRLEVEL CHANGE, REORG PLUS makes an incremental image copy. This copy records the changes that resulted from applying the log records that were taken after the full image copy was made. You must have one incremental copy data set for each full copy data set that is defined.

The incremental copy ddnames identify the output data sets that will contain an incremental image copy either of the table space or of each partition in the table space that you are reorganizing. The existence of the incremental copy ddnames in your JCL determines the number of copies made. For more information, see Table 95 on page 655.

--- WARNING ---

REORG PLUS does not support stacking incremental copy data sets on tape. If you choose to stack incremental copy data sets with the corresponding full data sets, unpredictable results can occur with subsequent recoveries.

Overriding the default ddnames

To override the default incremental copy ddnames or ddname prefixes in your installation options module, use the ICDDN (page 275) and RECOVERYICDDN (page 277) command options. For performance and tuning information when using these options, see “Incremental copy data sets” on page 578 and “Improving performance” on page 656.

Registration

The ddname itself controls the registration information that is placed in the ICBACKUP column of SYSIBM.SYSCOPY. REORG PLUS forces REGISTER ALL for a SHRLEVEL CHANGE reorganization.

Table 94 on page 654 describes the incremental copy data sets that, when registered, correspond to the DB2 local and remote copies. It also states when they are required.
Allocating incremental copy data sets

Use one of the methods described in “Methods for allocating copy and work data sets” on page 351 to allocate your copy data sets and determine the appropriate size for those data sets. REORG PLUS determines the optimal block size of incremental copy data sets based on the device type that contains the data set. Incremental copy data sets can be on different device types as long as the data sets are cataloged.

You can override the default block size by allocating the incremental copy data sets in the JCL with a block size that is greater than 0 and is an even multiple of 4 KB. If you are making more than one copy, the block size for all incremental copy data sets for that object is the block size that REORG PLUS determined was optimal for the primary local copy.

The size that REORG PLUS needs for incremental copy data sets is usually less than or equal to the size of the full copy data sets, and depends on the number of pages updated during the log apply process.

Considerations
The following considerations apply to incremental copy data sets:

- When dynamically allocating copy data sets for partition-by-growth table spaces, REORG PLUS uses the DSSIZE value to estimate the size.

- Reorganization jobs that invoke DSNUTILB do not produce incremental copies.

Dynamic allocation
Based on the information in Table 95 on page 655, specify dynamic allocation options that result in the same number and type of incremental copy data sets as if you specified DD statements in your JCL. For more information, see “Dynamic data set allocation” on page 90.
Allocating copy data sets in your JCL

If you do not use dynamic allocation, you must allocate the incremental copy data sets in your JCL by specifying DD statements as described in Table 95. The table lists the incremental copy DD statements that REORG PLUS requires based on the table space and type of reorganization.

<table>
<thead>
<tr>
<th>Table space being copied</th>
<th>Type of reorganization</th>
<th>Incremental copy DD statements required</th>
</tr>
</thead>
<tbody>
<tr>
<td>nonpartitioned</td>
<td>any</td>
<td>Specify one DD statement for each copy type that you want REORG PLUS to make. Do not use \textit{nn} in the ddname. You should not specify separate data sets for nonpartitioned, multi-data-set table spaces.</td>
</tr>
</tbody>
</table>
| partitioned, including partition-by-growth | full (entire table space) | Specify one DD statement for each copy type that you want REORG PLUS to make. Do not use \textit{nn} in the ddname. 

*or*

For each partition, specify one DD statement for each copy type that you want REORG PLUS to make. Use \textit{nn} for all ddnames, where \textit{nn} matches the partition number.  

If necessary for partition-by-growth table spaces, specify additional DD statements as discussed in “Partition-by-growth table spaces” on page 358.

| partial (selected partitions using PART option) | Specify one DD statement for each copy type that you want REORG PLUS to make. Do not use \textit{nn} in the ddname. This option is valid only when you specify a single subset of physically contiguous partitions.  

*or*

For each partition, specify one DD statement for each copy type that you want REORG PLUS to make. Use \textit{nn} for all ddnames, where \textit{nn} matches the partition number. (The \textit{nn} is not required if you are performing a partial reorganization with only one partition.)  

If necessary for partition-by-growth table spaces, specify additional DD statements as discussed in “Partition-by-growth table spaces” on page 358.

\textit{a} For more information, see “Partition-level copies” on page 357.

\textit{b} The value of the COPYSUBSET installation option must be YES.
Improving performance

When reorganizing a partitioned table space, REORG PLUS multitasks the incremental image copy process whenever possible. This multitasking decreases the elapsed time needed to run the reorganization. To facilitate multitasking, provide a unique incremental image copy data set for each partition that you are reorganizing, and ensure that each image copy data set resides on DASD.

When running a full reorganization of a partitioned table space, you can specify how the dynamically allocated incremental copy data sets are assigned. However, REORG PLUS does not allow dynamically allocated copy data sets to be stacked on tape. For more information, see “COPYLVL” on page 269.

You can use the SIZEPCT option (page 311) to reduce the size of the incremental copy data sets. Because REORG PLUS uses the high-used relative byte address (HURBA) of the table space to determine the primary quantity, the incremental copy data sets can be larger than needed. If you use your estimate of the percentage of pages in the table space that will be updated during the reorganization as the primary SIZEPCT value, less space will be allocated for the data sets.

Copy registration failure during a SHRLEVEL CHANGE reorganization

If an error occurs that prevents the registration of the copy data sets for a SHRLEVEL CHANGE reorganization, the setting of the CPYRFAIL installation option determines the action that REORG PLUS takes:

- If CPYRFAIL=TERM, REORG PLUS backs out any renames that it completed and terminates the reorganization. In this case, the original data sets, with no modifications, are available for use by DB2. The condition of your objects is the same as if the reorganization had never run.

- If CPYRFAIL=COPYPEND is specified, REORG PLUS sets the copy pending (COPY) status for the objects and starts them with their original status. You will need to make an image copy of the objects in order to allow applications to have RW access again.

If either of the following conditions exists, REORG PLUS changes CPYRFAIL=COPYPEND to CPYRFAIL=TERM at the beginning of the job. If the copy registration fails, REORG PLUS backs out the renames and terminates the reorganization.

- You are reorganizing a catalog object.
- You are using a DDLIN data set that alters the object that you are reorganizing. In this case, REORG PLUS also backs out the ALTER statement.
Restart considerations for a SHRLEVEL CHANGE reorganization

The restart considerations for a SHRLEVEL CHANGE reorganization depend on whether you want to restart the job before the UTILTERM phase or after the job enters the UTILTERM phase. For information about terminating or canceling a SHRLEVEL CHANGE reorganization, see “Terminating or canceling a job” on page 396.

Restarting before the UTILTERM phase

For a SHRLEVEL CHANGE reorganization, restarting is not allowed (nor is it necessary) before the UTILTERM phase, with the exception of a restart following an ANALYZE PAUSE. If you specify ANALYZE PAUSE, REORG PLUS analyzes the objects to be reorganized, provides an informational report, and saves the information for restart.

Restarting is not allowed during the UNLOAD, RELOAD, REORG, LOGAPPLY, or LOGFINAL phase. If a reorganization stops during one of these phases, a large quantity of log records could be written in the interim to the DB2 log for the objects that you are reorganizing, making it impractical to continue the reorganization in a timely manner. Generally, you should rerun the reorganization from the start. You do not need to restart, because the original data sets containing the DB2 table space and index space objects are not modified by REORG PLUS until the UTILTERM phase.

Not completing the reorganization before the UTILTERM phase

If REORG PLUS turns on Data Capture Changes and the job abnormally terminates before the UTILTERM phase, REORG PLUS attempts to turn off Data Capture Changes. To determine whether REORG PLUS was able to turn off Data Capture Changes, examine the output from the job. REORG PLUS was unsuccessful if the output includes the following messages:

BMC50811I taskNumber: ALTER TABLE tableName DATA CAPTURE ON/CHANGES

BMC50205S UNEXPECTED SQL ERROR, SQLCODE=n, STMID='stmid'

Determine why the failure occurred and manually run the SQL statement to reset Data Capture Changes.
Restarting in the UTILTERM phase

If REORG PLUS is canceled or ends during the UTILTERM phase of a SHRLEVEL CHANGE reorganization, you should restart the utility. This restart ensures that REORG PLUS successfully completes the following tasks:

- completes the data set rename or FASTSWITCH process
- removes pending statuses
- alters limit keys
- registers the reorganization and image copies
- updates the primary and secondary quantities in the DB2 catalog

If REORG PLUS cannot complete all of these operations, you might need to perform recovery actions. However, REORG PLUS is designed to be restartable in this case to complete these operations successfully.

The following considerations apply when restarting in the UTILTERM phase:

- REORG PLUS automatically backs out the reorganization when any of the following conditions exists:
  - A failure occurs during processing of limit-key ALTER statements and the value of the ALTRFAIL installation option is TERM.
  - A failure occurs while REORG PLUS is setting restrictive statuses, and a restarted reorganization might create invalid indexes.
  - You are reorganizing a catalog object and a failure occurs during rename processing.
- REORG PLUS does not update statistics when you restart in the UTILTERM phase.

On restart, REORG PLUS does not update statistics if, in the original job, any participating table space partitions were completely loaded or any participating index partitions were completely built.

Not completing the reorganization in the UTILTERM phase

For SHRLEVEL CHANGE, if you terminate or cancel the job during the UTILTERM phase, or if the UTILTERM phase cannot complete processing, BMC recommends that you restart your job.
Performance considerations

REORG PLUS uses virtual memory in the address space in which it is running and additional data spaces to hold the log records and the RID translation map. The amount of memory that you make available to REORG PLUS affects the performance of your job. The sections that follow provide information about the memory requirements for the following items:

- the RID translation map
- log records
- spill data sets
- log data

As with other REORG PLUS jobs, BMC recommends that you specify REGION=0M in the JOB or EXEC statement of the execution JCL whenever you run SHRLEVEL CHANGE. If your data center constrains region sizes or the use of REGION=0M, specify a region size that will allow the most virtual memory, both above and below the 16-megabyte line. Using a region size that is less than optimal risks the following potential issues:

- running less efficiently, which could result in additional CPU and elapsed time
- encountering memory failures or jobs that fail when new versions implement changes that require additional memory

**NOTE**

If you specify a value for REGION other than 0M, ensure that you have an appropriate value set for the MEMLIMIT parameter, either as your site’s default SMF option or on your JOB statement or EXEC statement.

BMC makes the following recommendations for the MEMLIMIT option:

- Specify NOLIMIT to allow unlimited above-the-bar memory.
- If you are unable to specify NOLIMIT, specify at least 4 GB; if you are reorganizing LOB or XML data, specify at least 32 GB.
Sizing memory for the RID translation map

REORG PLUS uses the RID translation map to translate the RIDs of the original objects to the RIDs of the reorganized objects. The RID map memory is the memory that holds the completed RID translation map. REORG PLUS allocates this memory in one or more data spaces. If REORG PLUS cannot build the entire RID map in the memory that you specified on the RIDMAPMEM command option (or RMAPMEM installation option), REORG PLUS spills the remaining portion of the RID map to DASD. During processing, as it needs space, REORG PLUS determines the amount of DASD that it needs for the spill data sets and allocates them for you.

Allowing REORG PLUS to calculate memory

BMC strongly recommends that you specify 0 as the value for RIDMAPMEM (and RMAPMEM) to allow REORG PLUS to compute the amount of memory for the RID map automatically. To determine the value, REORG PLUS multiplies the value of the installation option RIDMMAXD by the value of the installation option RIDMDSSZ. The result is the maximum amount of memory that REORG PLUS can allocate. However, REORG PLUS uses only as much memory as needed to hold the RID map.

RIDMMAXD specifies the maximum number of data spaces, and RIDMDSSZ specifies the maximum size for each data space. For more information about these options, see Appendix A, “REORG PLUS installation options.”

Specifying the amount of memory

Although BMC recommends that you specify 0 for RIDMAPMEM (and RMAPMEM), you can specify a nonzero value. REORG PLUS first ensures that the value you supplied is sufficient to satisfy the minimum storage requirements. If the value is not sufficient, REORG PLUS issues message BMC50887I and changes the value for RIDMAPMEM to the minimum required storage amount.

During execution, REORG PLUS allocates memory only as needed, up to the current value of RIDMAPMEM. If RIDMAPMEM is insufficient, REORG PLUS issues message BMC50885I and spills to the spill data set.

To calculate the RIDMAPMEM value, use the formula described in “Using formulas for the calculations” on page 661. Regardless of the value that you specify for RIDMAPMEM, REORG PLUS never allocates more memory than the value of the RIDMMAXD installation option multiplied by the value of the RIDMDSSZ installation option. To achieve optimal performance, make sure that the memory that you specify for RIDMAPMEM can hold the entire RID map.
Using formulas for the calculations

This section contains formulas that can help you calculate the maximum amount of memory (in kilobytes) that REORG PLUS might need for each component of the RID translation map. If you specify more memory than REORG PLUS needs, REORG PLUS does not use the extra memory. Under certain circumstances, REORG PLUS might be able to compress the RID map data and use significantly less memory.

In the formulas that follow, the variable \( r \) is the number of rows in the table space that are participating in the reorganization, including the estimated number of rows that applications might insert during the LOGAPPLY phase.

Formula A is sufficient for most table space reorganizations:

\[
A = (0.0078 \times r) + 2048
\]

However, if you have fewer than 20 rows per page, consider using Formula B for a more exact computation:

\[
B = (0.0132 \times \text{number of pages participating in the reorganization}) + (0.0071 \times r) + 2048
\]

REORG PLUS displays the total amount of memory (data space memory and DASD) that it used in the STORAGE field in message BMC50880I (SUMMARY version of the message).

```
BMC50880I RIDMAP: SUMMARY: ROWS=r, PAIRS=p, STORAGE=sK, MEM WAITS=mw, WAIT TIME=wt
```

Regardless of the amount that you specify for RIDMAPMEM or RMAPMEM, REORG PLUS never allocates more memory than the value of RIDMMAXD multiplied by the value in RIDMDSSZ.
Sizing memory for log records

The total memory that REORG PLUS uses for log records consists of

- the data spaces that you allocate with the LOGMEM command or installation option
- the DASD that you allocate with the LOGSPILL command option or LOGSPILL installation option

REORG PLUS stores log records (data and index components) in one or more data spaces that it allocates. These data spaces are separate from those used for the RID maps. When the memory becomes full, REORG PLUS spills the log records to the spill data sets.

For guidelines for specifying the spill data sets, see “Sizing the spill data sets” on page 664. For guidelines for using the LOGMEM and LOGSPILL options together, see “Calculating storage requirements for log data” on page 664.

Allowing REORG PLUS to calculate memory

BMC strongly recommends that you specify a value of 0 for the LOGMEM installation or command option. Doing so enables REORG PLUS to calculate memory automatically, using the formulas discussed in “Using formulas for the calculations” on page 663.

Specifying the amount of memory

Although BMC recommends specifying 0 for LOGMEM, you can specify a nonzero value for the LOGMEM option to control the amount of memory that is available for the log records. When that memory becomes full, REORG PLUS spills the log records to the spill data sets and issues the following message:

BMC50869I LOG RECORD: SPILLING TO DATASET 'dataSetName'

At the end of the LOGAPPLY phase, the following message displays the amount of time that REORG PLUS waited to store log records:

BMC50860I TOTAL WAIT TIME FOR LOG RECORD STORE MEMORY = seconds
DATA = seconds INDEX = seconds

To determine an appropriate amount of memory to specify, use the formulas in the next section to assist you. If the total wait time shown in message BMC50860I indicates that REORG PLUS spent a significant amount of time waiting, increase the value of LOGMEM to improve performance.
Using formulas for the calculations

This section contains formulas that can help you calculate the LOGMEM value. These are also the formulas that REORG PLUS uses when you specify 0 for LOGMEM. Use Table 96 to help determine each number in the formula.

Table 96  Calculating memory for LOGMEM

<table>
<thead>
<tr>
<th>Reorganization type</th>
<th>Value to use for ( p )</th>
<th>Value to use for ( c )</th>
<th>Value to use for ( n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>full table space</td>
<td>number of partitions that you are reorganizing</td>
<td>number of partitioned indexes times the number of partitions that you are reorganizing</td>
<td>number of nonpartitioned indexes that are defined on the table space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This number is the same as the number of table space partitions that you are reorganizing.</td>
<td></td>
</tr>
<tr>
<td>partial table space without nonpartitioned indexes</td>
<td>number of partitions that you are reorganizing</td>
<td>number of partitioned indexes times the number of partitions that you are reorganizing</td>
<td>0</td>
</tr>
<tr>
<td>partial table space with nonpartitioned indexes</td>
<td>total number of partitions in the table space</td>
<td>number of partitioned indexes times the number of partitions that you are reorganizing</td>
<td>number of nonpartitioned indexes that are defined on the table space</td>
</tr>
<tr>
<td>full or partial index (partitioned index)</td>
<td>number of index partitions that you are reorganizing</td>
<td>number of index partitions that you are reorganizing</td>
<td>0</td>
</tr>
<tr>
<td>index (nonpartitioned index)</td>
<td>total number of partitions in the table space</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

To determine the \textit{minimum} amount of data space to specify for LOGMEM, use formula A:

\[
A = ((p + c + n) \times 128) + 2048
\]

If formula A results in a value that is less than 20480 KB, BMC strongly recommends that you use 20480 KB for LOGMEM to achieve optimal performance.

To determine the \textit{recommended} amount of data space to specify for LOGMEM, use formula B:

\[
B = 20480 + (p \times 1024) + ((c + n) \times 512)
\]
Sizing the spill data sets

The more updates the application makes, the more space REORG PLUS needs for the spill data sets. When the amount of space that REORG PLUS needs to store the RID maps is greater than the space that you specified on the RIDMAPMEM command or RMAPMEM installation option, REORG PLUS uses the spill data sets. Therefore, it is important to have enough DASD to support spilling. When the amount of space that REORG PLUS needs to store the log records is greater than the space that you specified on the LOGMEM command or installation option, REORG PLUS uses a different set of spill data sets.

For log records, REORG PLUS allocates the spill data sets by using the extent size that you specify as primary on the LOGSPILL command or LOGSPIL installation option. After the spill data set becomes full, REORG PLUS allocates and uses additional secondary extents until it reaches the VSAM extent limit, or until the data set reaches a size of 4 gigabytes (GB). Then, REORG PLUS allocates another data set, and so on, for a maximum of 256 data sets. The following message displays the amount of memory that REORG PLUS actually used for the log record spill data sets:

BMC50867I LOG RECORD STORE SPILL REQUESTS = n, HIGH SPILL PAGE = n, SPILL DATASETS CREATED = n

For optimal performance, BMC recommends allocating relatively large primary and secondary allocations to reduce the number of extents that are created. For more information, see “Calculating storage requirements for log data.”

For spill data sets for RID maps, REORG PLUS calculates the primary and secondary extent sizes for you. To find the amount of storage that REORG PLUS used for the spill data sets, subtract the value displayed for USED MEM in message BMC50882I from the value displayed for STORAGE in message BMC50880I (the SUMMARY version of the message). For information about the options for spill data sets, see “Allocation of spill data sets” on page 648.

Calculating storage requirements for log data

REORG PLUS uses both data space memory (from the LOGMEM option) and DASD (from the LOGSPILL option) to store the log record data. The combined memory of these two options must be enough to support any DB2 updates that are being performed. The following factors affect the total amount of memory that REORG PLUS requires for log data:

- length of the rows
- number of indexes
- number of updates (including inserts and deletes) that are being made to the tables
Formula A provides a rough estimate of the total amount of memory REORG PLUS requires (memory and DASD) for the log data:

\[ A = \text{number of updates} \times \text{row length} \times 3 \]

Use formula B if the table space has more than two nonpartitioned indexes:

\[ B = \text{number of updates} \times \text{row length} \times \left( \frac{\text{sum of key lengths for all indexes}}{\text{row length}} \right) + 3 \]

For a multi-table table space, use the length of the longest row as the row length in the calculation.

**Copying nonpartitioned indexes during a partial table space reorganization**

You can improve the performance of a partial SHRLEVEL CHANGE reorganization of a table space containing nonpartitioned indexes by using the Instant Snapshot technology provided by SUF or XBM. To use this technology, you must specify AUTO or YES on the SIXSNAP command or installation option and have the supported hardware available. Instant Snapshot copies nonpartitioned indexes to the staging data sets almost instantaneously on intelligent storage devices, which is much faster than making copies by using software-based snapshots.

However, using Instant Snapshot can have performance consequences. If you use software-based snapshots, REORG PLUS holds the drain only while registering the data sets. If you use Instant Snapshot, REORG PLUS holds the drain during both the registration and the copy. If the table space has a large number of nonpartitioned indexes, an application could time out before all of the copies are made.

**Making inline copies during a SHRLEVEL CHANGE reorganization**

If you specify INLINE YES with SHRLEVEL CHANGE, REORG PLUS uses more memory than it would with INLINE NO because it keeps all of the image copy data sets open until the LOGFINAL phase completes processing.
REORG PLUS installation options

This appendix presents the following topics:

- **Overview** ................................................................. 667
- **Basic REORG PLUS installation options** ......................... 668
- **Dynamic allocation installation options** ......................... 740
- **DYNALOC installation option** ..................................... 761

**Overview**

The REORG PLUS product is installed by using the BMC Software Installation System. During this installation, the customization process generates a customized installation data set. This data set contains customized jobs that install REORG PLUS into your specific DB2® environment. Two of these jobs establish the default processing option values that REORG PLUS uses:

- `$C30DOPT` establishes the defaults for REORG PLUS processing options.
- `$C32SOPT` contains options for the BMC Software BMCSORT technology.

These jobs assemble options macros. The macros contain the REORG PLUS processing options and the values for those options that are shipped with REORG PLUS and BMCSORT. When the Installation System-generated customization job is submitted, it links the ARU$OPTS installation options module in the APF-authorized library that is designated by your site.

You can customize the installation of REORG PLUS by changing the values for the REORG PLUS installation options. However, if you change any of the values in `$C30DOPT` or `$C32SOPT` after REORG PLUS has been installed, you must rerun the jobs for these changes to take effect.
You can also create additional options modules that allow you to use different values of these options for different executions of REORG PLUS. For example, you might use the default installation options module for most jobs but create another options module with customized values for special situations. For information about specifying an options module at runtime, see Chapter 4, “Building and executing REORG PLUS jobs.” For more information about customizing your installation of REORG PLUS, see the BMC Products and Solutions for DB2 Configuration Guide.

To find a description of each options macro, use Table 97.

### Table 97 REORG PLUS installation macros

<table>
<thead>
<tr>
<th>Job</th>
<th>Macro name</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C30DOPT</td>
<td>$ARUOPTS</td>
<td>basic options</td>
<td>page 668</td>
</tr>
<tr>
<td></td>
<td>$ARUDYNA</td>
<td>options for dynamic data set allocation</td>
<td>page 740</td>
</tr>
<tr>
<td>C32SOPT</td>
<td>$AUPSMAC</td>
<td>options for BMCSORT</td>
<td>page 761</td>
</tr>
</tbody>
</table>

### Basic REORG PLUS installation options

Table 98 shows the options contained in the $ARUOPTS macro in $C30DOPT. For each option, the table provides the value that ships with this version of REORG PLUS (or No value if the option is shipped without a value), a brief description, and a reference to more details. If an option ships with no value, the table shows a recommended value or example value.

### Table 98 Basic REORG PLUS installation options (part 1 of 6)

<table>
<thead>
<tr>
<th>Option</th>
<th>Shipped value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTRFAIL</td>
<td>RCVRPEND</td>
<td>which action to take when a failure occurs during limit-key ALTER processing</td>
<td>page 674</td>
</tr>
<tr>
<td>ANALMAX</td>
<td></td>
<td>maximum number analyze tasks to start</td>
<td>page 674</td>
</tr>
<tr>
<td>ARC</td>
<td>NO</td>
<td>whether to use the AR/CTL interface</td>
<td>page 674</td>
</tr>
<tr>
<td>ARCHDDN</td>
<td>SYSARC</td>
<td>default ddname for the archive data set</td>
<td>page 675</td>
</tr>
<tr>
<td>ARCHFORMAT</td>
<td>BMC</td>
<td>default format for the archive data set</td>
<td>page 675</td>
</tr>
<tr>
<td>AUXREORG</td>
<td>Not applicable</td>
<td>obsolete</td>
<td>page 676</td>
</tr>
<tr>
<td>AVAILPAGEPCT</td>
<td>100</td>
<td>percentage of available 4-KB pages reported by the system that REORG PLUS can allocate to BMCSORT for sort processing</td>
<td>page 676</td>
</tr>
<tr>
<td>BILDMAX</td>
<td>300%</td>
<td>maximum number of tasks to start for the index build process</td>
<td>page 677</td>
</tr>
<tr>
<td>BMCHIST</td>
<td>YES</td>
<td>whether to update the BMCHIST table</td>
<td>page 678</td>
</tr>
<tr>
<td>CBUFFS</td>
<td>30</td>
<td>multiplier that controls the amount of buffer pool storage for each copy data set</td>
<td>page 678</td>
</tr>
<tr>
<td>CLONE</td>
<td>Not applicable</td>
<td>obsolete</td>
<td>page 678</td>
</tr>
</tbody>
</table>
Table 98  Basic REORG PLUS installation options (part 2 of 6)

<table>
<thead>
<tr>
<th>Option</th>
<th>Shipped value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONDEXEC</td>
<td>NO</td>
<td>whether REORG PLUS should consider performing a conditional reorganization</td>
<td>page 679</td>
</tr>
<tr>
<td>COPYDDN</td>
<td>(BCPY,BCPZ)</td>
<td>default ddname or prefix for the local primary and secondary copy data sets</td>
<td>page 680</td>
</tr>
<tr>
<td>COPYLVL</td>
<td>PART</td>
<td>whether to assign full or partition-level copy data sets for partitioned table spaces</td>
<td>page 680</td>
</tr>
<tr>
<td>COPYMAX</td>
<td>*</td>
<td>maximum number of tasks to start during the image copy process</td>
<td>page 681</td>
</tr>
<tr>
<td>COPYSUBSET</td>
<td>NO</td>
<td>whether to allow a single copy of a subset of partitions</td>
<td>page 682</td>
</tr>
<tr>
<td>CPYRFAIL</td>
<td>TERM</td>
<td>which action to take if the image copies cannot be registered in SYSIBM.SYSCOPY</td>
<td>page 682</td>
</tr>
<tr>
<td>DATACAP</td>
<td>NO</td>
<td>whether to turn on the Data Capture Changes flag</td>
<td>page 683</td>
</tr>
<tr>
<td>DDLDDN</td>
<td>DDLIN</td>
<td>default ddname or prefix of the DDLIN data set</td>
<td>page 683</td>
</tr>
<tr>
<td>DEADLINE</td>
<td>NONE</td>
<td>time by which the LOGFINAL phase should finish applying log records</td>
<td>page 683</td>
</tr>
<tr>
<td>DELFILES</td>
<td>YES</td>
<td>whether to delete data sets on completion</td>
<td>page 684</td>
</tr>
<tr>
<td>DESCcce</td>
<td>(3,7)</td>
<td>descriptor codes to control how WTO messages are displayed on a console</td>
<td>page 685</td>
</tr>
<tr>
<td>DRAINTYP</td>
<td>ALL</td>
<td>type of drain that REORG PLUS issues when it enters the LOGFINAL phase, after reaching the MAXRO threshold</td>
<td>page 686</td>
</tr>
<tr>
<td>DRNDELAY</td>
<td>1</td>
<td>minimum number of seconds that REORG PLUS waits before it retries to obtain the drain after a drain times out</td>
<td>page 686</td>
</tr>
<tr>
<td>DRNRETRY</td>
<td>255</td>
<td>maximum number of times that REORG PLUS attempts to obtain a drain before terminating</td>
<td>page 686</td>
</tr>
<tr>
<td>DRNWAIT</td>
<td>NONE</td>
<td>drain timeout value to use</td>
<td>page 687</td>
</tr>
<tr>
<td>DSNUEXit</td>
<td>(NONE,ASM)</td>
<td>name of a user exit for creating dynamically allocated data set names and the language in which the exit is written</td>
<td>page 688</td>
</tr>
<tr>
<td>DSNUTILB</td>
<td>YES</td>
<td>whether to use DSNUTILB to run the job when a feature requires DSNUTILB</td>
<td>page 689</td>
</tr>
<tr>
<td>DSPLOCKS</td>
<td>DRNFAIL</td>
<td>whether to display claims and locks if a drain attempt times out</td>
<td>page 689</td>
</tr>
<tr>
<td>DSRSEXit</td>
<td>(NONE,REXX)</td>
<td>name of a user exit for managing the redefinition of DB2 VSAM data sets and the programming language in which the exit is written</td>
<td>page 690</td>
</tr>
<tr>
<td>EXCLDUMP</td>
<td>(X37,X22,X06)</td>
<td>system codes to ignore during the dump process</td>
<td>page 690</td>
</tr>
</tbody>
</table>
Basic REORG PLUS installation options

Table 98  Basic REORG PLUS installation options (part 3 of 6)

<table>
<thead>
<tr>
<th>Option</th>
<th>Shipped value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>FASTSWITCH</td>
<td>NO</td>
<td>whether to bypass the VSAM rename process and directly update the DB2 catalog to use the staging data set names</td>
<td>page 691</td>
</tr>
<tr>
<td>FILECHK</td>
<td>FAIL</td>
<td>which action to take when encountering a temporary data set</td>
<td>page 692</td>
</tr>
<tr>
<td>FORCE</td>
<td>NONE</td>
<td>whether to cancel DB2 threads that might prevent a drain process from completing</td>
<td>page 693</td>
</tr>
<tr>
<td>FORCE_AT</td>
<td>(START,3)</td>
<td>when to cancel DB2 threads that might prevent the drain process from completing</td>
<td>page 693</td>
</tr>
<tr>
<td>FORCE_RPT</td>
<td>NO</td>
<td>whether to display the canceled threads</td>
<td>page 694</td>
</tr>
<tr>
<td>FSFALLBACK</td>
<td>NO</td>
<td>whether to perform rename processing when FASTSWITCH processing fails</td>
<td>page 695</td>
</tr>
<tr>
<td>FSTHRESHOLD</td>
<td>0</td>
<td>the minimum number of staging data sets that will trigger FASTSWITCH processing</td>
<td>page 695</td>
</tr>
<tr>
<td>HASHAX</td>
<td>YES</td>
<td>whether to enable reorganizing table spaces that contain tables defined as ORGANIZE BY HASH (using DSNUTILB processing)</td>
<td>page 696</td>
</tr>
<tr>
<td>ICDDN</td>
<td>(BICY,BICZ)</td>
<td>ddnames or ddname prefixes for the local copy data sets that receive an incremental image copy</td>
<td>page 696</td>
</tr>
<tr>
<td>ICTYPE</td>
<td>AUTO</td>
<td>whether to update the full image copy data sets or create incremental image copy data sets</td>
<td>page 697</td>
</tr>
<tr>
<td>IDCACHE</td>
<td>10000</td>
<td>cache size of document ID values</td>
<td>page 699</td>
</tr>
<tr>
<td>INDREFLM</td>
<td>10</td>
<td>limit over which REORG PLUS is to reorganize the specified table space</td>
<td>page 699</td>
</tr>
<tr>
<td>INLINECP</td>
<td>YES</td>
<td>whether to create an inline image copy</td>
<td>page 700</td>
</tr>
<tr>
<td>INLOB</td>
<td>Not applicable</td>
<td>obsolete</td>
<td>page 701</td>
</tr>
<tr>
<td>IXINCLCOL</td>
<td>Not applicable</td>
<td>obsolete</td>
<td>page 701</td>
</tr>
<tr>
<td>IXONEX</td>
<td>NO</td>
<td>whether to enable support for indexes that contain a key derived from certain expressions (using DSNUTILB processing)</td>
<td>page 702</td>
</tr>
<tr>
<td>IXRANDOM</td>
<td>NO</td>
<td>whether to enable support for indexes that contain keys with random ordering (using DSNUTILB processing)</td>
<td>page 702</td>
</tr>
<tr>
<td>KEEPDICTIONARY</td>
<td>NO</td>
<td>whether to keep an existing compression dictionary or build a new one</td>
<td>page 703</td>
</tr>
<tr>
<td>LEAFDSLM</td>
<td>200</td>
<td>limit over which REORG PLUS is to reorganize the index named in the reorganization</td>
<td>page 704</td>
</tr>
<tr>
<td>LOB</td>
<td>YES</td>
<td>whether to enable reorganizing LOB data when DSNUTILB is required</td>
<td>page 704</td>
</tr>
<tr>
<td>LOCKROW</td>
<td>YES</td>
<td>serialization method for BMCSYNC and BMCUTIL</td>
<td>page 705</td>
</tr>
<tr>
<td>LOGFINAL</td>
<td>NONE</td>
<td>time by which LOGFINAL is to start</td>
<td>page 706</td>
</tr>
<tr>
<td>LOGMEM</td>
<td>0</td>
<td>amount of memory for log records</td>
<td>page 706</td>
</tr>
</tbody>
</table>
### Table 98  Basic REORG PLUS installation options (part 4 of 6)

<table>
<thead>
<tr>
<th>Option</th>
<th>Shipped value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGSPIL (20000,10000)</td>
<td>space allocation for spill data sets that hold the log records</td>
<td>page 707</td>
<td></td>
</tr>
<tr>
<td>LOGTHRHSH 0</td>
<td>number of remaining log records</td>
<td>page 707</td>
<td></td>
</tr>
<tr>
<td>LONGLOG CONTINUE</td>
<td>which action to take when a long log condition occurs</td>
<td>page 708</td>
<td></td>
</tr>
<tr>
<td>LONGNAMETRUNC MIDDLE (or M)</td>
<td>long name truncation method for messages</td>
<td>page 708</td>
<td></td>
</tr>
<tr>
<td>MAPTEXIT (NONE,REXX)</td>
<td>name of a user-written mapping table exit and the language in which the exit is written</td>
<td>page 709</td>
<td></td>
</tr>
<tr>
<td>MAXNEWPARTS 2</td>
<td>maximum number of partitions that can be added to partition-by-growth table spaces</td>
<td>page 709</td>
<td></td>
</tr>
<tr>
<td>MAXRO 300</td>
<td>maximum number of seconds that REORG PLUS spends applying log records</td>
<td>page 710</td>
<td></td>
</tr>
<tr>
<td>MAXSORTMEMORY 0</td>
<td>maximum amount of memory that REORG PLUS can allocate to each sort task</td>
<td>page 711</td>
<td></td>
</tr>
<tr>
<td>MAXTAPE 3</td>
<td>maximum number of tape devices to allocate dynamically</td>
<td>page 711</td>
<td></td>
</tr>
<tr>
<td>MGEXTENT CONTINUE</td>
<td>how to allocate extents when extending to a new data set</td>
<td>page 712</td>
<td></td>
</tr>
<tr>
<td>MINSORTMEMORY 0</td>
<td>minimum amount of memory that REORG PLUS should allocate to each sort task</td>
<td>page 712</td>
<td></td>
</tr>
<tr>
<td>MSGLEVEL 1</td>
<td>default for the message level execution parameter</td>
<td>page 712</td>
<td></td>
</tr>
<tr>
<td>OFFPOSLM 10</td>
<td>OFFPOS limit for table space reorganizations</td>
<td>page 713</td>
<td></td>
</tr>
<tr>
<td>OPNDB2ID YES</td>
<td>whether to use the user’s RACF® ID instead of the DB2 RACF ID</td>
<td>page 713</td>
<td></td>
</tr>
<tr>
<td>ORIGDISP DELETE</td>
<td>disposition of original data set during staging data set renaming process</td>
<td>page 714</td>
<td></td>
</tr>
<tr>
<td>PENDDDL YES</td>
<td>whether to enable reorganizing table spaces and indexes that contain pending DDL changes (using DSNUTILB processing)</td>
<td>page 715</td>
<td></td>
</tr>
<tr>
<td>PLAN ARU1110</td>
<td>product plan name</td>
<td>page 715</td>
<td></td>
</tr>
<tr>
<td>PREFORMAT NO</td>
<td>whether to preformat unused data set pages</td>
<td>page 715</td>
<td></td>
</tr>
<tr>
<td>RCVICDDN (BIRY,BIRZ)</td>
<td>ddnames or ddname prefixes for the remote copy data sets that receive an incremental image copy</td>
<td>page 716</td>
<td></td>
</tr>
<tr>
<td>RCVYDDN (BRCY,BRCZ)</td>
<td>ddnames or ddname prefixes for the remote copy data sets that receive a full image copy</td>
<td>page 716</td>
<td></td>
</tr>
<tr>
<td>REDEFINE YES</td>
<td>whether to delete and redefine the VSAM data sets for the table space or index space as part of the reorganization</td>
<td>page 717</td>
<td></td>
</tr>
<tr>
<td>RENMMMAX 30</td>
<td>maximum number of tasks to start during the staging data set rename and delete process</td>
<td>page 717</td>
<td></td>
</tr>
<tr>
<td>RIDMDSSZ 2097152</td>
<td>maximum data space size for RID maps</td>
<td>page 718</td>
<td></td>
</tr>
<tr>
<td>RIDMMAXD 1</td>
<td>number of data spaces for RID maps</td>
<td>page 718</td>
<td></td>
</tr>
</tbody>
</table>
## Basic REORG PLUS installation options

<table>
<thead>
<tr>
<th>Option</th>
<th>Shipped value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMAPMEM</td>
<td>0</td>
<td>amount of memory for RID maps</td>
<td>page 718</td>
</tr>
<tr>
<td>RORGMAX</td>
<td>300%</td>
<td>maximum number of tasks to start for the REORG phase</td>
<td>page 719</td>
</tr>
<tr>
<td>ROUTCDE</td>
<td>(11,1)</td>
<td>WTO console routing codes</td>
<td>page 720</td>
</tr>
<tr>
<td>SCPYMAX</td>
<td>8</td>
<td>maximum number of tasks to start during the nonpartitioned index copy process</td>
<td>page 720</td>
</tr>
<tr>
<td>SDUMP</td>
<td>(ALLPSA,CSA, RGN, SQA,LSQA,SUM,TRT,IO)</td>
<td>system dump parameters</td>
<td>page 720</td>
</tr>
<tr>
<td>SHORTMEMORY</td>
<td>CONTINUE</td>
<td>which action to take when a memory shortage exists</td>
<td>page 721</td>
</tr>
<tr>
<td>SIXSNAP</td>
<td>NO</td>
<td>whether to use the Instant Snapshot technology of SUF or XBM to create a copy of storage-group-defined nonpartitioned indexes</td>
<td>page 723</td>
</tr>
<tr>
<td>SMAX</td>
<td>0</td>
<td>maximum number of sort processes to invoke concurrently</td>
<td>page 723</td>
</tr>
<tr>
<td>SMCORE</td>
<td>(0K,0K)</td>
<td>maximum amount of sort memory</td>
<td>page 724</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>(,SYSALLDA)</td>
<td>sort device types</td>
<td>page 724</td>
</tr>
<tr>
<td>SORTNUM</td>
<td>32</td>
<td>number of sort work files per sort task for BMCSORT to allocate dynamically</td>
<td>page 725</td>
</tr>
<tr>
<td>SPILDSNP</td>
<td>&amp;UID</td>
<td>pattern for spill data set name prefix</td>
<td>page 726</td>
</tr>
<tr>
<td>SPIILSCLS</td>
<td>NONE</td>
<td>SMS storage class for spill data sets</td>
<td>page 727</td>
</tr>
<tr>
<td>SPIILUNIT</td>
<td>WORK</td>
<td>DASD unit for spill data sets</td>
<td>page 728</td>
</tr>
<tr>
<td>SQLDELAY</td>
<td>3</td>
<td>number of seconds between retry attempts after SQLCODE -911</td>
<td>page 728</td>
</tr>
<tr>
<td>SQLRETRY</td>
<td>100</td>
<td>number of retry attempts after SQLCODE -911</td>
<td>page 728</td>
</tr>
<tr>
<td>STAGEDSN</td>
<td>BMC</td>
<td>naming convention to use for staging data sets</td>
<td>page 728</td>
</tr>
<tr>
<td>STOP@CMT</td>
<td>YES</td>
<td>whether to add the 'AT (COMMIT)' parameter to all DB2 STOP commands that REORG PLUS issues</td>
<td>page 729</td>
</tr>
<tr>
<td>STOPDELAY</td>
<td>1</td>
<td>number of seconds between checking to determine whether DB2 has stopped the object</td>
<td>page 729</td>
</tr>
<tr>
<td>STOPRETRY</td>
<td>300</td>
<td>number of times to check to determine whether DB2 has stopped the object</td>
<td>page 729</td>
</tr>
<tr>
<td>TAPEDISP</td>
<td>DELETE</td>
<td>tape file disposition</td>
<td>page 729</td>
</tr>
<tr>
<td>TASKMAX</td>
<td>1000%</td>
<td>global maximum number of tasks to start</td>
<td>page 730</td>
</tr>
<tr>
<td>TEMPRALDATA</td>
<td>Not applicable</td>
<td>obsolete</td>
<td>page 730</td>
</tr>
<tr>
<td>TERMEXIT</td>
<td>(NONE,REXX)</td>
<td>name of user-written TERM exit and the language in which the exit is written</td>
<td>page 730</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>TERM</td>
<td>which action to take when a drain attempt times out</td>
<td>page 731</td>
</tr>
</tbody>
</table>
Table 98  Basic REORG PLUS installation options (part 6 of 6)

<table>
<thead>
<tr>
<th>Option</th>
<th>Shipped value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTALPAGEPCT</td>
<td>0</td>
<td>percentage of total 4-KB pages reported by the system that REORG PLUS can allocate to BMCSORT for sort processing</td>
<td>page 732</td>
</tr>
<tr>
<td>TSPREC</td>
<td>Not applicable</td>
<td>obsolete</td>
<td>page 733</td>
</tr>
<tr>
<td>TSSAMPLEPCT</td>
<td>100</td>
<td>percentage of sampling to perform during statistics gathering</td>
<td>page 733</td>
</tr>
<tr>
<td>TSTZ</td>
<td>Not applicable</td>
<td>obsolete</td>
<td>page 734</td>
</tr>
<tr>
<td>UBUFFS</td>
<td>20</td>
<td>multiplier that controls the amount of buffer pool storage for unload data sets</td>
<td>page 734</td>
</tr>
<tr>
<td>UNLDDN</td>
<td>SYSREC</td>
<td>default ddname or ddname prefix for the output data set that contains the unloaded rows to be reorganized</td>
<td>page 734</td>
</tr>
<tr>
<td>UNLDMAX</td>
<td>300%</td>
<td>maximum number of unload tasks to start</td>
<td>page 735</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>RELOAD</td>
<td>whether to use single- or two-phase processing</td>
<td>page 735</td>
</tr>
<tr>
<td>UTILB_COLCCSID</td>
<td>UTILB</td>
<td>whether to enable support for columns that are defined with CCSID (using DSNUTILB processing)</td>
<td>page 736</td>
</tr>
<tr>
<td>UTILB_NULLIX</td>
<td>UTILB</td>
<td>whether to enable support for indexes that are defined with EXCLUDE NULL KEYS (using DSNUTILB processing)</td>
<td>page 736</td>
</tr>
<tr>
<td>UTSMEM</td>
<td>Not applicable</td>
<td>obsolete</td>
<td>page 737</td>
</tr>
<tr>
<td>UXSTATE</td>
<td>SUP</td>
<td>how to invoke DB2 user exits</td>
<td>page 737</td>
</tr>
<tr>
<td>WBUFSS</td>
<td>(20,10)</td>
<td>multiplier that controls the amount of buffer pool storage for each work data set</td>
<td>page 737</td>
</tr>
<tr>
<td>WORKDDN</td>
<td>SYSUT1</td>
<td>default ddname or ddname prefix for the index work data set</td>
<td>page 737</td>
</tr>
<tr>
<td>WORKUNIT</td>
<td>SYSALLDA</td>
<td>temporary unit for work data sets</td>
<td>page 738</td>
</tr>
<tr>
<td>XBMID</td>
<td>No value</td>
<td>XBM subsystem that REORG PLUS accesses when it uses XBM or SUF</td>
<td>page 738</td>
</tr>
<tr>
<td>XML</td>
<td>Not applicable</td>
<td>obsolete</td>
<td>page 739</td>
</tr>
<tr>
<td>ZIIP</td>
<td>ENABLED</td>
<td>whether to enable zIIP processing</td>
<td>page 739</td>
</tr>
</tbody>
</table>

This section describes each of the options. For more information about setting the values of some of these options at runtime, see Chapter 3, “Syntax of the REORG command.” For information about the performance implications of these options, see Chapter 6, “Tuning REORG PLUS jobs.”
**ALTFFAIL=RCVRPEND**

This option applies to the UTILTERM phase of a SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganization only.

The ALTFFAIL option specifies the action that REORG PLUS takes when a failure occurs during processing of limit-key ALTER statements. This option applies to ALTER statements that are processed either from a DDLIN data set or as a result of the REBALANCE option. You can specify one of the following values:

- TERM tells REORG PLUS to terminate and back out the ALTERs and renames so that the original data sets are restored. The condition of your objects is the same as if the reorganization had never run.
- RCVRPEND tells REORG PLUS to leave the newly reorganized data sets in use, but set the RECP or RBDP status and end with RC=12.

**ANALMAX=***

The ANALMAX option indicates the maximum number of tasks to start for the ANALYZE phase. You can use any of the following values:

- 0 tells REORG PLUS to determine the number of tasks to start.
- An asterisk (*) tells REORG PLUS to use the TASKMAX value (page 730).
- \( n \) tells REORG PLUS to start a maximum of \( n \) tasks (where \( n \) is a positive integer from 1 through 32767).
- \( n\% \) tells REORG PLUS that the maximum number of tasks that it can start is \( n\% \) of the number of CPUs on the system (where \( n \) is a positive integer from 1 through 32768).

For information about using the multitasking options, see “Multitasking installation options” on page 584.

**ARC=NO**

This option applies to SHRLEVEL CHANGE only.

The ARC option tells REORG PLUS whether to use the APPLICATION RESTART CONTROL (AR/CTL) interface. You can specify one of the following values:

- NO tells REORG PLUS not to use the AR/CTL interface.
- YES tell REORG PLUS to use the AR/CTL suspend-and-resume interface. Batch applications using the AR/CTL product and this interface will experience no outage for the DB2 objects that you are reorganizing.
ARCHDDN=SYSARC

This option applies to table space reorganizations only.

The ARCHDDN option specifies the default ddname for the archive data set. This data set contains the rows that REORG PLUS discards during a table space reorganization as a result of the following:

- a SELECT or DELETE operation
- rebalance of a table space where the last partition key has been altered and the table space is defined in one of the following ways:
  - LARGE (either by definition or default)
  - defined with DSSIZE
  - range-partitioned

If you change the default ddnames that BMC supplied, you must also change the names in your JCL. For information about specifying and using this data set, see “SYSARC data set” on page 370.

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS uses this option only to pass a value for the &DDNAME variable for the data set name pattern. REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

Overriding this option
You can override the value for this option by using the ARCHDDN command option (page 185).

ARCHFORMAT=BMC

This option applies to table space reorganizations only.

The ARCHFORMAT option specifies the format of your archive data set (SYSARC). For more information about the SYSARC data set, see “SYSARC data set” on page 370.

Specify one of the following options:

- BMC tells REORG PLUS to produce the archive data set in BMC internal format. You can reload the rows from the archive data set in this format by using the FORMAT BMC option of LOADPLUS. An archive data set produced with ARCHFORMAT BMC uses less space than one produced with ARCHFORMAT DB2.
Basic REORG PLUS installation options

- DB2 tells REORG PLUS to produce the archive data set in the same format as the IBM DB2 REORG utility produces when you specify UNLOAD ONLY. You can reload the rows from the archive data set in this format by using the FORMAT UNLOAD option of either LOADPLUS or the IBM DB2 LOAD utility.

*Overriding this option*
You can override the value for this option by using the ARCHFORMAT command option (page 185).

**AUXREORG (obsolete)**

In REORG PLUS version 10.2 and earlier, the AUXREORG option specified whether to reorganize (by invoking DSNUTILB) associated LOB table spaces while reorganizing the base table space.

Beginning with REORG PLUS version 11.1, you must specify the AUX command to reorganize the base table space and LOB table spaces in the same reorganization. For more information about the AUX command, see “AUX” on page 256.

**AVAILPAGEPCT=100**

The AVAILPAGEPCT option controls virtual storage above the 16-MB line that REORG PLUS allocates to BMCSORT for concurrent sort processing. AVAILPAGEPCT specifies the maximum percentage of available memory, in 4-KB pages in the system, that REORG PLUS can allocate. You can specify any integer from 0 through 100.

---

**NOTE**

REORG PLUS defines *available pages* as pages that have not been used. *Total pages* (which you can control with the TOTALPAGEPCT option), are pages that are underutilized and are available for use.

- A value of 0 tells REORG PLUS to ignore the number of available pages when allocating sort memory.
- A value of 1 through 100 tells REORG PLUS to use up to the specified percentage of available pages when allocating sort memory. For example, AVAILPAGEPCT 50 tells REORG PLUS to use no more than 50 percent of the available pages.

*Additional considerations*
The following additional information applies to the AVAILPAGEPCT option:

- Because available pages are rarely subject to system paging, changing this value will have a minimal effect, if any, on system performance.
When you specify values greater than 0 for both AVAILPAGEPCT and TOTALPAGEPCT, REORG PLUS uses the lesser of the two calculated results as the maximum amount of memory for sort processing.

If REORG PLUS is unable to perform an optimal sort due to an insufficient number of available pages that it is enabled to allocate, the SHORTMEMORY option (page 195 and page 721) controls the action that REORG PLUS takes.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**

You can override the value for this option by using the AVAILPAGEPCT command option (page 193).

**BILDMAX=300%**

The BILDMAX option indicates the maximum number of tasks to start for the index build process. You can specify any of the following values:

- 0 tells REORG PLUS to determine the number of tasks to start.
- An asterisk (*) tells REORG PLUS to use the TASKMAX value (page 730).
- \( n \) tells REORG PLUS to start a maximum of \( n \) tasks (where \( n \) is a positive integer from 1 through 32767).
- \( n\% \) tells REORG PLUS that the maximum number of tasks that it can start is \( n\% \) of the number of CPUs on the system (where \( n \) is a positive integer from 1 through 32768).

Regardless of the value that you specify, REORG PLUS will not start more than 16 tasks for this phase.

**NOTE**

For REORG PLUS to use BILDMAX, you must specify SMAX=0, and either not use MAXSORTS or specify MAXSORTS 0.

For information about using the multitasking options, see “Multitasking installation options” on page 584.
BMCHIST=YES

The BMCHIST option enables you to choose whether to insert a utility history row into the BMC Software BMCHIST table when the reorganization successfully completes. REORG PLUS updates the BMCHIST table in the UTILTERM phase.

- YES tells REORG PLUS to perform the insert.
- NO tells REORG PLUS to bypass the insert.

You can use the TERMEXIT option and user exit to dynamically control processing of updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS at execution time. For more information about the TERMEXIT installation option, see page 730. For more information about the TERMEXIT user exit, see “Using TERMEXIT to control BMCHIST and statistics updates” on page 849.

When invoking DSNUTILB, REORG PLUS ignores this option.

*Overriding this option*

You can override the value for this option by using the BMCHIST command option (page 208).

CBUFFS=30

The CBUFFS option specifies a multiple of 32 KB to use to define the amount of buffer pool storage that is allocated for each copy data set (full and incremental). If you are making multiple copies, the same set of buffers is used, not an additional set. For more information, see “Copy data sets” on page 577.

CLONE (obsolete)

In REORG PLUS version 10.1, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) the following types of objects:

- a clone object
- a base object that participates in a clone relationship when the instance number of the base table space is 2

Beginning with version 10.2, REORG PLUS reorganizes these objects natively and, therefore, no longer uses this option. If your installation options module contains this option, the module compiles successfully but completes with return code 4 and a message indicating that this option is obsolete.
CONDEXEC=NO

The CONDEXEC option specifies whether REORG PLUS should consider performing a conditional reorganization. For a description of how the installation and command options interact, see “Conditional reorganization” on page 136. For more information about conditional execution using the DASD MANAGER PLUS tables, see “Conditional reorganizations using the DASD MANAGER PLUS exceptions table” on page 138.

Valid values are YES, NO, and BMC:

- NO tells REORG PLUS to ignore the limit installation options (OFFPOSLM, INDREFLM, and LEAFDSLM) and perform the reorganization unless you override this option with the CONDEXEC or limit command options.

- YES tells REORG PLUS to examine the values of the limit command options (if specified) and the limit installation options in conjunction with the values in the DB2 catalog to determine whether to perform the reorganization, and to issue a report with the information.

- BMC tells REORG PLUS to issue a report and reorganize the object of the current job if an exception for the object exists in the DASD MANAGER PLUS exceptions table. REORG PLUS issues one of the following return codes after examining the DASD MANAGER PLUS exceptions table:

  — 1, which indicates that no calculated value exceeded an exception value. REORG PLUS does not perform a reorganization.

  — 2, which indicates that a calculated value exceeded an exception value. REORG PLUS performs a reorganization. If REORG PLUS issues any return code greater than 2 during processing, that return code supersedes return code 2.

Restrictions

The following restrictions apply to CONDEXEC:

- CONDEXEC=BMC is available only if you are using REORG PLUS as a component of the Database Performance for DB2 solution.

- If CONDEXEC=BMC or CONDEXEC=YES is in effect, REORG PLUS terminates if you are reorganizing a LOB table space.

Additional considerations

The following additional considerations apply to CONDEXEC:

- When invoking DSNUTILB, REORG PLUS ignores CONDEXEC=BMC.

- To obtain the report that recommends objects for reorganization without performing any reorganizations, specify REPORTONLY in the command.
**Overriding this option**  
You can override the value for this option by using the CONDEXEC command option (page 230).

**COPYDDN=(BCPY, BCPZ)**

The COPYDDN option specifies default ddnames or ddname prefixes for the local copy data sets that receive an image copy or DSN1COPY-type copy of the table space or partitions that you are reorganizing. If you are registering the copies, the first name is the local primary, and the second name is the local backup.

When using this name as a prefix for a ddname for a partitioned table space, ensure that you allow sufficient bytes for the partition number to be added and still have a valid ddname (eight bytes or less). In addition, if you change the default ddnames that BMC supplied, you must also change the names in your JCL. For more information, see “Copy data sets” on page 352.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

**Overriding this option**  
You can override the value for this option by using the COPYDDN command option (page 271).

**COPYLVL=PART**

The COPYLVL option specifies how REORG PLUS is to assign copy data sets when reorganizing partitioned data sets. You can specify one of the following values:

- **COPYLVL=PART** tells REORG PLUS to allocate individual full copy data sets and incremental copy data sets (if created) for each partition that you are reorganizing. If you also specify a tape device for the UNIT option, REORG PLUS allocates a tape unit for each partition.

REORG PLUS appends the partition number to the ddname prefixes that you specify in the COPYDDN (page 680), ICDDN (page 696), RCVYDDN (page 716), and RCVICDDN (page 716) options. The values that you specify for these options plus the highest partition number must not exceed eight characters. For more information, see “Specifying ddname prefixes” on page 93.

When invoking DSNUTILB, REORG PLUS changes COPYLVL PART to COPYLVL FULL when you specify COPY YES.
COPYLVL=FULL tells REORG PLUS to assign a single full copy data set to contain all of the partitions that you are reorganizing in either of the following cases:

— You are reorganizing all partitions.
— You are reorganizing a subset of contiguous partitions (specified on the PART command option) and you specify a value of YES for the COPYSUBSET installation option (page 682).

If incremental copy data sets are created, REORG PLUS also allocates a single incremental copy data set to contain all of the partitions that you are reorganizing.

Additional considerations
The following considerations apply to the COPYLVL option:

If you change ICTYPE in the options module to UPDATE, leave the COPYLVL installation option set to FULL, and do not specify either option on the command, REORG PLUS changes COPYLVL to PART to prevent the job from terminating.

If you anticipate reorganizing a large number of partitions, consider specifying COPYLVL FULL. For more information, see “Large number of partitions” on page 108.

Overriding this option
You can override the value for this option by using the COPYLVL command option (page 269).

COPYMAX=*  

The COPYMAX option indicates the maximum number of tasks to start during the image copy process. You can specify any of the following values:

■ 0 to have REORG PLUS determine the number of tasks to start

■ asterisk (*) to use the TASKMAX value (page 730)

■ n to specify that REORG PLUS can start a maximum of n tasks (where n is a positive integer from 1 through 32767)

■ n% to specify that the maximum number of tasks that REORG PLUS can start is n% of the number of CPUs on the system (where n is a positive integer from 1 through 32768)

For information about using the multitasking options, see “Multitasking installation options” on page 584.
**COPYSUBSET=NO**

The COPYSUBSET option tells REORG PLUS whether to use a single copy data set for a subset of partitions.

COPYSUBSET=NO does not allow this single copy data set. If COPYLVL FULL is in effect in this case, REORG PLUS changes it to COPYLVL PART.

COPYSUBSET=YES enables a single copy data set. You must also meet the following requirements:

- Specify a single subset of physically contiguous partitions with the PART command option.
- For the primary local image copy, either activate dynamic allocation or allocate a single copy data set in your JCL.
- If you dynamically allocate your copy data sets, specify FULL for the COPYLVL installation or command option.

**CPYRFAIL=TERM**

*This option applies to the UTILTERM phase of a SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganization only.*

The CPYRFAIL option specifies the action that REORG PLUS takes if the image copies cannot be registered in SYSIBM.SYSCOPY:

- CPYRFAIL=TERM tells REORG PLUS to terminate and back out the renames, if necessary, so that the original data sets are restored. The condition of your objects is the same as if the reorganization had never run.

- CPYRFAIL=COPYPEND tells REORG PLUS to leave the newly reorganized data sets in use, but set the COPY (copy pending) status and end with RC=4.

REORG PLUS changes COPYPEND to TERM and continues terminating the job when either of the following conditions exists. This value change ensures that REORG PLUS can return the table space to its original state.

- You are using a DDLIN data set to alter limit key values.
- You are reorganizing a catalog object.
**DATACAP=NO**

_This option applies to SHRLEVEL CHANGE only._

The DATACAP option determines whether REORG PLUS turns on the Data Capture Changes flag.

- NO tells REORG PLUS not to turn on the Data Capture Changes flag. Using DATACAP= NO decreases the possibility that contention will occur between REORG PLUS and the user’s application. Therefore, BMC recommends that you always use NO as the value for the DATACAP option.

- YES tells REORG PLUS to turn on the Data Capture Changes flag when you are performing a table space or index space reorganization and any one of the table space partitions is compressed.

  If you specify YES and do not specify KEEPDICTIONARY, you can improve performance. However, these settings increase the possibility that contention will occur between REORG PLUS and the user’s application.

**DDLDDN=DDLIN**

The DDLDDN option allows you to specify the default ddname of the DDLIN data set. For information about using a DDLIN data set in your REORG PLUS jobs, see page 360.

**Overriding this option**
You can also specify the DDLIN data set name by using the DDLDDN command option (page 187).

**DEADLINE=NONE**

_This option applies to SHRLEVEL CHANGE only._

The DEADLINE option specifies the time by which the LOGFINAL phase should finish applying log records. You can specify NONE or a time:

- NONE indicates no deadline.

- You can specify a time in _hh:mm:ss_ format as the deadline, using a 24-hour clock. REORG PLUS replaces the time portion of the current system timestamp with the specified time to create a new timestamp. If the calculated timestamp time has already passed, REORG PLUS updates the date to the next day. Then, it uses this new timestamp to determine when the LOGFINAL phase should stop applying log records.
**Additional considerations**
The following considerations apply to the DEADLINE option:

- If REORG PLUS calculates timestamps for LOGFINAL and DEADLINE, the DEADLINE timestamp must be later than the LOGFINAL timestamp.

- You can use the XBM Utility Monitor function or the MVS operator console to dynamically change the DEADLINE values while the reorganization is in progress.

- The DEADLINE option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 633.

**Overriding this option**
You can override the value for this option by using the DEADLINE command option (page 327).

**DELAY=1200**

*This option applies to SHRLEVEL CHANGE only.*

The DELAY option specifies the number of seconds, as a positive integer (0 or greater), that are to elapse from the time REORG PLUS detects a longlog condition until the time it performs the action specified on the LONGLOG option. If the longlog condition no longer exists at the end of the time period specified by DELAY, the DELAY value is reset.

During the window provided by the DELAY option, you can use the XBM Utility Monitor or the MVS operator console to terminate the reorganization or change the criteria that determines when LOGFINAL processing begins.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 633.

**Overriding this option**
You can override the value for this option by using the DELAY command option (page 330).

**DELFILES=YES**

The DELFILES option tells REORG PLUS whether to delete your work files after the reorganization.

- If you specify DELFILES=YES, the response from REORG PLUS depends on whether the reorganization completes successfully:
  
  — If a non-DSNUTILB reorganization completes successfully, DELFILES=YES tells REORG PLUS to delete all physical sequential data sets whose ddnames match the SYSREC, SYSUT1, and SORTWK ddnames or ddname prefixes.
— For successful DSNUTILB reorganizations, REORG PLUS deletes the following
data sets:

- the SYSREC and SYSUT1 data sets that were used during the reorganization
- all SYSREC, SYSUT1, and SORTWK data sets allocated in your JCL

— If a non-DSNUTILB reorganization completes unsuccessfully, DELFILES=YES tells REORG PLUS to delete any unregistered full and incremental copy data sets (as well as the SYSREC, SYSUT1, and SORTWK data sets) when either of the following conditions exists:

- You specify ON FAILURE phase TERMINATE UTILITY, where phase is UNLOAD, RELOAD, LOGAPPLY, or UTILTERM, and REORG PLUS terminates in the phase that you specified.

- A drain timeout occurs and you specified TIMEOUT TERM as a command or installation option.

In contrast, REORG PLUS does not delete the data sets if

- You did not specify ON FAILURE or TIMEOUT TERM.
- REORG PLUS ends with a STOP status.

— If a DSNUTILB reorganization completes unsuccessfully, REORG PLUS does not perform DELETEFILES processing.

- DELFILES=NO tells REORG PLUS not to delete any files after either a successful or unsuccessful reorganization.

If you are running the reorganization job in a worklist environment, REORG PLUS ignores the value that you set in the installation option and processes the job as if you had specified DELFILES=NO. REORG PLUS does this so that the data sets will exist for subsequent executions in the job. If you want to delete the data sets, you must specify DELETEFILES YES on the REORG command in the worklist.

*Overriding this option*

You can override the value for this option by using the DELETEFILES command option (page 215).

**DESCCDE=(3,7)**

The DESCCE option specifies the descriptor codes to control how write-to-operator (WTO) messages are displayed on a console. You can specify up to six descriptor codes. The values (3,7) specify eventual action required and retain action message for life-of-task. REORG PLUS uses this option only for WTOs that require an action.

For a complete listing of valid values, see the appropriate IBM reference manual.
Basic REORG PLUS installation options

DRAINTYP=ALL

This option applies to SHRLEVEL CHANGE only.

The DRAINTYP option determines the type of drain that REORG PLUS issues when it enters the LOGFINAL phase, after reaching the MAXRO threshold.

You can specify one of the following values:

- ALL (the default) tells REORG PLUS to drain all readers and writers.

  BMC recommends DRAINTYP=ALL if either of the following conditions exists:
  
  — SQL queries might be running that do not commit often enough to allow a drain of the readers to complete successfully in a timely fashion.

  — SQL updaters might be running that require a minimum possible outage, and these updaters might time out if they are all held while awaiting the readers to be drained.

- WRITERS tells REORG PLUS to drain only writers.

Overriding this option
You can override the value for this option by using the DRAIN command option (page 340).

DRNDELAY=1

After a drain times out, the DRNDELAY option specifies the minimum number of seconds that you want REORG PLUS to wait before it tries again to obtain the drain. The number of seconds can range from 1 through 1800.

Overriding this option
You can override the value for this option by using the RETRY_DELAY command option (page 246).

DRNRETRY=255

The DRNRETRY option specifies the maximum number of times to attempt to obtain a drain before terminating the job. The number of retry attempts can range from 0 through 255.

Overriding this option
You can override the value for this option by using the RETRY command option (page 246).
**DRNWAIT=NONE**

The DRNWAIT option specifies the drain timeout value to use. Specify one of the following values:

- **NONE**, which means that the drain request issued by REORG PLUS times out immediately if the drain cannot acquire the lock.

  NONE prevents any application transactions from being queued during the drain process. BMC recommends that you specify NONE in high-transaction environments.

- **UTIL**, which tells REORG PLUS to use the standard DB2 utility timeout value defined in DSNZPARMs for your site (IRLMRWT multiplied by UTIMOUT).

  The wait time applies to each object involved in the reorganization.

- **SQL**, which tells REORG PLUS to use the standard SQL timeout value (IRLMRWT) as the drain timeout value.

  The wait time applies to each object involved in the reorganization.

- Any integer value from 0 through 1800.
  
  — 0 is equivalent to the value UTIL.
  
  — 1 through 1800 specifies the number of seconds to wait to obtain the drain for each drain retry before timing out.

If REORG PLUS cannot drain all of the objects within the time period specified by DRNWAIT, REORG PLUS completes the following process:

1. releases the drains that it has obtained so far

2. waits the length of time that you specify in the DRNDELAY installation option (or RETRY_DELAY command option)

3. tries again to drain the objects for the number of times that you specify in the DRNRETRY installation option (or RETRY command option)

**NOTE**

The FORCE and FORCE_AT options tell REORG PLUS whether to cancel DB2 threads that might prevent a successful drain and, if so, when to cancel them. The value that you specify for the FORCE and FORCE_AT options might affect the drain process described here. For more information, see page 693.
**DSNUTILB reorganizations**

When invoking DSNUTILB, REORG PLUS converts the DRNWAIT option based on the DSNZPARM settings, and passes this option to the IBM DB2 REORG utility as DRNWAIT=integer, where integer is defined as follows:

- For DRNWAIT=UTIL, integer equals the product of SPRTMTOUT multiplied by SPRMUTO.
- For DRNWAIT=NONE, integer equals 1 (one second).
- For DRNWAIT=SQL, integer equals SPRTMTOU1T.

**Overriding this option**

You can override the value for this option by using the DRAIN_WAIT command option (page 245).

**DSNUEXIT=(NONE,ASM)**

The DSNUEXIT option specifies the name of a user-defined exit that creates data set name patterns and the programming language in which the exit is written. This user exit provides the user-defined variables for constructing name patterns for data sets that REORG PLUS dynamically allocates with either the DSNPAT or SPIILDSNP installation option or the DSNPAT (page 315) or SPIILDSNPAT (page 331) command option.

The possible values for the language are

- ASM
- COBOL2
- LE_COBOL
- C
- LE_C

When invoking DSNUTILB, REORG PLUS ignores this option.

For more information about the DSNUEXIT user exit, see “Using DSNUEXIT to construct data set name patterns” on page 789.

**Overriding this option**

You can override the value for this option by using the DSNUEXIT command option (page 240).
DSNUTILB=YES

The DSNUTILB option tells REORG PLUS whether to invoke DSNUTILB to pass processing to the IBM DB2 REORG utility. REORG PLUS uses this option to enable support for certain features. For the list of features, see “Reorganization jobs that invoke DSNUTILB” on page 72.

**NOTE**
REORG PLUS invokes DSNUTILB to enable new features quickly. REORG PLUS generally provides native support for these features (without invoking DSNUTILB) in a later version of the product or via PTF. You cannot use the DSNUTILB option (or any other option) to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively.

You can specify one of the following values:

- **YES** tells REORG PLUS to invoke DSNUTILB when DSNUTILB is required to support the object type that is involved in the reorganization. For a table space reorganization, REORG PLUS invokes DSNUTILB for the job if any table in or index on the table space that you are reorganizing uses a feature that REORG PLUS supports via DSNUTILB.

- **NO** tells REORG PLUS to not invoke DSNUTILB. If the type of reorganization job that you are running requires DSNUTILB, REORG PLUS issues message BMC50178E and terminates.

For requirements and restrictions when REORG PLUS invokes DSNUTILB, see “Reorganization jobs that invoke DSNUTILB” on page 72.

**Overriding this option**
You can override the value for this option by using the DSNUTILB command option (page 237).

DSPLOCKS=DRNFAIL

The DSPLOCKS option tells REORG PLUS what action to take regarding displaying claims and locks if a drain attempt times out:

- **DRNFAIL** tells REORG PLUS to display the claims and locks once, after the final attempt to obtain the drain times out.

- **NONE** tells REORG PLUS not to display any claims or locks.

- **RETRY** tells REORG PLUS to display the claims and locks after each drain timeout.

**Restriction**
When invoking DSNUTILB, REORG PLUS ignores this option.
**Overriding this option**

You can override the value for this option by using the DSPLOCKS command option (page 247).

**DSRSEXIT=(NONE,REXX)**

The DSRSEXIT option specifies the name of a user-defined exit for managing the redefinition of DB2 VSAM data sets, and the programming language in which the exit is written. This exit must be written in REXX.

You can use this user exit to

- change the primary and secondary quantities of the object that you are reorganizing

  Unless you indicate otherwise, this exit also updates the DB2 catalog with the changed values for use with subsequent allocations.

- selectively tell REORG PLUS to specify REDEFINE NO for an object

- specify the order of the storage group volumes to use for each object

  You can also restrict the volume list to a subset of the original storage group volume list.

- add SMS classes

  If no SMS classes exist in the DSNZPARMs or storage-group definition, you can add the classes by using the DSRSEXIT user exit. However, you cannot change existing specified SMS classes by using the DSRSEXIT user exit.

For more information about the DSRSEXIT user exit, see Appendix C, “REORG PLUS user exits.”

**Overriding this option**

You can override the value for this option by using the DSRSEXIT command option (page 240).

**EXCLDUMP=(X37,X22,X06)**

The EXCLDUMP option enables you to limit the conditions under which REORG PLUS generates a system dump when the SDUMP option contains values other than NO. The EXCLDUMP option tells REORG PLUS to exclude the listed abend codes when generating the dump.
Specify a three-digit abend code, such as 806 or 222, or prefix an 'X' to a two-digit abend code to exclude all abend codes that end in those two digits. For example, if you specify X37, REORG PLUS will not generate a system dump for all abend codes that end in 37 (such as B37 or E37). You can specify up to 10 abend codes with the EXCLDUMP option.

Specify EXCLDUMP=0 (without parentheses) if you want all abend codes to be candidates for a system dump.

**FASTSWITCH=NO**

*This option applies to SHRLEVEL REFERENCE or SHRLEVEL CHANGE only.*

The FASTSWITCH option determines the action that REORG PLUS takes in the UTILTERM phase regarding the staging data sets.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

**Additional considerations**

Consider the following information about the FASTSWITCH process:

- You can use the FSFALLBACK (page 695) and FSTHRESHOLD (page 695) options to control whether REORG PLUS performs FASTSWITCH or rename processing under particular conditions.

- BMC recommends specifying FASTSWITCH YES when you are reorganizing objects with more than 200 data sets, such as a segmented table space that contains many tables that each have an index.

- The FASTSWITCH process requires updates to the DB2 catalog and directory. In some environments, REORG PLUS might have difficulty acquiring the necessary locks, which results in contention or deadlocks. Minimizing use of the FASTSWITCH process can reduce this contention and allow the process to be more effective when it is needed. Frequently reorganizing the DB2 catalog and directory can also help prevent contention.

- BMC recommends that you do not specify FASTSWITCH YES when your staging data sets are named according to the STAGEDSN=BMC method.

You can specify one of the following values:

- NO tells REORG PLUS to rename the staging data sets to the original data set names. This value can help to prevent potential contention issues between the DB2 catalog and directory.
- YES tells REORG PLUS to bypass the VSAM rename process and directly update the DB2 catalog to use the staging data set names.

  When reorganizing either of the following types of objects, REORG PLUS changes FASTSWITCH=YES to FASTSWITCH=NO:

  — clone objects
  — catalog objects

- ZPARM tells REORG PLUS to use the value that is set in IBM DSNZPARM for FASTSWITCH. FASTSWITCH=ZPARM enables you to use DB2 parameters to manage the FASTSWITCH setting without changing your REORG PLUS installation options module.

  When reorganizing clone objects, REORG PLUS changes this value to NO if the value of the DSNZPARM for FASTSWITCH is YES.

For more information about FASTSWITCH processing, see “Staging data sets and the FASTSWITCH process” on page 105.

**Overriding this option**
You can override the value for this option by using the FASTSWITCH command option (page 218).

**FILECHK=FAIL**

The FILECHK option specifies the action that REORG PLUS should take when encountering a work file that has been allocated in one of the following ways:

- as a temporary data set

  For a definition of temporary data sets within REORG PLUS, see “Check for data set attributes” on page 96.

- as DD DUMMY or DSN=NULLFILE

Specify one of the following options:

- FAIL tells REORG PLUS to terminate.
- WARN tells REORG PLUS to issue a warning message and continue processing.

**Restriction**
When invoking DSNUTILB, REORG PLUS ignores this option.
FORCE=NONE

The FORCE option specifies whether to cancel DB2 threads that might prevent a drain process from completing. Specify one of the following options:

- NONE tells REORG PLUS not to cancel DB2 threads.
- READERS tells REORG PLUS to cancel read claimers at the point specified by the FORCE_AT option.
- ALL tells REORG PLUS to cancel both read and write claimers at the point specified by the FORCE_AT option.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the FORCE command option (page 248). The FORCE command option provides an additional option, REPORTONLY, that tells REORG PLUS to display a report of the threads that would have been canceled if you had specified FORCE ALL.

FORCE_AT=(START,3)

The FORCE_AT option tells REORG PLUS when to cancel DB2 threads that might prevent the drain process from completing. This option is applicable only when FORCE READERS, FORCE ALL, or FORCE REPORTONLY is in effect.

The first parameter of this option indicates at which point during the drain process to begin canceling threads. The second parameter indicates how long to delay the start of the cancelation process from the specified point in the drain process.

To obtain the same behavior as the FORCE option of the IBM DB2 REORG TABLESPACE utility, specify FORCE_AT=(LASTRETRY,0).

For the first parameter of this option, specify one of the following values:

- START tells REORG PLUS to start canceling threads when the drain request begins.
- RETRY tells REORG PLUS to start canceling threads the first time the drain process times out and REORG PLUS attempts to retry the drain.
- LASTRETRY tells REORG PLUS to start canceling threads at the beginning of the last retry attempt following a drain process timeout. You can control the number of drain retry attempts with the RETRY command option (page 246) or DRNRETRY installation option (page 686).
For the second parameter of this option, specify an integer value to indicate hundredths of a second. For example, specify 7 to tell REORG PLUS to wait .07 seconds. Specify an integer value of 0 or greater. The default, 0, tells REORG PLUS to start the cancelation process immediately upon reaching the point specified by the FORCE_AT option.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for the first parameter of this option by using the FORCE_AT suboption of the FORCE command option (page 248). You can override the value for the second parameter by using the FORCE_DELAY suboption of the FORCE command option (page 249).

FORCE_RPT=NO

The FORCE_RPT option tells REORG PLUS whether to display a report of the canceled threads.

Specify one of the following values for this option:

- YES tells REORG PLUS to display the report. This value is applicable only when FORCE READERS or FORCE ALL is in effect.
  
  If your JCL includes a BMCFORCE DD statement, REORG PLUS sends the canceled threads report to that data set. Otherwise, the report is displayed in the REORG PLUS SYSPRINT.

- NO tells REORG PLUS not to display the report.

Restrictions
REORG PLUS ignores this option when either of the following conditions exists:

- REORG PLUS is invoking DSNUTILB.
- You specify FORCE REPORTONLY on the REORG command.
**Overriding this option**

You can override the value for this option by using the FORCE\_RPT suboption of the FORCE command option (page 249).

### FSFALLBACK=NO

*This option applies to SHRLEVEL REFERENCE or SHRLEVEL CHANGE.*

The FSFALLBACK option tells REORG PLUS whether to perform rename processing of your staging data sets when FASTSWITCH processing fails:

- **NO** tells REORG PLUS to terminate instead of falling back to rename processing if FASTSWITCH processing fails.
- **YES** tells REORG PLUS to fall back to rename processing if FASTSWITCH processing fails.

**Restrictions**

The following restrictions apply to FSFALLBACK:

- For the following types of reorganizations, REORG PLUS falls back to rename processing only after successfully backing out the FASTSWITCH changes:
  - an index reorganization
  - a partial table space reorganization with a participating nonpartitioned index
- REORG PLUS ignores this option when any of the following conditions exists:
  - FASTSWITCH NO is in effect.
  - REORG PLUS is invoking DSNUTILB.
  - REORG PLUS is converting objects from BRF to RRF.

**Overriding this option**

You can override the value for this option by using the FSFALLBACK command option (page 220).

### FSTHRESHOLD=0

*This option applies to SHRLEVEL REFERENCE or SHRLEVEL CHANGE.*

The FSTHRESHOLD option provides a threshold above which REORG PLUS performs FASTSWITCH processing. If the number of staging data sets used in a single reorganization is greater than this value, REORG PLUS performs FASTSWITCH processing. Otherwise, REORG PLUS performs rename processing.

You can specify a value from 0 through 32767. A value of 0 tells REORG PLUS to always perform FASTSWITCH processing.
Basic REORG PLUS installation options

**Restriction**
REORG PLUS ignores this option when FASTSWITCH NO is in effect.

**Overriding this option**
You can override the value for this option by using the FSTHRESHOLD command option (page 221).

**HASHAX=YES**

The HASHAX option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) a table space that contains a table that is defined as ORGANIZE BY HASH:

- YES tells REORG PLUS to reorganize this table space by using DSNUTILB processing. To enable this feature, DSNUTILB YES must also be in effect.
- NO tells REORG PLUS not to reorganize this table space. REORG PLUS terminates.

**NOTE**
REORG PLUS natively reorganizes indexes on tables that are defined as ORGANIZE BY HASH.

You cannot use this option to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively. For more information about running a DSNUTILB job, see “Reorganization jobs that invoke DSNUTILB” on page 72.

In a future release, REORG PLUS will no longer require or support the HASHAX installation option.

**ICDDN=(BICY, BICZ)**

This option applies to SHRLEVEL CHANGE only.

The ICDDN option specifies the ddnames or ddname prefixes for the local copy data sets that receive an incremental image copy of the table space or partitions that you are reorganizing. The first name is the local primary, and the second name is the local backup. You must define a corresponding incremental data set for each copy data set that you specify if the value of ICTYPE is INCREMENTAL.

When using this name as a prefix for a ddname for a partitioned table space, ensure that you allow sufficient bytes for the partition number to be added and still have a valid ddname (eight bytes or less). In addition, if you change the default ddname that BMC supplied, you must also change the names in your JCL. For more information, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 653.
**Restriction**
When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**
You can override the value for this option by using the ICDDN command option (page 275).

**ICTYPE=AUTO**

This option applies to SHRLEVEL CHANGE only.

The ICTYPE option tells REORG PLUS to update the full image copy data sets or to create incremental image copy data sets during a SHRLEVEL CHANGE reorganization. Updating only full copies gives the best performance; REORG PLUS does not have to create the incremental copies during the LOGFINAL phase, thus reducing the time that application updates are prevented.

You can specify AUTO, UPDATE, or INCREMENTAL:

- A value of AUTO tells REORG PLUS to decide whether to update the full copy data sets or to create incremental copy data sets. BMC recommends that you use AUTO to allow REORG PLUS to decide which type of copy is best.

  REORG PLUS updates full copies when either of the following conditions exists:

  - You specify INLINECP=YES in the installation options or INLINE YES on the REORG command.

  - All of the full copy data sets are on DASD, and one full copy data set exists for each partition that you are reorganizing.

  If neither statement is true, REORG PLUS creates incremental copy data sets. You must define or dynamically allocate these incremental copy data sets.

  When invoking DSNUTILB, REORG PLUS ignores ICTYPE=AUTO and DSNUTILB uses ICTYPE UPDATE.
If you specify UPDATE (or you specify AUTO and REORG PLUS chooses UPDATE), REORG PLUS always updates the full copy data sets.

If you specify INLINECP=YES or INLINE YES, REORG PLUS appends the updated pages to the full copy data sets. With INLINE or INLINECP set to YES, the copy data sets can be on tape.

If you specify UPDATE and the value of the INLINECP installation option or INLINE command option is NO, REORG PLUS updates the full copy data sets in place. If you specify INLINE NO and any of the following statements are true, REORG PLUS terminates:

— Any of the data sets are not on DASD.
— The incremental copy data sets are defined as striped data sets.
— You are attempting to create a single image copy for multiple partitions, as in the following scenarios:
  ■ You allocate a single copy data set in your JCL but specify multiple partitions.
  ■ You specify COPYLVL FULL and COPYSUBSET=YES.

If you specify INCREMENTAL (or you specify AUTO and REORG PLUS chooses INCREMENTAL), REORG PLUS creates incremental copy data sets. You must define or dynamically allocate these data sets. If the data sets do not exist, REORG PLUS terminates.

If you specify SHRLEVEL CHANGE and ICTYPE=INCREMENTAL, REORG PLUS behaves as if you specified INLINE NO, regardless of the value that you specified for INLINE or INLINECP.

When invoking DSNUTILB, REORG PLUS ignores ICTYPE=INCREMENTAL and DSNUTILB uses ICTYPE UPDATE.

If your installation allows SMS data sets to go to tape for the SMS classes that you specify for dynamically allocated copy data sets, you must specify ICTYPE=INCREMENTAL if all of the following statements are true:

■ You specify SHRLEVEL CHANGE.
■ You specify SMS YES.
■ The value of the SMSUNIT installation or command option is NO.
■ The value of the INLINE command or INLINECP installation option is NO.

**NOTE**

REORG PLUS supports updating SMS-managed striped data sets for the copy data sets when you specify ICTYPE AUTO or UPDATE and INLINE YES (or INLINECP=YES).
**Basic REORG PLUS installation options**

**IDCACHE=10000**

The IDCACHE option allows you to specify the size of the cache that REORG PLUS uses when populating document ID columns. Specify the size as the number of values to reserve. REORG PLUS reserves this cache in SYSIBM.SYSSEQUENCES and reserves at least one cache for each unload and log apply task that requires it. The valid values for this option are 1 through 2147483647.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**
You can override the value for this option by using the IDCACHE command option (page 255).

**INDREFLM=10**

*This option does not apply to an index-only reorganization.*

If the value of the CONDEXEC command or installation option is YES, REORG PLUS issues a report recommending which objects in a table space should be reorganized, and uses the value in the INDREFLM option to conditionally perform the reorganization. Valid values are 0 through 100 or NONE.

For the table space named in the REORG command, REORG PLUS performs the following calculation. The calculation applies to the specified partitions in SYSIBM.SYSTABLEPART for the table space.

\[
\text{integer} = \frac{(\text{NEARINDREF} + \text{FARINDREF}) \times 100}{\text{CARDF}}
\]

If any calculated integer value exceeds the INDREFLM value, REORG PLUS reorganizes the object.

If the value of the CONDEXEC command or installation option is YES and you specify NONE in the INDREFLM installation option, REORG PLUS performs a conditional reorganization based on the values of the other limit options but not the value of INDREFLM.

**Overriding this option**
You can override the value for this option by using the INDREFLIMIT command option (page 233). If you specify the INDREFLIMIT command option with no value, REORG PLUS uses the value in this installation option as the value for that command option. For more information about the interaction between the limit installation and command options, see “Conditional reorganization” on page 136.
The INLINECP option tells REORG PLUS whether to create inline copies. You can specify YES or NO:

- **INLINECP=YES** (the default) specifies that REORG PLUS should create an inline image copy while reloading the table space. This requires that you also specify COPY YES on the REORG command.

If *any* of the following conditions exists for a particular reorganization, REORG PLUS behaves as if you specified NO, regardless of the value that you specify for INLINECP or for the INLINE command option:

- The copy data sets are on a stacked tape.
- The size of the table space page is greater than 4 KB, you are reorganizing multiple partitions, and you have a single image copy data set.
- You specify SHRLEVEL CHANGE and ICTYPE INCREMENTAL.
- You restart a SHRLEVEL NONE or SHRLEVEL REFERENCE table space reorganization and *all* of the following statements are true:
  
  - The table space is partitioned.
  - You have a single image copy data set.
  - At least one (but not all) of the partitions was reloaded before the failure.

If you specify INLINECP=YES for a SHRLEVEL CHANGE reorganization, REORG PLUS uses more memory than it would with INLINECP=NO because it keeps all of the image copy data sets open until the LOGFINAL phase completes processing.

When invoking DSNUTILB, REORG PLUS passes INLINECP=YES to the IBM DB2 REORG utility, which creates inline, single data set copies.

---

**TIP**

You can obtain the report that recommends objects for reorganization *without* performing any reorganizations. To do so, specify REPORTONLY in the command.
- INLINECP=NO specifies that REORG PLUS should not make an inline image copy while reloading the table space, but should create an image copy after reloading the table space.

REORG PLUS terminates if you also specify ICTYPE UPDATE and you attempt to create a single image copy for multiple partitions, as in the following scenarios:

- You allocate a single copy data set in your JCL but specify multiple partitions.
- You specify COPYLVL FULL and COPYSUBSET=YES.

**Overriding this option**
You can override the value for this option by using the INLINE command option (page 265).

**INLOB (obsolete)**

In REORG PLUS version 10.2 and earlier, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) a table space that contains an inline LOB column.

Beginning with version 11.1, REORG PLUS reorganizes table spaces that contain inline LOB columns natively in most cases and, therefore, no longer uses this option. If your installation options module contains this option, the module compiles successfully but completes with return code 4 and a message indicating that this option is obsolete.

**NOTE**
REORG PLUS invokes DSNUTILB when the length of an inline LOB column was altered before the reorganization. In this case, specify YES on the LOB installation option.

**IXINCLCOL (obsolete)**

In REORG PLUS version 10.1, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) an index that contains non-key columns or a table space that contains this type of index.
Beginning with version 10.2, REORG PLUS reorganizes these objects natively and, therefore, no longer uses this option. If your installation options module contains this option, the module compiles successfully but completes with return code 4 and a message indicating that this option is obsolete.

**IXONEX=NO**

The IXONEX option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) indexes that contain a key derived from certain expressions or table spaces that contain this type of index. These indexes include indexes on an inline LOB column and indexes that contain columns defined with a CCSID specification.

For more information about which expressions REORG PLUS supports natively, see “Indexes on expression” on page 148.

**NOTE**

REORG PLUS natively reorganizes indexes defined on table spaces that contain inline LOB data when the index is not on an inline LOB column.

You can specify one of the following options:

- **YES** tells REORG PLUS to use DSNUTILB processing to reorganize this type of index or a table space that contains this type of index. To enable this feature, DSNUTILB YES must also be in effect.

- **NO** tells REORG PLUS not to reorganize this type of index or a table space that contains this type of index. If REORG PLUS encounters this type of index, REORG PLUS issues message BMC50920E and terminates.

You cannot use this option to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively. For more information about running a DSNUTILB reorganization, see “Reorganization jobs that invoke DSNUTILB” on page 72.

In a future release, REORG PLUS will no longer require or support the IXONEX installation option.

**IXRANDOM=NO**

The IXRANDOM option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) indexes that contain a key with random ordering or table spaces that contain this type of index:

- **YES** tells REORG PLUS to use DSNUTILB processing to reorganize this type of index or a table space that contains this type of index. To enable this feature, DSNUTILB YES must also be in effect.
NO tells REORG PLUS not to reorganize this type of index or a table space that contains this type of index. If REORG PLUS encounters this type of index, REORG PLUS issues message BMC50920E and terminates.

For more information about running a DSNUTILB reorganization, see “Reorganization jobs that invoke DSNUTILB” on page 72.

In a future release, REORG PLUS will no longer require or support the IXRANDOM installation option.

**KEEPDICTIONARY=NO**

This option applies to a table space reorganization only.

The KEEPDICTIONARY option tells REORG PLUS whether to keep the existing compression dictionary. For more information about how REORG PLUS handles compression dictionaries, see “Table space compression” on page 145.

KEEPDICTIONARY=NO tells REORG PLUS to build a new compression dictionary, even if one already exists. REORG PLUS builds the dictionary and compresses the data during the UNLOAD phase.

KEEPDICTIONARY=YES tells REORG PLUS to keep the existing compression dictionary. If a dictionary does not exist, REORG PLUS builds one.

REORG PLUS ignores KEEPDICTIONARY=YES and operates as if you had specified KEEPDICTIONARY=NO when

— You are performing partition rebalancing.
— You are reorganizing a partition-by-growth table space.

When invoking DSNUTILB, REORG PLUS passes KEEPDICTIONARY=YES to the IBM DB2 REORG utility as KEEPDICTIONARY, and ignores KEEPDICTIONARY=NO.

**Restrictions**
The following restrictions apply to the KEEPDICTIONARY option:

- The KEEPDICTIONARY option is valid only if the table space or partition that you are reorganizing has the COMPRESS YES attribute.
- If a table space is compressed and a REORG PLUS job would convert the row format from BRF to RRF, REORG PLUS builds a new dictionary except when both of the following options are in effect:

— The value of the REORG PLUS KEEPDICTIONARY option is YES.
— The value of the DB2 subsystem parameter HONOR_KEEPDICTIONARY is YES.
**Basic REORG PLUS installation options**

*Overriding this option*
You can override the value for this option by using the KEEPDICTIONARY command option (page 206).

**LEAFDSLM=200**

*This option applies to index-only reorganizations.*

If the value of the CONDEXEC command or installation option is YES, REORG PLUS issues a report recommending which objects in a table space should be reorganized. REORG PLUS then uses the value in the LEAFDSLM option to conditionally perform the reorganization. Valid values are any positive integer (0 or greater) or NONE.

For the partitions that you are reorganizing for the specified index, REORG PLUS compares the specified LEAFDSLM value to the value of LEAFDIST in SYSIBM.SYSINDEXPART. If any LEAFDIST value exceeds the LEAFDSLM value, REORG PLUS reorganizes the index.

If the value of the CONDEXEC command or installation option is YES and you specify NONE in the LEAFDSLM installation option, REORG PLUS performs a conditional reorganization based on the values of the other limit options but not the value of LEAFDSLM.

*Overriding this option*
You can override the value for this option by using the LEAFDISTLIMIT command option (page 234). If you specify the LEAFDISTLIMIT command option with no value, REORG PLUS uses the value in this installation option as the value for that command option. For more information about the interaction between the limit installation and command options, see “Conditional reorganization” on page 136.

**TIP**
You can obtain the report that recommends objects for reorganization *without* performing any reorganizations. To do so, specify REPORTONLY in the command.

**LOB=YES**

The LOB option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces that contain a LOB column, LOB table spaces, or both when a DSNUTILB reorganization is required. For information about when REORG PLUS invokes DSNUTILB for LOB data, see “Reorganization jobs that invoke DSNUTILB” on page 72.

You can specify one of the following options:

- YES tells REORG PLUS to invoke DSNUTILB to reorganize these objects when required. The DSNUTILB option must be YES.
- **NO** tells REORG PLUS to not reorganize these objects when DSNUTILB is required. REORG PLUS terminates.

You cannot use this option to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively. For more information about running a DSNUTILB reorganization, see “Reorganization jobs that invoke DSNUTILB” on page 72.

In a future release, REORG PLUS will no longer require or support the LOB installation option.

**LOCKROW=YES**

The LOCKROW option tells REORG PLUS which serialization method to use when updating the BMCSYNC and BMCUTIL tables:

- **YES** tells REORG PLUS to use MVS enqueues instead of the SQL LOCK TABLE statements for serialization. Using LOCKROW=YES should prevent most SQL -911 return codes that occur when multiple BMC products concurrently update the BMCSYNC and BMCUTIL tables.

- **NO** tells REORG PLUS to use the SQL LOCK TABLE statements for serialization when updating the BMCSYNC and BMCUTIL tables.

**Additional considerations**

The following considerations apply to the LOCKROW option:

- Using LOCKROW=YES requires row-level locking. You must define the BMCSYNC and BMCUTIL tables with LOCKSIZE ROW to achieve row-level locking.

- The following BMC products must use the same LOCKROW value within a subsystem: CHECK PLUS, COPY PLUS, LOADPLUS, RECOVER PLUS, REORG PLUS, and UNLOAD PLUS. Regardless of the value assigned for LOCKROW, COPY PLUS and RECOVER PLUS always behave as if LOCKROW=YES.

- In a future version, REORG PLUS will no longer require or support the LOCKROW installation option.
LOGFINAL=NONE

This option applies to SHRLEVEL CHANGE only.

The LOGFINAL option specifies when you want REORG PLUS to start the LOGFINAL phase. Through this option, you can control when updates are prevented to the table space or index space that you are reorganizing. You can specify NONE or a time:

- NONE indicates no set time, and the other log apply options will control the process. For information, see “Control of the log apply process” on page 633.

- You can specify a time in hh:mm:ss format as the value, using a 24-hour clock. REORG PLUS replaces the time portion of the current system timestamp with the time specified to create a new timestamp. If the calculated timestamp time has already passed, REORG PLUS updates the date to the next day. Then, it uses this new timestamp to determine when it will begin the LOGFINAL phase.

Additional considerations
The following considerations apply to the LOGFINAL option:

- REORG PLUS does not check the LOGFINAL value until the LOGAPPLY phase begins. At that point, if the current time is greater than the time specified on the LOGFINAL option, REORG PLUS immediately begins the LOGFINAL phase.

- You can use the XBM Utility Monitor function or the MVS operator console to dynamically change LOGFINAL values while the reorganization is in progress.

- When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the LOGFINAL command option (page 338).

LOGMEM=0

This option applies to SHRLEVEL CHANGE only.

The LOGMEM option specifies the amount of data space memory (in kilobytes) needed to hold the internal structures that REORG PLUS uses to store the log records (data and index components). The number must be 0 or a positive integer.

BMC strongly recommends that you use the value 0. A value of 0 tells REORG PLUS to automatically compute the amount of data space memory that it needs. To do so, REORG PLUS uses the formula described in “Sizing memory for log records” on page 662.
If you specify a nonzero value for LOGMEM, during execution REORG PLUS first ensures that your specified value is sufficient to satisfy the minimum memory requirements. If the value is not sufficient, REORG PLUS issues message BMC50864I and changes the value for LOGMEM to the minimum required memory amount.

If you specify a nonzero positive integer, REORG PLUS allocates up to that amount of data space memory.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**
You can override the value for this option by using the LOGMEM command option (page 337).

\[
\text{LOGSPIL} = (20000, 10000)
\]

This option applies to SHRLEVEL CHANGE only.

The LOGSPIL option specifies the primary and secondary space allocation (in kilobytes) for the spill data sets that hold the log records. The number must be a nonzero positive integer. REORG PLUS uses these data sets when the memory allocated with LOGMEM becomes full.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**
You can override the value for this option by using the LOGSPILL command option (page 338).

\[
\text{LOGTHRS}=0
\]

This option applies to SHRLEVEL CHANGE only.

The LOGTHRS option specifies the number of log records as a positive integer (0 or greater). When the number of log records remaining to be applied is less than or equal to this value, REORG PLUS begins the LOGFINAL phase. The LOGFINAL phase prevents updates to the objects and applies the remaining log records.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 633.

**Overriding this option**
You can override the value for this option by using the LOGTHRESHLD command option (page 326).
LONGLOG=CONTINUE

This option applies to SHRLEVEL CHANGE only.

The LONGLOG option specifies the action to take if REORG PLUS determines that the DB2 subsystem is generating log records for the objects that you are reorganizing faster than REORG PLUS is applying them:

- CONTINUE tells REORG PLUS to continue performing the reorganization, even if it detects a longlog condition after the time specified by the DELAY option expires.
- TERM tells REORG PLUS to terminate the reorganization.
- DRAIN tells REORG PLUS to begin the LOGFINAL phase, prevent updates to the objects that you are reorganizing, and apply the remaining log records.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 633.

Overriding this option
You can override the value for this option by using the LONGLOG command option (page 329).

LONGNAMETRUNC=MIDDLE

The LONGNAMETRUNC option tells REORG PLUS where to truncate names that are longer than the area that is available in a REORG PLUS report-style message. Table 99 describes the valid values and the symbol that appears in the message based on that value.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIDDLE (or M)</td>
<td>truncates outward from the middle of the name</td>
<td>&gt;&lt;</td>
</tr>
<tr>
<td>BEGINNING (or B)</td>
<td>truncates from the beginning of the name</td>
<td>&lt;&lt;</td>
</tr>
<tr>
<td>END (or E)</td>
<td>truncates from the end of the name</td>
<td>&gt;&gt;</td>
</tr>
</tbody>
</table>

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the LONGNAMETRUNC command option (page 178).
MAPTEXIT=(NONE,REXX)

(For DB2 Versions 9 and 10) The MAPTEXIT option allows you to specify the name of a user-written exit that creates mapping objects for use with a DSNUTILB reorganization. This exit also drops the mapping objects after DSNUTILB processing completes successfully.

For details about the MAPTEXIT user exit, see Appendix C, “REORG PLUS user exits.”

Restrictions
The following restrictions apply to the MAPTEXIT option and the MAPTEXIT user exit:

- REORG PLUS uses this exit only when invoking DSNUTILB for a SHRLEVEL CHANGE reorganization.
- REORG PLUS ignores this option when any of the following conditions exists:
  - You are running on DB2 Version 11 or later.
  - You also specify the MAPPINGTABLE command option.
  - You specify REORG INDEX.

Overriding this option
You can override the value for this option by using the MAPTEXIT command option (page 243).

MAXNEWPARTS=2

This option applies to partition-by-growth table spaces only.

The MAXNEWPARTS option specifies the following limits:

- the maximum number of partitions that REORG PLUS can add during a reorganization

  Partition extension is also constrained by the DB2 MAXPARTITIONS value with which the table space was created.

- the maximum number of partitions added by DB2 that REORG PLUS can support (only applies to a SHRLEVEL CHANGE reorganization of a table space or an index defined on the table)

  If DB2 adds more partitions than the value specified for MAXNEWPARTS, REORG PLUS terminates and issues message BMC53025E.
Basic REORG PLUS installation options

You can specify any integer from 0 through 4096 for MAXNEWPARTS. However, BMC recommends that you specify only the number of new partitions that you expect the table space to require. A smaller MAXNEWPARTS value helps you to avoid performance problems and reduces the number of additional partition-level data sets that you must allocate. For information about data set requirements, see “Copy data sets” on page 352 and “SYSREC data sets” on page 379.

REORG PLUS ignores this option when either of the following conditions exists:

- REORG PLUS is invoking DSNUTILB.
- You are reorganizing a catalog object.

Overriding this option
You can override the value for this option by using the MAXNEWPARTS command option (page 177).

MAXRO=300

This option applies to SHRLEVEL CHANGE only.

The MAXRO option specifies the maximum number of seconds that REORG PLUS spends applying log records in the LOGFINAL phase. This estimate does not include the time spent producing incremental image copies and running the UTILTERM phase. Specify the value as a positive integer (0 or greater). When the estimated number of seconds required to apply the remaining log records is less than this value, REORG PLUS ends the LOGAPPLY phase and begins the LOGFINAL phase.

Alternatively, you can specify MAXRO=DEFER, which tells REORG PLUS to continue applying log records indefinitely. The LOGFINAL phase will not begin unless one of the following conditions occurs:

- You specify a value for the DEADLINE option and the value is reached.
- You change DEFER to a number of seconds using the XBM Utility Monitor or the MVS operator console.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 633.

Overriding this option
You can override the value for this option by using the MAXRO command option (page 326).
MAXSORTMEMORY=0

The MAXSORTMEMORY option specifies the maximum amount of memory, in kilobytes, that REORG PLUS can allocate to each sort task. The shipped value is 0, which tells REORG PLUS to automatically compute the maximum amount of memory that is needed to perform each sort task. BMCSORT might increase the amount of memory that REORG PLUS allocates if necessary, to successfully complete the sort (if additional memory is available).

In addition to 0, you can specify any number of kilobytes between the value that you specify for the MINSORTMEMORY installation or command option and 2097152 (see page 197 or page 712).

--- NOTE ---
BMC recommends that you use a value of 0.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the MAXSORTMEMORY command option (page 197).

MAXTAPE=3

The MAXTAPE option specifies the maximum number of tape devices to dynamically allocate at one time. The value must be greater than 0.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the MAXTAPE command option (page 293).
Basic REORG PLUS installation options

MGEXTENT=CONTINUE

The MGEXTENT option specifies how to allocate the extents of a DB2 object when creating a new data set during extend processing. Specify one of the following options:

- CONTINUE tells REORG PLUS to allocate the extents as follows:
  
  - Allocate the primary extent with the greater of the original primary quantity or the last secondary quantity of the previous data set.
  
  - Allocate the first secondary extent with the last secondary quantity of the previous data set.

- RESET tells REORG PLUS to allocate the primary and first secondary extents by using the original values from the DB2 object allocation.

With the exception of this option, REORG PLUS allocates secondary extents in the same way that DB2 does. REORG PLUS uses a sliding-scale calculation for secondary extents, if applicable, that is similar to the method that DB2 uses. For more information, see the DB2 documentation that describes the sliding-scale algorithm.

MINSORTMEMORY=0

The MINSORTMEMORY option specifies the minimum amount of memory, in kilobytes, that REORG PLUS should allocate to each sort task. The shipped and BMC-recommended value is 0, which tell REORG PLUS to automatically compute the minimum amount of memory that is needed to optimally perform each sort task. In addition to 0, you can specify any number of kilobytes between 1024 and the value that you specify for the MAXSORTMEMORY installation or command option (page 197 and page 711).

For information about how this option interacts with the SMCORE installation option, see SMCORE on page 724.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**

You can override the value for this option by using the MINSORTMEMORY command option (page 197).

MSGLEVEL=1

The MSGLEVEL option controls which messages are returned to the user in the SYSPRINT and SYSPRIN2 data sets. MSGLEVEL=0 returns minimal messages. MSGLEVEL=1 returns additional messages to help you diagnose problems and fine-tune performance.
**Overriding this option**
You can override the value for this option by using the MSGLEVEL parameter on the EXEC statement (page 347).

**OFFPOSLM=10**

This option does not apply to an index-only reorganization.

If the value of the CONDEXEC command or installation option is YES, REORG PLUS issues a report that recommends which objects in a table space should be reorganized. REORG PLUS then uses the value in the OFFPOSLM option to conditionally perform the reorganization. Valid values are 0 through 100, or NONE.

For every table in the table space named in the REORG command, REORG PLUS performs the following calculation. The calculation applies to the specified partitions in SYSIBM.SYSINDEXPART for the table’s explicit clustering index.

\[(\text{NEAROFFPOSF} + \text{FAROFFPOSF}) \times 100 / \text{CARDF}=integer\]

If any calculated integer value exceeds the OFFPOSLM value, REORG PLUS reorganizes the object.

If the value of the CONDEXEC command or installation option is YES and you specify NONE in the OFFPOSLM installation option, REORG PLUS performs a conditional reorganization based on the values of the other limit options but not the value of OFFPOSLM.

**Overriding this option**
You can override the value for this option by using the OFFPOSLIMIT command option (page 231). If you specify the OFFPOSLIMIT command option with no value, REORG PLUS uses the value in this installation option as the value for that command option. For more information about the interaction between the limit installation and command options, see “Conditional reorganization” on page 136.

---

**TIP**
You can obtain the report that recommends objects for reorganization without performing any reorganizations. To do so, specify REPORTONLY in the command.

**OPNDB2ID=YES**

The OPNDB2ID option tells REORG PLUS whether to use the DB2 RACF ID or the user’s RACF ID.

- YES tells REORG PLUS to use the DB2 RACF ID (instead of the RACF ID of the user running REORG PLUS) when opening or performing Access Method Services (AMS) functions on DB2 data sets.
Basic REORG PLUS installation options

- **NO** tells REORG PLUS to use the RACF ID of the user running REORG PLUS. If you specify NO, the user must have the appropriate RACF authority.

For any security system other than RACF, specify OPNDB2ID=NO to have REORG PLUS use the security authorization ID of the user who is running REORG PLUS.

For more information, see “Required authorization” on page 66.

**TIP**

Using OPNDB2ID=NO can improve performance depending on the size of your data set profiles and the number of VSAM data sets that are involved in this reorganization.

**ORIGDISP=DELETE**

The ORIGDISP option specifies the action that REORG PLUS should take after it has successfully renamed the staging data sets and completed a SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganization:

- **DELETE** tells REORG PLUS to delete the original data sets.
- **RENAME** tells REORG PLUS to rename the original data sets to the staging data set names.

RENAME enables you to preserve the space that was initially allocated for the original data sets by renaming them to the staging data set names. As a result, the staging data sets are ready to be used in a subsequent reorganization.

REORG PLUS ignores a value of RENAME if FASTSWITCH YES is in effect. In this case, no rename is needed. The data sets keep their original names, and the space is preserved for use in a subsequent reorganization.

For information about the naming conventions and a description of how the names are changed, see “Staging data sets” on page 101.

When invoking DSNUTILB, REORG PLUS ignores this option.

*Overriding this option*

You can override the value for this option by using the ORIGINALDISP command option (page 217).
**PENDDDL=** **YES**

The PENDDDL option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces or indexes that contain pending DDL changes. If the IBM DB2 REORG utility would not materialize the pending changes, REORG PLUS ignores this option and reorganizes the object natively. For information about the conditions under which the DB2 REORG utility would not materialize the pending changes, see the documentation for the DB2 REORG utilities.

You can specify one of the following options:

- **YES** tells REORG PLUS to invoke DSNUTILB when the following conditions exist:
  - REORG PLUS encounters a table, table space, or index that contains pending DDL changes.
  - The DB2 REORG utility would materialize the changes.

  To enable this feature, DSNUTILB YES must also be in effect.

  If the pending change affects a LOB object or an inline LOB column, specify LOB=YES.

- **NO** tells REORG PLUS not to invoke DSNUTILB. REORG PLUS terminates.

For more information about running a DSNUTILB job, see “Reorganization jobs that invoke DSNUTILB” on page 72.

**PLAN=** **ARU1110**

The PLAN option specifies the name of the product plan. This plan contains the packages that provide the various capabilities of REORG PLUS.

**PREFORMAT=** **NO**

The PREFORMAT option tells REORG PLUS whether to preformat unused data set pages.

- **NO** tells REORG PLUS not to preformat unused pages in a data set.
- **YES** tells REORG PLUS to write full pages initialized with zeros up to the high-allocated RBA of the table space and index spaces that it just reorganized.

**Considerations**

The following considerations apply to the PREFORMAT option:

- When reorganizing a catalog object, REORG PLUS changes PREFORMAT YES to PREFORMAT NO.
When invoking DSNUTILB, REORG PLUS passes PREFORMAT=YES to the IBM DB2 REORG utility as PREFORMAT and ignores PREFORMAT=NO.

**Overriding this option**
You can override the value for this option by using the PREFORMAT command option (page 227).

**RCVICDDN=(BIRY, BIRZ)**

This option applies to SHRLEVEL CHANGE only.

The RCVICDDN option specifies the ddnames or ddname prefixes for the remote copy data sets that receive an incremental image copy of the table space or partitions that you are reorganizing. The first name is the remote primary, and the second name is the remote backup. You must define the corresponding incremental data set for each copy data set that you specify if the value of ICTYPE is INCREMENTAL.

When using this name as a prefix for a ddname for a partitioned table space, ensure that you allow sufficient bytes for the partition number to be added and still have a valid ddname (eight bytes or less). In addition, if you change the default ddnames that BMC supplies, you must also change the names in your JCL. For more information, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 653.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**
You can override the value for this option by using the RECOVERYICDDN command option (page 277).

**RCVYDDN=(BRCY, BRCZ)**

The RCVYDDN option specifies default ddnames or ddname prefixes for remote copy data sets. These are the data sets that receive an image copy or a DSN1COPY-type copy of the table space or partitions that you are reorganizing. If you are registering the copies, the first name is the remote primary copy, and the second name is the remote backup copy.

When using this name as a prefix, ensure that you allow sufficient bytes for the partition number to be added and still have a valid ddname (eight bytes or less). In addition, if you change the default ddnames, you must change the names in your JCL. For more information, see “Copy data sets” on page 352.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

**Overriding this option**
You can override the value for this option by using the RECOVERYDDN command option (page 273).
Basic REORG PLUS installation options

REDEFINE=YES

The REDEFINE option tells REORG PLUS whether to delete and redefine the VSAM data sets for the table space or index space as part of the reorganization. REORG PLUS can delete and redefine both user-defined (VCAT-defined) data sets and data sets defined in DB2 storage groups (storage-group-defined).

- For SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY, REDEFINE=Yes tells REORG PLUS to delete and redefine the VSAM data sets for the space before reloading it. For SHRLEVEL REFERENCE or SHRLEVEL CHANGE, REORG PLUS deletes and defines the staging data sets.

- REDEFINE=NO tells REORG PLUS not to delete and redefine the existing VSAM data sets for the table space or indexes. Instead, REORG PLUS issues message BMC50391I, reuses the existing data sets, and resets the high-used relative byte address (HURBA).

Additional considerations
The following considerations apply to the REDEFINE option:

- If you are reorganizing a large number of partitions, consider specifying REDEFINE NO. This value minimizes the time that REORG PLUS requires to delete and redefine the existing VSAM data sets for the table space or indexes.

- When invoking DSNUTILB, REORG PLUS passes REDEFINE=NO to the IBM DB2 REORG utility as REUSE and ignores REDEFINE=YES.

Overriding this option
You can override the value for this option by using the REDEFINE command option (page 210). The command section also contains information about using the REDEFINE option with the DSRSEXIT user exit, the SYSIDCIN data set, and the UNLOAD command.

RENMMAX=30

The RENMMAX option indicates the maximum number of tasks to start during the rename and delete process for SHRLEVEL REFERENCE and SHRLEVEL CHANGE. You can specify any of the following values:

- 0 to have REORG PLUS determine the number of tasks to start

- asterisk (*) to use the TASKMAX value (page 730)

- $n$ to specify that REORG PLUS can start a maximum of $n$ tasks (where $n$ is a positive integer from 1 through 32767)
Basic REORG PLUS installation options

- \( n\% \) to specify that the maximum number of tasks that REORG PLUS can start is \( n\% \) of the number of CPUs on the system (where \( n \) is a positive integer from 1 through 32768)

When invoking DSNUTILB, REORG PLUS ignores this option.

For information about using the multitasking options, see “Multitasking installation options” on page 584.

**RIDMDSSZ=2097152**

*This option applies to SHRLEVEL CHANGE only.*

The RIDMDSSZ option specifies the maximum size (in kilobytes) for each data space that REORG PLUS uses to store the RID maps. The number must be greater than or equal to 20480 KB (20 megabytes) and not greater than 2097152 KB (2 gigabytes).

**RIDMMAXD=1**

*This option applies to SHRLEVEL CHANGE only.*

The RIDMMAXD option specifies the maximum number of data spaces that REORG PLUS can use to store the RID maps. The number must be a nonzero positive integer.

BMC recommends that you use as few data spaces as possible for RIDMMAXD, increasing instead the size of the data space (RIDMDSSZ), especially when spilling occurs. If you specify too many data spaces in RIDMMAXD, performance problems and system degradation can occur.

**RMAPMEM=0**

*This option applies to SHRLEVEL CHANGE only.*

The RMAPMEM option specifies the amount of data space memory (in kilobytes) that REORG PLUS can use to store the RID translation maps. The number must be 0 or a positive integer.

BMC strongly recommends that you use the value 0. The value 0 tells REORG PLUS to automatically calculate the RMAPMEM value for you. To determine the value, REORG PLUS multiplies the value of the installation option RIDMMAXD by the value of the installation option RIDMDSSZ. The result is the maximum amount of storage that REORG PLUS can allocate. However, REORG PLUS uses only as much memory as needed to hold the RID map, up to the calculated value.

*Restriction*
When invoking DSNUTILB, REORG PLUS ignores this option.
**Additional considerations**
The following considerations apply to the RMAPMEM option:

- If you specify a nonzero value for RMAPMEM, during execution REORG PLUS first ensures that your specified value is sufficient to satisfy the minimum storage requirements. If the value is not sufficient, REORG PLUS issues message BMC50887I and changes the value for RMAPMEM to the minimum required storage amount.

- To calculate the RMAPMEM value, use the formula described in “Using formulas for the calculations” on page 661. However, if you specify a value greater than the result of multiplying the RIDMMAXD value by the RIDMDSSZ value, REORG PLUS reduces the value that you specify to the product of those two installation options.

- During execution, REORG PLUS allocates memory only as needed, up to the current value of RMAPMEM. If RMAPMEM is insufficient, REORG PLUS issues message BMC50885I and spills to the spill data set.

**Overriding this option**
You can override the value for this option by using the RIDMAPMEM command option (page 336).

**RORGMAX=300%**
The RORGMAX option indicates the maximum number of tasks to start for the REORG phase. You can specify any of the following values:

- 0 to have REORG PLUS determine the number of tasks to start

- asterisk (*) to use the TASKMAX value (page 730)

- \( n \) to specify that REORG PLUS can start a maximum of \( n \) tasks (where \( n \) is a positive integer from 1 through 32767)

- \( n\% \) to specify that the maximum number of tasks that REORG PLUS can start is \( n\% \) of the number of CPUs on the system (where \( n \) is a positive integer from 1 through 32768)

Regardless of the value that you specify, REORG PLUS will not start more than 16 tasks for this phase.

**NOTE**
For REORG PLUS to use RORGMAX, you must specify SMAX=0, and either not use MAXSORTS or specify MAXSORTS 0.
For information about using the multitasking options, see “Multitasking installation options” on page 584.

**ROUTCDE=(11,1)**

The ROUTCDE option specifies the routing codes to route write-to-operator (WTO) messages to the designated console. The first subparameter is the code for informational WTOs. The second subparameter is the code for WTOs that require an action. The values (11,1) route informational WTOs to programmer information (11) and WTOs requiring an action to the master console (1).

For a complete listing of valid values, see the appropriate IBM reference manual.

**SCPYMAX=8**

The SCPYMAX option indicates the maximum number of tasks to start during the nonpartitioned index copy process for a SHRLEVEL REFERENCE or SHRLEVEL CHANGE partial reorganization. You can specify any of the following values:

- 0 to have REORG PLUS determine the number of tasks to start
- asterisk (*) to use the TASKMAX value (page 730)
- n to specify that REORG PLUS can start a maximum of n tasks (where n is a positive integer from 1 through 32767)
- n% to specify that the maximum number of tasks that REORG PLUS can start is n% of the number of CPUs on the system (where n is a positive integer from 1 through 32768)

For information about using the multitasking options, see “Multitasking installation options” on page 584.

**SDUMP=(ALLPSA,CSA,RGN,SQA,LSQA,SUM,TRT,IO)**

The SDUMP option tells REORG PLUS to generate a system dump, using the information listed in the option, if the job abnormally terminates. The values listed provide diagnostic information to BMC Customer Support. Although you can specify any values that IBM allows for the SDATA parameter on the SDUMPX macro, BMC recommends that you do not change the values that were shipped with the product. For a complete list and description of the values that you can specify for this option, see the IBM SDUMPX macro description.

REORG PLUS uses the system-defined dump data set to hold the data. In cases where multiple abends occur, REORG PLUS generates the dump for only the first abend. The SDUMP option allows you to generate a system dump, regardless of your access to storage keys 0 through 7.
You can also specify SDUMP=NO (without parentheses) to tell REORG PLUS not to generate a system dump.

**NOTE**
REORG PLUS displays SDUMP=YES in message BMC50471I in the SYSPRINT list of options when SDUMP contains one or more values.

To limit the conditions under which REORG PLUS generates the system dump, you can exclude selected abend codes by using the EXCLDUMP option (page 690).

**SHORTMEMORY=CONTINUE**

The SHORTMEMORY option controls the action that REORG PLUS takes when one of the following memory shortages exists during sort processing:

- The system contains insufficient available pages of memory for REORG PLUS to perform an optimal sort
- The region contains insufficient memory for REORG PLUS to perform a minimum number of tasks, given the amount of memory required to perform an optimal sort. However, the region *does* contain at least 1024 KB of memory, or the minimum amount of memory specified by the MINSORTMEMORY option. If the region contain at least 1024 KB of memory *and* the amount of memory specified by the MINSORTMEMORY option is available, REORG PLUS uses the greater amount of memory.

You can specify one of the following values for the SHORTMEMORY option:

- CONTINUE, the shipped value, indicates that, when a memory shortage exists, REORG PLUS should issue message BMC50364I and continue sort processing.
- FAIL indicates that REORG PLUS should fail when a memory shortage exists.

Be aware when specifying FAIL that sufficient memory might exist to sort during the UNLOAD phase of a two-phase reorganization or the REORG phase of a single-phase reorganization. However, because of other system conditions, insufficient available pages might exist during the index build process, which occurs during the RELOAD process (two-phase reorganization) or following the REORG process (single-phase reorganization).
Table 100 shows the relationship between SHORTMEMORY values and the following conditions:

- memory data obtained from the system
- memory in the region
- value specified for the SMCOORE installation option (page 724)
- value specified for the MINSORTMEMORY installation or command option (page 712 and page 197)

<table>
<thead>
<tr>
<th>Location</th>
<th>Condition</th>
<th>SHORTMEMORY value</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory in the system</td>
<td>insufficient to run one optimal sort task based on the amount of data to be sorted</td>
<td>REORG PLUS runs one task with 1024 KB of memory or the amount of memory that you specified with MINSORTMEMORY, whichever is greater.</td>
</tr>
<tr>
<td></td>
<td>insufficient as specified by MINSORTMEMORY</td>
<td>REORG PLUS fails.</td>
</tr>
<tr>
<td>virtual memory in the region</td>
<td>insufficient to run one optimal sort task based on the amount of data to be sorted but sufficient as specified by MINSORTMEMORY or at least 1024 KB, whichever is greater</td>
<td>REORG PLUS runs one task with the available memory.</td>
</tr>
<tr>
<td></td>
<td>insufficient as specified by MINSORTMEMORY or less than 1024 KB, whichever is greater</td>
<td>REORG PLUS fails.</td>
</tr>
<tr>
<td></td>
<td>insufficient as specified by the first parameter of SMCOORE</td>
<td></td>
</tr>
</tbody>
</table>

When invoking DSNUTILB, REORG PLUS ignores this option.

*Overriding this option*

You can override the value for this option by using the SHORTMEMORY command option (page 195).
SIXSNAP=NO

This option applies to a SHRLEVEL REFERENCE or SHRLEVEL CHANGE partial table space reorganization only.

The SIXSNAP option determines whether REORG PLUS uses the Instant Snapshot technology of the EXTENDED BUFFER MANAGER (XBM) product or the SNAPSHOT UPGRADE FEATURE (SUF) of XBM to create a copy of storage-group-defined nonpartitioned indexes. Using Instant Snapshot can improve performance because Instant Snapshot makes a hardware-based copy of the entire index at one time. To use Instant Snapshot, you must have the supported intelligent storage devices available.

- NO tells REORG PLUS to use the software-based copy method to copy each nonpartitioned index.

- AUTO tells REORG PLUS to use Instant Snapshot.

  If the Instant Snapshot copy fails, REORG PLUS copies the index using the software copy method.

- YES tells REORG PLUS to use only Instant Snapshot technology.

  If the Instant Snapshot copy fails, REORG PLUS terminates.

Although using SIXSNAP can improve the performance of the reorganization, SIXSNAP can cause the application to time out. The order of volumes allocated can differ from the order in your list, depending on the hardware vendor. For more information, see “Considerations for SIXSNAP use” on page 124.

For detailed information about the SIXSNAP function, see “Instant Snapshot with nonpartitioned indexes” on page 122. For a list of the supported devices, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the SIXSNAP command option (page 251).

SMAX=0

The SMAX option specifies the number of sort tasks that can run concurrently during a reorganization. If sorting is required to perform the reorganization, SMAX also limits the number of concurrent sorts. Limiting the number of concurrent sorts can have a significant impact on performance because sorting requires a substantial amount of system resources.
If you specify a value for SMAX, REORG PLUS starts only one task per CPU. If you want to improve performance by starting more than one task per CPU, use the default of SMAX=0, and use the multitasking options. For more information about using the multitasking options, see “Multitasking installation options” on page 584.

For information about the hierarchy of the SMAX, MAXSORTS, and multitasking options, see Table 90 on page 586. For more information, see “Controlling the number of sort processes” on page 583.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**
You can override the value for this option by using the MAXSORTS command option (page 189).

**SMCORE=(0K,0K)**

The SMCORE option specifies the amount of memory that you want each invocation of BMCSORT to use. BMC strongly recommends that you use the values 0K and 0K for this option. The values 0K and 0K indicate that REORG PLUS is to determine the appropriate amount of memory to use for each sort process. However, other valid values are:

- for the first parameter, 4096K through 65536K (or 0M or 4M through 64M)

  The first value specifies the total amount of memory to use both above and below the 16-megabyte line for each sort.

- for the second parameter, 256K through 4096K (or 0M or 1M through 4M)

  The second value specifies the amount of memory to use below the 16-megabyte line for each sort.

When invoking DSNUTILB, REORG PLUS ignores this option.

For more information about how REORG PLUS uses SMCORE and the other sort optimization options, see “Controlling memory usage” on page 581.

**SORTDEVT=(,SYSALLDA)**

The SORTDEVT option specifies the device type for the sort work files that are allocated dynamically.

The first parameter of this option is the device type to use for non-DSNUTILB jobs. This parameter overrides the first parameter of the BMCSORT DYNALOC installation option. If the value of the third parameter in the BMCSORT DYNALOC installation option is OFF, specifying a value for this parameter turns BMCSORT dynamic allocation on.
The second parameter of this option is the device type to use for DSNUTILB jobs. When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

**Overriding this option**
You can override the values for both parameters of this option by using the SORTDEVT command option (page 187).

### SORTNUM=32

The SORTNUM option affects the allocation of sort work files in the following cases. The shipped value is 32, and you can specify any integer value from 0 through 255.

**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS passes this value to the IBM DB2 REORG utility as the number of sort work files to allocate dynamically. For this type of reorganization, the value must be 2 or greater.

**All other reorganization jobs**
This value is in effect when BMCSORT is allocating your sort work files dynamically. Table 101 describes the action that BMCSORT takes for each value that you can specify for this option. The table also provides any additional considerations for these values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Additional considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>BMCSORT honors the value of the third parameter of the BMCSORT DYNALOC installation option. (This parameter tells BMCSORT whether to dynamically allocate sort work files.)</td>
<td>For more information about how this parameter affects dynamic allocation, see “Dynamically allocating SORTWK data sets” on page 367. For more information about the parameter itself, see “DYNALOC installation option” on page 761.</td>
</tr>
<tr>
<td>1–32</td>
<td>BMCSORT dynamically allocates the number of sort work files that it needs, up to 32 minus any preallocated sort work files. This number is per sort task.</td>
<td>Preallocated sort work files include sort work files that are allocated in your JCL and any sort work files that REORG PLUS dynamically allocates.</td>
</tr>
<tr>
<td>33–255</td>
<td>BMCSORT dynamically allocates the number of sort work files that it needs, up to the number that you specified minus any preallocated sort work files. This number is per sort task.</td>
<td></td>
</tr>
</tbody>
</table>

If the value of the third parameter in the BMCSORT DYNALOC installation option is OFF, specifying a value greater than 0 for the SORTNUM option turns BMCSORT dynamic allocation on and BMCSORT allocates sort work files as needed. For information about when BMCSORT allocates your sort work files dynamically, see the “SORTWK data sets” on page 366.
Overriding this option
You can override the value for this option with the SORTNUM command option (page 188).

SPIILDSNP=&&UID

This option applies to SHRLEVEL CHANGE only.

The SPIILDSNP option tells REORG PLUS to use a particular pattern of variables and text to create a prefix for the spill data set names. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates the spill data sets as needed. The spill data sets are VSAM objects.

The pattern must resolve to a prefix that is 22 bytes or less and does not end in a period. You can use text or any of the symbolic variables listed in Table 102 to construct your pattern. You can also provide user-defined variables from a user exit (specified with the DSNUEXIT installation or command option). When specifying a pattern in your installation options, you must precede each REORG variable with an additional ampersand (&) in your pattern.

REORG PLUS removes any trailing blanks in the result.

Table 102 Symbolic variables for the SPIILDSNP installation option (part 1 of 2)

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;DATE</td>
<td>current date (in the form MMDDYY)</td>
<td>6 bytes</td>
</tr>
<tr>
<td>&amp;DATEJ</td>
<td>current Julian date (in the form YYYYDDD)</td>
<td>7 bytes</td>
</tr>
<tr>
<td>&amp;DB</td>
<td>database containing the space for this data set allocation</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;GRPNM</td>
<td>DB2 data-sharing group name</td>
<td>4 bytes</td>
</tr>
<tr>
<td></td>
<td>In a non-data-sharing environment, GRPNM contains the DB2 SSID.</td>
<td></td>
</tr>
<tr>
<td>&amp;JDATE</td>
<td>current Julian date (in the form YYDDD)</td>
<td>5 bytes</td>
</tr>
<tr>
<td>&amp;JOBNAME</td>
<td>JOB name in the JCL</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;RTYPE</td>
<td>REORG type (TS or IX)</td>
<td>2 bytes maximum</td>
</tr>
<tr>
<td>&amp;SSID</td>
<td>DB2 subsystem ID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>&amp;STEPNAME</td>
<td>STEP name used in the JCL</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS ignores PROC names.</td>
<td></td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>current time (in the form HHMMSS)</td>
<td>6 bytes</td>
</tr>
<tr>
<td>&amp;TIME4</td>
<td>current time (in the form HHMM)</td>
<td>4 bytes</td>
</tr>
<tr>
<td>&amp;TSIX</td>
<td>table space or index space specified in your REORG command</td>
<td>8 bytes maximum</td>
</tr>
</tbody>
</table>
For more information and guidelines for specifying data set name patterns, see page 331. User exits are discussed in Appendix C, “REORG PLUS user exits.”

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**
You can override the value for this option by using the SPILLDSNPAT command option (page 331).

**SPILSCLS=NONE**

This option applies to SHRLEVEL CHANGE only.

The SPILSCLS option specifies the SMS storage class that REORG PLUS uses to allocate spill data sets. You can specify a valid SMS storage class name not exceeding eight characters, or NONE. The spill data sets are VSAM objects.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**
You can override the value for this option by using the SPILLSTORCLAS command option (page 331).

---

**Table 102  Symbolic variables for the SPILDSNP installation option (part 2 of 2)**

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of result</th>
</tr>
</thead>
</table>
| &USERID or &UID   | job user ID  
You must have a security package to use the job user ID variable. | 8 bytes maximum            |
| &UTIL             | BMC utility ID  
REORG PLUS truncates longer utility IDs to eight characters. | 8 bytes maximum            |
| &UTILPFX          | BMC utility ID prefix  | 8 bytes maximum            |
| &UTILSFX          | BMC utility ID suffix | 8 bytes maximum            |
| &VCAT             | VCATNAME specified in the DB2 catalog for the table space that you are reorganizing; or, if the table space is partitioned, the VCAT name from the first partition that you are reorganizing | 8 bytes maximum            |
**SPILUNIT=WORK**

This option applies to SHRLEVEL CHANGE only.

The SPILUNIT option specifies the DASD unit to which REORG PLUS can allocate spill data sets. The unit name must be a valid DASD unit name not exceeding eight characters, or NONE. The spill data sets are VSAM objects.

When invoking DSNUTILB, REORG PLUS ignores this option.

*Overriding this option*

You can override the value for this option by using the SPILLUNIT command option (page 330).

**SQLDELAY=3**

The SQLDELAY option specifies the number of seconds that REORG PLUS should wait between retry attempts after receiving an SQL –911 return code. This interval is in addition to the time that elapses when DB2 waits for a timeout or deadlock. The number of seconds can range from 1 through 655.

**SQLRETRY=100**

The SQLRETRY option specifies the number of times that REORG PLUS should retry an SQL statement after it receives an SQL –911 return code. The number of retry attempts can range from 1 through 255.

**STAGEDSN=BMC**

The STAGEDSN option tells REORG PLUS which naming convention you want it to use for the staging data sets. STAGEDSN=BMC tells REORG PLUS to allocate staging data sets with a second node of BMCDBC and BMCDBD for the cluster and data nodes.

STAGEDSN=DSN tells REORG PLUS to allocate staging data sets with a second node of DSNDBC and DSNDBD and a fifth node of I or J (whichever letter is not in the original data set name).

If FASTSWITCH YES is in effect, REORG PLUS changes STAGEDSN=BMC to STAGEDSN=DSN.

For more information about staging data sets and their naming conventions, see “Staging data sets” on page 101.
STOP@CMT=YES

This option applies to SHRLEVEL NONE and SHRLEVEL REFERENCE only.

The STOP@CMT option specifies whether to add the 'AT (COMMIT)' parameter to all DB2 STOP commands that REORG PLUS issues. YES adds the parameter and NO does not.

Considerations
BMC recommends that you specify the value for STOP@CMT based on your environment and your availability goals. Under certain circumstances, using STOP@CMT=N might result in the object being placed in stop pending status (STOPP) and failure of the reorganization. REORG PLUS issues the following message in this case:

BMC50266E SPACE ‘databaseName.spaceName’ CANNOT BE STOPPED. IT MAY BE IN USE

This situation might result when one of the following conditions exists:

- You are reorganizing one partition of a partitioned table space while an application thread is using other partitions of the same table space.

- You are reorganizing an index while an application thread is using the table space.

By using STOP@CMT=YES, you might avoid this problem, but application threads might be affected. When a thread performs a commit, DB2 might stop the object, although the thread continues to run. For details and implications of the 'AT(COMMIT)' parameter, see the IBM DB2 command reference manual.

STOPDELAY=1

The STOPDELAY option specifies the number of seconds that REORG PLUS waits before it checks again to see if DB2 has stopped the object. The number of seconds can be any nonzero positive integer.

STOPRETRY=300

The STOPRETRY option specifies the number of times that REORG PLUS checks to see if DB2 has changed the status of an object from stop pending (STOPP) to stopped (STOP). The number of checks can be any nonzero positive integer.

TAPEDISP=DELETE

The TAPEDISP option specifies the final disposition of tape data sets when you specify YES for the DELFILES installation or DELETFILES command option. TAPEDISP=DELETE specifies that each tape file will have a disposition of OLD,DELETE,DELETE when the tape is deallocated.
If you specify TAPEDISP=UNCATLG, each tape file will have a disposition of OLD,UNCATLG,UNCATLG when the tape is deallocated. Depending on your tape management environment, using UNCATLG can prevent a tape remount.

**TASKMAX=1000%**

The TASKMAX option sets the default for all of the other multitasking options. TASKMAX does not directly provide the number of tasks to use. Instead, TASKMAX provides a default value for any option that refers to it. You can specify any of the following values:

- 0 to specify that REORG PLUS determines the number of tasks to start
- $n$ to specify that REORG PLUS can start a maximum of $n$ tasks (where $n$ is a positive integer from 1 through 32767)
- $n\%$ to specify that the maximum number of tasks that REORG PLUS can start is $n\%$ of the number of online CPUs on the system (where $n$ is a positive integer from 1 through 32768)

For more information about using the multitasking options, see “Multitasking installation options” on page 584.

**TEMPRALDATA (obsolete)**

In REORG PLUS version 10.1, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces or indexes for certain types of temporal data.

Beginning with version 10.2, REORG PLUS reorganizes these objects natively and, therefore, no longer uses this option. If your installation options module contains this option, the module compiles successfully but completes with return code 4 and a message indicating that this option is obsolete.

**TERMEXIT=(NONE,REXX)**

The TERMEXIT option allows you to specify the name of a user-written exit that gives you dynamic control over several options at termination time. Use the TERMEXIT user exit to dynamically control processing of updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS. This exit must be written in REXX.

For details about the TERMEXIT user exit, see Appendix C, “REORG PLUS user exits.”
Restrictions
The following restrictions apply to the TERMEXIT option and to using a TERMEXIT user exit:

- When invoking DSNUTILB, REORG PLUS ignores this option.

- You cannot use a user exit supplied by the TERMEXIT option to override BMCSTATS NO or UPDATEDB2STATS NO to YES.

Overriding this option
You can override the value for this option by using the TERMEXIT command option (page 242).

TIMEOUT=TERM

The TIMEOUT option specifies the action that REORG PLUS should take after it has exhausted all retry attempts to obtain a drain.

- If you specify TIMEOUT=TERM (or TERM, rc) and a timeout condition occurs, REORG PLUS
  - issues messages BMC50020I and BMC50285E
  - leaves the objects in their original state
  - terminates the utility
  - (non-DSNUTILB reorganizations only) ends with the return code you specified
  - When you specify an integer of 31 or less, REORG PLUS issues the return code.
  - When you specify an integer that is 32 or greater, REORG PLUS issues a user abend that is equal to the specified integer.
  - deletes the work files and unregistered copy data sets if you set the value for the DELFILES installation option to YES, or you specify DELETEFILES YES on the command

- If you specify TIMEOUT=ABEND and a timeout condition occurs, REORG PLUS
  - abnormally ends (abends) with user code 3200
  - leaves the objects in their original state
  - leaves an entry in the BMCUTIL table

If this condition occurs when you are running a SHRLEVEL CHANGE reorganization, you must resubmit the job with TERM instead of NEW on the EXEC statement.
Basic REORG PLUS installation options

The value of the return code depends on the value that you specified for the TIMEOUT installation option and the ON FAILURE command option, as described in Table 103.

Table 103 Return code hierarchy for the TIMEOUT installation option

<table>
<thead>
<tr>
<th>TIMEOUT installation option with TERM</th>
<th>ON FAILURE with RETCODE specified</th>
<th>Return code is returned from</th>
</tr>
</thead>
<tbody>
<tr>
<td>TERM</td>
<td>yes</td>
<td>ON FAILURE: return code that you specified with RETCODE</td>
</tr>
<tr>
<td>TERM</td>
<td>no</td>
<td>TIMEOUT: return code 8</td>
</tr>
<tr>
<td>TERM, rc</td>
<td>does not matter</td>
<td>TIMEOUT: return code that you specified with TERM</td>
</tr>
<tr>
<td>no value specified</td>
<td>no</td>
<td>REORG: return code 8</td>
</tr>
</tbody>
</table>

Overriding this option

You can override the value for this option by specifying the TIMEOUT command option (page 238).

TOTALPAGEPCT=0

The TOTALPAGEPCT option controls memory above the 16-MB line that REORG PLUS allocates to BMCSORT for concurrent sort processing. TOTALPAGEPCT specifies the maximum percentage of total 4-KB pages, as obtained from the system, that REORG PLUS can allocate. You can specify any integer from 0 through 100.

NOTE

REORG PLUS defines total pages as pages that are underutilized and are available for use. Available pages (which you can control with the AVAILPAGEPCT option), are pages that have not been used.

A value of 0 tells REORG PLUS to ignore the number of total pages when allocating sort memory.

A value of 1 through 100 tells REORG PLUS to use up to the specified percentage of total pages when allocating sort memory. For example, TOTALPAGEPCT 50 tells REORG PLUS to use no more than 50 percent of the total pages.

Additional considerations

The following additional information applies to the TOTALPAGEPCT option:

- When you specify values greater than 0 for both TOTALPAGEPCT and AVAILPAGEPCT, REORG PLUS uses the lesser of the two calculated results as the maximum amount of memory for sort processing.
If REORG PLUS is unable to start any tasks because of restraints on sort memory caused by a low number of total or available pages, the SHORTMEMORY installation option (page 721) or command option controls the action that REORG PLUS takes.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**
You can override the value for this option by specifying the TOTALPAGEPCT command option (page 194).

**TSPREC (obsolete)**

In REORG PLUS version 10.1, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces that contain a timestamp column that is defined with a precision (number of microseconds) other than 6.

Beginning with version 10.2, REORG PLUS reorganizes these objects natively and, therefore, no longer uses this option. If your installation options module contains this option, the module compiles successfully but completes with return code 4 and a message indicating that this option is obsolete.

**TSSAMPLEPCT=100**

This option applies to a table space reorganization only.

The TSSAMPLEPCT option enables you to specify a percentage of table space pages that you want REORG PLUS to sample when gathering statistics. The following values are valid:

- 1 through 50 tells REORG PLUS to sample the specified percentage of the table space pages.

- 100 tells REORG PLUS to read all table space pages instead of sampling.

**NOTE**
Values 51 through 99 are not valid.

*Restrictions*
REORG PLUS ignores the TSSAMPLEPCT option for either of the following types of reorganizations:

- a DSNUTILB reorganization
- an index reorganization
Overriding this option

You can override the value for this option by using the TSSAMPLEPCT command option (page 263).

TSTZ (obsolete)

In REORG PLUS version 10.1, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces that contain a timestamp column that is defined as TIMESTAMP WITH TIME ZONE.

Beginning with version 10.2, REORG PLUS reorganizes these objects natively and, therefore, no longer uses this option. If your installation options module contains this option, the module compiles successfully but completes with return code 4 and a message indicating that this option is obsolete.

UBUFFS=20

The UBUFFS option specifies a multiple of 32 KB to use to define the amount of buffer pool storage that is allocated for the unload (SYSREC) data sets. For more information, see “SYSREC data set” on page 575.

UNLDDN=SYSREC

The UNLDDN option specifies the default ddname or ddname prefix for the output data set that contains the unloaded rows to be reorganized.

When using this name as a prefix for a ddname for a partitioned table space, ensure that you allow sufficient bytes for the partition number to be added and still have a valid ddname (eight bytes or less). In addition, if you change the default ddname that BMC supplies, you must also change the name in your JCL. For information about specifying and using this data set, see “SYSREC data sets” on page 379.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS uses this option only to pass a value for the &DDNAME variable for the data set name pattern. REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

Overriding this option

You can override the value for this option by using the UNLDDN command option (page 182).
UNLDMAX=300%

The UNLDMAX option indicates the maximum number of tasks to start per CPU for the UNLOAD phase. You can specify any of the following values:

- 0 to have REORG PLUS determine the number of tasks to start
- asterisk (*) to use the TASKMAX value (page 730)
- $n$ to specify that REORG PLUS can start a maximum of $n$ tasks (where $n$ is a positive integer from 1 through 32767)
- $n\%$ to specify that the maximum number of tasks that REORG PLUS can start is $n\%$ of the number of CPUs on the system (where $n$ is a positive integer from 1 through 32768)

Regardless of the value that you specify, REORG PLUS will not start more than 16 tasks for this phase.

**NOTE**

For REORG PLUS to use UNLDMAX, you must specify SMAX=0, and either not use MAXSORTS or specify MAXSORTS 0.

For information about using the multitasking options, see “Multitasking installation options” on page 584.

UNLOAD=RELOAD

The UNLOAD option tells REORG PLUS whether to use single-phase or two-phase processing. You can specify either RELOAD or CONTINUE. For more information about these options, see the UNLOAD command option on page 204.

- RELOAD (the default) tells REORG PLUS to use single-phase processing. Single-phase processing is usually much faster than two-phase processing. However, if the job fails, you might not be able to restart the job. For more information, see “Single-phase reorganization” on page 144.

- CONTINUE tells REORG PLUS to continue the reorganization with two-phase processing after the UNLOAD phase has unloaded the data. If the reorganization fails during two-phase processing, you can restart the job.

**NOTE**

REORG PLUS changes UNLOAD CONTINUE to UNLOAD RELOAD when you are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect.
**Basic REORG PLUS installation options**

*Overriding this option*
You can override the value for this option by using the UNLOAD command option (page 204). An additional value is available with the command option.

**UTILB_COLCCSID=UTILB**

The UTILB_COLCCSID option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) a table space that contains columns defined with a CCSID specification:

- UTILB tells REORG PLUS to reorganize this table space by using DSNUTILB processing. To enable this feature, DSNUTILB YES must also be in effect.

  If a participating index contains columns defined with a CCSID specification, you must also specify IXONEX=YES.

- FAIL tells REORG PLUS not to reorganize this table space. REORG PLUS terminates.

For more information about running a DSNUTILB reorganization, see “Reorganization jobs that invoke DSNUTILB” on page 72.

In a future release, REORG PLUS will no longer require or support the UTILB_COLCCSID installation option.

**UTILB_NULLIX=UTILB**

The UTILB_NULLIX option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces that contain indexes that are defined with EXCLUDE NULL KEYS:

- UTILB tells REORG PLUS to use DSNUTILB processing to reorganize a table space that contains this type of index. To enable this feature, DSNUTILB YES must also be in effect.

- FAIL tells REORG PLUS not to reorganize a table space that contains this type of index. If REORG PLUS encounters this type of index, REORG PLUS issues message BMC50920E and terminates.

**NOTE**

REORG PLUS natively reorganizes indexes that are defined with EXCLUDE NULL KEYS.

For more information about running a DSNUTILB reorganization, see “Reorganization jobs that invoke DSNUTILB” on page 72.
In a future release, REORG PLUS will no longer require or support the UTILB_NULLIX installation option.

**UTSMEM (obsolete)**

In REORG PLUS version 10.1, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) universal table spaces that are defined with MEMBER CLUSTER.

Beginning with version 10.2, REORG PLUS reorganizes these table spaces natively and, therefore, no longer uses this option. If your installation options module contains this option, the module compiles successfully but completes with return code 4 and a message indicating that this option is obsolete.

**UXSTATE=SUP**

The UXSTATE option tells REORG PLUS how to invoke DB2 user exits:

- SUP specifies that REORG PLUS should call EDITPROCs in supervisor state (and PSW key=7).
- PROB tells REORG PLUS to call EDITPROCs in problem state and PSW key=7.

The requirements of the exits dictate the UXSTATE setting. Check with the exit author (or vendor) before changing the value of this option.

**WBUFFS=(20,10)**

The WBUFFS option specifies a multiple of 32 KB to use to define the amount of buffer pool storage that is allocated for each work (SYSUT1) data set. REORG PLUS uses the first number if you specify only one work data set. REORG PLUS uses the second number for each work data set if you specify multiple work data sets. For more information, see “SYSUT1 data set” on page 576.

**WORKDDN=SYSUT1**

The WORKDDN option specifies the default ddname or ddname prefix for the index work data set that contains the unloaded index keys and serves as input for BMCSORT.

When using this name as a prefix, allow sufficient bytes for the number of indexes to be added and still have a valid ddname (eight bytes or less). In addition, if you change the default ddname that BMC supplies, you must also change the name in your JCL. For information about specifying and using this data set, see “SYSUT1 data sets” on page 382.
**DSNUTILB reorganization jobs**

When invoking DSNUTILB, REORG PLUS uses this option only to pass a value for the &DDNAME variable for the data set name pattern. REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

**Overriding this option**

You can override the value for this option by using the WORKDDN command option (page 183).

**WORKUNIT=SYSALLDA**

The WORKUNIT option specifies the unit to use for a temporary work data set. VIO is an acceptable value for this option.

**XBMID=**

The XBMID option specifies the XBM subsystem (SSID) that REORG PLUS accesses when it uses XBM or SUF. Specify a value for this option if you want to use a specific XBM subsystem for snapshot processing and zIIP processing in REORG PLUS.

The SSID is the unique identifier that was specified when XBM or SUF was installed. If you are using XBM or SUF in a DB2 data sharing environment, you can use the value of the XBMGROUP parameter instead of the XBM SSID. The XBMGROUP name is the name of the cross-system coupling facility (XCF) group that is defined to the XBM subsystem, and its default value is XBMGROUP.

If you specify an XBM subsystem, it must be:

- available
- at a supported maintenance level
- enabled for the required function

If you do not specify an XBM subsystem (either here or with the XBMID installation option), REORG PLUS automatically searches for an XBM subsystem that meets the same criteria. If you have multiple subsystems that meet these criteria, REORG PLUS can use any one of these subsystems.

For more information about using XBM or SUF with REORG PLUS, see “XBM and SUF considerations” on page 151. For more information about XBM and SUF, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

**Restriction**

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**

You can override the value for this option by using the XBMID command option (page 253).
XML (obsolete)

In REORG PLUS version 10.1, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces that contain XML columns that support XML versions, or associated XML table spaces.

Beginning with version 10.2, REORG PLUS reorganizes these table spaces natively and, therefore, no longer uses this option. If your installation options module contains this option, the module compiles successfully but completes with return code 4 and a message indicating that this option is obsolete.

ZIIP=ENABLED

The ZIIP option tells REORG PLUS whether to attempt to use IBM System z® Integrated Information Processors (zIIPs). REORG PLUS can use enclave service request blocks (SRBs) to enable zIIP processing automatically while running jobs. Using zIIP processing can reduce the overall CPU time for REORG PLUS jobs.

You can specify one of the following values:

- ENABLED tells REORG PLUS to attempt to offload eligible processing to an available zIIP. If the zIIP is busy or not available, normal processing continues on a general-purpose processor.
- DISABLED tells REORG PLUS to not attempt to use zIIP processing.

To enable and use zIIP processing with REORG PLUS, you must

- have an installed authorized version of XBM or SUF
- start and maintain an XBM subsystem in your environment

**NOTE**

You can specify a particular XBM subsystem to use by specifying a value for the XBMID installation or command option. For more information, see “XBMID” on page 253 or page 738.

- have a zIIP available in your environment

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

**Restriction**

When invoking DSNUTILB, REORG PLUS ignores this option.
Dynamic allocation installation options

Overriding this option
You can override the value for this option by using the ZIIP command option (page 252).

Dynamic allocation installation options

The $ARUDYNA macros produce the installation options for dynamic allocation (one macro for each data set type). Table 104 shows the options contained in the $ARUDYNA macros. For each option, the table provides the value that ships with this version of REORG PLUS (or No value if the option is shipped without a value), a brief description, and a reference to more details. If an option ships with no value, the table shows a recommended value or example value.

Your $C30DOPT member must contain a separate macro for each DDTYPE.

<table>
<thead>
<tr>
<th>Option</th>
<th>Shipped value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDTYPE</td>
<td>Each macro has a different default value. For example, the default value for local primary copy data sets is LOCPFCPY.</td>
<td>data set type to which the remaining dynamic allocation options apply</td>
<td>page 741</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>Each DDTYPE has a different default value. For example, the default value for DDTYPE UNLOAD is YES.</td>
<td>whether to dynamically allocate this type</td>
<td>page 742</td>
</tr>
<tr>
<td>ALLOC</td>
<td>ANY</td>
<td>method to use when dynamically allocating sort work files</td>
<td>page 744</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This installation option is valid only for the SORTWORK work file type.</td>
<td></td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
<td>average space available for data on volumes that are used for dynamic allocation</td>
<td>page 744</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>(NONE,NONE)</td>
<td>SMS data class to use</td>
<td>page 745</td>
</tr>
<tr>
<td>DSNPAT</td>
<td>Each DDTYPE has a different default value. For example, the default value for DDTYPE UNLOAD is &amp;&amp;UID.&amp; JOBNAME.&amp; TSIX.&amp;&amp;DDNAME</td>
<td>data set name pattern</td>
<td>page 745</td>
</tr>
<tr>
<td>DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>data set type</td>
<td>page 750</td>
</tr>
<tr>
<td>EXPDT</td>
<td>No value</td>
<td>expiration date for copy and archive data sets</td>
<td>page 751</td>
</tr>
<tr>
<td></td>
<td>example value: 2011087</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GDGEMPTY</td>
<td>NO</td>
<td>whether to uncatalog all data sets when the limit is reached</td>
<td>page 751</td>
</tr>
</tbody>
</table>
### Table 104  REORG PLUS dynamic allocation installation options (part 2 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Shipped value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDGLIMIT</td>
<td>5</td>
<td>number of GDG generations to keep</td>
<td>page 752</td>
</tr>
<tr>
<td>GDGSCRATCH</td>
<td>NO</td>
<td>whether to delete uncataloged data sets</td>
<td>page 752</td>
</tr>
<tr>
<td>IFALLOC</td>
<td>USE</td>
<td>whether to use DDs in the JCL if they are coded</td>
<td>page 752</td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
<td>maximum extent size</td>
<td>page 753</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>SMS management class to use</td>
<td>page 754</td>
</tr>
<tr>
<td>RETPD</td>
<td>No value</td>
<td>number of days to retain copy and archive data sets</td>
<td>page 754</td>
</tr>
<tr>
<td></td>
<td>example value: 30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIZEPCT</td>
<td>■ (5,100) for incremental copy DDTYPEs (LOCPICPY, LOCBICPY, REMPICPY, and REMBICPY)</td>
<td>percentage of the REORG PLUS calculated space to use for allocation</td>
<td>page 755</td>
</tr>
<tr>
<td></td>
<td>■ (100,100) for all other DDTYPEs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>NO</td>
<td>whether to perform SMS allocations</td>
<td>page 755</td>
</tr>
<tr>
<td>SMSUNIT</td>
<td>NO</td>
<td>whether to pass the unit value to SMS</td>
<td>page 756</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>(NONE,NONE)</td>
<td>SMS storage class to use</td>
<td>page 756</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>0</td>
<td>whether to use secondary units, classes, number and size of volumes, and extent size</td>
<td>page 757</td>
</tr>
<tr>
<td>UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>unit names used for dynamic allocation</td>
<td>page 758</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>(0,0)</td>
<td>number of devices to dynamically allocate</td>
<td>page 759</td>
</tr>
<tr>
<td>VOLCNT</td>
<td>(25,25)</td>
<td>largest number of volumes to process</td>
<td>page 760</td>
</tr>
</tbody>
</table>

This section describes each of the dynamic data set allocation options shown in Table 104. You can also specify most of these options with the DDTYPE command option, enabling you to override the defaults established at installation. For more information, see “Dynamic allocation options” on page 293.

**DDTYPE=UNLOAD**

The DDTYPE option specifies the data set type for which you are establishing dynamic allocation options. Table 105 on page 742 lists the valid values for the DDTYPE option. Each instance of the $ARUDYNA macro must contain a different value for this option.
Dynamic allocation installation options

Overriding this option
You can override each instance of the DDTYPE installation option by using the DDTYPE command option (page 294).

**ACTIVE=NO or ACTIVE=YES**

The ACTIVE option tells REORG PLUS whether to dynamically allocate the specified data sets. A value of YES activates dynamic allocation. A value of NO inactivates dynamic allocation. Each DDTYPE has its own default value for the ACTIVE option:

Table 106  ACTIVE option default values by DDTYPE (part 1 of 2)

<table>
<thead>
<tr>
<th>DDTYPE</th>
<th>ACTIVE value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLOAD</td>
<td>YES</td>
</tr>
<tr>
<td>WORK</td>
<td>YES</td>
</tr>
<tr>
<td>SORTWORK</td>
<td>NO</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>NO</td>
</tr>
<tr>
<td>SYSPUNCH</td>
<td>YES</td>
</tr>
<tr>
<td>LOCPFCPY</td>
<td>YES</td>
</tr>
<tr>
<td>LOCPICPY</td>
<td>YES</td>
</tr>
<tr>
<td>LOCBFCPY</td>
<td>NO</td>
</tr>
<tr>
<td>LOCBICPY</td>
<td>NO</td>
</tr>
<tr>
<td>REMPLCPY</td>
<td>for future use</td>
</tr>
<tr>
<td>REMBLCPY</td>
<td>for future use</td>
</tr>
</tbody>
</table>
The following considerations apply to the ACTIVE option:

- For a DSNUTILB reorganization, you must enable dynamic allocation for the following data sets by specifying ACTIVE YES for those DDTYPEs:
  - all work file DDTYPEs that the reorganization job requires
  - if you specify COPY YES, the LOCPFCPY DDTYPE (and other copy DDTYPEs if you need them)

  Additionally, if the IBM DB2 REORG utility job requires data sets for discarded rows and for LOAD control statements for those discarded rows, you must enable dynamic allocation for the ARCHIVE and SYSPUNCH DDTYPEs.

  If you specify any of these data sets in your JCL, REORG PLUS ignores them, regardless of your IFALLOC specification.

  The following additional considerations apply to DSNUTILB reorganization jobs:

  - All copy data sets for a DSNUTILB reorganization are dynamically allocated, even if you specify ACTIVE YES for only the primary local copy data set.

  - REORG PLUS ignores this option for ARCHIVE and SYSPUNCH DDTYPEs when invoking DSNUTILB to reorganize a LOB table space.

- When running in a worklist environment, REORG PLUS ignores the ACTIVE option in your installation options module. REORG PLUS dynamically allocates your data sets only if the invoking product (DASD MANAGER PLUS, CATALOG MANAGER, or CHANGE MANAGER) supplies the ACTIVE YES syntax.

Overriding this option
You can override the value for this option by using the ACTIVE command option (page 296).
**ALLOC=ANY**

The ALLOC option tells REORG PLUS what method to use when dynamically allocating your sort work files. This option is valid only for sort work files. The following values are valid:

- **OPTIMIZED** tells REORG PLUS to allocate as much DASD as necessary for the best performance results.
- **MINIMAL** tells REORG PLUS to allocate the smallest amount of DASD necessary to process the job.
- **ANY** (the default) tells REORG PLUS to first attempt to allocate as much DASD as necessary for the best performance results. If it cannot obtain an optimal allocation, REORG PLUS reduces DASD allocation and continues processing.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**
You can override the value for this option by using the ALLOC command option (page 299).

**AVGVOLSP=((30000,TRK),(30000,TRK))**

The AVGVOLSP option enables you to specify the average amount of space that is available on each device. Use this option to reflect the average space available on volumes that are eligible to contain the dynamically allocated data set.

Specify one or two integer values, and include the unit of measure as follows:

- K for kilobytes
- TRK for tracks (the default)
- CYL for cylinders

REORG PLUS uses AVGVOLSP only when you specify a value of AUTO for the corresponding first or second parameter of the VOLCNT installation or command option (see page 760 or page 304).

REORG PLUS uses the second parameter (integer2) when the value for the THRESHLD installation option or command is exceeded (see page 757 or page 312). This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).
Restrictions
The following restrictions apply to the AVGVOLSP option:

- The AVGVOLSP option is not valid for DDTYPE SORTWORK.
- When invoking DSNUTILB, REORG PLUS ignores this option.

Additional considerations
Note the following additional information about the AVGVOLSP option:

- You should not use AVGVOLSP to specify the maximum space on all devices or volumes unless the volumes to be used are empty.
- If you specify a value for AVGVOLSP that is too small, REORG PLUS computes a value for VOLCNT that is too large.
- If you specify a value for AVGVOLSP that is too large, REORG PLUS computes a value for VOLCNT that is too small.

Overriding this option
You can override the value for this option by using the AVGVOLSP command option (page 306).

DATACLAS=(NONE,NONE)

The DATACLAS option specifies the primary and secondary SMS data classes that REORG PLUS uses for an SMS allocation. The two class names must be valid SMS data class names, not exceeding eight characters each, or NONE.

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the secondary data class in that template. For information about when REORG PLUS builds a secondary template, see page 757.

Overriding this option
You can override the value for this option by using the DATACLAS command option (page 325).

DSNPAT=&UUID.&&JOBNAME.&TSIX.&&DDNAME

The DSNPAT option specifies the pattern that REORG PLUS uses to generate data set names during dynamic data set allocation.

For sort work data sets only, you can also specify the value NONE. Do not enclose NONE in single quotes with the DSNPAT installation option (even though the quotes are required in the DSNPAT command option).
Each DDTYPE has its own default value for the DSNPAT option:

### Table 107  DSNPAT option default values by DDTYPE

<table>
<thead>
<tr>
<th>DDTYPE</th>
<th>DSNPAT value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLOAD</td>
<td>&amp;&amp;UID.&amp;JOBNAME.&amp;TSIX.&amp;DDNAME</td>
</tr>
<tr>
<td>WORK</td>
<td>&amp;&amp;UID.&amp;JOBNAME.&amp;TSIX.&amp;DDNAME</td>
</tr>
<tr>
<td>SORTWORK</td>
<td>NONE</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>&amp;&amp;UID.&amp;UTILPFX.&amp;DDNAME</td>
</tr>
<tr>
<td>SYSPPUNCH</td>
<td>&amp;&amp;UID.&amp;UTILPFX.&amp;DDNAME</td>
</tr>
<tr>
<td>LOCPFCPY</td>
<td>&amp;&amp;UID.&amp;DDNAME.&amp;TSIX.F&amp;PART.T&amp;TIME</td>
</tr>
<tr>
<td>LOCPICPY</td>
<td>&amp;&amp;UID.&amp;DDNAME.&amp;TSIX.I&amp;PART.T&amp;TIME</td>
</tr>
<tr>
<td>LOCBFCPY</td>
<td>&amp;&amp;UID.&amp;DDNAME.&amp;TSIX.F&amp;PART.T&amp;TIME</td>
</tr>
<tr>
<td>LOCBICPY</td>
<td>&amp;&amp;UID.&amp;DDNAME.&amp;TSIX.I&amp;PART.T&amp;TIME</td>
</tr>
<tr>
<td>REMPFCPY</td>
<td>&amp;&amp;UID.&amp;DDNAME.&amp;TSIX.F&amp;PART.T&amp;TIME</td>
</tr>
<tr>
<td>REMPICPY</td>
<td>&amp;&amp;UID.&amp;DDNAME.&amp;TSIX.I&amp;PART.T&amp;TIME</td>
</tr>
<tr>
<td>REMBFCPY</td>
<td>&amp;&amp;UID.&amp;DDNAME.&amp;TSIX.F&amp;PART.T&amp;TIME</td>
</tr>
<tr>
<td>REMBICPY</td>
<td>&amp;&amp;UID.&amp;DDNAME.&amp;TSIX.I&amp;PART.T&amp;TIME</td>
</tr>
</tbody>
</table>

### General considerations

The following considerations apply to the DSNPAT option:

- The pattern that you specify in your DSNPAT option must allow REORG PLUS to generate unique data set names. If REORG PLUS encounters non-unique data set names, processing terminates. For copy data sets, you might need to include additional variables such as &amp;VCAT, &amp;DATEJ, and &amp;TIME4 to generate unique names across multiple reorganizations.

- You can use text or any of the symbolic variables in Table 108 on page 747 to construct your pattern. You can also provide user-defined variables from a user exit (specified with the DSNUEXIT installation or command option). For more information about using symbolic variables to construct a data set name pattern, see “DSNPAT” on page 315.

- When specifying a pattern in your installation options, you must precede each REORG variable with an additional ampersand (&amp;) in your pattern.

- The maximum total length allowed for a data set is 44 bytes.

- REORG PLUS removes any trailing blanks in the resolved pattern.

### DSNUTILB reorganizations

When invoking DSNUTILB, REORG PLUS includes this pattern in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. Any variables that you include in your pattern for this type of reorganization must be either valid for the
TEMPLATE control statement or translatable (as shown in Table 108) to a valid TEMPLATE variable. User-defined variables are not valid for a DSNUTILB reorganization. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

**Overriding this option**

You can also specify the data set name pattern by using the DSNPAT command option (page 315), which overrides any default. Additional ampersands are not allowed with the command option. However, if you use the keyword NONE with the DSNPAT command option, you must enclose NONE with single quotation marks.

### Table 108 Symbolic variables for the DSNPAT installation option (part 1 of 3)

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of result</th>
<th>DSNUTILB reorganization</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;DATE</td>
<td>current date (in the form MMDDYY)</td>
<td>6 bytes</td>
<td>variable passed</td>
</tr>
<tr>
<td>&amp;DATEJ</td>
<td>current Julian date (in the form YYYYDDD)</td>
<td>7 bytes</td>
<td>variable translated to the IBM &amp;JDATE variable</td>
</tr>
<tr>
<td>&amp;DB</td>
<td>database containing the space being used for this data set allocation</td>
<td>8 bytes maximum</td>
<td>variable passed</td>
</tr>
<tr>
<td>&amp;DDNAME</td>
<td>ddname being used for this data set allocation</td>
<td>8 bytes maximum</td>
<td>value passed</td>
</tr>
<tr>
<td>&amp;GRPNM</td>
<td>DB2 data-sharing group name</td>
<td>4 bytes</td>
<td>value passed</td>
</tr>
<tr>
<td></td>
<td>In a non-data-sharing environment, GRPNM contains the DB2 SSID.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp;JDATE</td>
<td>current Julian date (in the form YYDDD)</td>
<td>5 bytes</td>
<td>variable translated to the IBM &amp;JDATE(3,5) variable</td>
</tr>
<tr>
<td>&amp;JOBNAME</td>
<td>JOB name used in the JCL</td>
<td>8 bytes maximum</td>
<td>variable passed</td>
</tr>
<tr>
<td>&amp;PART</td>
<td>partition being used for this data set allocation</td>
<td>3 bytes for table spaces with 999 partitions or less, 4 bytes for table spaces with 1000 through 4096 partitions</td>
<td>variable passed</td>
</tr>
</tbody>
</table>

For all other data sets, REORG PLUS substitutes the value 000 or 0000.
Table 108  Symbolic variables for the DSNPAT installation option  (part 2 of 3)

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of result</th>
<th>DSNUTILB reorganization</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;PART5</td>
<td>partition being used for this data set allocation</td>
<td>5 bytes</td>
<td>variable passed</td>
</tr>
<tr>
<td></td>
<td>You can use this variable for any data set. However, REORG PLUS substitutes the partition number for only copy and unload data sets. For all other data sets, REORG PLUS substitutes the value 00000. REORG PLUS generates 5-character partition numbers as follows: partition 1 = 00001 partition 10 = 00010 partition 100 = 00100 partition 1000 = 01000 nonpartitioned = 00000 Example: REORG TABLESPACE PART 4096 DDTYPE UNLOAD ACTIVE YES DSNPAT 'ABC.DSN1.DA.&amp;DB.&amp;TSIX..P&amp;PART5' REORG PLUS generates the following 5-character partition number for partition 4096: ABC.DSN1.DA.DBNAME.TSNAME.P04096</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp;RTYPE</td>
<td>REORG type (TS or IX)</td>
<td>2 bytes maximum</td>
<td>job terminated</td>
</tr>
<tr>
<td>&amp;SSID</td>
<td>DB2 subsystem ID</td>
<td>4 bytes</td>
<td>variable passed</td>
</tr>
<tr>
<td>&amp;STEPNAME</td>
<td>STEP name used in the JCL</td>
<td>8 bytes maximum</td>
<td>variable passed</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS ignores PROC names.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>current time (in the form HHMMSS)</td>
<td>6 bytes</td>
<td>variable passed</td>
</tr>
<tr>
<td>&amp;TIME4</td>
<td>current time (in the form HHMM)</td>
<td>4 bytes</td>
<td>variable passed</td>
</tr>
<tr>
<td>&amp;TSIX</td>
<td>table space or index space specified in your REORG command</td>
<td>8 bytes maximum</td>
<td>variable translated to the IBM &amp;SN variable</td>
</tr>
<tr>
<td>&amp;USERID or &amp;UID</td>
<td>job user ID</td>
<td>8 bytes maximum</td>
<td>variable passed</td>
</tr>
<tr>
<td></td>
<td>You must have a security package to use this variable.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp;UTILa</td>
<td>BMC utility ID</td>
<td>8 bytes maximum</td>
<td>variable translated to the IBM &amp;UTILID variable</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS truncates longer utility IDs to eight characters.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
For your dynamically allocated SYSARC, SYSPUNCH, and copy data sets, you can also specify a pattern that contains a GDG name. Each DDTYPE must have a different GDG base.

The GDG format that you use to construct a data set name is the same as the format that you use in JCL when you use DD statements to allocate your copy data sets. Simply append the generation number in parentheses. The open parenthesis tells REORG PLUS that the pattern is a GDG name. The generation number must be an integer from 1 through 255.

If the base does not exist, REORG PLUS creates it for you using everything in the pattern up to the open parenthesis as the base name. For more information about GDG names and options, see “Generating data set names” on page 92.

The following example shows a valid GDG name:

'&UTILPFX.&DDNAME..COPY(+1)'

If you are using a substitution variable as the last variable before the open parenthesis, you must include a period before the open parenthesis. For example:

'&UTILPFX.&DDNAME.(+1)'

If you specify COPYLVL PART on the REORG command, each partition must have a different GDG base. To specify a pattern that includes a partition, the partition must not be in parentheses. The following example shows a valid name:

'&UTILPFX.&DDNAME..P&PART.(+1)'

---

**Table 108  Symbolic variables for the DSNPAT installation option  (part 3 of 3)**

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of result</th>
<th>DSNUTILB reorganization</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;UTILPFX&lt;sup&gt;a&lt;/sup&gt;</td>
<td>BMC utility ID prefix</td>
<td>8 bytes maximum</td>
<td>value passed</td>
</tr>
<tr>
<td>&amp;UTILSFX&lt;sup&gt;b&lt;/sup&gt;</td>
<td>BMC utility ID suffix</td>
<td>8 bytes maximum</td>
<td>value passed</td>
</tr>
<tr>
<td>&amp;VCAT</td>
<td>VCATNAME specified in the DB2 catalog for the table space that you are reorganizing; or, if the table space is partitioned, the VCAT name from the first partition that you are reorganizing</td>
<td>8 bytes</td>
<td>job terminated</td>
</tr>
</tbody>
</table>

<sup>a</sup> Utility IDs that include special characters might cause REORG PLUS to generate invalid data set names. For more information, see page 318.
You cannot specify a pattern that contains a partitioned data set (PDS) name. The following example shows an invalid name:

`'&UTILPFX.&DDNAME..(P&PART)'`

For more information and guidelines for specifying data set name patterns, see page 315.

**DSNTYPE=(NONE,NONE)**

The DSNTYPE option enables you to specify the type of data set that you want REORG PLUS to allocate. If you specify a second value, REORG PLUS uses that value when the value that is in effect for the THRESHLD option is exceeded (see page 757 or page 312).

The following values are valid:

- **NONE** (the default) tells REORG PLUS to not use any extended attributes for this data set allocation.
- **LARGE** tells REORG PLUS to allocate this data set as a large format sequential data set. This option enables data sets larger than 65,535 tracks.
- **BASIC** tells REORG PLUS to allocate this data set as a basic sequential data set. This data set will be limited to 65,535 tracks.
- **EXTREQ** tells REORG PLUS to allocate this data set as an extended format data set.

This option is ignored if either of the following conditions exists:

- The SMS option is NO.
- The DDTYPE is SORTWORK.

- **EXTPREF** indicates that you prefer that REORG PLUS allocate this data set as an extended format data set. If this allocation is not possible, the data set is allocated as a basic format data set.

This option is ignored if either of the following conditions exists:

- The SMS option is NO.
- The DDTYPE is SORTWORK.

**Restrictions**
The following restrictions apply to the DSNTYPE option:

- The DSNTYPE option is not valid for tape data sets.
- When invoking DSNUTILB, REORG PLUS ignores this option.
Overriding this option
You can override the value for this option by using the DSNTYPE command option (page 309).

EXPDT=

The EXPDT option specifies the expiration date for the SYSARC, SYSPUNCH, or copy data set that you are dynamically allocating. The date must be either blank or in the format yyyyddd:

- yyyy is the 4-digit year (1900 through 2155).
- ddd is the 3-digit Julian day (000 through 366).

A blank value means that REORG PLUS does not use an expiration date.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

When you specify the EXPDT installation option, it takes precedence over the RETPD installation option.

REORG PLUS ignores this option if you specify it for any other data set that you are dynamically allocating.

Overriding this option
You can override the EXPDT installation option by using the EXPDT or RETPD command option. For information about the commands, see “EXPDT” on page 322 and “RETPD” on page 322.

GDGEMPTY=NO

The GDGEMPTY option specifies whether to uncatalog all data sets when the GDGLIMIT is reached:

- NO indicates that the system uncatalog only the oldest GDG data set when the GDGLIMIT is reached.
- YES indicates that the system uncatalog all preexisting generations of this data set when the limit is reached.

REORG PLUS honors this option only when creating the GDG base for the SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating. REORG PLUS ignores this option if you specify it for any other data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS ignores this option.
**GDGLIMIT=5**

The GDGLIMIT option specifies the number of generations to keep for the SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating as a GDG data set. The number must be an integer in the range 1 through 255.

REORG PLUS honors this option only when creating the GDG base. REORG PLUS ignores this option if you specify it for any other data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

*Overriding this option*

You can override the value for this option by using the GDGLIMIT command option (page 323).

**GDGSCRATCH=NO**

The GDGSCRATCH option specifies whether to delete uncataloged data sets:

- **NO** specifies that the system should not delete an entry that is uncataloged as a result of the GDGEMPTY option.

- **YES** specifies that the system should delete the GDG entry from the volume’s table of contents (VTOC) when uncataloging the data set. The space on the volume is then available to other users.

REORG PLUS honors this option only when creating the GDG base for the SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating. REORG PLUS ignores this option if you specify it for any other data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS ignores this option.

**IFALLOC=USE**

The IFALLOC option tells REORG PLUS how to handle any data sets that are specified in your JCL:

- **USE** tells REORG PLUS to use the data sets that you allocated in your JCL. If the number of data sets that you specify in your JCL is insufficient for processing, REORG PLUS (or BMCSORT in the case of sort work data sets) dynamically allocates the additional data sets that your job needs.
FREE tells REORG PLUS to free the data sets that you allocated in your JCL and use only dynamically allocated data sets.

When invoking DSNUTILB, REORG PLUS ignores this option.

**Overriding this option**

You can override the value for this option by using the IFALLOC command option (page 297).

**MAXEXTSZ=((0,K),(0,K))**

For any extent that REORG PLUS allocates for a dynamically allocated data set, this option enables you to specify the maximum allowable value for the primary space allocation. Because the secondary quantity cannot exceed the primary quantity, MAXEXTSZ also controls the maximum secondary quantity.

Specify one of the following values for the first value of each parameter:

- 0 if you do not want to set a limit for space allocation
- an integer for the units specified with the UNIT installation or command option (see page 758 or page 302)

For the second value of each parameter, specify a unit of measure as follows:

- K for kilobytes (the default)
- TRK for tracks
- CYL for cylinders

REORG PLUS uses the second parameter when the value specified for the THRESHLD installation or command option is exceeded (see page 757 or page 312). This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

Regardless of the amount of space that REORG PLUS determines that it needs, REORG PLUS will not allocate more than your specified MAXEXTSZ limit for either the primary or the secondary quantity:

- If the amount of required space that REORG PLUS calculates is greater than the MAXEXTSZ limit for the primary quantity, REORG PLUS uses the secondary extents to hold the remainder of the required primary space.

- If the amount of required space that REORG PLUS calculates cannot be accommodated because of MAXEXTSZ restrictions, the job might terminate with an out-of-space condition on the data set.
**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS includes the first parameter of this option as the MAXPRIME value in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For DSNUTILB, the unit of measure is always cylinders.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second parameter of this option as the MAXPRIME value in that template. For information about when REORG PLUS builds a secondary template, see page 757.

**Additional considerations**
Note the following additional information about the MAXEXTSZ option:

- The MAXEXTSZ option is not valid for DDTYPE SORTWORK.
- REORG PLUS ignores MAXEXTSZ when you specify SMS YES.
- REORG PLUS checks the value of MAXEXTSZ after applying SIZEPCT to the allocation amount.

**Overriding this option**
You can override the value for this option by using the MAXEXTSZ command option (page 307).

**MGMTCLAS=(NONE,NONE)**

The MGMTCLAS option specifies the primary and secondary SMS management classes that REORG PLUS uses for an SMS allocation. The two class names must be valid SMS management class names, not exceeding eight characters each, or NONE.

**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the secondary management class in that template. For information about when REORG PLUS builds a secondary template, see page 757.

**Overriding this option**
You can override the value for this option by using the MGMTCLAS command option (page 325).

**RETPD=**

Use RETPD to specify the retention period (in days) for the SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating. The number of days must be either blank or in the range 0 through 9999. A blank value means that REORG PLUS does not use a retention period.
REORG PLUS ignores this option if you specify it for any other data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

**Overriding this option**
You can override the value for this option by using the RETPD (page 322) or EXPDT command option (page 322). If you specify the EXPDT installation option, the EXPDT installation option takes precedence over the RETPD installation option.

**SIZEPCT=(100,100) or SIZEPCT=(5,100)**

The SIZEPCT option provides the percentages of the primary and secondary space to use for allocation. The values that you specify must be greater than 0.

The first number indicates the percentage of the primary quantity (calculated by REORG PLUS) to allocate. The second number indicates the percentage of the secondary quantity to allocate.

For the incremental image copy DDTYPEs (LOCPICPY, LOCICPY, REMPICPY, and REMBICPY), the default value is SIZEPCT=(5,100). For all other DDTYPEs, the default value is SIZEPCT=(100,100).

When you specify a value for both the SIZEPCT installation or command option and the ARCROWS command option, REORG PLUS ignores SIZEPCT for the archive data set and uses only ARCROWS to determine the archive data set size.

**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS includes the first parameter of this option as the PCTPRIME value in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. If you specify a value greater than 100, REORG PLUS converts it to 100.

**Overriding this option**
You can override the value for this option by using the SIZEPCT command option (page 311).

**SMS=NO**

The SMS option tells REORG PLUS whether to pass the SMS classes and extended data set types (EXTREQ and EXT PREF) to SMS during dynamic allocation. Whether REORG PLUS actually performs an SMS allocation depends on your site.
You can specify one of the following values:

- **YES** tells REORG PLUS to pass SMS classes and extended data set types during dynamic allocation.

  The following considerations apply to SMS YES:
  
  — When you specify SMS YES, REORG PLUS ignores the value that you specify for MAXEXTSZ.
  
  — To pass the UNIT value to SMS during dynamic allocation, also specify SMSUNIT=YES.
  
  — When invoking DSNUTILB, REORG PLUS includes the SMS classes in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

- **NO** tells REORG PLUS not to pass SMS classes and extended data set types during dynamic allocation.

  When you specify SMS NO, REORG PLUS always passes the UNIT value during dynamic allocation.

**Overriding this option**

You can override the value for this option by using the SMS command option (page 300).

**SMSUNIT=NO**

The SMSUNIT option tells REORG PLUS whether to pass the UNIT value in the SMS allocation parameter list to SMS during dynamic allocation. REORG PLUS does not modify any other parameters based on this option. If the value of the SMS option is NO, REORG PLUS ignores the SMSUNIT option.

- **NO** tells REORG PLUS not to pass the value for the UNIT option.
- **YES** tells REORG PLUS to pass the value for the UNIT option.

  When invoking DSNUTILB, REORG PLUS includes the UNIT option values in the TEMPLATE control statements that it builds for the IBM DB2 REORG utility.

**Overriding this option**

You can override the value for this option by using the SMSUNIT command option (page 301).

**STORCLAS=(NONE,NONE)**

The STORCLAS option specifies the primary and secondary SMS storage classes that REORG PLUS uses for an SMS allocation. The two class names must be valid SMS storage class names, not exceeding eight characters each, or NONE.
**DSNUTILB reorganization jobs**

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the secondary storage class in that template. For information about when REORG PLUS builds a secondary template, see page 757.

**Overriding this option**

You can override the value for this option by using the STORCLAS command option (page 325).

**THRESHLD=0**

The THRESHLD option allows you to specify a threshold value, in kilobytes, above which REORG PLUS applies secondary values to allocated data sets. REORG PLUS tests this threshold for each data set to be allocated. If the size for a particular data set is greater than the threshold, REORG PLUS performs the following tasks:

- When SMS is NO, REORG PLUS uses the secondary values of the following options for the allocated data sets:
  - UNIT (unit name)
  - UNITCNT (unit count)
  - VOLUMECNT (volume count)
  - AVGVOLSP (average volume space)
  - MAXEXTSZ (extent size)
  - DSNTYPE (data set type), unless DSNTYPE is EXTREQ or EXTPREF

- When SMS is YES, REORG PLUS uses the secondary values of the following options (or passes them to SMS if applicable):
  - STORCLAS, MGMTCLAS, or DATACLAS (class name for the SMS classes)
  - VOLUMECNT (volume count)
  - AVGVOLSP (average volume space)
  - DSNTYPE (data set type)
  - if SMSUNIT is YES
    - UNIT (unit name)
    - UNITCNT (unit count)

If you specify 0 or if the threshold is not exceeded, REORG PLUS uses the primary values for these options.
**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS handles this option differently, depending on the type of data set that you are allocating:

- For copy data sets, REORG PLUS translates this option to a LIMIT value in the TEMPLATE control statement that REORG PLUS builds for the IBM DB2 REORG utility. REORG PLUS also builds the secondary TEMPLATE control statement to which the DB2 REORG utility will switch when the LIMIT is exceeded.

  REORG PLUS assumes that the value that you supply for THRESHLD is in kilobytes. REORG PLUS translates that value to the appropriate value and unit of measure for the LIMIT keyword. Note the following additional information about this value:

  — REORG PLUS rounds down to the nearest whole value.
  — If you specify a value that would cause REORG PLUS to translate to a value less than 1 cylinder, REORG PLUS builds the template with a LIMIT value of 1 CYL.

- For all other data sets, REORG PLUS ignores this option.

**Additional consideration**
If you use THRESHLD to send larger data sets to tape, consider setting the MAXTAPE option to limit the number of tapes that you use. For examples of using the THRESHLD option with other dynamic allocation options, see page 312.

**Overriding this option**
You can override the value for this option by using the THRESHLD command option (page 312).

**UNIT=(SYSALLDA, SYSALLDA)**
For non-SMS-managed data sets, the UNIT option specifies the primary and secondary unit names that REORG PLUS uses for dynamic data set allocation. These unit names cannot exceed eight characters each.

For SMS-managed data sets when you specify SMS YES and SMSUNIT YES, this option supplies the unit names that REORG PLUS passes in the SMS allocation parameter list. DSNUTILB reorganization jobs function differently, as described in the DSNUTILB reorganization jobs section.

The THRESHLD command or installation option controls which of the specified unit names REORG PLUS selects. For more information about the THRESHLD installation option, see page 757.

**DSNUTILB reorganization jobs**
When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.
If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second UNIT parameter in that template. For information about when REORG PLUS builds a secondary template, see page 757.

**Overriding this option**

You can override the value for this option by using the UNIT command option (page 302).

**UNITCNT=(0,0)**

The UNITCNT option enables you to specify the number of devices to allocate when dynamically allocating data sets. Valid values are 0 through 59. A value of 0 tells REORG PLUS to use the system default.

If you specify a second value, REORG PLUS uses that value when the value for THRESHLD (page 757) is exceeded. This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

**DSNUTILB reorganization jobs**

When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second UNITCNT parameter in that template. For information about when REORG PLUS builds a secondary template, see page 757.

**Additional considerations**

The following additional considerations apply to the UNITCNT option:

- REORG PLUS ignores this option for SORTWORK files.
- To avoid performance problems, specify only the number of devices that you need.

**Overriding this option**

You can override the value for this option by using the UNITCNT command option (page 302).
VOLCNT=(25,25)

The VOLCNT option enables you to specify the maximum number of volumes to use during dynamic data set allocation. Valid values are:

- 0, which tells REORG PLUS to not specify a volume count for dynamic allocation
- integer values 1 through 255 to specify the number of volumes
- AUTO, which tells REORG PLUS to compute the volume count based on the amount of data, adjusted for the estimated space required

REORG PLUS computes the volume count by dividing the size estimate by the value specified for the AVGVOLSP installation or command option (see page 744 or page 306).

If you specify a second value (integer2), REORG PLUS uses that value when the value for the THRESHLD installation or command option is exceeded (page 757 and page 312). This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second VOLCNT parameter in that template. For information about when REORG PLUS builds a secondary template, see page 757.

Additional considerations
The following additional considerations apply to the VOLCNT option:

- This option is not available for DDTYPE SORTWORK. If you specify a value for this option for sort work files, REORG PLUS changes the value to 1.
- Specify only the number of volumes that you need.
- If the DDTYPE will be an SMS-managed data set, BMC recommends that you specify a value of 0 if your ACS routines are set up to provide a volume count.

Overriding this option
You can override the value for this option by using the VOLCNT command option (page 304).
DYNALOC installation option

The DYNALOC installation option provides information for dynamically allocating SORTWK data sets. BMCSORT deallocates these data sets at the end of each sort. The content of the $AUPSMAC macro in $C32SOPT follows, showing DYNALOC and the values that are shipped with BMCSORT.

$AUPSMAC DYNALOC=(SYSDA,3,ON,ON,6000000,3000000,3390,SC=,RETRY=(0,0))

The values that you specify in this macro apply to all invocations of BMCSORT. BMCSORT uses the same options module for all BMC products that invoke BMCSORT.

Table 109 describes each parameter of the DYNALOC option. These parameters are positional. The values that you specify for these parameters should correspond to your site’s standards for any system sort routine.

BMCSORT overrides the values that you supplied if BMCSORT determines that it can complete sorting more efficiently than the specified values allow. An invoking product’s options might also override the BMCSORT options values that you specify when either of the following conditions exists:

- The values in the invoking product’s dynamic allocation installation options or corresponding command options conflict with the values that you specify.
- You turn on BMCSORT SORTWK dynamic allocation from the product that invokes BMCSORT, and you specify OFF for the position 3 parameter.

BMCSORT dynamically allocates SORTWK files as necessary.

Table 109  DYNALOC parameters  (part 1 of 3)

<table>
<thead>
<tr>
<th>Parameter name or position</th>
<th>Description</th>
<th>Initial value</th>
<th>Valid values</th>
</tr>
</thead>
<tbody>
<tr>
<td>position 1</td>
<td>This parameter specifies the generic unit name from which REORG PLUS for DB2 should dynamically allocate SORTWK data sets. This parameter applies only when the Data Facility Storage Management Subsystem (DFSMS) product from IBM is not installed or is not active for temporary DASD work data sets. If DFSMS is active, use the SC parameter.</td>
<td>SYSDA</td>
<td>Use a unit name up to 8 characters.</td>
</tr>
<tr>
<td>position 2</td>
<td>Do not change this value. REORG PLUS for DB2 does not use this parameter, but the parameter is required for proper assembly of the installation options macro.</td>
<td>3</td>
<td>Do not change this value.</td>
</tr>
</tbody>
</table>
### Table 109 DYNALOC parameters (part 2 of 3)

<table>
<thead>
<tr>
<th>Parameter name or position</th>
<th>Description</th>
<th>Initial value</th>
<th>Valid values</th>
</tr>
</thead>
</table>
| position 3                | This parameter tells REORG PLUS for DB2 whether to dynamically allocate SORTWK files. **Note**: BMC recommends that you not change this value. | ON | - ON dynamically allocates SORTWK.  
- OFF does not dynamically allocate SORTWK. |
| position 4                | Do not change this value. REORG PLUS for DB2 does not use this parameter, but the parameter is required for proper assembly of the installation options macro. | ON | Do not change this value. |
| position 5                | Do not change this value. REORG PLUS for DB2 does not use this parameter, but the parameter is required for proper assembly of the installation options macro. | 6000000 | Do not change this value. |
| position 6                | Do not change this value. REORG PLUS for DB2 does not use this parameter, but the parameter is required for proper assembly of the installation options macro. | 3000000 | Do not change this value. |
| position 7                | This parameter specifies the DASD type with the smallest track capacity that a dynamically allocated SORTWK data set might encounter at your site. | 3390 | - 3380, track capacity of 47968  
- 3390, track capacity of 56664  
- 9345, track capacity of 46456  |
| SC                        | This parameter specifies the name of the DFSMS storage class from which to dynamically allocate SORTWK. If DFSMS is active and you do not specify a value for this parameter, REORG PLUS for DB2 uses the value from the first DYNALOC parameter. **Note**: If your installation has an automatic class selection (ACS) routine, it can override this specification. | blank | Use any valid DFSMS storage class. |
DYNALOC installation option

**Table 109  DYNALOC parameters (part 3 of 3)**

<table>
<thead>
<tr>
<th>Parameter name or position</th>
<th>Description</th>
<th>Initial value</th>
<th>Valid values</th>
</tr>
</thead>
</table>
| RETRY                     | This parameter specifies how you want REORG PLUS for DB2 to handle retry attempts for SORTWK dynamic allocation:  
  - The first subparameter indicates the number of times that you want REORG PLUS for DB2 to retry the request.  
  - The second subparameter indicates the number of minutes to wait between each retry.  
  Using this parameter allows you to avoid a capacity-exceeded condition when disk space is not immediately available for a SORTWK dynamic allocation request.  
  BMC recommends that you do not change this value because it can affect the elapsed time of your jobs. However, if you currently use SyncSort and rely on the retry function, BMC recommends that you use the same values as your SyncSort RETRY installation parameter. | (0,0) | If you use this parameter, BMC recommends that you specify the same values as your SyncSort RETRY installation parameter. The following values are valid for this parameter:  
  - 0 through 16 for the first subparameter. 0 indicates that you do not want REORG PLUS for DB2 to retry the request.  
  - 0 through 15 for the second subparameter. 0 indicates that you do not want REORG PLUS for DB2 to retry the request. |
Common utility tables

This appendix presents the following topics:

Overview .......................................................... 765
  Considerations and warnings for common utility tables ............. 766
  Managing common utility tables .................................. 767
BMCDICT table ....................................................... 769
  Considerations for the BMCDICT table ............................ 769
  Maintaining the BMCDICT table .................................. 770
BMCHIST table ...................................................... 770
  Maintaining the BMCHIST table .................................. 772
BMCLGRNX table .................................................... 772
BMCSYNC table ...................................................... 773
  Considerations for the BMCSYNC table ............................. 775
  Maintaining the BMCSYNC table .................................. 776
BMCTRANS table .................................................... 777
BMCUTIL table ...................................................... 778
  Maintaining the BMCUTIL table .................................. 780
BMXCOPY table ...................................................... 781
  Maintaining the BMXCOPY table .................................. 785

Overview

The BMC common utility tables contain information about the BMC utilities that you generate and submit through a BMC utility product. Table 110 on page 766 lists the tables that each utility uses and each table’s default name and synonym.
Considerations and warnings for common utility tables

Note the following considerations when using the common utility tables:

- Some columns in the tables are present for compatibility with specific BMC utilities and are not used by all of the utilities.

- If you have applications that depend on the structure or content of these tables, be aware that these tables are subject to change.

- In general, the utility tables should not require maintenance, with the exception of BMCHIST.

- You should back up the BMC table spaces on a regular basis to enable recoveries. If you use COPY PLUS as the copy utility, you must use SHRLEVEL CHANGE for the following spaces:
  - BMCUTIL
  - BMCHIST
  - BMCSYNC
  - BMCXCOPY
Managing common utility tables

This section provides basic procedures for working with the common utility tables:

**To determine your site’s table names**

The names of the common utility tables can be changed during installation. To determine the names that your site uses, perform one of the following actions:

- Use your utility to run a job with restart parameters of MAINT and MSGLEVEL(1).

  Specifying MSGLEVEL(1) with MAINT prints the names of the BMC tables that your utility uses and identifies the applied maintenance. The utility does not perform any other processing, and the job ends without affecting any utility that is running.

- Run the following SQL statement, replacing `tableName` with a BMC common utility table name (listed in Table 110 on page 766):

  ```sql
  SELECT CREATOR, NAME FROM SYSIBM.SYSTABLES
  WHERE TSNAME='tableName';
  ```

- Get the names from your DB2 system administrator.

---

**WARNING**

The following warnings apply:

- Do not run LOADPLUS, REORG PLUS, or UNLOAD PLUS against the BMC common utility tables or table spaces. Doing so can cause unpredictable results.

- Because RECOVER PLUS uses BMC tables during the recovery process, you cannot use RECOVER PLUS to recover the BMC tables, with the exception of the BMCHIST table.

- Do not run the RUNSTATS utility against the BMC common utility tables. Doing so can negatively impact utility performance.

- BMC strongly recommends that you use the ISOLATION (UR) bind option and issue SQL COMMIT statements when querying the tables in the BMC database. If objects in the BMC database are restricted for UPDATE, the executing BMC utilities might not be able to complete successfully.
To query the tables

Run SQL statements similar to the following examples:

--- EXAMPLE ---

This example queries the BMCXCOPY table to access information about the rows in an index space:

```sql
SELECT *
FROM creatorName.CMN_BMCXCOPY
WHERE DBNAME = 'databaseName'
AND  IXNAME = 'indexSpaceName'
ORDER BY START_RBA;
```

This example identifies (from the BMCHIST table) the database name, table space name, elapsed time, and when the utility completed:

```sql
SELECT DBNAME, SPNAME, CHAR(ELAPSED, ISO), CHAR(TIME, ISO)
FROM creatorName.CMN_BMCHIST
WHERE UTILID = 'utilityID';
```

To display BMC utility status

To display the status of all BMC utilities that are executing or awaiting restart for a given table space or index space, use the following SQL statements:

```sql
SELECT * FROM creatorName.CMN_BMCUTIL
WHERE DBNAME = 'databaseName'
AND SPNAME = 'tableSpaceName'
SELECT * FROM creatorName.CMN_BMCSYNC
WHERE NAME1 = 'databaseName'
AND NAME2 = 'spaceName';
```

To terminate a BMC utility

To terminate a BMC utility that is executing, use the following SQL statements:

```sql
DELETE FROM creatorName.CMN_BMCUTIL
WHERE UTILID = 'utilityID';
DELETE FROM creatorName.CMN_BMCSYNC
WHERE UTILID = 'utilityID';
DELETE FROM creatorName.CMN_BMCDICT -- for LOADPLUS and REORG PLUS
WHERE UTILID = 'utilityID';
```

The utility terminates with return code 8 when the next checkpoint is taken.
To clean up a BMC utility that is not executing, run the utility with the correct utility ID and specify TERM as the restart parameter.

BMCDICT table

Table 111 describes the BMCDICT table, which stores the compression dictionary during load or reorganization processing.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>utility identifier</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>database name</td>
</tr>
<tr>
<td>TSNAME</td>
<td>CHAR(8)</td>
<td>table space name</td>
</tr>
<tr>
<td>PARTITION</td>
<td>SMALLINT</td>
<td>partition number</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT</td>
<td>sequence number</td>
</tr>
<tr>
<td>DICTDATA</td>
<td>VARCHAR(4000)</td>
<td>dictionary data</td>
</tr>
</tbody>
</table>

**Considerations for the BMCDICT table**

Note the following considerations:

- If you are processing a large number of compressed partitions, you might need to increase the size of the BMCDICT table space significantly from the standard size that was allocated during installation. To estimate the allocation, multiply 64 KB by the number of compressed partitions that you are processing concurrently (loading with LOADPLUS and reorganizing with REORG PLUS).

- LOADPLUS inserts rows into the BMCDICT table during the PRELOAD phase and deletes those rows following compression processing in the LOAD phase.

- REORG PLUS inserts rows into the BMCDICT table during the UNLOAD phase and deletes those rows following compression processing in the RELOAD phase.
Maintaining the BMCDICT table

If LOADPLUS or REORG PLUS abends during the time between building the compression dictionary and completing compression, rows might remain in the BMCDICT table.

If you need to control the expansion of this table, use the following procedure:

1. Delete any rows in the BMCUTIL table that you know are no longer valid.
   - Do not delete any rows for instances of utilities that are awaiting restart.

2. Use the following SQL statement to delete rows from the BMCDICT table:

   ```sql
   DELETE
   FROM creatorName.CMN_BMCDICT
   WHERE UTILID NOT IN
     (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
   ```

**NOTE**
The names of the BMCUTIL and BMCDICT tables might have been changed at your site during installation.

BMCHIST table

Table 112 on page 771 describes the BMCHIST table, which contains information about completed executions of the BMC utilities for DB2. The following configuration or installation options control use of the BMCHIST table:

- **HISTORY** (for COPY PLUS, RECOVER PLUS, and UNLOAD PLUS)
- **BMCHIST** (for REORG PLUS)

If the option value is NO, the utility bypasses any updates to the BMCHIST table. If the value is YES (or the utility does not use a configuration or installation option), the utility inserts rows into the BMCHIST table during the UTILTERM phase.
### Table 112  BMCHIST table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>name of the database that contains the table or index space</td>
</tr>
<tr>
<td>SPNAME</td>
<td>CHAR(8)</td>
<td>name of the table or index space</td>
</tr>
<tr>
<td>UTILNAME</td>
<td>CHAR(8)</td>
<td>name of the utility:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LOAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- RECOVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- REORG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UNLOAD</td>
</tr>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>utility identifier</td>
</tr>
<tr>
<td>AUTHID</td>
<td>CHAR(8)</td>
<td>user ID that ran the utility</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>date that the utility completed</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>time that the utility completed</td>
</tr>
<tr>
<td>ELAPSED</td>
<td>TIME</td>
<td>elapsed time of the utility</td>
</tr>
<tr>
<td>PARTITION</td>
<td>LONG VARCHAR</td>
<td>ALL, or the partition numbers as specified by the DSNUM option (for COPY PLUS) or the PART option</td>
</tr>
</tbody>
</table>

- For UNLOAD PLUS, if you specified LOGICAL PART, these partitions are the physical partitions that correspond to the logical partitions that you specified.
- This column lists only three-digit partitions (any loaded partitions 1 through 999). Four-digit partitions (any loaded partitions from 1000 through 4096) are not stored in this column. For jobs that load only four-digit partitions, this column is empty.
- If the list of partitions exceeds 1011 bytes, the utility truncates the value that is stored in this column.

<table>
<thead>
<tr>
<th>OBJNAME</th>
<th>VARCHAR(27)</th>
<th>fully qualified object name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHASE_1</td>
<td>CHAR(8)</td>
<td>name of utility phase 1</td>
</tr>
<tr>
<td>ELAPSED_1</td>
<td>TIME</td>
<td>elapsed time of phase 1</td>
</tr>
<tr>
<td>PHASE_2</td>
<td>CHAR(8)</td>
<td>name of utility phase 2</td>
</tr>
<tr>
<td>ELAPSED_2</td>
<td>TIME</td>
<td>elapsed time of phase 2</td>
</tr>
<tr>
<td>PHASE_3</td>
<td>CHAR(8)</td>
<td>name of utility phase 3</td>
</tr>
<tr>
<td>ELAPSED_3</td>
<td>TIME</td>
<td>elapsed time of phase 3</td>
</tr>
<tr>
<td>PHASE_4</td>
<td>CHAR(8)</td>
<td>name of utility phase 4</td>
</tr>
<tr>
<td>ELAPSED_4</td>
<td>TIME</td>
<td>elapsed time of phase 4</td>
</tr>
<tr>
<td>PHASE_5</td>
<td>CHAR(8)</td>
<td>name of utility phase 5</td>
</tr>
<tr>
<td>ELAPSED_5</td>
<td>TIME</td>
<td>elapsed time of phase 5</td>
</tr>
</tbody>
</table>
Maintaining the BMCHIST table

When a utility completes successfully, it inserts a row into the BMCHIST table. Periodically, review BMCHIST and delete old rows to control its expansion.

To delete selected rows from the BMCHIST table based on the date that the utility completed, use the following sample SQL statement:

```
DELETE
FROM creatorName.CMN_BMCHIST
WHERE DATE < 'yyyy-mm-dd';
```

You can also use the TERMEXIT user exit to control inserts into the BMCHIST table. For more information, see Appendix C, “REORG PLUS user exits.”

BMCLGRNX table

Table 113 describes the contents of the BMCLGRNX table, which contains log ranges that show when a table space was open for updates.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGRDBID</td>
<td>CHAR(2)</td>
<td>DBID of the modified object</td>
</tr>
<tr>
<td>LGRPSID</td>
<td>CHAR(2)</td>
<td>OBID of the modified object</td>
</tr>
<tr>
<td>LGRUCDT</td>
<td>CHAR(6)</td>
<td>modification date (mmdyy)</td>
</tr>
<tr>
<td>LGRUCTM</td>
<td>CHAR(8)</td>
<td>modification time (hhmmssst)</td>
</tr>
<tr>
<td>LGRSRBA</td>
<td>CHAR(6)</td>
<td>starting RBA</td>
</tr>
<tr>
<td>LGRELSPBA</td>
<td>CHAR(6)</td>
<td>stopping RBA</td>
</tr>
<tr>
<td>LGRPART</td>
<td>SMALLINT</td>
<td>table space partition number</td>
</tr>
<tr>
<td>LGRSLRSN</td>
<td>CHAR(6)</td>
<td>starting LRSN of update log records for data sharing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For non-data-sharing, the value is X'000000000000'.</td>
</tr>
<tr>
<td>LGRELRSN</td>
<td>CHAR(6)</td>
<td>ending LRSN of update log records for data sharing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For non-data-sharing, the value is X'000000000000'.</td>
</tr>
<tr>
<td>LGRMEMBER</td>
<td>CHAR(2)</td>
<td>data sharing member ID of the modifying DB2 subsystem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For non-data-sharing, the value is X'0001'.</td>
</tr>
</tbody>
</table>
BMCSYNC table

Table 114 describes the BMCSYNC table, which contains information about the status of the objects that the currently executing utilities are accessing. The BMCSYNC table synchronizes and controls access to DB2 spaces by concurrently executing BMC utility products. If you have more than one BMC utility installed, all of these utilities should share the same BMCSYNC table.

The utilities insert rows into the BMCSYNC table during the UTILINIT phase. While the job executes, the utilities update the table as the status of the object changes. The utilities delete rows from the BMCSYNC table during the UTILTERM phase.

Table 114  BMCSYNC table (part 1 of 3)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>utility identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For RECOVER PLUS, this column is blank when a RECOVER UNLOADKEYS command creates the row and then a RECOVER BUILDINDEX command reads and deletes the row.</td>
</tr>
<tr>
<td>NAME1</td>
<td>CHAR(8)</td>
<td>database name or creator namea</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For DASD MANAGER PLUS, the value is the database name.</td>
</tr>
<tr>
<td>NAME2</td>
<td>CHAR(18)</td>
<td>space, table, or index namea</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For DASD MANAGER PLUS, the BMCSTATS utility always inserts the space name (limited to a maximum of 8 characters).</td>
</tr>
<tr>
<td>KIND</td>
<td>CHAR(2)</td>
<td>type of object:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ IP (index partition)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ IX (index)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ TB (table)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ TP (table space partition)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ TS (table space)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ DD, DW, D1, D2 (dynamic work file allocation)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ CI (copy information)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RD (restart data set block)</td>
</tr>
<tr>
<td>PARTITION</td>
<td>SMALLINT</td>
<td>physical partition number:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ null or 0 for a single data set nonpartitioned space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ data set number for a multi-data-set, nonpartitioned space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ partition number for a partitioned space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COPY PLUS, LOADPLUS, UNLOAD PLUS, CHECK PLUS, DASD MANAGER PLUS, and REORG PLUS use null or 0 for any nonpartitioned space.</td>
</tr>
<tr>
<td>BMCID</td>
<td>SMALLINT</td>
<td>internal identifier of the object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
</tbody>
</table>
Table 114  BMCSYNC table (part 2 of 3)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILNAME</td>
<td>CHAR(8)</td>
<td>name of the executing utility:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- STATS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LOAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- RECOVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- REORG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UNLOAD</td>
</tr>
<tr>
<td>SHRLEVEL</td>
<td>CHAR(1)</td>
<td>degree to which utilities can share this object:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Blank means that no status is requested, and any other utility can obtain any status.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- S allows sharing among any number of SHRLEVEL S utilities.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- X indicates that exclusive control is required. No other utility can run with SHRLEVEL X.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see Table 10 on page 83.</td>
</tr>
<tr>
<td>STATUS</td>
<td>CHAR(1)</td>
<td>status of the utility or object:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- blank (indicates no processing has been done)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- C (for CHECK PLUS, indicates checked)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- L (for LOADPLUS, indicates loaded)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- U (for UNLOAD PLUS, indicates unloaded)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- R (for REORG PLUS, indicates reloaded)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>XCOUNT</td>
<td>INTEGER</td>
<td>number of rows or keys processed in the current phase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>DDNAME</td>
<td>CHAR(8)</td>
<td>check, load, unload, or work ddname</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>INTEGER</td>
<td>number of blocks for the check, load, unload, or work data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>ORIG_STATUS</td>
<td>CHAR(8)</td>
<td>encoded representation of the original DB2 status of the space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For RECOVER PLUS, this column restores the DB2 status of a space after recovery, if necessary.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>EXTRBA</td>
<td>CHAR(6)</td>
<td>(RECOVER PLUS) log point at which this space was externalized</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> RECOVER PLUS no longer uses EXTRBA.</td>
</tr>
</tbody>
</table>

DASD MANAGER PLUS does not use this column.
Considerations for the BMCSYNC table

Note the following considerations:

- You might need to increase the size of the BMCSYNC table space from the standard size that was allocated during installation when any of the following conditions exists:
  
  — You are processing a large number of partitions.

  Estimate this allocation based on the following factors:

  ■ number of utilities that you are executing concurrently
  ■ number of partitions that you are processing concurrently
  ■ number of files that you are allocating dynamically

  — You are loading or unloading XML data and the XML table space is partition-by-growth.

  Estimate this allocation based on the following factors:

  ■ number of utilities that you are executing concurrently
  ■ number of XML columns that you are loading or unloading
  ■ value of MAXPARTITIONS (a minimum of 256 partitions in this case)
  ■ number of files that you are allocating dynamically

---

Table 114  BMCSYNC table (part 3 of 3)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATE</td>
<td>LONG VARCHAR</td>
<td>restart information for the space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, the STATE indicates the object state and sync information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>INSTANCE</td>
<td>SMALLINT</td>
<td>(RECOVERY MANAGER and RECOVER PLUS) the instance number of the current base</td>
</tr>
<tr>
<td></td>
<td></td>
<td>objects (table and index)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is 1. The other utilities do not use this column.</td>
</tr>
</tbody>
</table>

a (LOADPLUS, UNLOAD PLUS, CHECK PLUS, and REORG PLUS) If the value for NAME1 would exceed 8 bytes or the value for NAME2 would exceed 18 bytes, NAME1 contains the DBID for the object; NAME2 contains the table OBID or index ISOBID of the object in hexadecimal format.
— You are loading or unloading LOB data.

Estimate this allocation based on the following factors:

- number of utilities that you are executing concurrently
- number of LOB columns that you are loading or unloading
- number of partitions in the base table space
- number of files that you are allocating dynamically

- If BMCSTATS is processing multiple objects and encounters an object that is held by another utility, the BMCSTATS job issues a warning. The warning identifies the object and the utility that is using it. BMCSTATS continues processing the next object.

- If BMCSTATS is processing an object and another utility requires exclusive control of that object, the other utility stops execution at initialization time.

Maintaining the BMCSYNC table

When a utility abends, rows might remain in the BMCSYNC table. If you need to control expansion of this table, use one of the following methods to delete rows:

- Use the TERM restart parameter on the EXEC statement to delete rows from the BMCUTIL and BMCSYNC tables. Do not delete any rows for instances of utilities that are awaiting restart.

- Delete invalid rows in the BMCUTIL table. Do not delete any rows for instances of utilities that are awaiting restart.

Then use the following SQL statement to delete rows from the BMCSYNC table:

```
DELETE
FROM creatorName.CMN_BMCSYNC
WHERE UTILID NOT IN
  (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
```

**NOTE**

The names of the BMCUTIL and BMCSYNC tables might have been changed at your site during installation.
Table 115 describes the contents of the BMCTRANS table, which contains information that RECOVERY MANAGER and Log Master use for transaction recovery. The table contains one row for each execution of Log Master (one row for each log scan).

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERID</td>
<td>CHAR(8) NOT NULL</td>
<td>transaction creator</td>
</tr>
<tr>
<td>TRANID</td>
<td>VARCHAR(18) NOT NULL</td>
<td>transaction ID</td>
</tr>
<tr>
<td>STARTTIME</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>transaction start time</td>
</tr>
<tr>
<td>PITRBA</td>
<td>CHAR(6) NOT NULL FORBIT DATA</td>
<td>RBA for point-in-time recovery</td>
</tr>
<tr>
<td>OUTDSNAME</td>
<td>VARCHAR(35) NOT NULL</td>
<td>output data set prefix for SQL statements or the logical log</td>
</tr>
<tr>
<td>STATE³</td>
<td>SMALLINT NOT NULL</td>
<td>level of recovery analysis performed</td>
</tr>
<tr>
<td>PITTIME</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>timestamp for the PIT RBA</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT NOT NULL</td>
<td>sequence number of the filter text</td>
</tr>
<tr>
<td>PITWKEST</td>
<td>FLOAT NOT NULL</td>
<td>work estimate</td>
</tr>
<tr>
<td>FILTERLINE</td>
<td>VARCHAR(1040) NOT NULL</td>
<td>text of the filter (may span more than one row)</td>
</tr>
<tr>
<td>UNDONUMROWSUPD</td>
<td>FLOAT</td>
<td>number of unique rows (RIDs) that are selected by the filter of the log scan</td>
</tr>
<tr>
<td>UNDOSUBSEQUPDROWS</td>
<td>FLOAT</td>
<td>total number of anomaly log records relating to one of the rows (RIDs) selected by the log scan</td>
</tr>
<tr>
<td>UNDOLOGRECROWS</td>
<td>FLOAT</td>
<td>number of unique rows (RIDs) that are affected by an anomaly log record</td>
</tr>
<tr>
<td>UNDOJOBSTATUS</td>
<td>SMALLINT</td>
<td>code indicating the status of an UNDO log scan:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0 (no action taken)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 (Log Master execution started)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 (Log Master execution completed successfully with return code 0,4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 3 (Log Master execution completed unsuccessfully with return code 8,12)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 4 (Log Master execution abnormally ended)</td>
</tr>
</tbody>
</table>
Table 116 describes the BMCUTIL table, which contains information about utilities that are currently running or started. The utilities use the table to control the use of utility IDs. Each BMC utility must have a unique ID for restart purposes. If you have more than one BMC utility installed, all of these utilities should share the same BMCUTIL table.

The utilities insert rows into the BMCUTIL table during the UTILINIT phase and update the table as the job status changes. The utilities delete rows from the BMCUTIL table during the UTILTERM phase.

Table 116  BMCUTIL table  (part 1 of 3)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>utility identifier</td>
</tr>
<tr>
<td>STATUS</td>
<td>CHAR(1)</td>
<td>execution status of the utility:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- A (active, not executing command)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I (initializing)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- P (pausing or pause-stopped)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- S (stopped)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- T (terminating)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- X (executing command)</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>UTILNAME</td>
<td>CHAR(8)</td>
<td>name of the executing utility:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- STATS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LOAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- RECOVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- REORG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UNLOAD</td>
</tr>
<tr>
<td>PHASE</td>
<td>CHAR(8)</td>
<td>current phase of the utility</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- COPY PLUS does not use this column.</td>
</tr>
<tr>
<td>USERID</td>
<td>CHAR(8)</td>
<td>user ID executing the utility</td>
</tr>
<tr>
<td>SSID</td>
<td>CHAR(4)</td>
<td>DB2 subsystem where the utility is running</td>
</tr>
<tr>
<td>RESTART</td>
<td>CHAR(1)</td>
<td>restart option:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- N (not restart)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- P (RESTART(PHASE))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Y (RESTART)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Although UNLOAD PLUS accepts the RESTART, RESTART(PHASE), NEW/RESTART, and NEW/RESTART(PHASE) parameters, the utility executes as though you had specified the NEW parameter.</td>
</tr>
<tr>
<td>NOTEID</td>
<td>CHAR(8)</td>
<td>TSO user ID to be notified</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>(RECOVER PLUS and REORG PLUS) name of the database containing the table or index space for which the last checkpoint was taken</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- This value can be blank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The other utilities do not use this column.</td>
</tr>
<tr>
<td>SPNAME</td>
<td>CHAR(8)</td>
<td>(RECOVER PLUS and REORG PLUS) name of the table or index space for which the last checkpoint was taken</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- This value can be blank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The other utilities do not use this column.</td>
</tr>
<tr>
<td>SPSTATUS</td>
<td>CHAR(5)</td>
<td>(REORG PLUS) space status before the utility stopped</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The other utilities do not use this column.</td>
</tr>
</tbody>
</table>
Maintaining the BMCUTIL table

When a utility abends, rows might remain in the BMCUTIL table. If you need to control expansion of this table, use one of the following methods to delete rows:

- Use the TERM restart parameter on the EXEC statement to delete rows from the BMCUTIL and BMCSYNC tables. Do not delete any rows for instances of utilities that are awaiting restart.

- Delete invalid rows in the BMCUTIL table. Do not delete any rows for instances of utilities that are awaiting restart.

Then use the following SQL statement to delete rows from the BMCSYNC table.

```
DELETE
FROM  creatorName.CMN_BMCSYNC
WHERE UTILID NOT IN
    (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
```

**NOTE**

The names of the BMCUTIL and BMCSYNC tables might have been changed at your site during installation.

---

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMANDNO</td>
<td>SMALLINT</td>
<td>not used (always 0)</td>
</tr>
<tr>
<td>COMMAND</td>
<td>VARCHAR(256)</td>
<td>first 256 characters of the utility command text</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVER PLUS, DASD MANAGER PLUS, and COPY PLUS do not use this column.</td>
</tr>
<tr>
<td>STATE</td>
<td>LONG VARCHAR</td>
<td>utility state and sync information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>starting timestamp of the utility</td>
</tr>
</tbody>
</table>
BMCXCOPY table

Table 117 describes the contents of the BMCXCOPY table, which the BMC utilities use for tracking the following types of registered copies:

- Indexes that COPY PLUS has copied:
  - COPY NO index copies
  - DSNUM n index (nonpartitioned) copies
  - Incremental index copies
  - Index copies that are made at data set level

- Instant Snapshots made by COPY PLUS with the BMC EXTENDED BUFFER MANAGER (XBM) product or its SNAPSHOT UPGRADE FEATURE (SUF) technology, and any standard copies made in association with the Instant Snapshot

- Online consistent copies

- Cabinet copies

- Encrypted copies

The BMCXCOPY table functions like SYSIBM.SYSCOPY except that IXNAME replaces TSNAME in BMCXCOPY. You must control authorization and access to users for BMCXCOPY through standard DB2 authorization.

If you have more than one BMC utility installed, all of these utilities should share the same BMCXCOPY table.

Table 117  BMCXCOPY table (part 1 of 5)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>name of the database</td>
</tr>
<tr>
<td>IXNAME</td>
<td>CHAR(8)</td>
<td>name of the index space or table space for Instant Snapshots and associated copies</td>
</tr>
<tr>
<td>DSNUM</td>
<td>INTEGER</td>
<td>data set number within the index or table space</td>
</tr>
<tr>
<td>ICTYPE</td>
<td>CHAR(1)</td>
<td>operation type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- F (COPY FULL YES; for COPY PLUS version 8.1 and later, online consistent copies)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I (COPY FULL NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- W (REORG LOG NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- B (REBUILD INDEX)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- P (POINT-IN-TIME RECOVERY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- C (for COPY PLUS version 7.3 and earlier, online consistent copies)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- s (used by COPY PLUS to track system pages)</td>
</tr>
</tbody>
</table>
### BMXCOPY table (part 2 of 5)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICDATE</td>
<td>CHAR(6)</td>
<td>date of the entry ( \text{yyyyMMdd} )</td>
</tr>
</tbody>
</table>
| START_RBA     | VARCHAR(10)  | the relative byte location of a point in the DB2 recovery log \[The indicated point is as follows:
- for ICTYPE F, the starting point for all updates since the image copy was taken
- for COPY_TYPE O, the minimum of the consistent point and the oldest inflight URID
- \((\text{RECOVERY MANAGER})\) for ICTYPE C, the consistent log point for the copy
  - RBA for non-data-sharing systems
  - LRSN for data sharing systems]
| FILESEQNO     | INTEGER      | tape file sequence number of the copy                                                                                                       |
| DEVTYPE       | CHAR(8)      | type of device on which the copy resides                                                                                                     |
| IBMREQD       | CHAR(1)      | whether the row came from the basic machine-readable material (MRM) tape:  
  - N (NO)
  - Y (YES)
| DSNAME        | CHAR(44)     | name of the data set \[If STYPE V, DSNAME is the name of the VSAM data component.\]                                                            |
| ICTIME        | CHAR(6)      | time at which this row was inserted \( \text{hhmmss} \) \[The insertion takes place after the completion of the operation that the row represents.\] |
| SHRLEVEL      | CHAR(1)      | SHRLEVEL parameter on COPY if ICTYPE F:  
  - C (change)
  - R (reference)
| DSVOLSER      | VARCHAR(1784)| volume serial numbers of the data set \[Commas separate items in a list of 6-byte numbers. This column is blank if the data set is cataloged.\] |
| TIMESTAMP     | TIMESTAMP    | date and time when the row was inserted \[This is the date and time that are recorded in ICDATE and ICTIME. The use of TIMESTAMP over ICDATE and ICTIME is recommended, because later DB2 releases might not support the latter two columns.\] |
Table 117  BMCXCOPY table (part 3 of 5)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICBACKUP</td>
<td>CHAR(2)</td>
<td>type of image copy contained in the data set:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ LB (data set contains local backup data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RP (data set contains recovery system main data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RB (data set contains recovery system backup data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ blank (data set contains local system main data or is not one of multiple copies)</td>
</tr>
<tr>
<td>ICUNIT</td>
<td>CHAR(1)</td>
<td>media on which the image copy data set is stored:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ D (DASD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ T (tape)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ blank (medium is neither tape nor DASD)</td>
</tr>
<tr>
<td>STYPE</td>
<td>CHAR(1)</td>
<td>type of copy:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ blank (for ICTYPE=F)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ V (Instant Snapshot or a VSAM data set)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ e (encrypted copy)</td>
</tr>
<tr>
<td>PIT_RBA</td>
<td>VARCHAR(10)</td>
<td>point-in-time recovery:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ X'000000000000' (for ICTYPE=F)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ consistent point (for COPY_TYPE=O)</td>
</tr>
<tr>
<td>GROUP_MEMBER</td>
<td>CHAR(8)</td>
<td>data-sharing group member (the name of the SSID where the copy was made)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This column is blank if you are not using data sharing.</td>
</tr>
<tr>
<td>OTYPE</td>
<td>CHAR(1)</td>
<td>type of object:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ T (table)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ I (index)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ i (compressed index)</td>
</tr>
<tr>
<td>LOWDSNUM</td>
<td>INTEGER</td>
<td>not used</td>
</tr>
<tr>
<td>HIGHDSNUM</td>
<td>INTEGER</td>
<td>not used</td>
</tr>
<tr>
<td>COPY_PAGESF</td>
<td>FLOAT(53)</td>
<td>number of pages written to the copy data set</td>
</tr>
<tr>
<td>NPAGESF</td>
<td>FLOAT(53)</td>
<td>high-used RBA divided by the page size</td>
</tr>
<tr>
<td>CPAGESF</td>
<td>FLOAT(53)</td>
<td>total number of changed pages</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>CHAR(8)</td>
<td>job name</td>
</tr>
<tr>
<td>AUTHID</td>
<td>CHAR(8)</td>
<td>authorization ID</td>
</tr>
<tr>
<td>OLDEST_VERSION</td>
<td>SMALLINT</td>
<td>when ICTYPE= B, F, I, S, W, or X, the version number of the oldest format of data for an object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For other values of ICTYPE, the value is –1.</td>
</tr>
<tr>
<td>LOGICAL_PART</td>
<td>INTEGER</td>
<td>logical partition number</td>
</tr>
</tbody>
</table>
Table 117  BMCXCOPY table (part 4 of 5)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGGED</td>
<td>CHAR(1)</td>
<td>logging attribute of the table space:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (logged)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (not logged)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ blank (row inserted prior to DB2 version 9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For a non-LOB table space or index space, blank indicates that the logging attribute is logged.</td>
</tr>
<tr>
<td>TTYPE</td>
<td>CHAR(8)</td>
<td>row format for the table space or partition:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RRF (reordered row format)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ BRF (basic row format)</td>
</tr>
<tr>
<td>INSTANCE</td>
<td>SMALLINT</td>
<td>instance number of the current base objects (table and index)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is 1.</td>
</tr>
<tr>
<td>RELCREATED</td>
<td>CHAR(1)</td>
<td>DB2 release that created the object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the release is earlier than version 9, the value is blank.</td>
</tr>
<tr>
<td>COPY_TYPE</td>
<td>CHAR(1)</td>
<td>type of copy:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ C (cabinet copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ O (online consistent copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ blank (default value)</td>
</tr>
<tr>
<td>NOTE_VALUE</td>
<td>CHAR(4)</td>
<td>encoded value that quickly locates data for a specific space in a cabinet copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is blank.</td>
</tr>
<tr>
<td>NOTE_TYPE</td>
<td>CHAR(1)</td>
<td>type of NOTE (issued by COPY PLUS):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ A (ABS — tape)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ R (REL — disk)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ F (frame)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ blank (default value)</td>
</tr>
<tr>
<td>OCC_COPY_RBA</td>
<td>VARCHAR(10)</td>
<td>original START_RBA of an online consistent copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is blank.</td>
</tr>
<tr>
<td>OCC_LOCKRULE</td>
<td>CHAR(1)</td>
<td>locking rule for a table space (not used for indexes):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ A (for page level)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ R (for row level)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ blank (default value)</td>
</tr>
<tr>
<td>OCC_SPACE_ALTERED</td>
<td>CHAR(1)</td>
<td>whether the space was altered:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (altered)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (not altered)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ blank (default value)</td>
</tr>
<tr>
<td>CAB_BLOCKS</td>
<td>INTEGER</td>
<td>total number of frames written for a cabinet copy</td>
</tr>
</tbody>
</table>
Periodically, you should review BMCXCOPY and delete old rows to control its expansion. To delete all rows from the BMCXCOPY table that are older than 30 days, use the following statement as an example:

```
DELETE
FROM creatorName.CMN_BMCXCOPY
WHERE DAYS(CURRENT_TIMESTAMP) - DAYS(TIMESTAMP) > 30;
```
REORG PLUS user exits

This appendix presents the following topics:

Overview ................................................................. 788
Accessing the sample user exits ................................... 788
Using DSNUEXIT to construct data set name patterns .......... 789
   DSNUEXIT requirements and considerations .................. 789
   DSNUEXIT return codes ........................................... 790
   DSNUEXIT user-defined variables ............................... 791
   DSNUEXIT assembler user exit ................................... 791
   DSNUEXIT COBOL II and LE COBOL user exit ................. 805
   DSNUEXIT C user exit .............................................. 814
   DSNUEXIT LE C user exit ......................................... 823
Using a DSRSEXIT, TERMEXIT, or MAPTEXIT user exit ....... 832
   DSRSEXIT, TERMEXIT, and MAPTEXIT common restrictions . 832
   DSRSEXIT, TERMEXIT, and MAPTEXIT common variables and return codes 832
Using DSRSEXIT to manage VSAM data set redefinition .......... 834
   Resizing DB2 objects ............................................... 835
   Setting REDEFINE NO ............................................. 835
   Using the DSRSEXIT user exit to order storage group volumes 836
   Using the DSRSEXIT user exit to add SMS classes ............ 836
   DSRSEXIT requirements and restrictions ....................... 836
   Running the DSRSEXIT exit ....................................... 837
   DSRSEXIT variables ................................................ 838
   Modifying DSRSEXIT variables ................................... 841
   Sample DSRSEXIT REXX user exit ............................... 844
Using TERMEXIT to control BMCHIST and statistics updates ...... 849
   Additional TERMEXIT restrictions ............................... 850
   Running the TERMEXIT exit ....................................... 850
   TERMEXIT variables ................................................. 851
   Sample TERMEXIT REXX user exit ............................... 852
Using MAPTEXIT to create mapping objects ....................... 857
   Additional MAPTEXIT requirements and restrictions .......... 858
   Supplying MAPTEXIT in your STEPLIB ........................... 858
   Supplying MAPTEXIT in a SYSEXEC DD statement ............ 859
   MAPTEXIT variables ................................................. 859
   Sample MAPTEXIT REXX exit ...................................... 861
The REORG PLUS for DB2 product allows you to customize your reorganization processing by using user-written exits. Table 118 lists the following information:

- sample user exits that REORG PLUS provides
- corresponding option to specify on the REORG command
- languages in which you can write the exit

The exits are listed in the order in which REORG PLUS passes control to them.

### Table 118  User exits that REORG PLUS provides

<table>
<thead>
<tr>
<th>Exit routine purpose</th>
<th>Command option</th>
<th>Language</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>creating data set name patterns</td>
<td>DSNUEXIT</td>
<td>Assembler, COBOL II, LE COBOL, C, LE C</td>
<td>789</td>
</tr>
<tr>
<td>managing the redefinition of DB2® VSAM data sets</td>
<td>DSRSEEXIT</td>
<td>REXX</td>
<td>832, 834</td>
</tr>
<tr>
<td>controlling updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS updates</td>
<td>TERMEXIT</td>
<td>REXX</td>
<td>832, 849</td>
</tr>
<tr>
<td>creating mapping objects for use with DSNUTILB</td>
<td>MAPTEEXIT</td>
<td>REXX</td>
<td>832, 857</td>
</tr>
</tbody>
</table>

### Accessing the sample user exits

Source code for the sample user exits is distributed in the REORG PLUS HLQ.LLQ.SAMP library (where HLQ is the high-level qualifier specified during installation and LLQ is the low-level qualifier or prefix set during installation). Table 119 lists the library members that contain the sample user exits.

### Table 119  Library member names of user exits (part 1 of 2)

<table>
<thead>
<tr>
<th>User exit</th>
<th>Language</th>
<th>Library member name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUEXIT</td>
<td>Assembler</td>
<td>ARUEDSNA</td>
</tr>
<tr>
<td></td>
<td>COBOL II and LE COBOL</td>
<td>ARUEDSN2</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>ARUEDSC</td>
</tr>
<tr>
<td></td>
<td>LE C</td>
<td>ARUEDSL</td>
</tr>
<tr>
<td>DSRSEEXIT</td>
<td>REXX</td>
<td>DSRSREXX</td>
</tr>
</tbody>
</table>
Using DSNUEXIT to construct data set name patterns

The user exit for data set name patterns, DSNUEXIT, provides REORG PLUS with user-defined variables that you can use in constructing data set name patterns for the DSNPAT command or installation option, and the SPIILDSNP installation or SPIILDSNPAT command option. For information about the command options, see Chapter 3, “Syntax of the REORG command.” For information about the installation options, see Appendix A, “REORG PLUS installation options.”

To use a sample DSNUEXIT user exit from the HQL.LLQSAMP library, copy the appropriate member as the base code for your exit and edit the code. Then, compile or assemble and link the edited code into an authorized library so that the exit can be loaded during execution.

DSNUEXIT requirements and considerations

The DSNUEXIT user exit has the following requirements and considerations:

- REORG PLUS supports DSNUEXIT user exits that are written in the following languages:
  - assembler
  - COBOL II
  - IBM Language Environment COBOL (LE COBOL)
  - C
  - IBM Language Environment C (LE C)

- The library in which this exit resides must be included in your system’s LINKLIST or in your JOBLIB or STEPLIB.

- For COBOL II and C programs, the appropriate runtime libraries must be authorized and must be in your LINKLIST, JOBLIB, or STEPLIB.

---

Table 119  Library member names of user exits (part 2 of 2)

<table>
<thead>
<tr>
<th>User exit</th>
<th>Language</th>
<th>Library member name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPTEXIT</td>
<td>REXX</td>
<td>MAPTREXX</td>
</tr>
<tr>
<td>TERMEXIT</td>
<td>REXX</td>
<td>TERMREXX</td>
</tr>
</tbody>
</table>
For LE COBOL and LE C programs, the appropriate language environment runtime libraries must be authorized and must be in your LINKLIST, JOBLIB, or STEPLIB.

Your routine must be reentrant.

You cannot use these exit routines with a DSNUTILB reorganization.

Dynamic data set allocation must be active.

You can supply the module name and program language in the DSNUEXIT installation or command option. The DSNUEXIT command option overrides any value in the installation options. For more information, see page 240 or page 688.

The phase in which REORG PLUS invokes your user-written exit depends on whether you run a two-phase or single-phase reorganization:

— For a two-phase reorganization, REORG PLUS invokes your user-written exit at the beginning of the UNLOAD phase.

— For a single-phase reorganization, REORG PLUS invokes your user-written exit at the beginning of the REORG phase.

### DSNUEXIT return codes

REORG PLUS communicates with the DSNUEXIT exit by passing parameters to the exit and receiving a return code and user-defined variables from the exit. Table 120 lists where you can find the return code.

<table>
<thead>
<tr>
<th>Language</th>
<th>Return code location</th>
</tr>
</thead>
<tbody>
<tr>
<td>assembler</td>
<td>contents of register 15</td>
</tr>
<tr>
<td>COBOL II</td>
<td>contents of the RETURN-CODE variable</td>
</tr>
<tr>
<td>LE_COBOL</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>returned by the return function</td>
</tr>
<tr>
<td>LE_C</td>
<td></td>
</tr>
</tbody>
</table>

Valid return codes and their results are as follows:

- 0 indicates a good return and processing continues.
- Other return codes indicate an error and the utility terminates the job.
DSNUEXIT user-defined variables

The DSNUEXIT user exit returns user-defined variables to REORG PLUS. You can use these variables with the DSNPAT installation or command option to construct data set name patterns. REORG PLUS uses these patterns and the values returned for any user-defined variables to create data set names during dynamic data set allocation. You can also use these variables with the SPIILDSNP installation option or SPIILDSNPAT command option to construct patterns for data set name prefixes. REORG PLUS uses these patterns and the values returned for any user-defined variables to create the data set name prefix for the spill data sets used when running a SHRLEVEL CHANGE reorganization.

The following rules apply for creating a user-defined exit to create your variables:

- variable names must meet the following requirements:
  - begin with an underscore character
  - begin in the leftmost byte of the XPUVNAME field
  - be padded on the right with blanks if the variable name is shorter than 9 bytes
  - not contain embedded blanks
  - contain the following national characters for nonblank characters after the underscore:
    - A through Z
    - 0 through 9
    - #, @, $

- variable data must meet the following requirements:
  - begin in the leftmost byte of the XPUVDATA field
  - be padded on the right with blanks if the variable data is shorter than 8 bytes
  - not contain embedded blanks
  - consist of the following characters for nonblank characters:
    - A through Z
    - 0 through 9
    - #, @, $, –, .

DSNUEXIT assembler user exit

This section provides a sample exit parameter block, variable mapping structure, and exit that you can use as an example of how to write this user exit in assembler.
Exit parameter block DSECT

The exit parameter block DSECT (Figure 70) contains both input and output fields. Input fields pass vital information to the user exit, such as the database name and user ID. If you make any modifications to these input fields, they are disregarded on return. The output fields pass information about your user variables back to REORG PLUS.

Figure 70   DSNUEXIT assembler exit parameter block (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPJOBN</td>
<td>CL8</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>XPSTEP</td>
<td>CL8</td>
<td>STEPNAME</td>
</tr>
<tr>
<td>XPDBNAME</td>
<td>CL8</td>
<td>DATABASE NAME</td>
</tr>
<tr>
<td>XPSPNAME</td>
<td>CL8</td>
<td>SPACE NAME</td>
</tr>
<tr>
<td>XPRTYPE</td>
<td>CL2</td>
<td>REORG TYPE (TS OR IX)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DS CL2 RESERVED FOR REORG PLUS</td>
</tr>
<tr>
<td>XPUSER</td>
<td>CL8</td>
<td>USER ID</td>
</tr>
<tr>
<td>XPSSID</td>
<td>CL4</td>
<td>DB2 SUBSYSTEM ID</td>
</tr>
<tr>
<td>XPDATE</td>
<td>CL6</td>
<td>UTILITY EXECUTION DATE MMDDYY</td>
</tr>
<tr>
<td>XPTIME</td>
<td>CL6</td>
<td>UTILITY EXECUTION TIME HHMMSS</td>
</tr>
<tr>
<td>XPUTILID</td>
<td>CL16</td>
<td>UTILITY ID</td>
</tr>
<tr>
<td>XPDATE8</td>
<td>CL8</td>
<td>UTILITY EXECUTION DATE MMDDYYYY</td>
</tr>
<tr>
<td>XPGRPNM</td>
<td>CL4</td>
<td>DATA SHARING GROUP NAME</td>
</tr>
<tr>
<td>XPVCAT</td>
<td>CL8</td>
<td>VCAT NAME (FROM 1ST PART IF PARTITNED)</td>
</tr>
<tr>
<td>XPDATEJ</td>
<td>CL7</td>
<td>UTILITY EXECUTION DATE CCYYDDD</td>
</tr>
<tr>
<td></td>
<td>CL13</td>
<td>RESERVED FOR REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER WORK AREA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XSUSIWD1 DS F USER WORD 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XSUSIWD2 DS F USER WORD 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XSUSIWD3 DS F USER WORD 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XSUSIWD4 DS F USER WORD 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XSUSIWD5 DS F USER WORD 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XSUSIWD6 DS F USER WORD 6</td>
</tr>
</tbody>
</table>
## DSECT fields

Table 121 describes the major DSECT fields for the DSNUEXIT assembler user exit parameter block and their uses.

### Table 121  Major DSECT fields for the DSNUEXIT assembler user exit parameter block (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPJOBN</td>
<td>job name</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>XSTEP</td>
<td>step name</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>XPDBNAME</td>
<td>database name</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>XSPNAME</td>
<td>name of the table space or index space from the REORG command</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>XPRTYPE</td>
<td>type of reorganization being performed, table space (TS) or index (IX)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>XUSER</td>
<td>USERID of the user running REORG PLUS</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>XSSID</td>
<td>DB2 subsystem ID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>XDATE</td>
<td>date of the execution of the utility, in the format MMDDYY</td>
<td>6 bytes</td>
</tr>
<tr>
<td>XTIME</td>
<td>time of the execution of the utility, in the format HHMMSS</td>
<td>6 bytes</td>
</tr>
<tr>
<td>XPUTILID</td>
<td>utility ID</td>
<td>up to 16 bytes</td>
</tr>
<tr>
<td>XDATE8</td>
<td>date of the execution of the utility, in the format MMDDYYYY</td>
<td>8 bytes</td>
</tr>
<tr>
<td>XGRPNM</td>
<td>DB2 data sharing group name</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>In a non-data sharing environment, the field contains the DB2 SSID.</td>
<td></td>
</tr>
<tr>
<td>XVCAT</td>
<td>VCATNAME specified in the DB2 catalog for the table space being reorganized, or for the first partition if the table space is partitioned</td>
<td>Not applicable</td>
</tr>
<tr>
<td>XDATEJ</td>
<td>Julian date of the execution of the utility, in the format CCYYDDD</td>
<td>7 bytes</td>
</tr>
<tr>
<td>XUSRWD1...</td>
<td>work space.</td>
<td>up to 4 bytes each</td>
</tr>
</tbody>
</table>

---

**Figure 70  DSNUEXIT assembler exit parameter block (part 2 of 2)**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>XPUSRWD7</td>
<td>DS</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER WORD 7</td>
</tr>
<tr>
<td>XPUSRWD8</td>
<td>DS</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER WORD 8</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>OUTPUT AREA</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>XPUVAREA</td>
<td>DS</td>
<td>100CL17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER VARIABLE AREA</td>
</tr>
<tr>
<td>XPUVENT#</td>
<td>EQU</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NUMBER OF ENTRIES IN OUTPUT AREA</td>
</tr>
<tr>
<td>XP$</td>
<td>EQU</td>
<td>*-ARUDSNXP</td>
</tr>
</tbody>
</table>
Variable mapping block DSECT

The variable mapping block DSECT, as shown in Figure 71, defines the output area of the exit parameter DSECT. Using this DSECT allows you to easily address the variable definition table. The number of entries in this table (each entry consists of both a variable name and its value) must not exceed 100.

Table 121  Major DSECT fields for the DSNUEXIT assembler user exit parameter block (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPUVAREA</td>
<td>area containing user-defined variable information</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>For information about establishing user-defined variables, see page 791.</td>
<td></td>
</tr>
<tr>
<td>XPUVENT#</td>
<td>maximum number of entries in the output variable area</td>
<td>up to 100 entries</td>
</tr>
</tbody>
</table>

Figure 71  DSNUEXIT assembler variable mapping block (part 1 of 2)

---

* XPUVARS DEFINES THE OUTPUT AREA OF THE ARUDSNXP DSECT AT LABEL XPUVAREA.
* YOU MAY USE THIS DSECT TO EASILY ADDRESS THE TABLE DEFINED AT XPUVAREA.

* NOTE:

* THE NUMBER OF ENTRIES IN THE XPUVAREA TABLE MUST NOT EXCEED 100
* OR YOU WILL ADDRESS BEYOND THE END OF THE GETMAIN STORAGE PROVIDED FOR THIS ROUTINE.

* VARIABLE NAME:

* 1. MUST BEGIN WITH AN UNDERSCORE CHARACTER C'_' X'6D'
* 2. MUST BEGIN IN THE LEFTMOST BYTE OF XPUVNAME FIELD
* 3. IF LESS THAN 9 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS
* 4. MAY NOT CONTAIN EMBEDDED BLANKS
* 5. NON-BLANK CHARACTERS AFTER THE LEFTMOST BYTE MUST BE NATIONAL CHARACTERS WHICH ARE A THROUGH Z, 0 THROUGH 9, #, @, AND $

* VARIABLE DATA:

* 1. MUST BEGIN IN THE LEFTMOST BYTE OF XPUVDATA FIELD
* 2. IF LESS THAN 8 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS
* 3. MAY NOT CONTAIN EMBEDDED BLANKS
* 4. NON-BLANK CHARACTERS MUST BE 'NATIONAL' CHARACTERS WHICH ARE A THROUGH Z, 0 THROUGH 9, #, @, $, -, AND .
Table 122 describes the major DSECT fields for the DSNUEXIT assembler variable mapping block and their uses.

**Table 122  Major DSECT fields for the DSNUEXIT assembler variable mapping block**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPUVNAME</td>
<td>name of the variable you are defining</td>
</tr>
<tr>
<td>XPUVDATA</td>
<td>value of the variable you are defining</td>
</tr>
<tr>
<td>XPUVENT$</td>
<td>length of each of these variable entries in the table</td>
</tr>
</tbody>
</table>

**Sample assembler user exit**

Figure 72 shows an example of this user exit written in assembler.

**Figure 72  DSNUEXIT sample assembler user exit (part 1 of 11)**

```
ARUEDSNA TITLE 'ARUEDSNA - REORG PLUS USER EXIT USER EXAMPLE - V42'
ARUEDSNA CSECT
ARUEDSNA AMODE 31
ARUEDSNA RMODE 24

*---------------------------------------------------------------------*
*                        D I S C L A I M E R                          *
*---------------------------------------------------------------------*
*                                                                     *
*  THIS IS A SAMPLE REORG PLUS USER EXIT. THIS EXIT WOULD BE USED    *
*  IN ORDER TO DEFINE USER VARIABLES FOR BUILDING DATASET NAMES FOR  *
*  DYNAMIC WORKFILE ALLOCATION WITH THE DSNPAT KEYWORD AND FOR        *
*  SHRLEVEL CHANGE WITH THE SPIILDSNPAT KEYWORD.                      *
*                                                                     *
*  THIS EXIT WILL ONLY BE INVOKED WHEN IT IS SPECIFICALLY NAMED       *
*  IN THE ARU$OPTS DSNUEXIT PARAMETER OR IN THE DSNUEXIT PARAMETER OF  *
*  THE REORG COMMAND. THE SPECIFICATION IN THE REORG COMMAND WILL     *
*  OVERRIDE THE SPECIFICATION IN THE ARU$OPTS MACRO. IF THE NAME       *
*  SPECIFIED IS 'NONE' WITHOUT THE SINGLE QUOTE MARKS, THE EXIT WILL   *
*  NOT BE INVOKED.                                                    *
*                                                                     *
*  NOTE: PLEASE REVIEW THE DOCUMENTATION IN THE REFERENCE MANUAL,     *
*  AND THE FOLLOWING USAGE NOTES BEFORE IMPLEMENTING THIS EXIT.       *
*                                                                     *
*  PLEASE CALL BMC SOFTWARE CUSTOMER SUPPORT WITH ANY QUESTIONS YOU   *
```
**DSNUEXIT assembler user exit**

Figure 72  DSNUEXIT sample assembler user exit (part 2 of 11)

| * MAY HAVE IN THIS AREA.                           |
| *                                              |
| * PHONE: 1-800-537-1813                          |
| *                                              |
| SPACE                                           |
| NOTES                                          |
| * ARUEDSNA IS CALLED ONCE AT THE BEGINNING OF THE UNLOAD PHASE FOR A |
| * TWO-PHASE REORGANIZATION OR AT THE BEGINNING OF THE REORG PHASE FOR |
| * A SINGLE-PHASE REORGANIZATION.                |
| * WHEN INVOKED, R1 CONTAINS THE ADDRESS OF A USER EXIT BLOCK |
| * DESCRIBED BY THE ARUDSNXP DSECT.              |
| * R13 CONTAINS THE ADDRESS OF THE CALLER’S SAVE AREA |
| * R14 CONTAINS THE CALLER’S RETURN ADDRESS      |
| * R15 CONTAINS THE ENTRY POINT ADDRESS OF THIS EXIT |
| * UPON EXIT R15 CONTAINS THE RETURN CODE FROM THIS EXIT |
| * R15=0 GOOD RETURN                             |
| * R15=OTHER ERROR RETURN. TERMINATE UTILITY    |
| * ARUDSNXP DSECT THE AREA MAPPED BY THIS DSECT CONTAINS BOTH INPUT |
| * AND OUTPUT FIELDS. INPUT FIELDS ARE PROVIDED TO |
| * PASS VITAL INFORMATION TO THIS EXIT. ANY MODIFI- |
| * CATIONS TO THESE FIELDS WILL BE DISREGARDED UPON |
| * RETURN TO THE CALLER. THE OUTPUT FIELDS ARE |
| * PROVIDED TO ALLOW THE EXIT TO PASS INFORMATION |
| * BACK TO THE CALLER.                           |
| EJECT                                           |
| * INTERNAL REGISTER USAGE                       |
| *                                              |
| * 0 -                                          |
| * 1 - ON ENTRY TO MODULE = A(ARUDSNXP BLOCK) COPIED TO R10 |
| * 2 -                                          |
| * 3 -                                          |
| * 4 -                                          |
| * 5 -                                          |
| * 6 -                                          |
| * 7 -                                          |
| * 8 -                                          |
**Figure 72  DSNEXIT sample assembler user exit (part 3 of 11)**

* 9 - *
* 10 - *
* 11 - *
* 12 - BASEREG *
* 13 - SAVEAREA *
* 14 - *
* 15 - *
* *
*
* R0       EQU   0                  R *
* R1       EQU   1                   E *
* R2       EQU   2                    G *
* R3       EQU   3                     I *
* R4       EQU   4                      S *
* R5       EQU   5                       T *
* R6       EQU   6                         R *
* R7       EQU   7                               *
* R8       EQU   8                                               *
* R9       EQU   9                         E *
* R10      EQU   10                      Q *
* R11      EQU   11                         U *
* R12      EQU   12                              *
* R13      EQU   13                                      *
* R14      EQU   14                                               *
* R15      EQU   15                                S *
* *
* ARUDSNXP DEFINES THE USER VARIABLE USER EXIT PARM BLOCK. *
* *
* NOTE: *
* *
* YOU MAY NOT MODIFY THE FIELDS IN FRONT OF THE USER AREA. *
* *
* YOU MAY NOT MODIFY THE 'STRUCTURE' OF THIS DSECT. *
* *
* ARUDSNXP DSECT ,                        PARMS PASSED TO EXIT *
* *
* INPUT AREA *
* *
XPJOBN DS CL8  JOBNAME  
XPSTEP DS CL8  STEPNANE  
XPDBNAME DS CL8  DATABASE NAME  
XPSPNAME DS CL8  SPACE NAME  
XPRTYPE DS CL2  REORG TYPE (TS OR IX)  
             DS CL2  RESERVED FOR REORG PLUS  
XPUSER DS CL8  USER ID  
XPSSID DS CL4  DB2 SUBSYSTEM ID  

**DSNUEXIT assembler user exit**
DSNUEXIT assembler user exit

Figure 72  DSNUEXIT sample assembler user exit (part 4 of 11)

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPDATE</td>
<td>CL6</td>
<td>Utility execution date MMDDYY</td>
</tr>
<tr>
<td>XPTIME</td>
<td>CL6</td>
<td>Utility execution time HHMMSS</td>
</tr>
<tr>
<td>XPUTILID</td>
<td>CL16</td>
<td>Utility ID</td>
</tr>
<tr>
<td>XPDATE2</td>
<td>CL8</td>
<td>Utility execution date MMDDYYYY</td>
</tr>
<tr>
<td>XGRPNAME</td>
<td>CL4</td>
<td>Data sharing group name</td>
</tr>
<tr>
<td>XPVCAT</td>
<td>CL8</td>
<td>VCAT name (from 1st part if partitioned)</td>
</tr>
<tr>
<td>XPDATEJ</td>
<td>CL7</td>
<td>Utility execution date CCYYDM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved for reorg plus</td>
</tr>
</tbody>
</table>

* USER WORK AREA

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPUSRWD1</td>
<td>F</td>
<td>USER WORD 1</td>
</tr>
<tr>
<td>XPUSRWD2</td>
<td>F</td>
<td>USER WORD 2</td>
</tr>
<tr>
<td>XPUSRWD3</td>
<td>F</td>
<td>USER WORD 3</td>
</tr>
<tr>
<td>XPUSRWD4</td>
<td>F</td>
<td>USER WORD 4</td>
</tr>
<tr>
<td>XPUSRWD5</td>
<td>F</td>
<td>USER WORD 5</td>
</tr>
<tr>
<td>XPUSRWD6</td>
<td>F</td>
<td>USER WORD 6</td>
</tr>
<tr>
<td>XPUSRWD7</td>
<td>F</td>
<td>USER WORD 7</td>
</tr>
<tr>
<td>XPUSRWD8</td>
<td>F</td>
<td>USER WORD 8</td>
</tr>
</tbody>
</table>

* OUTPUT AREA

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPUVAREA</td>
<td>USER VARIABLE AREA</td>
</tr>
<tr>
<td>XPUVENT#</td>
<td>NUMBER OF ENTRIES IN OUTPUT AREA</td>
</tr>
<tr>
<td>XP$</td>
<td>*-ARUDSNXP</td>
</tr>
</tbody>
</table>

*---------------------------------------------------------------------*

*XPUVARS defines the output area of the ARUDSNXP DSECT at label *
*XPUVAREA. You may use this DSECT to easily address the table *
*defined at XPUVAREA.*

* NOTE: *
* THE NUMBER OF ENTRIES IN THE XPUVAREA TABLE MUST NOT EXCEED 100 *
* OR YOU WILL ADDRESS BEYOND THE END OF THE GETMAINED STORAGE *
* PROVIDED FOR THIS ROUTINE. *

* VARIABLE NAME:

* 1. MUST BEGIN WITH AN UNDERSCORE CHARACTER C'_' X'6D' *
* 2. MUST BEGIN IN THE LEFTMOST BYTE OF XPUVNAME FIELD *
* 3. IF LESS THAN 9 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS *
* 4. MAY NOT CONTAIN EMBEDDED BLANKS *
* 5. NON-BLANK CHARACTERS AFTER THE LEFTMOST BYTE MUST BE NATIONAL *
  CHARACTERS WHICH ARE A THROUGH Z, 0 THROUGH 9, #, @, AND $ *

* VARIABLE DATA:

* 1. MUST BEGIN IN THE LEFTMOST BYTE OF XPUVDATA FIELD *
* 2. IF LESS THAN 8 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS *
Figure 72  DSNUEXIT sample assembler user exit (part 5 of 11)

* 3. MAY NOT CONTAIN EMBEDDED BLANKS
* 4. NON-BLANK CHARACTERS MUST BE 'NATIONAL' CHARACTERS WHICH ARE
  A THROUGH Z, 0 THROUGH 9, #, @, $, -, AND .
* * * * *
  XPUVARS DSECT ,
  XPUVNAME DS  CL9
  XPUVDATA DS  CL8
  XPUVENT$ EQU  *-XPUVNAME
  EJECT
* * * * *
* LOCAL PROGRAM WORK AREA DSECT POINTED TO BY R11
* * * * *
* WORKAREA DSECT ,
  SAVE     DS    18F
  DWORK    DS    D
  CWORK    DS    CL8
  WRKAREA$ EQU  *-WORKAREA
  EJECT
* * * * *

  P R O G R A M  S T A R T
  *
  *
  *
  *
  *
  *
  *
  *
  *
  *

ARUEDSNA CSECT
* *
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
* * UNCOMMENT THE NEXT TWO INSTRUCTIONS TO DISABLE THIS EXIT.  *
* * *
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
* *
* LA    R15,0
* BSM   0,R14
* RC=0, GOOD RETURN
* JUST RETURN TO CALLER
* *
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
* * UNCOMMENT THE PREVIOUS TWO INSTRUCTIONS TO DISABLE THIS EXIT.  *
* * *
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
* *
  STM   R14,R12,12(R13)
  LR    R12,R15
  USING ARUEDSNA,R12
  SAVE CALLERS REGS
  R12 IS MY BASEREG
  ESTABLISH ADDRESSABLITY
  *
LR   R10,R1    GET A(PASSED BLOCK)  
USING ARUDSNXP,R10 MAP IT

*  
* GET SOME WORKING STORAGE  
*  
*  
XR   R11,R11      ZERO R11
GETMAIN RC,lv=WRKAREA$,LOC=BELOW GET WORK AREA
LTR   R15,R15     OK?
BNZ   DSNXRC08    ..NO, ERROR
LR    R11,R1      R11 = A(USER WORK AREA)
ST    R11,XPUSRWD1 SAVE A(USER WORK AREA)
USING WORKAREA,R11 ADDRESS WORKAREA DSECT
LA    R15,SAVE    MY SAVE AREA@
ST    R15,4(,R13) SAVE CALLERS SAVE AREA IN MINE
ST    R13,4(,R15) SAVE CALLERS SAVE AREA@ IN MINE
LR    R13,R15     SET OURS CURRENT
EJECT

*  
*  
* CREATE USER-DEFINED VARIABLES  
*  
* THE FOLLOWING USER VARIABLES WILL BE CREATED:  
*  
*   _JDATE    - JULIAN DATE IN THE FORM OF Dyyddd BUILT FROM THE 
*                REORG SUPPLIED VARIABLE &DATE8  
*  
*   _JCDATE   - JULIAN DATE IN THE FORM OF Dyyyyddd BUILT FROM THE 
*                REORG SUPPLIED VARIABLE &DATE8  
*  
*   _UTILPFX  - UP TO 8 BYTES OF THE FIRST NODE OF THE UTILITY ID  
*  
*   _UTILSFX  - UP TO 8 BYTES OF THE REMAINING PORTION OF THE 
*                UTILITY ID  
*  
*  
LA    R3,XPUVAREA @(OUTPUT VARIABLE AREA)
USING XPUVARS,R3 ESTABLISH BASE

*  
*  
* BUILD _JDATE AND _JCDATE USER VARIABLES  
*  
*  
MVC  XPUVNAME(XPUVENT$),BLANKS CLEAR ENTRY
Figure 72  DSNUEXIT sample assembler user exit  (part 7 of 11)

<table>
<thead>
<tr>
<th>Line</th>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MVC XPUVNAME,=CL9'_JDATE'</td>
<td>SET VAR NAME</td>
</tr>
<tr>
<td>2</td>
<td>LA R2,MONTHTAB</td>
<td>@(MONTH TABLE - NO LEAP YEAR)</td>
</tr>
<tr>
<td>3</td>
<td>PACK DWORK(8),XPDATE8+4(4)</td>
<td>GET YEARS</td>
</tr>
<tr>
<td>4</td>
<td>CVB R4,DWORK</td>
<td>INTO R4</td>
</tr>
<tr>
<td>5</td>
<td>LR R7,R4</td>
<td>SAVE A COPY IN R7</td>
</tr>
<tr>
<td>6</td>
<td>SRDL R4,32</td>
<td>PREPARE FOR DIVIDE</td>
</tr>
<tr>
<td>7</td>
<td>D R4,=F'4'</td>
<td>DIVIDE BY 4</td>
</tr>
<tr>
<td>8</td>
<td>LTR R4,R4</td>
<td>LEAP YEAR?</td>
</tr>
<tr>
<td>9</td>
<td>BNZ DSNX0110</td>
<td>NO - &gt;</td>
</tr>
<tr>
<td>10</td>
<td>LR R4,R7</td>
<td>GET YEAR AGAIN</td>
</tr>
<tr>
<td>11</td>
<td>SRDL R4,32</td>
<td>PREPARE FOR DIVIDE</td>
</tr>
<tr>
<td>12</td>
<td>D R4,=F'100'</td>
<td>DIVIDE BY 100</td>
</tr>
<tr>
<td>13</td>
<td>LTR R4,R4</td>
<td>LEAP YEAR?</td>
</tr>
<tr>
<td>14</td>
<td>BNZ DSNX0100</td>
<td>YES - &gt;</td>
</tr>
<tr>
<td>15</td>
<td>LR R4,R7</td>
<td>GET YEAR AGAIN</td>
</tr>
<tr>
<td>16</td>
<td>SRDL R4,32</td>
<td>PREPARE FOR DIVIDE</td>
</tr>
<tr>
<td>17</td>
<td>D R4,=F'400'</td>
<td>DIVIDE BY 400</td>
</tr>
<tr>
<td>18</td>
<td>LTR R4,R4</td>
<td>LEAP YEAR?</td>
</tr>
<tr>
<td>19</td>
<td>BNZ DSNX0110</td>
<td>NO - &gt;</td>
</tr>
</tbody>
</table>

DSNX0100 DS OH

* LA R2,LEAPTAB  @(MONTH TABLE - LEAP YEAR)

DSNX0110 DS OH

* PACK DWORK(8),XPDATE8(2)  GET MONTH
* CVB R4,DWORK              INTO R4
* BCTR R4,0                 SUBTRACT 1
* SLL R4,1                  MULTIPLY BY 2 FOR OFFSET
* LH R6,(R2,R4)             GET DAYS FOR PRIOR MONTHS
* PACK DWORK(8),XPDATE8+2(2)GET DAYS
* CVB R4,DWORK              INTO R4
* AR R6,R4                  TOTAL DAYS IN R6
* LR R5,R7                  GET YEAR
* M R4,=F'1000'             SHIFT THE YEAR OVER
* AR R5,R6                  MAKE IT YYDDD FORMAT
* CVD R5,DWORK              MAKE IT PACKED
* MVC CWORK(L'PL4PAT),PL4PATGET EDIT PATTERN
* ED CWORK(L'PL4PAT),DWORK+4EDIT YYDDD

* MVI XPUVDATA,C'D'         SET VAR DATA
* MVC XPUVDATA+1(5),CWORK+3  SET VAR DATA

* LA R3,XPUVENT$(,R3)       @(NEXT ENTRY)
* MVC XPUVNAME(XPUVENT$),BLANKSCLEAR ENTRY

* MVC XPUVNAME,=CL9'_JCDATE' SET VAR NAME

* CVD R5,DWORK              MAKE IT PACKED
### Figure 72 DSNUEXIT sample assembler user exit (part 8 of 11)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVC</td>
<td>CWORK(L'PL4PAT),PL4PAT GET EDIT PATTERN</td>
</tr>
<tr>
<td>ED</td>
<td>CWORK(L'PL4PAT),DWORK+4 EDIT YYYYDDD</td>
</tr>
<tr>
<td>MVI</td>
<td>XPUVDATA,C'D' SET VAR DATA</td>
</tr>
<tr>
<td>MVC</td>
<td>XPUVDATA+1(7),CWORK+1 SET VAR DATA</td>
</tr>
<tr>
<td></td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>BUILD _UTILPFX AND _UTILSFX USER VARIABLES</td>
</tr>
<tr>
<td></td>
<td>------------------</td>
</tr>
<tr>
<td>LA</td>
<td>R3,XPUVENT$(,R3) @(NEXT ENTRY)</td>
</tr>
<tr>
<td>MVC</td>
<td>XPUVNAME(XPUVENT$),BLANKS CLEAR ENTRY</td>
</tr>
<tr>
<td></td>
<td>------------------</td>
</tr>
<tr>
<td>MVC</td>
<td>XPUVNAME.=CL9'_UTILPFX' SET VAR NAME</td>
</tr>
<tr>
<td>TRT</td>
<td>XPUTILID,TRTAB LOOK FOR DELIMITER</td>
</tr>
<tr>
<td>BZ</td>
<td>DSNX0200 NONE -&gt;</td>
</tr>
<tr>
<td></td>
<td>------------------</td>
</tr>
<tr>
<td>LA</td>
<td>R2,XPUTILID @(UTIL ID)</td>
</tr>
<tr>
<td>LR</td>
<td>R4,R1 SAVE @(DELMETER)</td>
</tr>
<tr>
<td>SR</td>
<td>R1,R2 LENGTH OF FIRST PART</td>
</tr>
<tr>
<td>BZ</td>
<td>DSNX0200 NONE -&gt;</td>
</tr>
<tr>
<td></td>
<td>------------------</td>
</tr>
<tr>
<td>C</td>
<td>R1.=F'8' MORE THAN 8 BYTES LONG</td>
</tr>
<tr>
<td>BNH</td>
<td>DSNX0130 NO -&gt;</td>
</tr>
<tr>
<td>LA</td>
<td>R1,8 MAKE IT 8 BYTES</td>
</tr>
<tr>
<td>DSNX0130</td>
<td>OH</td>
</tr>
<tr>
<td>BCTR</td>
<td>R1,0 SUBTRACT 1 FOR EXECUTED MVC</td>
</tr>
<tr>
<td>LA</td>
<td>R5,XPUTILID @(UTILID)</td>
</tr>
<tr>
<td>EX</td>
<td>R1.DSNXMCU MOVE UTILID PREFIX</td>
</tr>
<tr>
<td></td>
<td>------------------</td>
</tr>
<tr>
<td>LA</td>
<td>R3,XPUVENT$(,R3) @(NEXT ENTRY)</td>
</tr>
<tr>
<td>MVC</td>
<td>XPUVNAME(XPUVENT$),BLANKS CLEAR ENTRY</td>
</tr>
<tr>
<td>MVC</td>
<td>XPUVNAME.=CL9'_UTILSFX' SET VAR NAME</td>
</tr>
<tr>
<td>LA</td>
<td>R5.1(C,R4) @(1ST CHAR PAST DELIMITER)</td>
</tr>
<tr>
<td>LA</td>
<td>R1,XPUTILID+L'XPUTILID @(END OF UTILID)</td>
</tr>
<tr>
<td>SR</td>
<td>R1,R5 LENGTH OF SECOND PART OF UTILID</td>
</tr>
<tr>
<td>BZ</td>
<td>DSNXRCC00 NONE -&gt;</td>
</tr>
<tr>
<td>C</td>
<td>R1.=F'8' MORE THAN 8 BYTES LONG</td>
</tr>
<tr>
<td>BNH</td>
<td>DSNX0140 NO -&gt;</td>
</tr>
<tr>
<td>LA</td>
<td>R1,8 MAKE IT 8 BYTES</td>
</tr>
<tr>
<td>DSNX0140</td>
<td>OH</td>
</tr>
<tr>
<td>BCTR</td>
<td>R1,0 SUBTRACT 1 FOR EXECUTED MVC</td>
</tr>
<tr>
<td>EX</td>
<td>R1.DSNXMCU MOVE UTILID PREFIX</td>
</tr>
</tbody>
</table>
Figure 72  DSNUEXIT sample assembler user exit  (part 9 of 11)

*  B  DSNXRC00  RETURN RC=0

DSNX0200  DS  OH

  MVC  XPUVDATA,XPUTILID  GET FIRST 8 BYTES

  LA  R3,XPUVENT$(,R3)  @(NEXT ENTRY)
  MVC  XPUVNAME(XPUVENT$),BLANKS  CLEAR ENTRY

  MVC  XPUVNAME,=CL9'_UTILSFX'  SET VAR NAME
  MVC  XPUVDATA,XPUTILID+8  GET NEXT 8 BYTES

  B  DSNXRC00  RETURN RC=0

DSNXMVCU  MVC  XPUVDATA(0),0(R5)  *** EXECUTED INSTRUCTION ***

  DROP  R3

  EJECT

*  ---------------------------------------------------------------------*

*  SET RETURN CODE AND EXIT  *

*  ---------------------------------------------------------------------*

DSNXRC00  DS  OH

  LA  R3,0  SAVE RETURN CODE
  B  DSNX9000  GOOD RETURN

DSNXRC08  DS  OH

  LA  R3,8  SAVE RETURN CODE
  DSNX9000  DS  OH

  LTR  R11,R11  DID WE GET STORAGE?
  BZ  DSNX9900  NO ->
  L  R13,SAVE+4
  FREEMAIN  RC,LV=WRKAREA$,A=(R11) FREE LOCAL WORK AREA

DSNX9900  DS  OH

  LR  R15,R3  RESTORE RETURN CODE
  L  R14,12(R,R13)
  LM  R0,R12.20(R13)
  BSM  0,R14  RETURN
  EJECT

*
**CONSTANTS**

```
*                0 1 2 3 4 5 6 7 8 9 A B C D E F *

TRTAB    DC    X'00000000000000000000000000000000' 0
DC    X'00000000000000000000000000000000' 1
DC    X'00000000000000000000000000000000' 2
DC    X'00000000000000000000000000000000' 3
DC    X'FF00000000000000000000FF0000FFFF' 4 (SP) . + |
DC    X'0000000000000000000000000000FF00' 5 ;
DC    X'FFFF0000000000000000FFFF00FF0000' 6 - / _
DC    X'00000000000000000000000000000000' 7 : =
DC    X'00000000000000000000000000000000' 8
DC    X'00000000000000000000000000000000' 9
DC    X'00000000000000000000000000000000' A
DC    X'00000000000000000000000000000000' B
DC    X'00000000000000000000000000000000' C
DC    X'00000000000000000000000000000000' D
DC    X'FF000000000000000000000000000000' E \nDC    X'00000000000000000000000000000000' F

*                0 1 2 3 4 5 6 7 8 9 A B C D E F *

MONTHTAB DC    H'0'                    JANUARY
DC    H'31'                   FEBRUARY
DC    H'59'                   MARCH
DC    H'90'                   APRIL
DC    H'120'                  MAY
DC    H'151'                  JUNE
DC    H'181'                  JULY
DC    H'212'                  AUGUST
DC    H'243'                  SEPTEMBER
DC    H'273'                  OCTOBER
DC    H'304'                  NOVEMBER
DC    H'334'                  DECEMBER

* LEAPTAB DC    H'0'                    JANUARY
DC    H'31'                   FEBRUARY
DC    H'60'                   MARCH
DC    H'91'                   APRIL
DC    H'121'                  MAY
DC    H'152'                  JUNE
DC    H'182'                  JULY
DC    H'213'                  AUGUST
DC    H'244'                  SEPTEMBER
DC    H'274'                  OCTOBER
DC    H'305'                  NOVEMBER
```
DSNUEXIT COBOL II and LE COBOL user exit

This section provides a sample exit parameter record, variable mapping record, and exit that you can use as an example of how to write the DSNUEXIT user exit in COBOL or LE COBOL.

COBOL II and LE COBOL exit parameter record

The COBOL II and LE COBOL exit parameter record (Figure 73) contains both input and output fields. Input fields pass vital information to the user exit, such as the database name and user ID. If you make any modifications to these input fields, they are disregarded on return. The output fields pass information about your user variables back to REORG PLUS.

Figure 73 DSNUEXIT COBOL II and LE COBOL exit parameter record (part 1 of 2)
DSNUEXIT COBOL II and LE COBOL exit parameter record (part 2 of 2)

```
15 EXIT-PREFIX   PIC X(8).
15 EXIT-SUFFIX   PIC X(8).
10 EXIT-DATE8.
  15 EXIT-DATE8-MM  PIC 9(2).
  15 EXIT-DATE8-DD  PIC 9(2).
  15 EXIT-DATE8-YEAR PIC 9(4).
15 FILLER REDEFINES EXIT-DATE8-YEAR.
  20 EXIT-DATE8-CC  PIC 9(2).
  20 EXIT-DATE8-YY  PIC 9(2).
10 EXIT-GRPNM    PIC X(4).
10 EXIT-VCAT     PIC X(8).
10 EXIT-DATEJ.
  15 EXIT-DATEJ-YEAR PIC 9(4).
  15 FILLER REDEFINES EXIT-DATEJ-YEAR.
  20 EXIT-DATEJ-CC  PIC 9(2).
  20 EXIT-DATEJ-YY  PIC 9(2).
  15 EXIT-DATEJ-DDD PIC 9(3).
10 EXIT-FILLER2  PIC X(13).
05 WORK-AREA-ADDRESSES.
  10 WORK-AREA-1   PIC 9(4).
  10 WORK-AREA-2   PIC 9(4).
  10 WORK-AREA-3   PIC 9(4).
  10 WORK-AREA-4   PIC 9(4).
  10 WORK-AREA-5   PIC 9(4).
  10 WORK-AREA-6   PIC 9(4).
  10 WORK-AREA-7   PIC 9(4).
  10 WORK-AREA-8   PIC 9(4).
```

Parameter record fields

Table 123 describes the major fields for the DSNUEXIT COBOL II and LE COBOL user exit parameter record and their uses.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXIT-JOBNAME</td>
<td>job name</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>EXIT-STEPNAME</td>
<td>step name</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>EXIT-DBNAME</td>
<td>database name</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>EXIT-TSNAME</td>
<td>name of the table space or index space from the REORG command</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>EXIT-REORG-TYPE</td>
<td>type of reorganization being performed; table space (TS) or index (IX)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>EXIT-USERID</td>
<td>USERID of the user running REORG PLUS</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>EXIT-DB2-SSID</td>
<td>DB2 subsystem ID</td>
<td>4 bytes</td>
</tr>
</tbody>
</table>
COBOL II and LE COBOL variable mapping record

The COBOL II and LE COBOL variable mapping record (Figure 74) defines the output area of the exit parameter record. Using this record allows you to easily address the variable definition table. The number of entries in this table (each entry consists of both a variable name and its value) must not exceed 100.

Table 123 Major parameter record fields for the DSNEXIT COBOL II and LE COBOL user exit (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXIT-DATE</td>
<td>date of the execution of the utility, in the format MMDDYY</td>
<td>6 bytes</td>
</tr>
<tr>
<td>EXIT-TIME</td>
<td>time of the execution of the utility, in the format HHMMSS</td>
<td>6 bytes</td>
</tr>
<tr>
<td>EXIT-UTILID-PARM</td>
<td>utility ID</td>
<td>up to 16 bytes</td>
</tr>
<tr>
<td>EXIT-DATE8</td>
<td>date of the execution of the utility, in the format MMDDYYYY</td>
<td>8 bytes</td>
</tr>
<tr>
<td>EXIT-GRPNM</td>
<td>DB2 data sharing group name</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>In a non-data-sharing environment, the field contains the DB2 SSID.</td>
<td></td>
</tr>
<tr>
<td>EXIT-VCAT</td>
<td>VCATNAME specified in the DB2 catalog for the table space being reorganized, or for the first partition, if the table space is partitioned</td>
<td>Not applicable</td>
</tr>
<tr>
<td>EXIT-DATEJ</td>
<td>Julian date of the execution of the utility, in the format CCYYDDD</td>
<td>7 bytes</td>
</tr>
<tr>
<td>WORK-AREA-1...</td>
<td>8 parameters for work space</td>
<td>up to 4 bytes each</td>
</tr>
<tr>
<td>WORK-AREA-8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Variable mapping record fields

Table 124 on page 808 describes the major variable mapping record fields for the DSNEXIT COBOL II and LE COBOL user exit and their uses.
### Table 124  Major fields for the DSNUEXIT COBOL II and LE COBOL variable mapping record

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER-DEFINED-VARIABLE-TABLE</td>
<td>table containing user-defined variable information For information about establishing user-defined variables, see page 791.</td>
</tr>
<tr>
<td>VARIABLE-NAME</td>
<td>name of the variable you are defining</td>
</tr>
<tr>
<td>VARIABLE-VALUE</td>
<td>value of the variable you are defining</td>
</tr>
</tbody>
</table>

### Sample COBOL II and LE COBOL user exit

Figure 75 shows an example of this user exit written in COBOL.

#### Figure 75  DSNUEXIT sample COBOL II and LE COBOL user exit (part 1 of 7)

```
*-----------------------------------------------------------------  
* ALL COBOL MODULES MUST BE COMPILED WITH DATA(31) AND DYNAM TO  
* EXECUTE PROPERLY!!!!!!!!!!                                     
*-----------------------------------------------------------------  
IDENTIFICATION DIVISION.  
*-----------------------------------------------------------------  
PROGRAM-ID.     ARUEDSN2.                                        
AUTHOR.         BMC SOFTWARE                                      
DATE-WRITTEN.   AUGUST 1995.                                    
DATE-COMPILED.  
*-----------------------------------------------------------------  
* ARUEDSN2 IS A SAMPLE DB2 COBOL II USER EXIT.                   
*-----------------------------------------------------------------  
* THIS IS A SAMPLE REORG PLUS USER EXIT. THIS EXIT WOULD BE USED  
* IN ORDER TO DEFINE USER VARIABLES FOR BUILDING DATA SET NAMES  
* FOR DYNAMIC WORKFILE ALLOCATION WITH THE DSNPAT KEYWORD AND FOR  
* SHRLEVEL CHANGE WITH THE SPILLDSPAT KEYWORD.                   
*-----------------------------------------------------------------  
* ARUEDSN2 IS CALLED ONLY ONCE PER EXECUTION OF ARUUMAIN          
* THE MODULE IS CALLED AT THE BEGINNING OF ---                    
* 1. THE UNLOAD PHASE FOR A TWO-PHASE REORGANIZATION OR          
* 2. THE REORG PHASE FOR A SINGLE-PHASE REORGANIZATION           
*-----------------------------------------------------------------  
* THIS EXIT WILL ONLY BE INVOKED WHEN IT IS THE VALUE SET FOR     
* DSNUEXIT PARM IN ARUSOPTS DURING INSTALLATION OR IN THE        
* DSNUEXIT PARAMETER OF THE REORG COMMAND. THE REORG COMMAND     
* OPTION WILL OVERRIDE THE PARM IN THE ARUSOPTS MACRO.            
*-----------------------------------------------------------------  
* PLEASE REVIEW ADDITIONAL DOCUMENTATION IN THE REFERENCE MANUAL  
*-----------------------------------------------------------------  
ENVIRONMENT DIVISION.                                            
INPUT-OUTPUT SECTION.                                            
```
**Figure 75  DSNEXIT sample COBOL II and LE COBOL user exit (part 2 of 7)**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FILE-CONTROL.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DATA DIVISION.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>FILE SECTION.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>EJECT</strong></td>
<td></td>
</tr>
<tr>
<td><strong>WORKING-STORAGE SECTION.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>01 FILLER</strong></td>
<td><strong>PIC X(16) VALUE 'WORKING STORAGE'.</strong></td>
</tr>
<tr>
<td><strong>01 MISCELLANEOUS.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>05 SAVE-DATA</strong></td>
<td><strong>PIC X(8) VALUE SPACES.</strong></td>
</tr>
<tr>
<td><strong>05 DAYS-SUBX</strong></td>
<td><strong>PIC 9(3) COMP VALUE ZERO.</strong></td>
</tr>
<tr>
<td><strong>05 SUBX</strong></td>
<td><strong>PIC 9(3) COMP VALUE ZERO.</strong></td>
</tr>
<tr>
<td><strong>05 ONE</strong></td>
<td><strong>PIC 9(1) COMP-3 VALUE +1.</strong></td>
</tr>
<tr>
<td><strong>05 TWO</strong></td>
<td><strong>PIC 9(1) COMP-3 VALUE +2.</strong></td>
</tr>
<tr>
<td><strong>05 FOUR</strong></td>
<td><strong>PIC 9(1) COMP-3 VALUE +4.</strong></td>
</tr>
<tr>
<td><strong>05 MAX-LENGTH</strong></td>
<td><strong>PIC 9(3) COMP-3 VALUE +16.</strong></td>
</tr>
<tr>
<td><strong>05 NINETEEN</strong></td>
<td><strong>PIC 9(3) COMP-3 VALUE +19.</strong></td>
</tr>
<tr>
<td><strong>05 TWENTY</strong></td>
<td><strong>PIC 9(3) COMP-3 VALUE +20.</strong></td>
</tr>
<tr>
<td><strong>05 ONE-HUNDRED</strong></td>
<td><strong>PIC 9(3) COMP-3 VALUE +100.</strong></td>
</tr>
<tr>
<td><strong>05 FOUR-HUNDRED</strong></td>
<td><strong>PIC 9(3) COMP-3 VALUE +400.</strong></td>
</tr>
<tr>
<td><strong>05 JULIAN-DATE-DESC</strong></td>
<td><strong>PIC X(9) VALUE '_JDATE'.</strong></td>
</tr>
<tr>
<td><strong>05 JULIAN-CDATE-DESC</strong></td>
<td><strong>PIC X(9) VALUE '_JCDATE'.</strong></td>
</tr>
<tr>
<td><strong>05 UTILITY-PREFIX</strong></td>
<td><strong>PIC X(9) VALUE '_UTILPFX'.</strong></td>
</tr>
<tr>
<td><strong>05 UTILITY-SUFFIX</strong></td>
<td><strong>PIC X(9) VALUE '_UTILSFX'.</strong></td>
</tr>
<tr>
<td><strong>05 UTILID-PREFIX</strong></td>
<td><strong>PIC X(8) VALUE SPACES.</strong></td>
</tr>
<tr>
<td><strong>05 UTILID-SUFFIX</strong></td>
<td><strong>PIC X(8) VALUE SPACES.</strong></td>
</tr>
<tr>
<td><strong>05 UTILID-POINTER</strong></td>
<td><strong>PIC 9(3) VALUE ZERO.</strong></td>
</tr>
<tr>
<td><strong>05 UTILID-COUNTER</strong></td>
<td><strong>PIC 9(3) VALUE ZERO.</strong></td>
</tr>
<tr>
<td><strong>05 UTILID-TALLY</strong></td>
<td><strong>PIC 9(3) VALUE ZERO.</strong></td>
</tr>
<tr>
<td><strong>01 DATE-WORK-AREA.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>05 CONVERSION-DATE.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>10 DATE-PREFIX</strong></td>
<td><strong>PIC X(1) VALUE 'D'.</strong></td>
</tr>
<tr>
<td><strong>10 JULIAN-CDATE</strong></td>
<td><strong>PIC 9(7) VALUE ZERO.</strong></td>
</tr>
<tr>
<td><strong>10 FILLER</strong></td>
<td><strong>REDEFINES JULIAN-CDATE.</strong></td>
</tr>
<tr>
<td><strong>15 JULIAN-CC</strong></td>
<td><strong>PIC 9(2).</strong></td>
</tr>
<tr>
<td><strong>15 JULIAN-DATE</strong></td>
<td><strong>PIC 9(5).</strong></td>
</tr>
<tr>
<td><strong>15 FILLER</strong></td>
<td><strong>REDEFINES JULIAN-DATE.</strong></td>
</tr>
<tr>
<td><strong>20 JULIAN-YY</strong></td>
<td><strong>PIC 9(2).</strong></td>
</tr>
<tr>
<td><strong>20 JULIAN-DAYS</strong></td>
<td><strong>PIC 9(3).</strong></td>
</tr>
<tr>
<td><strong>05 YEAR-ANSWER</strong></td>
<td><strong>PIC 9(3) COMP-3 VALUE ZERO.</strong></td>
</tr>
<tr>
<td><strong>05 YEAR-REMAINDER</strong></td>
<td><strong>PIC 9(5) COMP-3 VALUE ZERO.</strong></td>
</tr>
</tbody>
</table>
Figure 75  DSNUEXIT sample COBOL II and LE COBOL user exit (part 3 of 7)

05 filler redefines year-remainder.
   10 year-x                      pic x(5).

*-----------------------------------------------------------------
*    THE FOLLOWING TWO TABLES ARE USED TO CALCULATE THE JULIAN
*    DAY DEPENDING ON WHETHER THE YEAR IS A LEAP OR NOT.
*-----------------------------------------------------------------

01 no-leap-months.
  05 january               pic s9(3) comp-3 value 00.
  05 february              pic s9(3) comp-3 value 31.
  05 march                 pic s9(3) comp-3 value 59.
  05 april                 pic s9(3) comp-3 value 90.
  05 may                   pic s9(3) comp-3 value 120.
  05 june                  pic s9(3) comp-3 value 151.
  05 july                  pic s9(3) comp-3 value 181.
  05 august                pic s9(3) comp-3 value 212.
  05 september             pic s9(3) comp-3 value 243.
  05 october               pic s9(3) comp-3 value 273.
  05 november              pic s9(3) comp-3 value 304.
  05 december              pic s9(3) comp-3 value 334.
  01 filler redefines no-leap-months.
  05 month-days            pic s9(3) comp-3 occurs 12 times.

01 leap-months.
  05 leap-january          pic s9(3) comp-3 value 00.
  05 leap-february         pic s9(3) comp-3 value 31.
  05 leap-march            pic s9(3) comp-3 value 59.
  05 leap-april            pic s9(3) comp-3 value 90.
  05 leap-may              pic s9(3) comp-3 value 120.
  05 leap-june             pic s9(3) comp-3 value 151.
  05 leap-july             pic s9(3) comp-3 value 181.
  05 leap-august           pic s9(3) comp-3 value 212.
  05 leap-september        pic s9(3) comp-3 value 243.
  05 leap-october          pic s9(3) comp-3 value 273.
  05 leap-november         pic s9(3) comp-3 value 304.
  05 leap-december         pic s9(3) comp-3 value 335.
  01 filler redefines leap-months.
  05 leap-month-days       pic s9(3) comp-3 occurs 12 times.

*-----------------------------------------------------------------
*    LINKAGE SECTION.
*-----------------------------------------------------------------

*    THE TABLE CAN NOT OCCUR MORE THAN 100 TIME OR A STORAGE
*    OVERLAY WILL OCCUR.
*-----------------------------------------------------------------

01 reorg-exit-parms.
  05 fixed-parm-values.
     10 exit-jobname          pic x(8).
     10 exit-stepname         pic x(8).
     10 exit-dbname           pic x(8).
Figure 75  DSNEXIT sample COBOL II and LE COBOL user exit (part 4 of 7)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>EXIT-TSNAME</td>
</tr>
<tr>
<td>10</td>
<td>EXIT-REORG-TYPE</td>
</tr>
<tr>
<td>10</td>
<td>EXIT-FILLER1</td>
</tr>
<tr>
<td>10</td>
<td>EXIT-USERID</td>
</tr>
<tr>
<td>10</td>
<td>EXIT-DB2-SSID</td>
</tr>
<tr>
<td>10</td>
<td>EXIT-DATE.</td>
</tr>
<tr>
<td>15</td>
<td>EXIT-MM</td>
</tr>
<tr>
<td>15</td>
<td>EXIT-DD</td>
</tr>
<tr>
<td>15</td>
<td>EXIT-YY</td>
</tr>
<tr>
<td>10</td>
<td>EXIT-TIME</td>
</tr>
<tr>
<td>10</td>
<td>EXIT-UTILID-PARM</td>
</tr>
<tr>
<td>15</td>
<td>EXIT-PREFIX</td>
</tr>
<tr>
<td>15</td>
<td>EXIT-SUFFIX</td>
</tr>
<tr>
<td>10</td>
<td>EXIT-DATE8.</td>
</tr>
<tr>
<td>15</td>
<td>EXIT-DATE8-MM</td>
</tr>
<tr>
<td>15</td>
<td>EXIT-DATE8-DD</td>
</tr>
<tr>
<td>15</td>
<td>EXIT-DATE8-YEAR</td>
</tr>
<tr>
<td>20</td>
<td>EXIT-DATE8-CC</td>
</tr>
<tr>
<td>20</td>
<td>EXIT-DATE8-YY</td>
</tr>
<tr>
<td>15</td>
<td>EXIT-DATEJ-YEAR</td>
</tr>
<tr>
<td>20</td>
<td>EXIT-DATEJ-YYYY</td>
</tr>
<tr>
<td>20</td>
<td>EXIT-DATEJ-DDD</td>
</tr>
<tr>
<td>10</td>
<td>EXIT-FILLER2</td>
</tr>
<tr>
<td>05</td>
<td>WORK-AREA-ADDRESSES.</td>
</tr>
<tr>
<td>10</td>
<td>WORK-AREA-1</td>
</tr>
<tr>
<td>10</td>
<td>WORK-AREA-2</td>
</tr>
<tr>
<td>10</td>
<td>WORK-AREA-3</td>
</tr>
<tr>
<td>10</td>
<td>WORK-AREA-4</td>
</tr>
<tr>
<td>10</td>
<td>WORK-AREA-5</td>
</tr>
<tr>
<td>10</td>
<td>WORK-AREA-6</td>
</tr>
<tr>
<td>10</td>
<td>WORK-AREA-7</td>
</tr>
<tr>
<td>10</td>
<td>WORK-AREA-8</td>
</tr>
<tr>
<td>05</td>
<td>USER-DEFINED-VARIABLE-TABLE.</td>
</tr>
<tr>
<td>10</td>
<td>VARIABLE-NAME</td>
</tr>
<tr>
<td>10</td>
<td>VARIABLE-VALUE</td>
</tr>
<tr>
<td>10</td>
<td>FILLER REDEFINES VARIABLE-VALUE.</td>
</tr>
<tr>
<td>15</td>
<td>VARIABLE-PREFIX</td>
</tr>
<tr>
<td>15</td>
<td>VARIABLE-JUL-DATE</td>
</tr>
</tbody>
</table>

*-----------------------------------------------------------------
PROCEDURE DIVISION USING REORG-EXIT-PARMS.
*-----------------------------------------------------------------
0000-MAIN.
  MOVE ZERO TO RETURN-CODE.
  PERFORM 1000-PROCESS-DATE.
  GOBACK.

*-----------------------------------------------------------------*
*    THE DATE IS PASSED IN A MMDDYYYY FORMAT AND CONVERTED TO      *
*    A JULIAN-DATE(WITH NO CENTURY) FORMAT OR A JULIAN-DATE        *
*    (WITH THE CENTURY) FORMAT.                                   *
*-----------------------------------------------------------------*

1000-PROCESS-DATE.
  MOVE EXIT-DATE8-DD TO JULIAN-DAYS.
  MOVE EXIT-DATE8-MM TO DAYS-SUBX.
  MOVE EXIT-DATE8-CC TO JULIAN-CC.
  MOVE EXIT-DATE8-YY TO JULIAN-YY.
  MOVE EXIT-DATE8-YEAR TO WORK-YEAR.
  DIVIDE WORK-YEAR BY FOUR
    GIVING YEAR-ANSWER
    REMAINDER YEAR-REMAINDER.
  IF YEAR-REMAINDER > ZERO
    THEN
      PERFORM 1100-NO-LEAP-YEAR
    ELSE
      DIVIDE WORK-YEAR BY ONE-HUNDRED
        GIVING YEAR-ANSWER
        REMAINDER YEAR-REMAINDER
      IF YEAR-REMAINDER > ZERO
        THEN
          PERFORM 1200-LEAP-YEAR
        ELSE
          DIVIDE WORK-YEAR BY FOUR-HUNDRED
            GIVING YEAR-ANSWER
            REMAINDER YEAR-REMAINDER
          IF YEAR-REMAINDER > ZERO
            THEN
              PERFORM 1100-NO-LEAP-YEAR
            ELSE
              ADD LEAP-MONTH-DAYS(DAYS-SUBX) TO JULIAN-DAYS
              PERFORM 1200-LEAP-YEAR
            END-IF
          END-IF
        END-IF
      END-IF
    END-IF
  END-IF.

1000-PROCESS-DATE-EXIT.
  EXIT.
1100-NO-LEAP-YEAR.
   ADD MONTH-DAYS(DAYS-SUBX) TO JULIAN-DAYS.
   PERFORM 1300-CENTURY.

1100-NO-LEAP-YEAR-EXIT.
   EXIT.

1200-LEAP-YEAR.
   ADD LEAP-MONTH-DAYS(DAYS-SUBX) TO JULIAN-DAYS.
   PERFORM 1300-CENTURY.

1200-LEAP-YEAR-EXIT.
   EXIT.

1300-CENTURY.
   PERFORM 1400-EDIT-UTILID.

1300-CENTURY-EXIT.
   EXIT.

*-----------------------------------------------------------------
*    THE FULL 16 BYTES OF THE UTILITY ID PARM IS CHECKED FOR
*    A DELIMITER. IF ONE IS FOUND AFTER THE FIRST 8 BYTES, IT
*    IS INCLUDED IN THE SUFFIX. IF A DELIMITER IS FOUND IN THE
*    FIRST 8 BYTES, ONLY THOSE CHARACTERS/NUMBERS UP TO THE
*    DELIMITER, WILL BE MOVED INTO THE PREFIX FIELD.
*-----------------------------------------------------------------
*    THE DELIMITER CHARACTER IMMEDIATELY FOLLOWING THE FORWARD
*    SLASH ("/") IS A BROKEN VERTICAL BAR, HEX 6A.
*-----------------------------------------------------------------
1400-EDIT-UTILID.
   MOVE ONE TO UTILID-POINTER, UTILID-TALLY.
   UNSTRING EXIT-UTILID-PARM
      DELIMITED BY ' ' OR '.' OR '+' OR '|' OR ';' OR '-' OR '/' OR '|' OR ',' OR '_' OR ':'
      OR '=' OR '\' OR ' ' INTO UTILID-PREFIX
   COUNT IN UTILID-COUNTER
   WITH POINTER UTILID-POINTER.
   IF UTILID-COUNTER = MAX-LENGTH
      THEN
         MOVE EXIT-SUFFIX TO UTILID-SUFFIX
      ELSE
         MOVE UTILID-COUNTER TO UTILID-POINTER
         ADD TWO TO UTILID-POINTER
         UNSTRING EXIT-UTILID-PARM
         INTO UTILID-SUFFIX
         WITH POINTER UTILID-POINTER
      END-IF.
   PERFORM 1500-UTILID-PARMS.
This section provides a sample exit parameter structure, variable mapping structure, and exit that you can use as an example of how to write the DSNEXIT user exit in C.

### C exit parameter structure

The C exit parameter structure (Figure 76) contains both input and output fields. Input fields pass vital information to the user exit, such as the database name and user ID. If you make any modifications to these input fields, they are disregarded on return. The output fields pass information about your user variables back to REORG PLUS.

#### Figure 76  DSNEXIT C exit parameter structure  (part 1 of 2)

```c
struct arudsnpx {
    char   xpjobn(|8|);   /* jobname */
    char   xpstep(|8|);   /* stepname */
    char   xpdbname(|8|); /* database name */
    char   xpspname(|8|); /* space name */
    char   xprtype(|2|);  /* reorg type (ts or ix) */
}
```
Structure fields

Table 125 describes the major structure fields for the DSNUEXIT C user exit and their uses.

Table 125  Major structure fields for the DSNUEXIT C exit parameter (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>xpjobn</td>
<td>job name</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>xpstep</td>
<td>step name</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>xpdbname</td>
<td>database name</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>xpspname</td>
<td>name of the table space or index space from the REORG command</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>xptype</td>
<td>type of reorganization being performed, table space (TS) or index (IX)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>xuser</td>
<td>USERID of the user running REORG PLUS</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>xpssid</td>
<td>DB2 subsystem ID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>xdate</td>
<td>date of the execution of the utility, in the format MMDDYY</td>
<td>6 bytes</td>
</tr>
<tr>
<td>xptime</td>
<td>time of the execution of the utility, in the format HHMMSS</td>
<td>6 bytes</td>
</tr>
<tr>
<td>xputilid</td>
<td>utility ID</td>
<td>up to 16 bytes</td>
</tr>
<tr>
<td>xdate8</td>
<td>date of the execution of the utility, in the format MMDDYYYY</td>
<td>8 bytes</td>
</tr>
<tr>
<td>xpgrpm</td>
<td>DB2 data sharing group name</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

In a non-data sharing environment, the field contains the DB2 SSID.
C exit variable mapping structure

The C exit variable mapping structure, as shown in Figure 77, defines the output area of the exit parameter structure. Using this structure allows you to easily address the variable definition table. The number of entries in this table (each entry consists of both a variable name and its value) must not exceed 100.

Figure 77  DSNUEXIT C exit variable mapping structure

```
struct XPUVAREA {
   char xpuvname[9];
   char xpuvdata[8];
} xpuvars[XPUVENT];
```

Structure fields

Table 126 describes the major variable mapping structure fields for the DSNUEXIT C user exit and their uses.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>xpvcat</td>
<td>VCATNAME specified in the DB2 catalog for the table space being reorganized, or for the first partition if the table space is partitioned</td>
<td>Not applicable</td>
</tr>
<tr>
<td>xpldatej</td>
<td>Julian date of the execution of the utility, in the format CCYYDDD</td>
<td>7 bytes</td>
</tr>
<tr>
<td>xpusrwd1...xpusrwd8</td>
<td>8 parameters of work space</td>
<td>up to 4 bytes each</td>
</tr>
</tbody>
</table>

Table 126  Major structure fields for the DSNUEXIT C exit parameter (part 2 of 2)

Sample C user exit

Figure 78 on page 817 shows an example of this user exit written in C.
Figure 78  DSNEXIT sample C user exit (part 1 of 7)

/*--------------------------------------------------------------------*/
// DISCLAImER
/*--------------------------------------------------------------------*/
/* THIS IS A SAMPLE REORG PLUS USER EXIT. THIS EXIT WOULD BE USED */
/* IN ORDER TO DEFINE USER VARIABLES FOR BUILDING DATASET NAMES FOR */
/* DYNAMIC WORKFILE ALLOCATION WITH THE DSNPAT KEYWORD AND FOR */
/* SHRLEVEL CHANGE WITH THE SPILLDSNPAT KEYWORD */
/* */
/* THIS EXIT WILL ONLY BE INVOKED WHEN IT IS SPECIFICALLY NAMED */
/* IN THE ARU$OPTS DSNEXIT PARAMETER OR IN THE DSNEXIT PARAMETER OF */
/* THE REORG COMMAND. THE SPECIFICATION IN THE REORG COMMAND WILL */
/* OVERRIDE THE SPECIFICATION IN THE ARU$OPTS MACRO. IF THE NAME */
/* SPECIFIED IS 'NONE' WITHOUT THE SINGLE QUOTE MARKS, THE EXIT WILL */
/* NOT BE INVOKED. */
/* */
/* NOTE: PLEASE REVIEW THE DOCUMENTATION IN THE REFERENCE MANUAL, */
/* AND THE FOLLOWING USAGE NOTES BEFORE IMPLEMENTING THIS EXIT. */
/* */
/* PLEASE CALL BMC SOFTWARE CUSTOMER SUPPORT WITH ANY QUESTIONS YOU */
/* MAY HAVE IN THIS AREA. */
/* */
/* PHONE: 1-800-537-1813 */
/* */
/* NOTES */
/*--------------------------------------------------------------------*/
/* ARUEDSC IS CALLED ONCE AT THE BEGINNING OF THE UNLOAD PHASE FOR A */
/* TWO-PHASE REORGANIZATION OR AT THE BEGINNING OF THE REORG PHASE */
/* FOR A SINGLE-PHASE REORGANIZATION. */
/* */
/* WHEN INVOKED IT IS PASSED THE ADDRESS OF A USER EXIT BLOCK */
/* DESCRIBED BY THE ARUDSNXP STRUCT. */
/* THIS CALL IS FROM ASEMBLER MODULE DYNWDSNX */
/* */
/* UPON EXIT RETRUN CODE FROM THIS EXIT */
/* */
/* 0         GOOD RETURN */
/* */
/* OTHER ERROR RETURN, TERMINATE UTILITY */
/* */
/* ARUDSNXP STRUCT THIS STRUCT CONTAINS BOTH INPUT */
/* AND OUTPUT FIELDS. INPUT FIELDS ARE PROVIDED TO */
/* PASS VITAL INFORMATION TO THIS EXIT. ANY MODIFI- */
/* CATIONS TO THESE FIELDS WILL BE DISREGARDED UPON */
/* RETURN TO THE CALLER. THE OUTPUT FIELDS ARE */
/* PROVIDED TO ALLOW THE EXIT TO PASS INFORMATION */
/* BACK TO THE CALLER. */
/* */
XPUVARS DEFINES THE OUTPUT AREA OF THE ARUDSNXP STRUCT AT TAG XPUVAREA. YOU MAY USE THIS STRUCT TO EASILY REFERENCE THE ARRAY DEFINED AT XPUVAREA.

NOTE:

THE NUMBER OF ENTRIES IN THE XPUVAREA ARRAY MUST NOT EXCEED 100 OR YOU WILL ADDRESS BEYOND THE END OF THE STRUCTURE STORAGE PROVIDED FOR THIS ROUTINE.

VARIABLE NAME:

1. MUST BEGIN WITH AN UNDERSCORE CHARACTER C'_' X'6D'
2. MUST BEGIN IN THE LEFTMOST BYTE OF XPUVNAME FIELD
3. IF LESS THAN 9 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS
4. MAY NOT CONTAIN EMBEDDED BLANKS
5. NON-BLANK CHARACTERS AFTER THE LEFTMOST BYTE MUST BE NATIONAL CHARACTERS WHICH ARE A THROUGH Z, 0 THROUGH 9, #, @, AND $
6. MUST BE UPPER CASE.

VARIABLE DATA:

1. MUST BEGIN IN THE LEFTMOST BYTE OF XPUVDATA FIELD
2. IF LESS THAN 8 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS
3. MAY NOT CONTAIN EMBEDDED BLANKS
4. NON-BLANK CHARACTERS MUST BE 'NATIONAL' CHARACTERS WHICH ARE A THROUGH Z, 0 THROUGH 9, #, @, $, -, AND .

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define XPUVENT 100
void dynwdsnx(struct XPVAREA*):
static int monthtab(||) = {
0, /* january */
31, /* february */
59, /* march */
90, /* april */
120, /* may */
151, /* june */
181, /* july */
212, /* august */
243, /* september */
273, /* october */
304, /* november */
334 /* december */
}
static int leaptab(||) = {
    0,                   /*   january   */
    31,                  /*   february  */
    60,                  /*   march     */
    91,                  /*   april     */
    121,                 /*   may       */
    152,                 /*   june      */
    182,                 /*   july      */
    213,                 /*   august    */
    244,                 /*   september */
    274,                 /*   october   */
    305,                 /*   november  */
    335                  /*   december  */
};

struct arudsnpx {
    char       xpjobn(|8|);     /* jobname                         */
    char       xpstep(|8|);     /* stepname                        */
    char       xpdbname(|8|);   /* database name                   */
    char       xpspname(|8|);   /* space name                      */
    char       xprttype(|2|);   /* reorg type (ts or ix)           */
    char       xpresrv1(|2|);   /* reserved for REORG PLUS         */
    char       xuser(|8|);      /* user id                         */
    char       xssid(|4|);      /* db2 subsystem id                */
    char       xdate(|6|);      /* utility execution date mmddyy   */
    char       xtime(|6|);      /* utility execution time hhmmss   */
    char       xutilid(|16|);   /* utility id                      */
    char       xdate8(|8|);     /* utility execution date mmddyyyy */
    char       xgrpnm(|4|);     /* data sharing group name         */
    char       xpvcat(|8|);     /* vcat name (from 1st part if partitnd*/
    char       xdatej(|7|);     /* utility execution date ccyyddd */
    char       xpresrv2(|13|);  /* reserved for REORG PLUS         */
    int           xusrwd1;     /* user word 1*/
    int           xusrwd2;     /* user word 2*/
    int           xusrwd3;     /* user word 3*/
    int           xusrwd4;     /* user word 4*/
    int           xusrwd5;     /* user word 5*/
    int           xusrwd6;     /* user word 6*/
    int           xusrwd7;     /* user word 7*/
    int           xusrwd8;     /* user word 8*/
    struct XPUVAREA {
        char xpuvname(|9|);;
        char xpuvdata(|8|);
    } xpuvars(|XPUVENT|)
};

int aruedsc (struct arudsnpx *xparm)
{
    char yr(|5|);
    char day(|4|);
    char month(|4|);
    char wuid(|17|);
Figure 78  DSNUEXIT sample C user exit (part 4 of 7)

```c
int i;
int x;
int l;
int wday;
int wyyear;
int iyear;
int imonth;
int totdays;
int xdisable;
int pfxlen;
int sfxstrt;
int sfxlen;
struct XPUVAREA *tp;

tp = xparm->xpuvars;

x = l = wday = wyyear = iyear = imonth = totdays = 0;
xdisable = i = pfxlen = sfxstrt = sfxlen = 0;

/* to disable this exit set  xdisable = 1 */
if (xdisable)
  return(0);

/* CREATE USER-DEFINED VARIABLES */
/* THE FOLLOWING USER VARIABLES WILL BE CREATED: */
/* */
/* _JDATE - JULIAN DATE IN THE FORM OF DYYDDD BUILT FROM THE */
/* REORG SUPPLIED VARIABLE &DATE8 */
/* _JCDATE - JULIAN DATE IN THE FORM OF DYYYYDDD BUILT FROM THE */
/* REORG SUPPLIED VARIABLE &DATE8 */
/* _UTILPFX - UP TO 8 BYTES OF THE FIRST NODE OF THE UTILITY ID */
/* _UTILSFX - UP TO 8 BYTES OF THE REMAINING PORTION OF THE */
/* UTILITY ID */

memcpy(month,xparm->xpdate8,2);
memcpy(day,xparm->xpdate8+2,2);
memcpy(yr,xparm->xpdate8+4,4);
memcpyp(month,xparm->xpdate8+2.2);
memcpyp(day,xparm->xpdate8+2+2.2);
memcpyp(yr,xparm->xpdate8+4+4.4);
```
Figure 78  DSNEXIT sample C user exit (part 5 of 7)

```c
sscanf(yr, "%d", &wyear);
sscanf(day, "%d", &wday);
sscanf(month, "%d", &imonth);
imonth = imonth - 1;
if (wyear % 4 == 0 && wyear % 100 != 0 || wyear % 400 == 0)
totdays = wday + leaptab(|imonth|);
else
totdays = wday + monthtab(|imonth|);
sprintf(yr, "%d", wyear);
sprintf(day, "%d", totdays);
/*--------------------------------------------------------------------*/
/* clear the user variable area to blanks */
/*--------------------------------------------------------------------*/
memset(tp->xpuvname,' ', sizeof(struct XPUVAREA));
/*--------------------------------------------------------------------*/
/* clear the user variable area to blanks */
/*--------------------------------------------------------------------*/
memcpy(tp->xpuvname,"_JDATE", 6);
memcpy(tp->xpuvdata,"D",1);
memcpy(tp->xpuvdata+1,xparm->xpdate8+6,2);
memcpy(tp->xpuvdata+3,day,3);
/*--------------------------------------------------------------------*/
/* sprintf(tp->xpuvdata,"D%-2.2s%3.3s",xparm->xpdate8+6,day); */
/* do not use this format as the '\0' used by sprintf will cause */
/* BMC51239I USER VARIABLE '_JDATE' HAS INVALID DATA */
/*--------------------------------------------------------------------*/
/* tp++; */
/*--------------------------------------------------------------------*/
/* clear the user variable area to blanks */
/*--------------------------------------------------------------------*/
memset(tp->xpuvname,' ', sizeof(struct XPUVAREA));
memcpy(tp->xpuvname,"_JCDATE", 7);
memcpy(tp->xpuvdata,"D",1);
memcpy(tp->xpuvdata+1,yr,4);
memcpy(tp->xpuvdata+5,day,3);
/*--------------------------------------------------------------------*/
/* sprintf(tp->xpuvdata,"D%-4.4s%-3.3s",yr,day); */
/* do not use this format as the '\0' used by sprintf will cause */
/* BMC51239I USER VARIABLE '_JDATE' HAS INVALID DATA */
/*--------------------------------------------------------------------*/
/* */
/* build _utilpfx and _utilsfx user variables */
/* */
/* tp++; */
/* */
/* sprintf(wuid,"%16.16s",xparm->xputilid); */
/* */
/* get the length of xputilid */
/* find the first delimiter character */
/* limit _UTILPFX and UTILSFX to 8 characters - DSN node limit */
/* */
```
for (i=0; wuid(|i|) !="\0"; i++)
{
    switch (wuid(|i|))
    {
    case ':':
    case ',':
    case '+':
    case '|':
    case ':':
    case '=':
    case '\':
    if (pfxlen == 0)
    {
        pfxlen = i;
        sfxstrt = i + 1;
    }
    break;
    default:
    break;
    }
}
if (pfxlen == 0) /* no delimiter found */
{
    pfxlen = 8;
    sfxstrt = 8;
}
if (sfxlen > 8)
sfxlen = 8;
if (sfxlen > 8)
sfxlen = 8;
/* clear the user variable area to blanks */
memset(tp->xpuvname,' ', sizeof(struct XPUVAREA));
memcpy(tp->xpuvname, "_UTILPFX", 8);
memcpy(tp->xpuvdata, xparm->xputilid, pfxlen);
/* sprintf(tp->xpuvdata,"%-8.8s",xparm->xputilid); */
/* do not use this format as the '"\0" used by sprintf will cause */
/* BMC51239I USER VARIABLE '_UTILPFX' HAS INVALID DATA */
tp++;
/* clear the user variable area to blanks */
This section provides a sample exit parameter structure, variable mapping structure, and exit that you can use as an example of how to write the DSNUEXIT user exit in LE C.

### LE C exit parameter structure

The LE C exit parameter structure (Figure 79) contains both input and output fields. Input fields pass vital information to the user exit, such as the database name and user ID. If you make any modifications to these input fields, they are disregarded on return. The output fields pass information about your user variables back to REORG PLUS.

```c
struct arudsnpx {
    char       xpjobn[8];       /* jobname                        */
    char       xpstep[8];       /* stepname                        */
    char       xpdbname[8];     /* database name                   */
    char       xpspname[8];     /* space name                      */
    char       xprtype[2];      /* REORG type (ts or ix)           */
    char       xpresrv1[2];     /* reserved for REORG PLUS         */
    char       xpuser[8];       /* user id                         */
    char       xpssid[4];       /* db2 subsystem id                */
    char       xpdata[6];       /* utility execution date mmddyy  */
    char       xptime[6];       /* utility execution time hhmmss   */
    char       xputilid[16];    /* utility id                      */
    char       xdate8[8];       /* utility execution date mmddyyyy */
    char       xgrpnm[4];       /* data sharing group name        */
    char       xvcat[8];        /* vcat name (from 1st part if partitnd) */
    char       xdatej[7];       /* utility execution date ccyyddd  */
    char       xpresrv2[13];    /* reserved for REORG PLUS         */
    int           xpusrwd1;     /* user word 1*/
};
```
Structure fields

Table 127 describes the major structure fields for the DSNUEXIT LE C user exit and their uses.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>xpjobn</td>
<td>job name</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>xstep</td>
<td>step name</td>
<td></td>
</tr>
<tr>
<td>xdbname</td>
<td>database name</td>
<td></td>
</tr>
<tr>
<td>xspname</td>
<td>name of the table space or index space from the REORG command</td>
<td></td>
</tr>
<tr>
<td>xrtype</td>
<td>type of reorganization being performed, table space (TS) or index (IX)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>xuser</td>
<td>USERID of the user running REORG PLUS</td>
<td>up to 8 bytes</td>
</tr>
<tr>
<td>xssid</td>
<td>DB2 subsystem ID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>xdate</td>
<td>date of the execution of the utility, in the format MMDDYY</td>
<td>6 bytes</td>
</tr>
<tr>
<td>xtime</td>
<td>time of the execution of the utility, in the format HHMMSS</td>
<td></td>
</tr>
<tr>
<td>xutilid</td>
<td>utility ID</td>
<td>up to 16 bytes</td>
</tr>
<tr>
<td>xdate8</td>
<td>date of the execution of the utility, in the format MMDDYYYY</td>
<td>8 bytes</td>
</tr>
<tr>
<td>xgrpmn</td>
<td>DB2 data sharing group name</td>
<td>not applicable</td>
</tr>
<tr>
<td>xvcat</td>
<td>VCATNAME specified in the DB2 catalog for the table space being reorganized, or for the first partition if the table space is partitioned</td>
<td></td>
</tr>
<tr>
<td>xdatej</td>
<td>Julian date of the execution of the utility, in the format CCYDDD</td>
<td>7 bytes</td>
</tr>
<tr>
<td>xusrwd1...xusrwd8</td>
<td>8 parameters of work space</td>
<td>up to 4 bytes each</td>
</tr>
</tbody>
</table>

LE C exit variable mapping structure

The LE C exit variable mapping structure (Figure 80 on page 825) defines the output area of the exit parameter structure. Using this structure allows you to easily address the variable definition table. The number of entries in this table (each entry consists of both a variable name and its value) must not exceed 100.
Table 128 describes the major variable mapping structure fields for the DSNUEXIT LE C user exit and their uses.

### Table 128 Major variable mapping structure fields for the DSNUEXIT LE C user exit

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPUVAREA</td>
<td>area containing user-defined variable information</td>
</tr>
<tr>
<td></td>
<td>For information about establishing user-defined variables, see page 791.</td>
</tr>
<tr>
<td>xpuvname</td>
<td>name of the variable you are defining</td>
</tr>
<tr>
<td>xpuvdata</td>
<td>value of the variable you are defining</td>
</tr>
<tr>
<td>XPUVENT</td>
<td>maximum number of entries in the output variable area (up to 100 entries)</td>
</tr>
</tbody>
</table>

**Sample LE C user exit**

Figure 81 shows an example of this user exit written in LE C.
/* PLEASE CALL BMC SOFTWARE CUSTOMER SUPPORT WITH ANY QUESTIONS YOU MAY HAVE IN THIS AREA. */
/* PHONE: 1-800-537-1813 */
/* */
/* NOTES */
/* */
/* ARUEDSC IS CALLED ONCE AT THE BEGINNING OF THE UNLOAD PHASE FOR A */
/* TWO-PHASE REORGANIZATION OR AT THE BEGINNING OF THE REORG PHASE */
/* FOR A SINGLE-PHASE REORGANIZATION. */
/* */
/* WHEN INVOKED IT IS PASSED THE ADDRESS OF A USER EXIT BLOCK */
/* DESCRIBED BY THE ARUDSNXP STRUCT. */
/* THIS CALL IS FROM ASEMBLER MODULE DYNWDSNX */
/* */
/* UPON EXIT RETRUN CODE FROM THIS EXIT */
/* */
/* O GOOD RETURN */
/* */
/* OTHER ERROR RETURN, TERMINATE UTILITY */
/* */
/* ARUDSNXP STRUCT THIS STRUCT CONTAINS BOTH INPUT */
/* AND OUTPUT FIELDS. INPUT FIELDS ARE PROVIDED TO */
/* PASS VITAL INFORMATION TO THIS EXIT. ANY MODIFI- */
/* CATIONS TO THESE FIELDS WILL BE DISREGARDED UPON */
/* RETURN TO THE CALLER. THE OUTPUT FIELDS ARE */
/* PROVIDED TO ALLOW THE EXIT TO PASS INFORMATION */
/* BACK TO THE CALLER. */
/* */
/* XPUVARS DEFINES THE OUTPUT AREA OF THE ARUDSNXP STRUCT AT TAG */
/* XPUVAREA. YOU MAY USE THIS STRUCT TO EASILY REFERENCE THE ARRAY */
/* DEFINED AT XPUVAREA. */
/* */
/* NOTE: */
/* */
/* THE NUMBER OF ENTRIES IN THE XPUVAREA ARRAY MUST NOT EXCEED 100 */
/* OR YOU WILL ADDRESS BEYOND THE END OF THE STRUCTURE STORAGE */
/* PROVIDED FOR THIS ROUTINE. */
/* */
/* VARIABLE NAME: */
/* */
/* 1. MUST BEGIN WITH AN UNDERSCORE CHARACTER C’_’ X’6D’ */
/* 2. MUST BEGIN IN THE LEFTMOST BYTE OF XPUVNAME FIELD */
/* 3. IF LESS THAN 9 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS */
/* 4. MAY NOT CONTAIN EMBEDDED BLANKS */
DSNUEXIT sample LE C user exit (part 3 of 7)

/* 5. NON-BLANK CHARACTERS AFTER THE LEFTMOST BYTE MUST BE NATIONAL */
/* CHARACTERS WHICH ARE A THROUGH Z, 0 THROUGH 9, #, @, AND $ */
/* 6. MUST BE UPPER CASE. */
/* */
/* VARIABLE DATA: */
/* */
/* 1. MUST BEGIN IN THE LEFTMOST BYTE OF XPUVDATA FIELD */
/* 2. IF LESS THAN 8 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS */
/* 3. MAY NOT CONTAIN EMBEDDED BLANKS */
/* 4. NON-BLANK CHARACTERS MUST BE 'NATIONAL' CHARACTERS WHICH ARE */
/* A THROUGH Z, 0 THROUGH 9, #, @, $, -, AND . */
/* */
/* */
#define XPUVENT 100
static int monthtab[] = {
    0, /* january */
    31, /* february */
    59, /* march */
    90, /* april */
    120, /* may */
    151, /* june */
    181, /* july */
    212, /* august */
    243, /* september */
    273, /* october */
    304, /* november */
    334 /* december */
};
static int leaptab[] = {
    0, /* january */
    31, /* february */
    60, /* march */
    91, /* april */
    121, /* may */
    152, /* june */
    182, /* july */
    213, /* august */
    244, /* september */
    274, /* october */
    305, /* november */
    335 /* december */
};
struct arudsnpx {
    char xpjobn[8]; /* jobname */
    char xpstep[8]; /* stepname */
    char xpdbname[8]; /* database name */
    char xpspname[8]; /* space name */
};
char xprtype[2]; /* REORG type (ts or ix) */
char xpresrv1[2]; /* reserved for REORG PLUS */
char xpuser[8]; /* user id */
char xpsssid[4]; /* db2 subsystem id */
char xpdte[6]; /* utility execution date mmddyy */
char xptime[6]; /* utility execution time hhmmss */
char xputilid[16]; /* utility id */
char xpdte8[8]; /* utility execution date mmddyyyy */
char xpdtej[7]; /* utility execution date ccyyddd */
char xpresrv2[13]; /* reserved for REORG PLUS */
int xpusrwd1; /* user word 1*/
int xpusrwd2; /* user word 2*/
int xpusrwd3; /* user word 3*/
int xpusrwd4; /* user word 4*/
int xpusrwd5; /* user word 5*/
int xpusrwd6; /* user word 6*/
int xpusrwd7; /* user word 7*/
int xpusrwd8; /* user word 8*/
struct XPUVAREA {
    char xpuvname[9];
    char xpuvdata[8];
    } xpuvars[XPUVENT];

int main (int argc, char *argv[])
{
    char yr[5];
    char day[4];
    char month[4];
    char wuid[17];
    int i;
    int x;
    int l;
    int wday;
    int wyear;
    int iyear;
    int imonth;
    int totdays;
    int xdisable;
    int pfxlen;
    int sfxstrt;
    int sfxlen;
    struct XPUVAREA *tp;
    struct arudsnpx *xparm;
    xparm = (void *)argv[1];
    tp = xparm->xpuvars;
    x = l = wday = wyear = iyear = imonth = totdays = 0;
    xdisable = i = pfxlen = sfxstrt = sfxlen = 0;
    /**************************************************************************/
Appendix C  REORG PLUS user exits  829

Figure 81  DSNUEXIT sample LE C user exit (part 5 of 7)
memcpy(tp->xpuvname, "_JDATE", 6);
memcpy(tp->xpuvdata, "D", 1);
memcpy(tp->xpuvdata+1, xparm->xpdate8+6, 2);
memcpy(tp->xpuvdata+3, day, 3);
/*--------------------------------------------------------------------*/
/* sprintf(tp->xpuvdata,"D%-2.2s%3.3s",xparm->xpdate8+6,day); */
/* do not use this format as the '\0' used by sprintf will cause */
/* BMC51239I USER VARIABLE '_JDATE' HAS INVALID DATA */
/*--------------------------------------------------------------------*/
tp++;
/*--------------------------------------------------------------------*/
/* clear the user variable area to blanks */
/*--------------------------------------------------------------------*/
memset(tp->xpuvname, ' ', sizeof(struct XPUVAREA));
memcpy(tp->xpuvname, "_JCDATE", 7);
memcpy(tp->xpuvdata, "D", 1);
memcpy(tp->xpuvdata+1, yr, 4);
memcpy(tp->xpuvdata+5, day, 3);
/*--------------------------------------------------------------------*/
/* sprintf(tp->xpuvdata,"D%-4.4s%-3.3s",yr,day); */
/* do not use this format as the '\0' used by sprintf will cause */
/* BMC51239I USER VARIABLE '_JCDATE' HAS INVALID DATA */
/*--------------------------------------------------------------------*/
/* build _utilpfx and _utilsfx user variables */
/*--------------------------------------------------------------------*/
tp++;
sprintf(wuid, "%16.16s", xparm->xputilid);
/*--------------------------------------------------------------------*/
/* get the length of xputilid */
/* find the first delimeter character */
/* limit _UTILPFX and UTILSFX to 8 characters -DSN node limit */
/*--------------------------------------------------------------------*/
for (i=0; wuid[i] != '\0'; i++)
{
    switch (wuid[i])
    {
    case ' ': case ',', case '_': case ':': case '=':
    case ':': case ':': case ':': case ':': case ':': case ':': case ':': case ':': case ':': case ':': case ':': case ':': case ':':
    case ':': case ':': case ':': case ':':
case '\b':
    if (pfxlen == 0)
    {
        pfxlen = i;
        sfxstrt = i + 1;
        break;
    }
    default:
        break;
}
if (pfxlen == 0)  /* no delimiter found */
{
    pfxlen = 8;
    sfxstrt = 8;
}
    sfxlen = i - sfxstrt;
    if (pfxlen > 8)
        pfxlen = 8;
    if (sfxlen > 8)
        sfxlen = 8;
/*--------------------------------------------------------------------*/
/*      clear the user variable area to blanks                       */
/*--------------------------------------------------------------------*/
memset(tp->xpuvname,' ', sizeof(struct XPUVAREA));
memcpy(tp->xpuvname, "_UTILPFX", 8);
memcpy(tp->xpuvdata, xparm->xutilid, pfxlen);
/*--------------------------------------------------------------------*/
/*  sprintf(tp->xpuvdata, "%-8.8s", xparm->xutilid);              */
/*  do not use this format as the '\0' used by sprintf will cause  */
/*  BMC512391 USER VARIABLE '_UTILPFX' HAS INVALID DATA              */
/*--------------------------------------------------------------------*/
    tp++;
/*--------------------------------------------------------------------*/
/*      clear the user variable area to blanks                        */
/*--------------------------------------------------------------------*/
memset(tp->xpuvname,' ', sizeof(struct XPUVAREA));
memcpy(tp->xpuvname, "_UTILSFX", 8);
memcpy(tp->xpuvdata, xparm->xutilid+sfxstrt, sfxlen);
/*--------------------------------------------------------------------*/
/*  sprintf(tp->xpuvdata, "%-8.8s", xparm->xutilid+8);             */
/*  do not use this format as the '\0' used by sprintf will cause    */
/*  BMC512391 USER VARIABLE '_UTILSPFX' HAS INVALID DATA              */
/*--------------------------------------------------------------------*/
    return (0);
Using a **DSRSEXIT, TERMEXIT, or MAPTEXIT user exit**

You can use the following user exits written in REXX:

- **DSRSEXIT** to manage the redefinition of DB2 VSAM data sets (see page 834)
- **TERMEXIT** to control updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS (see page 849)
- **MAPTEXIT** to create mapping objects for use with DSNUTILB (see page 857)

This section describes the following information common to all REXX exits that you can use with REORG PLUS:

- requirements and restrictions
- variables and return codes

To use a sample REXX user exit from the *HQL.LLQSAMP* library, copy the appropriate member as the base code for your exit; edit the code if needed, and save the modified exit in your SYSEXEC library. For additional instructions, see the section for the specific exit.

**DSRSEXIT, TERMEXIT, and MAPTEXIT common restrictions**

The REXX user exits have the following restrictions:

- REORG PLUS does not permit any TSO commands, such as ALLOC, in a REXX exit.
- You cannot perform a call to another REXX program from within one of your REXX user exit programs. Doing so will cause values in your REXX variables to be lost.

**DSRSEXIT, TERMEXIT, and MAPTEXIT common variables and return codes**

REORG PLUS communicates with REXX exits through REXX variables. Table 129 on page 833 lists the variables that are common to all REXX exits. For the list of variables unique to a particular REXX exit, see the section about variables for that exit.
Your REXX exit should always return a return code of 0 to REORG PLUS. REORG PLUS uses the REXX variables returned from the exit, not the return code, to determine the intent of the exit. If you specifically code a nonzero return code, REORG PLUS treats it as an environment failure and terminates the job.

Table 129 lists the variables that REORG PLUS passes to all REXX user exits. You cannot change the value of any of these variables.

**Table 129  Variables common to all REXX exits (part 1 of 2)**

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_EXIT_POINT</td>
<td>name of the exit point</td>
</tr>
<tr>
<td>BMCUTILITY_NAME</td>
<td>REORG</td>
</tr>
<tr>
<td>BMC_REORG_TYPE</td>
<td>reorganization type, either TABLESPACE or INDEX</td>
</tr>
<tr>
<td>BMCUTILITY_SHRLEVEL</td>
<td>the SHRLEVEL of the reorganization:</td>
</tr>
<tr>
<td></td>
<td>■ N for SHRLEVEL NONE</td>
</tr>
<tr>
<td></td>
<td>■ U for SHRLEVEL REFERENCE UNLOADONLY</td>
</tr>
<tr>
<td></td>
<td>■ R for SHRLEVEL REFERENCE</td>
</tr>
<tr>
<td></td>
<td>■ C for SHRLEVEL CHANGE</td>
</tr>
<tr>
<td>BMC_UTIL_ID</td>
<td>UTILID from the EXEC parameters</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS truncates longer utility IDs to eight characters.</td>
</tr>
<tr>
<td>BMC_UTILPREFIX</td>
<td>BMC utility ID prefix</td>
</tr>
<tr>
<td>BMC_UTIL_SUFFIX</td>
<td>BMC utility ID suffix</td>
</tr>
<tr>
<td>BMC_DB2_RELEASE</td>
<td>three-character version, release, and maintenance number (VRM) of the DB2 release</td>
</tr>
<tr>
<td>BMC_JOBNAME</td>
<td>JOB name used in the JCL</td>
</tr>
<tr>
<td>BMC_STEPNAME</td>
<td>STEP name used in the JCL</td>
</tr>
<tr>
<td>BMC_DBNAME</td>
<td>database containing the space being used for this data set allocation</td>
</tr>
<tr>
<td>BMC_TSIX</td>
<td>table space or index space name</td>
</tr>
<tr>
<td>BMC_USERID</td>
<td>job user ID</td>
</tr>
<tr>
<td></td>
<td>You must have a security package to use this variable.</td>
</tr>
<tr>
<td>BMC_VCATNAME</td>
<td>VCAT NAME specified in the DB2 catalog for the table space being reorganized; or, if the table space is partitioned, the VCAT name from the first part being reorganized</td>
</tr>
<tr>
<td>BMC_DB2SSID</td>
<td>DB2 subsystem ID</td>
</tr>
<tr>
<td>BMC_DB2_GROUPNAME</td>
<td>DB2 data sharing group name</td>
</tr>
<tr>
<td></td>
<td>In a non-data sharing environment, this variable contains the DB2 SSID.</td>
</tr>
<tr>
<td>BMC_DATE</td>
<td>current date (in the form MMDDYY)</td>
</tr>
</tbody>
</table>
REORG PLUS calls the DSRSEXIT user exit for each DB2 object before redefining that object. REORG PLUS obtains information about the object from the DB2 and VSAM catalogs and passes this information to the exit in REXX variables. REORG PLUS redefines the objects by using the values that you set in the exit. Based on the information passed, you can

- change the size of DB2 virtual sequential access method (VSAM) data sets by changing the primary and secondary allocations for the new data sets within the exit (see “Resizing DB2 objects” on page 835)

- update the DB2 catalog with the changed values for use with subsequent allocations unless you indicate otherwise (see “Resizing DB2 objects” on page 835)

- selectively tell REORG PLUS to specify REDEFINE NO for an object (see “Setting REDEFINE NO” on page 835)

- specify the order of the storage group volumes to use for each object (see “Using the DSRSEXIT user exit to order storage group volumes” on page 836)

- restrict the volume list to a subset of the original storage group volume list (see “Using the DSRSEXIT user exit to order storage group volumes” on page 836)

- add the DATACLAS, MGMTCLAS, and STORCLAS for storage groups (see “Using the DSRSEXIT user exit to add SMS classes” on page 836)

**NOTE**

Within the exit, if you specify REDEFINE NO for an object and you also change the primary or secondary quantity for that same object, REORG PLUS ignores REDEFINE NO and issues message BMC50307I.
Resizing DB2 objects

Resizing your DB2 objects during a reorganization allows you to

- reclaim unused space
- provide additional space for growth
- manage the object size at the partition level
- use the new size for subsequent allocations by altering the DB2 catalog to reflect the changes

After it has completed object redefinition, REORG PLUS issues an SQL ALTER TABLESPACE or ALTER INDEX statement to update the DB2 catalog with the modified primary and secondary quantities for use with subsequent allocations. You can modify a variable to prevent REORG PLUS from issuing this statement. For information about how to resize objects, see “Modifying the variables for resizing the data sets” on page 842.

Setting REDEFINE NO

You can improve performance by telling REORG PLUS not to redefine an object during the current reorganization. For that particular object, REORG PLUS will change REDEFINE YES to REDEFINE NO, bypassing VSAM DELETE/DEFINE processing. This function can be particularly useful in an SAP environment where many indexes exist, but only a few are growing and need to be redefined. For information about how to bypass the redefinition of objects, see “Modifying the variable for selective REDEFINE” on page 843.

Considerations for redefining a nonpartitioned DB2 object

If DB2 determines that the capacity of the A001 data set of a nonpartitioned DB2 object has been exceeded, DB2 allocates another data set (A002) with the same size as the original, creating a multiple data set table space or index space. During object redefinition, REORG PLUS calls the DSRSEXIT user exit once for each of these data set components so that you can perform any of the functions in the exit for each one individually.

Even though you cannot change the size of the additional data set components with DB2, you can change the size with the DSRSEXIT user exit. By default (because the BMC_ALTER_DB2_CATALOG variable is set to YES), REORG PLUS alters the DB2 catalog for the first data set component (A001). Because the DB2 catalog reflects only
one set of values for a nonpartitioned data set, REORG PLUS sets BMC_ALTER_DB2_CATALOG to NO for any other data set components. If you set BMC_ALTER_DB2_CATALOG to YES for any component other than the first, REORG PLUS ignores the request and issues the following message:

BMC50305I REQUESTED SQL ALTER OF DB2 CATALOG IGNORED FOR dbname.spname COMPONENT Annn

Using the DSRSEXIT user exit to order storage group volumes

To help balance the distribution of your DB2 objects, you can use the DSRSEXIT user exit to perform the following functions for each object that you select:

- specify the order in which you want REORG PLUS to use your storage group volumes
- return only a subset of the volumes to REORG PLUS from the original list, in the order in which you want them to be used

Changes made by the user exit only apply to the data set that you are processing. For information about how to order the storage group volumes, see “Modifying the variables for the storage group volumes” on page 843.

Using the DSRSEXIT user exit to add SMS classes

If no SMS classes (DATACLAS, MGMTCLAS, and STORCLAS) exist in the DSNZPARMs or storage-group definition, you can add the classes by using the DSRSEXIT user exit. However, you cannot change existing specified SMS classes by using the DSRSEXIT user exit. REORG PLUS ignores values that you specify for a class in the DSRSEXIT user exit if a value already exists for that class. REORG PLUS truncates values that exceed eight characters.

DSRSEXIT requirements and restrictions

In addition to the restrictions common to DSRSEXIT, TERMEXIT, and MAPTEXIT, the following requirements and restrictions apply to the DSRSEXIT user exit:

- You cannot use the DSRSEXIT with a DSNUTILB reorganization.
You can use the DSRSEXIT user exit only for storage-group-defined objects. If you want to alter the size of VCAT-defined objects, use the SYSIDCIN data set. For more information, see “SYSIDCIN data set” on page 374.

To use the DSRSEXIT user exit with a value of YES (the default) for the BMC_ALTER_DB2_CATALOG variable, you must have certain authorizations. For more information, see “Additional authorizations for using DSRSEXIT” on page 68.

If you specify a DSRSEXIT user exit and the value of the REDEFINE command or installation option is YES, REORG PLUS calls the exit before the VSAM DEFINE for all objects being reorganized. For information about the REDEFINE command option, see “REDEFINE” on page 210.

If you specify a DSRSEXIT user exit and the value of the REDEFINE command or installation option is NO, REORG PLUS calls the exit before the VSAM DEFINE if it needs to define

— an expansion data set for multiple data set objects
— a staging data set that was not preallocated by you

In all other cases, REORG PLUS does not call the exit if you specify REDEFINE NO on the command.

Running the DSRSEXIT exit

Before starting this procedure, ensure that the exit that you plan to use resides in your SYSEXEC library.

To use the DSRSEXIT exit

1 Specify the name of your user exit in one of the following ways:

• On the DSRSEXIT installation option, replace NONE with the name of your REXX user exit. For information about the DSRSEXIT installation option, see Appendix A, “REORG PLUS installation options.”

• On the REORG command, specify the DSRSEXIT option with the name of your REXX user exit. For information about the DSRSEXIT command option, see “DSRSEXIT” on page 241.

2 Include the following DD statement in your JCL, where dataSetName is the name of the library containing the REXX exit:

```bash
//SYSEXEC DD DSN=dataSetName,DISP=SHR
```
3 Include the following DD statement, where all output from REXX ‘SAY’ statements will be routed:

```
//SYSTSPRT DD SYSOUT=*  
```

In a RACF® environment, REORG PLUS uses the DB2 RACF ID when writing to SYSTSPRT from the REXX exit if OPNDB2ID=YES. For more information, see “Using RACF and OPNDB2ID=YES” on page 69 and the OPNDB2ID=YES installation option on page 713.

### DSRSEXIT variables

In addition to the variables common to all REXX exits (Table 129 on page 833), REORG PLUS passes other variables to the DSRSEXIT user exit. Table 130 on page 839 describes the variables specific to this exit, grouping the variables according to whether they apply to any object type, a table space only, or an index only. Table 130 also lists the following information about these variables:

- variable name
- brief description of the variable

If the variable is found in the DB2 catalog or in the integrated catalog facility (ICF), the table lists the corresponding DB2 or ICF column name. If the variable is specific to REORG PLUS, the table lists the values that you can enter.

- whether you can update the variable within the exit (yes or no)

For a more complete explanation of a variable and its valid values, refer to the IBM documentation. For more information about the variables that you can change, see “Modifying the variables for resizing the data sets” on page 842.

REORG PLUS re-initializes the REXX environment and all REXX variables each time it calls the exit. You can create any additional variables for the exit to use, but REORG PLUS does not retain the variables for subsequent exit calls.
<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description and DB2 or ICF column name, or valid values</th>
<th>Update?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variables that apply to all objects</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC_HIGH_USED_RBA</td>
<td>high-used relative byte address (HURBA) \nThis field is accurate for VSAM data sets that are less than or equal to 4 GB. If the data set is greater than 4 GB, the value of this field is set to 4 GB and you should use the RBN field counterpart instead.</td>
<td>no</td>
</tr>
<tr>
<td>BMC_HIGH_ALLOC_RBA</td>
<td>high-allocated relative byte address \nThis field is accurate for VSAM data sets that are less than or equal to 4 GB. If the data set is greater than 4 GB, the value of this field is set to 4 GB, and you should use the RBN field counterpart, instead.</td>
<td>no</td>
</tr>
<tr>
<td>BMC_HIGH_USED_RBN</td>
<td>high-used relative block number (number of 4 KB blocks used in the data set) \nThis field is accurate for any size VSAM data set.</td>
<td>no</td>
</tr>
<tr>
<td>BMC_HIGH_ALLOC_RBN</td>
<td>high-allocated relative block number (number of 4 KB blocks allocated in the data set) \nThis field is accurate for any size VSAM data set.</td>
<td>no</td>
</tr>
<tr>
<td>BMC_PRIMARY_QUANTITY</td>
<td>primary quantity, in 4-KB pages, to be allocated (PRIQTY)</td>
<td>yes</td>
</tr>
<tr>
<td>BMC_SECONDARY_QUANTITY</td>
<td>secondary quantity, in 4-KB pages, to be allocated (SECQTY)</td>
<td>yes</td>
</tr>
<tr>
<td>BMC_PRIMARY_SPACE_ICF</td>
<td>primary space currently being used (PRIMSPAC)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SECONDARY_SPACE_ICF</td>
<td>secondary space currently being used (SCONSPAC)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_NBR_EXTENTS</td>
<td>number of extents (NOEXTNT)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_ICF_ALLOC_TYPE</td>
<td>allocation type in tracks (TRK), cylinders (CYL), or blocks (BLK) (SPACOPTN)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_DB2_STORTYPE</td>
<td>VCAT (E) or STOGROUP (I) indicator (STORTYPE)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_ESTIMATED_CARDINALITY</td>
<td>estimate of the number of rows involved in the reorganization, from ANALYZE (integer)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_OBJECT_PART_NUMBER</td>
<td>current partition number of the table space or index (PARTITION)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_OBJECT_COMPONENT_NUMBER</td>
<td>data set number from the last node of the DB2 linear data set name (for example, 001 from .A001)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_DB2_FREEPAGE</td>
<td>number of pages loaded before a page is left as free space (FREEPAGE)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_DB2_PCTFREE</td>
<td>percentage of each page left as free space (PCTFREE)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_DB2_VCATNAME</td>
<td>data set name HLQ node of storage group or ICF catalog used for space allocation (VCAT NAME)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_DB2_STORNAME</td>
<td>storage group name (STORNAME)</td>
<td>no</td>
</tr>
</tbody>
</table>
### Table 130  Variables that REORG PLUS passes to DSRSEXIT (part 2 of 3)

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description and DB2 or ICF column name, or valid values</th>
<th>Update?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_DB2_DATACLAS</td>
<td>name of the SMS DATACLAS, taken from the DSN6SPRM macro in DSNZPARM</td>
<td>no</td>
</tr>
<tr>
<td>BMC_DB2_MGMTCLAS</td>
<td>specify the name of the SMS MGMTCLAS to add</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You cannot use the DSRSEXIT to change an existing specified SMS MGMTCLAS.</td>
<td></td>
</tr>
<tr>
<td>BMC_DB2_STORCLAS</td>
<td>specify the name of the SMS STORCLAS to add</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You cannot use the DSRSEXIT to change an existing specified SMS STORCLAS.</td>
<td></td>
</tr>
<tr>
<td>BMC_DATASET_NAME</td>
<td>fully qualified data set name</td>
<td>no</td>
</tr>
<tr>
<td>BMC_OBJECT_TYPE_IND</td>
<td>table space (TS) or index space (IX)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_ALTER_DB2_CATALOG</td>
<td>whether to update the DB2 catalog (YES or NO)</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>The default is YES.</td>
<td></td>
</tr>
<tr>
<td>BMC_REDEFINE_OBJECT</td>
<td>whether REORG PLUS should use REDEFINE YES (YES) or REDEFINE NO (NO) for the current object</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>The default is YES.</td>
<td></td>
</tr>
<tr>
<td>BMC_REORDER_STOGROUP_VOLUMES</td>
<td>whether REORG PLUS should use the storage group volume list returned from the exit (YES) or ignore it (NO)</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>The default is NO.</td>
<td></td>
</tr>
<tr>
<td>BMC_STOGROUP_VOLUME.0</td>
<td>stem variable that contains the number of volumes in the storage group list.</td>
<td>yes</td>
</tr>
<tr>
<td>BMC_STOGROUP_VOLUME.(n)</td>
<td>$BMC_{\text{STOGROUP} _ \text{VOLUME}.0} = n$, where $n$ is the number of volumes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$BMC_{\text{STOGROUP} _ \text{VOLUME}.1}$ through $BMC_{\text{STOGROUP} _ \text{VOLUME}.n}$ contain the actual names of volumes in the storage group</td>
<td></td>
</tr>
<tr>
<td>Variables that apply only to a table space</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC_SYSTABLESPACE_PARTITIONS</td>
<td>number of partitions of the table space or index (PARTITIONS)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSTABLESPACE_PGSIZE</td>
<td>size of pages in the table space in KB (PGSIZE)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSTABLESPACE_SEGSIZE</td>
<td>number of pages in each segment of a segmented table space (0 if not segmented) (SEGSIZE)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSTABLESPACE_NTABLES</td>
<td>number of tables defined in the table space (NTABLES)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSTABLESPACE_MAXROWS</td>
<td>maximum number of rows that DB2 will place on a data page (MAXROWS)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSTABLESPACE_DSSIZE</td>
<td>maximum size of a data set in KB (DSSIZE)</td>
<td>no</td>
</tr>
</tbody>
</table>
Table 130 Variables that REORG PLUS passes to DSRSEXIT (part 3 of 3)

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description and DB2 or ICF column name, or valid values</th>
<th>Update?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_SYSTABLESPACE_NACTIVE</td>
<td>number of active pages in the table space (NACTIVEF)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSTABLEPART_COMPRESSED</td>
<td>indicator that table space is compressed (COMPRESS)</td>
<td>no</td>
</tr>
<tr>
<td>Variables that apply only to an index</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC_SYSINDEXES_NAME</td>
<td>name of the index (NAME)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSINDEXES_DBNAME</td>
<td>name of the data base that contains the index (DBNAME)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSINDEXES_CREATOR</td>
<td>authorization ID of the owner of the index (CREATOR)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSINDEXES_INDEXSPACE</td>
<td>name of the index space (INDEXSPACE)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSINDEXES_CLUSTERING</td>
<td>whether CLUSTER was specified when the index was created (CLUSTERING)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSINDEXES_CLUSTERED</td>
<td>whether the table is actually clustered by the index (CLUSTERED)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSINDEXES_FULLKEYCARD</td>
<td>number of distinct values of the key (FULLKEYCARD(F))</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSINDEXES_UNIQUERULE</td>
<td>whether the index is unique (UNIQUERULE)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSINDEXES_PIECESIZE</td>
<td>maximum size of a data set in KB for nonpartitioned indexes (PIECESIZE)</td>
<td>no</td>
</tr>
<tr>
<td>BMC_SYSINDEXPART_LEAFDIST</td>
<td>100 times the average number of leaf pages between successive active leaf pages of the index (LEAFDIST)</td>
<td>no</td>
</tr>
</tbody>
</table>

If you are redefining a nonpartitioned DB2 object, for more information, see “Sample DSRSEXIT REXX user exit” on page 844.

If you restart a failed reorganization, REORG PLUS passes the original information to the exit for any object that is being redefined.

**Modifying DSRSEXIT variables**

REORG PLUS calls the DSRSEXIT user exit for each DB2 object before deleting or redefining that object. REORG PLUS obtains information about the object from the DB2 and VSAM catalogs and passes this information to the exit in REXX variables. In the exit, you can examine the returned values and change certain variables. When control is returned to REORG PLUS, REORG PLUS uses the values that you set. You can change variables to accomplish the following tasks:

- resize the data set (see “Modifying the variables for resizing the data sets” on page 842)
bypass redefining the data set (see “Modifying the variable for selective REDEFINE” on page 843)

order the storage group volumes (see “Modifying the variables for the storage group volumes” on page 843)

**Modifying the variables for resizing the data sets**

You can use the information in the variables to help you determine the primary and secondary quantities to allocate. Because DB2 stores the primary and secondary values in 4-KB pages in its catalog, the values passed to the exit represent 4-KB pages. To modify the variables BMC_PRIMARY_QUANTITY or BMC_SECONDARY_QUANTITY within the exit, set the quantity to the number of 4-KB pages that you want allocated. REORG PLUS uses the values that you set when it redefines the DB2 VSAM data sets.

REORG PLUS also issues an SQL ALTER TABLESPACE or ALTER INDEX statement to update the DB2 catalog with the modified primary and secondary quantities unless you set the variable BMC_ALTER_DB2_CATALOG to NO. The DB2 catalog tables that REORG PLUS updates are SYSIBM.SYSTABLEPART and SYSIBM.SYSINDEXPART.

**NOTE**

If you are reorganizing a partition-by-growth table space, REORG PLUS does not issue the ALTER statement to update the DB2 catalog.

If you use the default of YES for the BMC_ALTER_DB2_CATALOG variable to have REORG PLUS update the DB2 catalog, you will also need the following authorities:

- For the ALTER TABLESPACE statement, you will need one of these authorities:
  - ownership of the table space
  - DBADM authority for its database
  - SYSADM or SYSCTRL authority

- For the ALTER INDEX statement, you will need one of these authorities:
  - ownership of the index
  - ownership of the table on which the index is defined
  - DBADM authority for the database that contains the table
  - SYSADM or SYSCTRL authority
Modifying the variable for selective REDEFINE

If you do not want REORG PLUS to perform VSAM DELETE/DEFINE processing for a particular object, set BMC_REDEFINE_OBJECT to NO. Bypassing this processing significantly improves performance and can be especially useful in an SAP environment, where many indexes exist but most of them have not grown.

If you use the DSRSEXIT user exit to alter the primary or secondary quantity for an object and also set BMC_REDEFINE_OBJECT to NO, REORG PLUS ignores the REDEFINE NO request and issues message BMC50307I.

If you set BMC_REDEFINE_OBJECT to NO for a staging data set, you must have predefined the data set. Otherwise, REORG PLUS terminates the job.

Modifying the variables for the storage group volumes

When the DSRSEXIT gets control, the variable BMC_STOGROUP_VOLUME.0 contains the number of volumes in the storage group (and therefore the number of stem variables containing volume names). The stem variables BMC_STOGROUP_VOLUME.1 through n contain the volume names in the order that DB2 returns them from SYSIBM.SYSVOLUMES. You can change this order by assigning new values to the list of stem variables. Any changes that you make to the storage group order are in effect only for the current object.

You can reorder and return the entire original list or any subset of the list, but you cannot add any volume that is not in the original storage group list. If you add a volume name that is not in the original list, REORG PLUS terminates the job.

To change the volume order

1. Set the BMC_REORDER_STOGROUP_VOLUMES variable to YES.

   If you do not set this variable to YES, REORG PLUS will ignore any changes that you make to the original volume list.

2. Set a stem variable BMC_STOGROUP_VOLUME.n to the name of a storage group volume.

   You will have one BMC_STOGROUP_VOLUME.n variable for each volume in your list, incrementing n for each volume.
For example, suppose REORG PLUS returned these volumes in the original storage group list:

- BMC_STOGROUP_VOLUME.0=5
- BMC_STOGROUP_VOLUME.1=BMC001
- BMC_STOGROUP_VOLUME.2=BMC002
- BMC_STOGROUP_VOLUME.3=BMC003
- BMC_STOGROUP_VOLUME.4=BMC004
- BMC_STOGROUP_VOLUME.5=BMC005

To have REORG PLUS allocate the data sets on volume BMC005 first, and then BMC004, followed by BMC001, BMC003, and BMC002, you would leave the stem variable BMC_STOGROUP_VOLUME.0 set to 5 and change the values in the remaining stem variables as follows:

- BMC_STOGROUP_VOLUME.1=BMC005
- BMC_STOGROUP_VOLUME.2=BMC004
- BMC_STOGROUP_VOLUME.3=BMC001
- BMC_STOGROUP_VOLUME.4=BMC003
- BMC_STOGROUP_VOLUME.5=BMC002

To use a subset of the original volume list, set the stem variable BMC_STOGROUP_VOLUME.0 to the number of volumes that you want REORG PLUS to consider. (The maximum value for this stem variable is the total number of volumes in the STOGROUP.)

To use volumes BMC002, BMC003, and BMC004 (a subset of the volumes listed in the preceding example), you would change the values in the stem variables as follows:

- BMC_STOGROUP_VOLUME.0=3
- BMC_STOGROUP_VOLUME.1=BMC002
- BMC_STOGROUP_VOLUME.2=BMC003
- BMC_STOGROUP_VOLUME.3=BMC004

---

**Sample DSRSEXIT REXX user exit**

Figure 82 on page 845 provides a sample REXX exit that shows one way of setting the variables in the DSRSEXIT user exit.
Sample DSRSEXIT REXX user exit

Figure 82  Sample DSRSEXIT REXX user exit (part 1 of 5)

/* REXX */
/* *************************************************************** */
/* SAMPLE REXX EXIT FOR THE DATASET REDDEFINE EXIT-POINT 'DSRSEXIT' */
/* * */
/* THIS EXIT IS CALLED WHEN 'REDEFINE YES' AND DSRSEXIT=(NAME) ARE */
/* * SPECIFIED IN OPTIONS MODULE OR REORG SYNTAX. */
/* */
/* THE EXIT IS CALLED ONCE FOR EACH OBJECT IMMEDIATELY PRIOR TO ITS */
/* * DELETE/DEFINE. DURING RESTART OF A FAILED REORG PLUS JOB, ANY */
/* * OBJECT THAT MAY BE RE-DEFINED WILL HAVE ITS ORIGINAL INFORMATION */
/* * PASSED TO THE EXIT. */
/* */
/* THE REXX ENVIRONMENT AND ALL REXX VARIABLES ARE RE-INITIALIZED */
/* ON EACH CALL TO THE EXIT. YOU MAY CREATE ANY VARIABLES YOU WISH */
/* FOR USE BY THE EXIT BUT THEY WILL NOT BE RETAINED FOR SUBSEQUENT */
/* EXIT CALLS. */
/* */
/* THE ONLY VARIABLES THAT WILL BE INSPECTED AFTER CONTROL RETURNS */
/* TO REORG PLUS ARE: 'BMC_PRIMARY_QUANTITY', */
/* 'BMC_SECONDARY_QUANTITY' AND 'BMC_ALTER_DB2 CATALOG'. */
/* */
/* IF YOU CHANGE THE VALUE OF 'BMC_PRIMARY_QUANTITY' OR */
/* 'BMC_SECONDARY_QUANTITY' THEN THE NEW VALUE(S) WILL BE USED FOR */
/* * ALLOCATION OF THAT OBJECT. */
/* */
/* THE DEFAULT FOR 'BMC_ALTER_DB2_CATALOG' IS 'YES'. IF YOU */
/* *DO NOT* WANT REORG PLUS TO ISSUE 'ALTER TABLESPACE/INDEX' SQL */
/* TO REFLECT YOUR NEW 'BMC_PRIMARY_QUANTITY' OR */
/* 'BMC_SECONDARY_QUANTITY' IN THE DB2 CATALOG YOU MUST SET */
/* 'BMC_ALTER_DB2_CATALOG' TO 'NO'. */
/* */
/* //SYSEXEC DD IS REQUIRED FOR ALL REXX EXITS. IT SPECIFIES THE */
/* * LIBRARY CONCATENATION WHERE THE REXX SOURCE (THIS PROGRAM) WILL */
/* * BE FOUND. */
/* */
/* //SYSTSPRT DD IS REQUIRED FOR ALL REXX EXITS. ALL REXX 'SAY' */
/* COMMAND OUTPUT IS ROUTED TO THIS DD (USUALLY A SYSOUT). */
/* */
/* THIS SAMPLE EXIT PERFORMS THE FOLLOWING FUNCTIONS: */
/* */
/* 1. DISPLAYS ALL REXX VARIABLES PASSED TO THE EXIT FROM REORG PLUS*/
/* 2. INCREASES PRIMARY AND SECONDARY QUANTITY VALUES 20% IF THE */
/* NUMBER OF EXTENTS IS GREATER THAN 5. */
/* 3. SETS 'BMC_REDEFINE_OBJECT' TO 'NO' IF THE */
/* NUMBER OF EXTENTS IS LESS-THAN-OR-EQUAL-TO 5. */
/* 4. DOES *NOT* CHANGE THE VALUE OF 'BMC_ALTER_DB2_CATALOG'. THUS */
/* ALL PRIMARY/SECONDARY QUANTITY CHANGES WILL AUTOMATICALLY BE */
/* STAGED FOR SQL ALTER AFTER THE RELOAD (OR REORG) PHASE */
/* IS COMPLETE. */
/* 5. DISPLAYS THE VOLUME LIST FOR THE CURRENT STOGROUP. */
/* */
/* *************************************************************** */
/* IMPORTANT NOTE REGARDING REXX EXITS:                             */
/*    YOU MUST *NOT* PERFORM A CALL TO ANOTHER REXX PROGRAM !!      */
/*    DOING SO WILL CAUSE VALUES IN YOUR REXX VARIABLES TO BE LOST. */
/*    IT *IS* OK TO USE 'CALL' TO REFERENCE LABELED SUBROUTINES       */
/*    LOCATED WITHIN THE SAME REXX PROGRAM.                          */
/*                                                                      */
/* **************************************************************** */
SAY '**** START *********** ' BMC_EXIT_POINT ' ************'            
SAY 'BMC_EXIT_POINT               ='     BMC_EXIT_POINT
SAY 'BMC_UTIL_ID                  ='     BMC_UTIL_ID
SAY 'BMC_UTIL_ID                  ='     BMC_UTIL_ID
SAY 'BMC_DB2_RELEASE              ='     BMC_DB2_RELEASE
SAY 'BMC_JOBNAME                  ='     BMC_JOBNAME
SAY 'BMC_STEPNAME                 ='     BMC_STEPNAME
SAY 'BMC_DBNAME                   ='     BMC_DBNAME
SAY 'BMC_TSIX                     ='     BMC_TSIX
SAY 'BMC_USERID                   ='     BMC_USERID
SAY 'BMC_VCATNAME                 ='     BMC_VCATNAME
SAY 'BMC_DB2_SSID                 ='     BMC_DB2_SSID
SAY 'BMC_DB2_GROUPNAME            ='     BMC_DB2_GROUPNAME
SAY 'BMC_DATE                     ='     BMC_DATE
SAY 'BMC_TIME                     ='     BMC_TIME
SAY 'BMC_DATE8                    ='     BMC_DATE8
SAY 'BMC_TIME4                    ='     BMC_TIME4
SAY 'BMC_DATE3                    ='     BMC_DATE3
SAY 'BMC_UTILPFX                  ='     BMC_UTILPFX
SAY 'BMC_UTILPFX                  ='     BMC_UTILPFX
SAY 'BMC_HIGH_USED_RBA            ='     BMC_HIGH_USED_RBA
SAY 'BMC_HIGH_ALLOC_RBA           ='     BMC_HIGH_ALLOC_RBA
SAY 'BMC_HIGH_USED_RBN            ='     BMC_HIGH_USED_RBN
SAY 'BMC_HIGH_ALLOC_RBN           ='     BMC_HIGH_ALLOC_RBN
SAY 'BMC_PRIMARY_QUANTITY         ='     BMC_PRIMARY_QUANTITY
SAY 'BMC_SECONDARY_QUANTITY       ='     BMC_SECONDARY_QUANTITY
SAY 'BMC_PRIMARY_SPACE_ICF        ='     BMC_PRIMARY_SPACE_ICF
SAY 'BMC_SECONDARY_SPACE_ICF      ='     BMC_SECONDARY_SPACE_ICF
SAY 'BMC_NBR_EXTENTS              ='     BMC_NBR_EXTENTS
SAY 'BMC_ICF_ALLOC_TYPE           ='     BMC_ICF_ALLOC_TYPE
SAY 'BMC_DB2_STORCLASS            ='     BMC_DB2_STORCLASS
SAY 'BMC_ESTIMATED_CARDINALITY    ='     BMC_ESTIMATED_CARDINALITY
SAY 'BMC_OBJECT_PART_NUMBER       ='     BMC_OBJECT_PART_NUMBER
SAY 'BMC_OBJECT_COMPONENT_NUMBER  ='     BMC_OBJECT_COMPONENT_NUMBER
SAY 'BMC_DB2_FREEPAGE             ='     BMC_DB2_FREEPAGE
SAY 'BMC_DB2_PCTFREE              ='     BMC_DB2_PCTFREE
SAY 'BMC_DB2_VCATNAME             ='     BMC_DB2_VCATNAME
SAY 'BMC_DB2_OBJECT_TYPE          ='     BMC_DB2_OBJECT_TYPE
SAY 'BMC_DB2_DATACLASS            ='     BMC_DB2_DATACLASS
SAY 'BMC_DB2_STORCLASS            ='     BMC_DB2_STORCLASS
SAY 'BMC_DB2_MGMTCLASS            ='     BMC_DB2_MGMTCLASS
SAY 'BMC_DATASET_NAME             ='     BMC_DATASET_NAME
SAY 'BMC_OBJECT_TYPE_IND          ='     BMC_OBJECT_TYPE_IND
SAY 'BMCUTILITY_SHRLEVEL          ='     BMCUTILITY_SHRLEVEL

IF BMC_OBJECT_TYPE_IND = 'TS' THEN DO    /*   IF    TS OBJECT   */
Sample DSRSEXIT REXX user exit (part 3 of 5)

```rexx
SAY "BMC_SYSTABLESPACE_PARTITIONS =" BMC_SYSTABLESPACE_PARTITIONS
SAY "BMC_SYSTABLESPACE_PGSIZE =" BMC_SYSTABLESPACE_PGSIZE
SAY "BMC_SYSTABLESPACE_SEGSIZE =" BMC_SYSTABLESPACE_SEGSIZE
SAY "BMC_SYSTABLESPACE_NTABLES =" BMC_SYSTABLESPACE_NTABLES
SAY "BMC_SYSTABLESPACE_MAXROWS =" BMC_SYSTABLESPACE_MAXROWS
SAY "BMC_SYSTABLESPACE_DSSIZE =" BMC_SYSTABLESPACE_DSSIZE
SAY "BMC_SYSTABLESPACE_NACTIVE =" BMC_SYSTABLESPACE_NACTIVE
SAY "BMC_SYSTABLEPART_COMPRESSED =" BMC_SYSTABLEPART_COMPRESSED
END

ELSE DO    /*   ELSE  IX OBJECT   */
SAY "BMC_SYSINDEXES_NAME =" BMC_SYSINDEXES_NAME
SAY "BMC_SYSINDEXES_DBNAME =" BMC_SYSINDEXES_DBNAME
SAY "BMC_SYSINDEXES_CREATOR =" BMC_SYSINDEXES_CREATOR
SAY "BMC_SYSINDEXES_INDEXSPACE =" BMC_SYSINDEXES_INDEXSPACE
SAY "BMC_SYSINDEXES_CLUSTERING =" BMC_SYSINDEXES_CLUSTERING
SAY "BMC_SYSINDEXES_CLUSTERED =" BMC_SYSINDEXES_CLUSTERED
SAY "BMC_SYSINDEXES_FULLKEYCARD =" BMC_SYSINDEXES_FULLKEYCARD
SAY "BMC_SYSINDEXES_UNIQUERULE =" BMC_SYSINDEXES_UNIQUERULE
SAY "BMC_SYSINDEXES_PIECESIZE =" BMC_SYSINDEXES_PIECESIZE
SAY "BMC_SYSINDEXPART_LEAFDIST =" BMC_SYSINDEXPART_LEAFDIST
END

/* ****************************************************************
/*                                                                  */
/* CHANGING 'BMC_PRIMARY_QUANTITY' OR 'BMC_SECONDARY_QUANTITY' IS */
/* ALL YOU HAVE TO DO TO CHANGE THE ACTUAL VSAM ALLOCATION WHEN    */
/* 'REDEFINE YES' (DEFAULT) IS SPECIFIED IN YOUR REORG SYNTAX.      */
/*                                                                  */
/* SQL ALTER WILL ALSO BE PERFORMED AUTOMATICALLY SO THE PRI/SEC   */
/* QUANTITY CHANGE(S) ARE REFLECTED IN THE DB2 CATALOG TABLES      */
/* SYSIBM.SYSTABLEPART AND/OR SYSIBM.SYSINDEXPART.                  */
/*                                                                  */
/* IF YOU *DO NOT* WANT SQL ALTER PERFORMED THEN SET              */
/* 'BMC_ALTER_DB2_CATALOG = NO' FOR THAT OBJECT.                   */
/*                                                                  */
/* ****************************************************************
/*                                                                  */
/* NOTE FOR NON-PARTITIONED DB2 DATASETS:                          */
/*                                                                  */
/* FOR MULTIDATASET INDEX COMPONENTS (.A002 -> .A00N) THAT ARE NOT */
/* COMPONENTS OF A PARTITIONED INDEX (PERHAPS CREATED BECAUSE OF    */
/* A PIECESIZE SPECIFICATION)                                      */
/*                                                                  */
/* OR -                                                            */
/*                                                                  */
/* FOR MULTIDATASET TABLESPACE COMPONENTS (.A002 -> .A00N) THAT ARE */
/* NOT PART OF A PARTITIONED TABLESPACE                            */
/*                                                                  */
/* YOU *CAN* CHANGE THE PRIMARY_QUANTITY AND/OR SECONDARY_QUANTITY */
/* VALUES FOR THESE COMPONENTS.                                    */
/*                                                                  */
/* YOU *CAN NOT* SPECIFY 'BMC_ALTER_DB2_CATALOG = YES' FOR THEM     */
/* SINCE THEY DO NOT HAVE ENTRIES IN SYSIBM.SYSTABLEPART OR         */
```
/* OF COURSE, YOU *CAN* SPECIFY 'BMC_ALTER_DB2_CATALOG = YES' FOR */
/* THE .A001 COMPONENTS OF MULTIDATASET DB2 SPACES. */
/* OF COURSE, YOU *CAN* SPECIFY 'BMC_ALTER_DB2_CATALOG = YES' FOR */
/* ANY COMPONENT OF A PARTITIONED TABLESPACE OR PARTITIONED INDEX. */
/* */
/* **************************************************************** */
IF BMC_NBR_EXTENTS > 5 THEN DO
  SAY '******** ALLOCATIONS CHANGED ********************'
  BMC_PRIMARY_QUANTITY   = TRUNC(BMC_PRIMARY_QUANTITY   * 1.2)
  BMC_SECONDARY_QUANTITY = TRUNC(BMC_SECONDARY_QUANTITY * 1.2)
  SAY 'BMC_PRIMARY_QUANTITY         ='     BMC_PRIMARY_QUANTITY
  SAY 'BMC_SECONDARY_QUANTITY       ='     BMC_SECONDARY_QUANTITY
  SAY 'BMC_ALTER_DB2_CATALOG        ='     BMC_ALTER_DB2_CATALOG
END                                     /* END IF > 5   */
ELSE DO        /* NUMBER EXTENTS <= 5 SO SET REDEFINE NO FOR OBJECT */
  BMC_REDEFINE_OBJECT = 'NO'
END                                     /* END ELSE DO  */
SAY 'BMC_REDEFINE_OBJECT          ='     BMC_REDEFINE_OBJECT

/* **************************************************************** */
/* THE FOLLOWING SECTION DISPLAYS ALL VOLUMES IN THE */
/* CURRENT STOGROUP IN THE ORDER IN WHICH THEY WILL BE USED. */
/* */
/* THIS PROGRAM CAN BE MODIFIED TO CHANGE THE ORDER OF THE */
/* VOLUMES IN THE "BMC_STOGROUP_VOLUME." STEM VARIABLE. */
/* */
/* YOU MAY ALSO RETURN ONLY A *SUBSET* OF THE ORIGINAL VOLUME LIST */
/* IF YOU WISH.  TO DO THIS YOU SHOULD UPDATE AS MANY OF THE */
/* BMC_STOGROUP_VOLUME.1 THROUGH BMC_STOGROUP_VOLUME.N VARIABLES */
/* AS YOU WISH, AND THEN CHANGE THE BMC_STOGROUP_VOLUME.0 VARIABLE */
/* TO REFLECT THE NBR OF VOLUMES (1 THROUGH BMC_STOGROUP_VOLUME.0) */
/* YOU WISH TO BE CONSIDERED FOR ALLOCATION.  FOR INSTANCE, IF */
/* YOU SET "BMC_STOGROUP_VOLUME.0 = 3" THEN ONLY THE FIRST THREE */
/* VOLUMES IN THE BMC_STOGROUP_VOLUME. STEM VARIABLE WILL BE USED */
/* FOR ALLOCATION (IN THE ORDER YOU RETURN THEM). */
/* */
/* YOU MAY *NOT* ADD VOLUME NAMES THAT WEREN'T IN THE ORIGINAL LIST! */
/* */
/* IF YOU CHANGE THE ORDER OF VOLUMES IN YOUR STOGROUP BE SURE TO */
/* SET BMC_REORDER_STOGROUP_VOLUMES = 'YES' OR REORG PLUS WILL */
/* IGNORE YOUR CHANGES.  ANY CHANGES TO THE STOGROUP VOLUME ORDER */
/* WILL BE IN EFFECT ONLY FOR THE CURRENT OBJECT. */
Using TERMEXIT to control BMCHIST and statistics updates

TERMEXIT is the user exit for controlling BMCHIST and statistics functions in the UTILTERM phase. TERMEXIT provides REORG PLUS with user-defined variables that allow you to dynamically control processing of updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS at execution time.

For information about the BMCHIST and TERMEXIT command options, see Chapter 3, “Syntax of the REORG command.” For information about the corresponding installation options, see Appendix A, “REORG PLUS installation options.”

Within this exit, you can also use SQL to maintain your BMCHIST tables. The example in Figure 83 deletes all rows over 90 days old.

Figure 83  Example maintaining BMCHIST tables using SQL

```
/*                                                                 */
/* **************************************************************** */
SAY '**** START *********** STOGROUP SECTION ************'
SAY 'BMC_REORDER_STOGROUP_VOLUMES = '  BMC_REORDER_STOGROUP_VOLUMES
SAY 'NUMBER OF VOLUMES IN STOGROUP = '  BMC_STOGROUP_VOLUME.0
INDX = 1
DO WHILE INDX <= BMC_STOGROUP_VOLUME.0
  SAY 'BMC_STOGROUP_VOLUME.'INDX ' = ' BMC_STOGROUP_VOLUME.INDX
  INDX = INDX + 1
END                                     /* END DO WHILE */
SAY '**** END   *********** STOGROUP SECTION ************'
SAY '**** END   *********** ' BMC_EXIT_POINT ' ************'
RETURN
```
Additional TERMEXIT restrictions

In addition to the common restrictions that apply to DSRSEXIT, TERMEXIT, and MAPTEXIT user exits ("DSRSEXIT, TERMEXIT, and MAPTEXIT common restrictions" on page 832), the following restrictions apply to the TERMEXIT user exit:

- You cannot use a TERMEXIT user exit with a DSNUTILB reorganization.
- You cannot use a TERMEXIT user exit to override BMCSTATS NO or UPDATEDB2STATS NO to YES.

Running the TERMEXIT exit

Before starting this procedure, ensure that the exit that you plan to use resides in your SYSEXEC library.

To use the TERMEXIT exit

1. Specify the name of your user exit in one of the following ways:
   - On the TERMEXIT installation option, replace NONE with the name of your REXX user exit. For information about the TERMEXIT installation option, see Appendix A, “REORG PLUS installation options.”
   - On the REORG command, specify the TERMEXIT option with the name of your REXX user exit. For information about the TERMEXIT command option, see “TERMEXIT” on page 242.

2. Include the following DD statement in your JCL, where dataSetName is the name of the library containing the REXX exit:

```
//SYSEXEC DD DSN=dataSetName,DISP=SHR
```

3. Include the following DD statement, where all output from REXX 'SAY' statements will be routed:

```
//SYSTSPRT DD SYSOUT=* 
```

In a RACF environment, REORG PLUS uses the DB2 RACF ID when writing to SYSTSPRT from the REXX exit if OPNDB2ID=YES. For more information, see “Using RACF and OPNDB2ID=YES” on page 69 and the OPNDB2ID=YES installation option on page 713.
TERMEXIT variables

In addition to the variables common to all REXX exits (Table 129 on page 833), REORG PLUS passes additional variables to the TERMEXIT user exit. Table 131 describes the variables specific to this exit and indicates whether you can update the variable (yes or no).

Table 131 Variables that REORG PLUS passes to TERMEXIT

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Update?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_OPT_BMCHIST</td>
<td>The value of this variable is populated with the value specified in the BMCHIST installation option or command option. You can modify this variable to dynamically override your BMCHIST option. Valid values are Yes and No.</td>
<td>yes</td>
</tr>
<tr>
<td>BMC_OPT_RTS</td>
<td>You can modify this variable to dynamically control the real-time statistics update function. The default value is Yes. Valid values are Yes and No.</td>
<td>yes</td>
</tr>
<tr>
<td>BMC_OPT_UPDATEDB2STATS</td>
<td>The initial value of this variable is populated with the value of the UPDATEDB2STATS command option. You can modify this variable to dynamically override the UPDATEDB2STATS option from UPDATEDB2STATS YES to UPDATEDB2STATS NO. The only valid value is No.</td>
<td>yes</td>
</tr>
<tr>
<td>BMC_OPT_BMCSTATS</td>
<td>The initial value of this variable is populated with the value of the BMCSTATS command option. You can modify this variable to dynamically override the BMCSTATS option from BMCSTATS YES to BMCSTATS NO. The only valid value is No.</td>
<td>yes</td>
</tr>
<tr>
<td>BMC_CREATOR_BMCHIST</td>
<td>The value of this variable is populated with your BMCHIST table CREATOR.</td>
<td>no</td>
</tr>
<tr>
<td>BMC_TBNAME_BMCHIST</td>
<td>The value of this variable is populated with your BMCHIST table NAME.</td>
<td>no</td>
</tr>
<tr>
<td>BMC_REORG_PART_NUMBERS</td>
<td>The value of this stem variable is populated with the partition numbers of any partition involved in the reorganization. BMC_REORG_PART_NUMBERS.0=ni, where ni is the number of partitions involved in the reorganization. BMC_REORG_PART_NUMBERS.1 through BMC_REORG_PART_NUMBERS.n contain the actual partition numbers, or 0 if the object is not partitioned.</td>
<td>no</td>
</tr>
</tbody>
</table>
Sample TERMEXIT REXX user exit

Figure 84 provides a sample REXX exit that shows one way of setting the variables in the TERMEXIT user exit.

**Figure 84  Sample TERMEXIT REXX user exit (part 1 of 6)**

```rexx
/* REXX */
/* **********************************************************
SAMPLE REXX EXIT FOR THE 'TERMEXIT' EXIT-POINT.

THE SAMPLE EXIT PERFORMS THE FOLLOWING FUNCTIONS:
  1. DISPLAYS ALL VALUES BUT MAKES NO CHANGES.
  2. DISPLAYS ALL PART NUMBERS INVOLVED IN THE REORG (OR ZERO).
  3. DISPLAYS LAST UTILITY INFORMATION FOR THIS UTILID FROM BMCHIST.
  4. DISPLAYS REAL-TIME-STATS REOGLASTTIME VALUE FOR TS/IXSPACE.

IF YOU CHANGE ANY OF THE _OPT_ YES/NO VALUES THEN REORG PLUS
WILL ISSUE MESSAGE BMC50308I AND YOUR NEW VALUE WILL BE IN EFFECT.

NOTE: 1. ERRORS IN THIS EXIT WILL NOT AFFECT THE FINAL RC OF THE REORG.
  2. INVALID VALUES RETURNED ARE SIMPLY IGNORED BY REORG PLUS.

********************************************************** */

SAY '**** START *********** ' BMC_EXIT_POINT ' ************'

SAY '* BMC_OPT_RTS             = ' BMC_OPT_RTS
SAY '* BMC_OPT_UPDATEDB2STATS  = ' BMC_OPT_UPDATEDB2STATS
SAY '* BMC_OPT_BMCSTATS        = ' BMC_OPT_BMCSTATS
SAY '* BMC_OPT_BMCHIST         = ' BMC_OPT_BMCHIST

SAY ' '.
SAY '* NUMBER OF PARTS IN REORG = ' BMC_REORG_PART_NUMBERS.0

DO I=1 TO BMC_REORG_PART_NUMBERS.0
   SAY ' ** PART ' BMC_REORG_PART_NUMBERS.I ' INVOLVED IN REORG'
END

SAY ' '.

/' SUBCOM DSNREXX'         /* HOST CMD ENV AVAILABLE? */
```
IF RC THEN S_RC = RXSUBCOM('ADD','DSNREXX','DSNREXX')

/* ***********************************************************************************/
/* CONNECT TO DB2 USING VARIABLE PASSED FROM REORG */
/* ***********************************************************************************/
CURR_FUNC = 'CONNECT TO DB2'
ADDRESS DSNREXX 'CONNECT' BMC_DB2_SSID

IF SQLCODE = 0 THEN DO
   SAY '*** CONNECTED TO SUBSYSTEM = ' BMC_DB2_SSID
END
ELSE DO
   SAY '*** CONNECT TO SUBSYSTEM = ' BMC_DB2_SSID ' *** FAILED ***'
   SIGNAL BAD_SQLCODE
END

/* *****************************************************************/
/* PREPARE STMT / DECLARE CURSOR / OPEN CURSOR */
/* *****************************************************************/
SQLHIST1 = 'SELECT COUNT(*), MAX(DATE) FROM ' ,
   STRIP(BMC_TBCREATOR_BMCHIST)||'.'||BMC_TBNAME_BMCHIST,
   ' WHERE UTILID = ? '
SQLHIST2 = 'SELECT MAX(TIME) FROM ' ,
   STRIP(BMC_TBCREATOR_BMCHIST)||'.'||BMC_TBNAME_BMCHIST,
   ' WHERE UTILID = ? AND DATE = ? '
CURR_FUNC = 'PREPARE SQL HIST1'             /* PREPARE SQLHIST1 */
ADDRESS DSNREXX ,
   'EXECSQL PREPARE S1 FROM :SQLHIST1'
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE
CURR_FUNC = 'PREPARE SQL HIST2'             /* PREPARE SQLHIST2 */
ADDRESS DSNREXX ,
   'EXECSQL PREPARE S2 FROM :SQLHIST2'
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE
CURR_FUNC = 'DECLARE CURSOR C1'             /* DECLARE C1 FOR HIST1 */
ADDRESS DSNREXX ,
   'EXECSQL DECLARE C1 CURSOR FOR S1'
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE
CURR_FUNC = 'DECLARE CURSOR C2'             /* DECLARE C2 FOR HIST2 */
ADDRESS DSNREXX ,
   'EXECSQL DECLARE C2 CURSOR FOR S2'
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE
CURR_FUNC = 'OPEN CURSOR C1'                /* OPEN CURSOR C1 */
ADDRESS DSNREXX ,
   'EXECSQL OPEN C1 USING :BMC_UTIL_ID'
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE

/* ****************************************************************
 * FETCH PRIOR BMCHIST UTILITY HISTORY
 * *******************************************************************/

SAY
SAY '** BMCHIST UTILITY HISTORY **'

CURR_FUNC = 'FETCH CURSOR C1'      /* GET UTILCOUNT AND LAST-DATE */
ADDRESS DSNREXX ,
   'EXECSQL FETCH C1 INTO :UTIL_COUNT, :LAST_UTIL_DATE:NULLIND'
IF SQLCODE /= 0 THEN SIGNAL BMCHIST_SELECT
IF UTIL_COUNT = 0 THEN SIGNAL BMCHIST_SELECT

CURR_FUNC = 'OPEN CURSOR C2'       /* OPEN CURSOR C2       */
ADDRESS DSNREXX ,
   'EXECSQL OPEN C2 USING :BMC_UTIL_ID, :LAST_UTIL_DATE'
IF SQLCODE /= 0 THEN SIGNAL BMCHIST_SELECT

CURR_FUNC = 'FETCH CURSOR C2'      /* GET LAST RUNTIME ON MAX_DATE */
ADDRESS DSNREXX ,
   'EXECSQL FETCH C2 INTO :LAST_UTIL_TIME:NULLIND'

BMCHIST_SELECT:
SELECT
   WHEN (SQLCODE = 0) & (UTIL_COUNT > 0) THEN DO
      SAY '*** FOR UTILID                  =' BMC_UTIL_ID
      SAY '*** LAST RUN DATE              =' LAST_UTIL_DATE
      SAY '*** LAST RUN TIME              =' LAST_UTIL_TIME
      SAY '*** TOTAL ROWS FOR THIS UTILID =' UTIL_COUNT
   END      /* END SQLCODE = 0   */

   WHEN (SQLCODE = 100) | (UTIL_COUNT = 0) THEN DO
      SAY '*** NO PRIOR HISTORY FOR UTILID = ' BMC_UTIL_ID
   END      /* END SQLCODE = 100 */

   OTHERWISE SIGNAL BAD_SQLCODE

END                         /*  END SELECT    */

SAY
ADDRESS DSNREXX  'EXECSQL CLOSE C1'
ADDRESS DSNREXX  'EXECSQL CLOSE C2'
ADDRESS DSNREXX  'EXECSQL  COMMIT'
/* ****************************************************************
PREPARE_STMT / DECLARE_CURSOR / OPEN_CURSOR FOR RTS DATA.
NOTE: RTS DATA IS ONLY AVAILABLE FOR DB2 V7 AND ABOVE.
******************************************************************* */

IF BMC_DB2_RELEASE < '710' THEN RETURN   /* EXIT IF NOT V7 OR ABOVE */

SQL_STMTS = 'SELECT REORGLASTTIME FROM SYSIBM.TABLESPACESTATS ',
           'WHERE DBNAME = ? AND NAME = ? AND PARTITION = ? '

SQL_STMT_MIX = 'SELECT REORGLASTTIME FROM SYSIBM.INDEXSPACESTATS ',
               'WHERE DBNAME = ? AND INDEXSPACE = ? AND PARTITION = ? '

CURR_FUNC = 'DECLARE_CURSOR C3'
ADDRESS DSNREXX,
  'EXECSQL DECLARE C3 CURSOR FOR S3'
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE

/* ****************************************************************
LOOP FOR EACH PART IN BMC_REORG_PART_NUMBERS.0
******************************************************************* */

SAY '** REAL-TIME-STATS HISTORY **'

DO PART_IX = 1 TO BMC_REORG_PART_NUMBERS.0    /* LOOP FOR NUMPARTS */
    CURR_PART = BMC_REORG_PART_NUMBERS.PART_IX
    IF BMC_REORG_TYPE = 'TABLESPACE' THEN DO    /* IF REORG TABLESPACE */
        CURR_FUNC = 'PREPARE_SQL_STMTS'
        ADDRESS DSNREXX,
            'EXECSQL PREPARE S3 FROM :SQL_STMTS'
        END
        ELSE DO                                      /* ELSE REORG INDEX   */
    CURR_FUNC = 'PREPARE_SQL_STMT_MIX'
    ADDRESS DSNREXX,
        'EXECSQL PREPARE S3 FROM :SQL_STMT_MIX'
    END

    IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE

    CURR_FUNC = 'OPEN_CURSOR C3'
    ADDRESS DSNREXX,
        'EXECSQL OPEN C3 USING :BMC_DBNAME, :BMC_TSIX, :CURR_PART'

    IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE

    /* ****************************************************************
FETCH RTS HISTORY FOR CURRENT PART NUMBER
NOTE: THIS EXAMPLE FETCHES ROWS FOR THE TARGET OF THE REORG ONLY.
YOU MAY ADD SELECTS FOR SUBORDINATE OBJECTS IF YOU WISH.
******************************************************************* */
Sample TERMEXIT REXX user exit

Figure 84  Sample TERMEXIT REXX user exit (part 5 of 6)

CURR_FUNC = 'FETCH CURSOR C3'
ADDRESS DSNREXX ,
    'EXECSQL FETCH C3 INTO :REORG_DATE:NULLIND'

SELECT
    WHEN (SQLCODE = 0) & (NULLIND = 0) THEN DO
        SAY '** LAST REORG RTS TIMESTAMP FOR ',
        STRIP(BMC_DBNAME)||.||STRIP(BMC_TSIX),
        ' PART ' CURR_PART ' = ' REORG_DATE
    END      /* END SQLCODE = 0 */

    WHEN (SQLCODE = 100) | (NULLIND /= 0) THEN DO
        SAY '** LAST REORG RTS TIMESTAMP FOR ',
        STRIP(BMC_DBNAME)||.||STRIP(BMC_TSIX),
        ' PART ' CURR_PART ' =  * NO REORG RTS HISTORY *'
    END      /* END SQLCODE = 100 */

    OTHERWISE SIGNAL BAD_SQLCODE
END                         /*  END SELECT  */

ADDRESS DSNREXX  'EXECSQL CLOSE C3'
ADDRESS DSNREXX  'EXECSQL  COMMIT'
END     /* END  DO PART_IX = 1 TO BMC_REORG_PART_NUMBERS.0  */

/* **************************** **************************************************
   DISCONNECT DB2 THREAD
   **************************** ************************************************** */

ADDRESS DSNREXX DISCONNECT /*  DISCONNECT INTERFACES */

SAY
SAY '**** END          *********** ' BMC_EXIT_POINT ' ************'
RETURN     /*  NORMAL EXIT FOR THIS PROGRAM */

/* **************************** **************************************************
   SQL ERROR - IF YOU GET HERE SOMETHING BAD HAPPENED
   **************************** ************************************************** */

BAD_SQLCODE:
    SAY
    SAY '*** ERROR DURING DB2 ''CURR_FUNC'' SQLCODE ' SQLCODE

    SAY
    SAY '******************** SQLCA CONTENTS *********************'

856  REORG PLUS for DB2 Reference Manual
Using MAPTEXIT to create mapping objects

MAPTEXIT creates a mapping table and index for use with SHRLEVEL CHANGE reorganizations that invoke DSNUTILB. Following a successful reorganization, this exit drops these objects. If the specified mapping table already exists, REORG PLUS uses that table.

NOTE

The default (sample) MAPTEXIT user exit creates the mapping objects in an implicit database and table space. As with all DB2 implicit objects, dropping the mapping table drops the table space but not the database.

You can supply this exit in one of the following ways:

- In your STEPLIB

  This method provides a mapping table, when needed for a DSNUTILB reorganization, without requiring you to change your JCL.

- In a SYSEXEC DD statement in your job
When you specify a MAPTEXIT user exit, REORG PLUS looks for it first in your STEPLIB, then in a SYSEXEC DD.

**Additional MAPTEXIT requirements and restrictions**

In addition to the common restrictions that apply to DSRSEXIT, TERMEXIT, and MAPTEXIT user exits, the following requirements and restrictions apply specifically to MAPTEXIT user exits:

- You must have additional authorizations as described in “Additional authorizations for using MAPTEXIT” on page 68.

- REORG PLUS ignores any MAPTEXIT user exit that you specify when any of the following conditions exists:
  - You are running on DB2 Version 11 or later.
  - You specify the MAPPINGTABLE option.
  - You specify REORG INDEX.

**Supplying MAPTEXIT in your STEPLIB**

To have REORG PLUS create and drop mapping objects when needed without requiring you to change your JCL, include a MAPTEXIT user exit library in your STEPLIB.

**To provide a MAPTEXIT user exit in your STEPLIB**

1. Compile the REXX source for the user exit by using the NOCEXEC option of the REXX compiler.

2. Link the resulting load module into an unformatted PDS.

3. Ensure that the data set is APF authorized.

4. Include the data set in your STEPLIB.

5. On the MAPTEXIT installation option, replace NONE with the name of your REXX user exit.

For information about the MAPTEXIT installation option, see Appendix A, “REORG PLUS installation options.”
Supplying MAPTEXIT in a SYSEXEC DD statement

Before starting this procedure, ensure that the exit that you plan to use resides in your SYSEXEC library.

To supply the MAPTEXIT user exit by using a SYSEXEC DD statement

1. Specify the name of your user exit in one of the following ways:
   - On the MAPTEXIT installation option, replace NONE with the name of your REXX user exit. For information about the MAPTEXIT installation option, see Appendix A, “REORG PLUS installation options.”
   - On the REORG command, specify the MAPTEXIT option with the name of your REXX user exit. For information about the MAPTEXIT command option, see “MAPTEXIT” on page 243.

2. Include the following DD statement in your JCL, where dataSetName is the name of the library containing the REXX exit:

   ```sysexec DD DSN=dataSetName,DISP=SHR```

3. (optional) Include the following DD statement, where all output from REXX 'SAY' statements will be routed:

   ```sysprint DD SYSOUT=*
   ```

   For a MAPTEXIT user exit, REORG PLUS dynamically allocates SYSTSPRT if your JCL does not include a SYSTSPRT DD statement.

   In a RACF environment, REORG PLUS uses the DB2 RACF ID when writing to SYSTSPRT from the REXX exit if OPNDB2ID=YES. For more information, see “Using RACF and OPNDB2ID=YES” on page 69 and the OPNDB2ID=YES installation option on page 713.

MAPTEXIT variables

In addition to the variables common to all REXX exits (Table 129 on page 833), REORG PLUS passes additional variables to the MAPTEXIT user exit. Table 132 on page 860 describes the variables specific to this exit and indicates whether you can update the variable (yes or no).
### Variable name | Description | Update?
--- | --- | ---
BMC_MAPT_NOCEXEC | Whether the exit was called from the STEPLIB (Y) or a SYSEXEC DD (N) | No
BMC_MAPT_TYPE | Whether the exit is to create a mapping table (when called prior to DSNUTILB execution) or drop a mapping table (when called after DSNUTILB completes successfully). | No
BMC_MAPT_AUTH | Authorization ID that the exit uses in the SET CURRENT SQLID SQL statement
The initial value of this variable is set to the authorization ID that is in effect for the REORG PLUS job.
If you change the value for this variable, the authorization ID that you supply must have the authorizations as described in “Additional authorizations for using MAPTEXIT” on page 68. | Yes
BMC_MAPT_CREATOR\textsuperscript{a} | Creator portion of the fully qualified mapping table name and mapping table index name
The initial value of this variable is set to the user ID of the job. | Yes
BMC_MAPT_SUFX | Suffix concatenated to the BMC_MAPT_TABLE and BMC_MAPT_INDEX values to ensure that fully qualified object names are unique to a job
The initial value of this variable is set to the job name. | Yes
BMC_MAPT_TABLE\textsuperscript{a} | Table name portion of the fully qualified mapping table name
The initial value of this variable is set to map_tbl (the default DB2 mapping table name). | Yes
BMC_MAPT_INDEX | Index name portion of the fully qualified mapping table index name.
The initial value of this variable is set to xmap_tbl (the default DB2 mapping table index name). | Yes
BMC_MAPT_CRLN_ADDR\textsuperscript{a} | Address of the creator length, used only when BMC_MAPT_NOCEXEC=Y | No
BMC_MAPT_CRNM_ADDR\textsuperscript{a} | Address of the creator name, used only when BMC_MAPT_NOCEXEC=Y | No
BMC_MAPT_TBLN_ADDR\textsuperscript{a} | Address of the table name length, used only when BMC_MAPT_NOCEXEC=Y | No
BMC_MAPT_TBNM_ADDR\textsuperscript{a} | Address of the table name, used only when BMC_MAPT_NOCEXEC=Y | No

\textsuperscript{a} The exit passes this value back to REORG PLUS to pass to DSNUTILB with the MAPPINGTABLE keyword.
Sample MAPTEXIT REXX exit

Figure 85 shows the sample MAPTEXIT REXX exit provided with REORG PLUS.

Figure 85 Sample MAPTEXIT REXX user exit (part 1 of 9)

/* REXX */
/* ******************************************************************
SAMPLE REXX EXIT FOR THE 'MAPTEXIT' EXIT-POINT.

THE SAMPLE EXIT PERFORMS THE FOLLOWING FUNCTIONS:
1. DISPLAYS ALL VALUES BUT MAKES NO CHANGES.
2. CONNECTS TO DB2 SUBSYSTEM USING DSNREXX.
3. CREATES MAPPING TABLE FOR IBM SHRLEVEL CHANGE REORG BEFORE
   ARUUMAIN CALL TO DSNUTILB OR...
   DROPS MAPPING TABLE FOR IBM SHRLEVEL CHANGE REORG AFTER
   ARUUMAIN CALL TO DSNUTILB.
4. DISCONNECTS FROM DB2 SUBSYSTEM
5. RETURNS BMC_MAPT_CREATOR AND BMC_MAPT_TABLE TO REORG PLUS TO BE
   USED IN BUILDING DSNUTILB COMMAND STATEMENT.

NOTE: 1. ERRORS IN THIS EXIT MAY CAUSE REORG PLUS OR CALL TO DSNUTILB
   TO FAIL.
   2. INVALID VALUES RETURNED ARE NOT VALIDATED BY REORG PLUS.

THERE ARE THREE WAYS THIS REXX CAN BE PASSED TO A REORG PLUS JOB:
1. AS A COMPILED REXX LOAD MODULE BY COMPILING THIS SOURCE USING
   THE REXX COMPILER WITH THE 'NOCEXEC' OPTION AND LINKING THE
   MODULE INTO AN UNFORMATTED PDS. SPECIFY THAT DATASET ON THE
   STEPLIB DD OF THE REORG PLUS JOB AND ENSURE THE DATASET IS
   APF AUTHORIZED.

2. AS INTERPRETED REXX BY COPYING THIS EXAMPLE INTO A FORMATTED
   SEQUENTIAL DATASET OR PDS. SPECIFY THAT DATASET ON THE SYSEXEC
   DD OF THE REORG PLUS JOB.

3. AS 'COMPILED EXEC' REXX BY COMPILING THIS SOURCE USING THE
   REXX COMPILER WITH THE 'CEXEC' OPTION AND LINKING THE MODULE
   INTO A FORMATTED SEQUENTIAL DATASET OR PDS (CAN BE THE SAME
   PDS AS INTERPRETED REXX). SPECIFY THAT DATASET ON THE SYSEXEC
   DD OF THE REORG PLUS JOB.

****************************************************************** */

ARG PARM
IF PARM == "" THEN DO
   BMC_MAPT_NOCEXEC = 'N'
END
ELSE DO
   BMC_MAPT_NOCEXEC = 'Y'
   PARSE UPPER VAR PARM BMC_EXIT_POINT,
   BMC_UTIL_ID,
   BMC_Utility_NAME,
   BMC_DB2_RELEASE,
SAY '**** START *********** ' BMC_EXIT_POINT ' ************'

IF BMC_MAPT_NOCEXEC = 'N' THEN DO
  SAY '* EXECUTING REXX VIA SYSEXEC DD'
END
ELSE DO
  SAY '* EXECUTING COMPILED REXX VIA STEPLIB DD'
END

/* **************.MODIFICATIONS AFTER THIS COMMENT ARE NOT RECOMMENDED.***************/

BMC_MAPT_TABLE = 'MAP_TBL_'||STRIP(BMC_MAPT_SUFX)
BMC_MAPT_INDEX = 'XMAP_TBL_'||STRIP(BMC_MAPT_SUFX)
SAY '  BMC_DB2_SSID               = ' BMC_DB2_SSID
SAY '  BMC_MAPT_TYPE              = ' BMC_MAPT_TYPE
SAY '  BMC_MAPT_AUTH              = ' BMC_MAPT_AUTH
SAY '  BMC_MAPT_CREATOR           = ' BMC_MAPT_CREATOR
SAY '  BMC_MAPT_SUFX              = ' BMC_MAPT_SUFX
SAY '  BMC_MAPT_TABLE             = ' BMC_MAPT_TABLE
SAY '  BMC_MAPT_INDEX             = ' BMC_MAPT_INDEX

SAY '  '  

/* ****************************************************************
  ENSURE DSNREXX COMMAND ENVIRONMENT AVAILABLE - ADD IT IF NOT
  ******************************************************************* */
'SUBCOM DSNREXX'                     /*  HOST CMD ENV AVAILABLE? */
IF RC THEN S_RC = RXSUBCOM('ADD','DSNREXX','DSNREXX')

/* ****************************************************************
  CONNECT TO DB2 USING VARIABLE PASSED FROM REORG
  ******************************************************************* */
CURR_FUNC = 'CONNECT TO DB2'
ADDRESS DSNREXX 'CONNECT' BMC_DB2_SSID

IF SQLCODE = 0 THEN DO  
  SAY '*** CONNECTED TO SUBSYSTEM = ' BMC_DB2_SSID
  SAY '  '  
END
ELSE DO  
  SAY '*** CONNECT TO SUBSYSTEM = ' BMC_DB2_SSID ' *** FAILED ***'
  SAY '  '  
  SIGNAL BAD_SQLCODE
END

CURR_FUNC = 'SET SQLID TO AUTHID'
SQLAUTH = 'SET CURRENT SQLID = ' BMC_MAPT_AUTH

SQLCALL = "EXECSQL " SQLAUTH
ADDRESS DSNREXX SQLCALL

IF BMC_MAPT_TYPE = 'CREATE' THEN DO  
  SIGNAL CREATE_TABLE
END
ELSE DO  
  SIGNAL DROP_TABLE
END

/* ****************************************************************
  CREATE MAPPING TABLE AND INDEX FOR DB2 VERSION 10 AND LESS
  ******************************************************************* */
CREATE_TABLE:
CURR_FUNC = 'CREATE TABLE'
SQLCRTTB = 'CREATE TABLE ',
    STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE,
    ' (TYPE         CHAR( 01 ) NOT NULL,
    '  SOURCE_RID   CHAR( 05 ) NOT NULL,
    '  TARGET_XRID  CHAR( 09 ) NOT NULL,
    '  LRSN         CHAR( 06 ) NOT NULL) '
SAY '*** CREATING MAPPING TABLE ***'
SAY ''
SQLCALL = "EXECSQL " SQLCRTTB
ADDRESS DSNREXX SQLCALL
SELECT
    WHEN (SQLCODE = -601) | (SQLCODE = 0) THEN DO
        IF SQLCODE = 0 THEN DO
            SAY '*   TABLE ' STRIP(BMC_MAPT_CREATOR)||'.',
            ||BMC_MAPT_TABLE ' CREATED'
            SAY ''
            END
        ELSE DO
            SIGNAL RESTART_PROCESS
            END
    END
    OTHERWISE SIGNAL BAD_SQLCODE
END
CURR_FUNC = 'COMMIT TABLE CREATE'
ADDRESS DSNREXX 'EXECSQL COMMIT'
IF SQLCODE = 0 THEN DO
    SIGNAL CREATE_INDEX
    END
ELSE DO
    SAY '*   COMMIT FAILED'
    SAY ''
    SIGNAL BAD_SQLCODE
    END

/* ***************************************************
   RESTART PROCESSING, DROP TABLE/INDEX UNLESS -666
   *************************************************** */
RESTART_PROCESS:
SAY '*   TABLE ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE,
    ' EXISTS FROM PREVIOUS EXECUTION, DETERMINE DROP OR REUSE...'  
SAY ''
CURR_FUNC = 'RESTART DROP MAPPING TABLE AND INDEX'
SQLDRPTB = 'DROP TABLE ' STRIP(BMC_MAPT_CREATOR)||'.BMC_MAPT_TABLE

SQLCALL = "EXEC SQL” SQLDRPTB
ADDRESS DSNREXX SQLCALL
SELECT
  WHEN (SQLCODE = -666) | (SQLCODE = 0) THEN DO
    IF SQLCODE = 0 THEN DO
      SAY '*   TABLE ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE,
       ' EXPLICITLY DROPPED.'
      SAY ' ';
      SAY '*   INDEX ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_INDEX,
       ' IMPLICITLY DROPPED.'
      SAY ' ';
      CURR_FUNC = 'RESTART COMMIT TABLE DROP/REUSE'
      ADDRESS DSNREXX 'EXEC SQL COMMIT'
      IF SQLCODE = 0 THEN DO
        SIGNAL DISCONNECT_FROM_DB2
      END
      ELSE DO
        SAY '*   COMMIT FAILED'
        SAY ' ';
        SIGNAL BAD_SQLCODE
      END
      END
    ELSE DO
      SAY '*   TABLE ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE,
       ' WILL BE REUSED.'
      SAY ' ';
      SAY '*   INDEX ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_INDEX,
       ' WILL BE REUSED.'
      SAY ' ';
      CURR_FUNC = 'RESTART COMMIT TABLE DROP/REUSE'
      ADDRESS DSNREXX 'EXEC SQL COMMIT'
      IF SQLCODE = 0 THEN DO
        SIGNAL DISCONNECT_FROM_DB2
      END
      ELSE DO
        SAY '*   COMMIT FAILED'
        SAY ' ';
        SIGNAL BAD_SQLCODE
      END
      END
  OTHERWISE SIGNAL BAD_SQLCODE
END
CURR_FUNC = 'RESTART TABLE CREATE'
SQLCALL = "EXEC SQL” SQLCRTTB
ADDRESS DSNREXX SQLCALL
IF SQLCODE = 0 THEN DO
  SAY '*   TABLE ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE,
   ' RECREATED FOR CURRENT EXECUTION'
  SAY ' ';
END
ELSE DO
  SIGNAL BAD_SQLCODE
END
CURR_FUNC = 'RESTART COMMIT TABLE CREATE'
ADDRESS DSNREXX 'EXEC SQL COMMIT'
IF SQLCODE = 0 THEN DO
  SIGNAL_CREATE_INDEX
END
CREATE MAPPING TABLE AND INDEX FOR DB2 VERSION 10 AND LESS
*******************************************************************************

CREATE_INDEX:

CURR_FUNC = 'CREATE INDEX'
SQLCRTIX = 'CREATE UNIQUE INDEX ',
            STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_INDEX,
            ' ON ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE,
            ' (SOURCE_RID ASC, TYPE, TARGET_XRID, LRSN)'.
            ' USING STOGROUP SYSDEFLT PRIQTY -1 SECQTY -1'

SAY '*** CREATING MAPPING INDEX ***'
SAY ''

SQLCALL = "EXECSQL " SQLCRTIX
ADDRESS DSNREXX SQLCALL

SELECT WHEN (SQLCODE = -601) | (SQLCODE = 0) THEN DO
  IF SQLCODE = 0 THEN DO
    SAY '*   INDEX ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_INDEX ' CREATED'
    SAY ''
  ELSE DO
    SIGNAL BAD_SQLCODE
  END
ELSE DO
  SAY '*   INDEX ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_INDEX ' EXISTS FROM PREVIOUS EXECUTION, DROPPING...'
  SAY ''
  CURR_FUNC = 'DROP   INDEX'
  SQLDRPIX = 'DROP INDEX ',
            STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_INDEX
  SQLCALL = "EXECSQL " SQLDRPIX
  ADDRESS DSNREXX SQLCALL
  IF SQLCODE = 0 THEN DO
    SAY '*   INDEX ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_INDEX, ' EXPLICITLY DROPPED.'
    SAY ''
  END
ELSE DO
  SIGNAL BAD_SQLCODE
END

SQLCALL = "EXECSQL " SQLCRTIX
ADDRESS DSNREXX SQLCALL
Figure 85 Sample MAPTEXIT REXX user exit (part 7 of 9)

```plaintext
IF SQLCODE = 0 THEN DO
    SAY '*   INDEX ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_INDEX ' RECREATED FOR CURRENT EXECUTION'
    SAY '
END
ELSE DO
    SIGNAL BAD_SQLCODE
END

OTHERWISE SIGNAL BAD_SQLCODE
END

CURR_FUNC = 'COMMIT INDEX CREATE'
ADDRESS DSNREXX 'EXECSQL COMMIT'
IF SQLCODE = 0 THEN DO
    SIGNAL DISCONNECT_FROM_DB2
END
ELSE DO
    SAY '*   COMMIT FAILED'
    SAY '
    SIGNAL BAD_SQLCODE
    END

/* ****************************************************************
  DROP MAPPING TABLE AND INDEX
*********************************************************************/
DROP_TABLE:

CURR_FUNC = 'DROP MAPPING TABLE AND INDEX'
SQLDRPTB = 'DROP TABLE ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE

SAY '*** DROPPING MAPPING TABLE AND INDEX ***'
SAY '

SQLCALL = "EXECSQL " SQLDRPTB
ADDRESS DSNREXX SQLCALL

SELECT
    WHEN (SQLCODE = -204) | (SQLCODE = 0) THEN DO
        IF SQLCODE = 0 THEN DO
            SAY '*   TABLE ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE, ' EXPLICITLY DROPPED.'
            SAY '
        END
        SAY '*   INDEX ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_INDEX, ' IMPLICITLY DROPPED.'
        SAY '
        END
    ELSE DO
        SAY '*   TABLE ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE, ' DOES NOT EXIST.'
    END
```
Cur_Func = 'COMMIT TABLE AND INDEX DROP'
ADDRESS DSNREXX 'EXECSQL COMMIT'

IF SQLCODE /= 0 THEN DO
  SAY '*   COMMIT FAILED'
  SAY ' ',
  SIGNAL BAD_SQLCODE
END

/* ****************************************************************
   DISCONNECT DB2 THREAD
******************************************************************* */
DISCONNECT_FROM_DB2:
ADDRESS DSNREXX DISCONNECT /*  DISCONNECT INTERFACES */

/* ****************************************************************
   FOR NOCEXEC COMPILED REXX, USE STORAGE FUNCTION TO PASS
   BMC_MAPT_CREATOR AND BMC_MAPT_TABLE BACK TO ARUUMAIN.

   FOR INTERPRETED AND CEXEC COMPILED REXX, ARUUMAIN MAKES A CALL
   TO IRXEXCOM TO RETRIEVE BMC_MAPT_CREATOR AND BMC_MAPT_TABLE
******************************************************************* */
IF BMC_MAPT_NOCEXEC = 'Y' THEN DO
  /* STORE TABLE CREATOR */
  BMC_MAPT_CREATOR = STRIP(BMC_MAPT_CREATOR)
  BMC_MAPT_CRLEN = LENGTH(BMC_MAPT_CREATOR)
  STCRLN = STORAGE(BMC_MAPT_CRLN_ADDR,,D2C(BMC_MAPT_CRLEN,2))
  STCRNM = STORAGE(BMC_MAPT_CRNM_ADDR,BMC_MAPT_CRLEN,BMC_MAPT_CREATOR)

  /* STORE TABLE NAME */
  BMC_MAPT_TABLE = STRIP(BMC_MAPT_TABLE)
  BMC_MAPT_TBLEN = LENGTH(BMC_MAPT_TABLE)
  STTBLN = STORAGE(BMC_MAPT_TBLN_ADDR,,D2C(BMC_MAPT_TBLEN,2))
  STTBNM = STORAGE(BMC_MAPT_TBNM_ADDR,BMC_MAPT_TBLEN,BMC_MAPT_TABLE)
END

SAY
SAY '**** END *********** ' BMC_EXIT_POINT ' ************'
SAY

RETURN     /*  NORMAL EXIT FOR THIS PROGRAM */
/* ****************************************************************
   SQL ERROR - IF YOU GET HERE SOMETHING BAD HAPPENED
******************************************************************* */
BAD_SQLCODE:
  SAY  '*** ERROR DURING DB2 ''CURR_FUNC'' SQLCODE ' SQLCODE
  SAY  '******************** SQLCA CONTENTS ********************'
  SAY 'SQLCODE     = ' SQLCODE
  SAY 'SQLERRMC    = ' SQLERRMC
  SAY 'SQLERRP     = ' SQLERRP

T_INDEX = 1
DO WHILE T_INDEX <= 6   /* LOOP THROUGH SQLERRD 1-6 */
  SAY 'SQLERRD.'T_INDEX'-1 = ' SQLERRD.T_INDEX
  T_INDEX = T_INDEX + 1
END

T_INDEX = 1
DO WHILE T_INDEX <= 9   /* LOOP THROUGH SQLWARN 1-9 */
  SAY 'SQLWARN.'T_INDEX'-1 = ' SQLWARN.T_INDEX
  T_INDEX = T_INDEX + 1
END

SAY 'SQLWARN.10  = ' SQLWARN.10  /* THIS KEEPS SYSPRINT ALIGNED */
SAY 'SQLSTATE    = ' SQLSTATE

SAY  '****** TERMINATING DUE TO SQL ERROR IN ' BMC_EXIT_POINT

EXIT 8
Index

Symbols

$ARUDYN A macro 668, 740, 741
$ARUOPTS macro 668
$AUPSMAC macro 668
$C30DOPT job 667, 668
$C32SOPT job 667, 668
$ORTPARM DD statement 386

&DATE variable
  with DSNPAT option 316, 747
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332

&DATEJ variable
  with DSNPAT option 316, 747
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332

&DB variable
  with DSNPAT option 316, 747
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332

&DDNAME variable 316, 747

&GRPNM variable
  with DSNPAT option 316, 747
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332

&JDATE variable
  with DSNPAT option 316, 747
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332

&JOBNAME variable 316, 726

&PART variable 317, 747

&PART5 variable 317, 748

&RTYPE variable
  with DSNPAT option 317, 748
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332

&SSID variable
  with DSNPAT option 317, 748
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332

&STEPNAME variable
  with DSNPAT option 317, 748
  with SPIILDSNP option 726

&TIME variable
  with DSNPAT option 317, 748
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332

&TIME4 variable
  with DSNPAT option 317, 748
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332

&TSIX variable
  with DSNPAT option 317, 748
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332

&UID variable
  with DSNPAT option 318, 748
  with SPIILDSNP option 727
  with SPIILDSNPAT option 332

&USERID variable
  with DSNPAT option 318, 748
  with SPIILDSNP option 727
  with SPIILDSNPAT option 332

&UTIL variable
  with DSNPAT option 318, 748
  with SPIILDSNP option 727
  with SPIILDSNPAT option 332

&UTILID variable 318, 748

&UTILPFX variable
  with DSNPAT option 318, 749
  with SPIILDSNP option 727
  with SPIILDSNPAT option 332

&UTILSFX variable
  with DSNPAT option 318, 749
  with SPIILDSNP option 727
  with SPIILDSNPAT option 333

&VCAT variable
  with DSNPAT option 318, 749
  with SPIILDSNP option 727
  with SPIILDSNPAT option 333

A

ABEND keyword of TIMEOUT option 239
abends
  and data capture flag 657
  excluding from a system dump 690
  JES3 386
  recovering from, using the ON FAILURE option 224
  restarting from 391
  restarting with SHRLEVEL CHANGE 657
  taking a system dump 720
  using ANALYZE statistics to avoid 570
above-the-bar storage 71
above-the-line storage 193, 582, 676
access, shared 774
accessibility of objects
  SHRLEVEL option 98, 172
tuning considerations 599
ACFORTSS option 33
activating dynamic allocation
  ACTIVE option 296, 742
    example 507
ACTIVE option
  command 296
  installation 742
  restart considerations 96
advisory pending trigger 142
advisory restart pending (AREST) status 84, 87
ALL keyword
  of DRAIN option 340
  of DRAINTYP option 686
  of FORCE option 250, 693
  of REGISTER option 267
ALLOC option 299, 744
allocating data sets
  See also dynamic allocation of data sets
  avoiding I/O queueing 589
copy data sets 354
dynamically. See dynamic allocation of data sets
estimating size with ANALYZE 386, 570, 601
incremental copy data sets 654
inline copy data sets 265, 700
multiple 584
pattern for spill data set names 726
reallocating during a job 201
SORTWK 369
staging data set requirements 101
VIO, restrictions 369
allocation method, specifying for dynamic allocation 299
allocation size
  adjusting 311, 755
  limiting 307, 753
ALTER authority 70
ALTER SQL statement
  backing out 674
  in DDLOUT data set 365
  with DDLIN data set 360
  with primary, secondary quantities 835
altered index compression definition 148
altering
  data set allocations 835
  limit keys 360
altering a SHRLEVEL CHANGE job dynamically 640
ALTRFAIL option 674
AMEND function of DATA PACKER for DB2 214
AMENDED option 214
ampersand
  in DSNPAT option 746
  in SPIILDSNPAT option 726
ANALMAX option
  considerations 586
  description 674
ANALYZE option
  description 199
dynamic allocation considerations 199, 387
  example 417, 481, 554
  meanings of subparameters 199
  retrieving DASD MANAGER PLUS statistics 202
ANALYZE phase
  allocation-related messages 572
  bypassing gathering statistics 572
data set allocation estimates 386
determining method to use 571
  multitasking considerations 586
  pausing and continuing after analysis 199
primary functions 52
  restarting 398
  specifying ON FAILURE 225
tuning considerations 574, 601
ANY keyword of ALLOC option 299
APF-authorization requirements 667
append inserts trigger 142
application defaults module 344
applying log records, online REORG 567
APPNDINS exception 142
AR/CTL (APPLICATION RESTART CONTROL)
  implementing the interface 651
  purpose 650
  requirements 650
  specifying 674
ARC option 674
ARCHDDN option
  command 185
  example 492
  installation 675
ARCHFORMAT option 185, 675
archive data sets 185, 370, 675
  See also SYSARC data sets
ARCHIVE keyword of DDTYPE option 294, 742
archive-enabled tables 279
ARCCROWS option 229
AREOPEND exception 142
AREST (advisory restart pending) status 84, 87
ARU$OPTS load module 667
$ARUDYNA installation options 740
$ARUOPTS installation options 668
ARUUMAIN module 342, 391
ASM keyword of DSNUEXIT option 240
assembler, sample DSNUEXIT exit 791
assigning copy data sets for partial reorganization 682
ASSOCIATE option of ORDER YES option 191
ASUSRPRRT data set 258, 351
attributes of data sets, checks for 96
authorizations
  ALTER authority 70
  APF 667
authorizations (continued)
  CONTROL authority 70
  DISPLAY 67, 69
  for canceling threads 69
  for XML reorganizations 68
  mechanisms, description 67
  MONITOR2 67
  needed to execute 66
  RACF 69
  TRACE 67

AUTO keyword
  of ICTYPE option 268
  of SIXSNAP option 251
  of VOLCNT option 304, 760

AUTOESTSPACE option 256
automatic backout of reorganization 402
AUX option 256
AUXREORG option 676
availability of objects
  SHRLEVEL option 98, 172
  tuning considerations 599
available pages 193, 676
AVAILPAGEPCT option 193, 676
average row length, determining 199
AVGVOLSP option 306, 744

B
backing out reorganization 402, 674
backing up BMC tables 766
base objects that participate in a clone relationship 171, 678
base table space instance number 678
BASIC keyword of DSNTYPE option 309, 750
basic row format 149
batch applications, using with AR/CTL 650
BEGINNING keyword of LONGNAMETRUNC option 178, 708
below-the-line storage 582, 583
benefits
  of an online reorganization 44, 48
  of REORG PLUS 44
BILDMAX option 587, 677
binary strings 287, 292
BMC database tables 765
BMC keyword
  of ARCHFORMAT option 186, 675
  of CONDEXEC option 230
  of NLPCTFREE option 210
  of STAGEDSN option 728
BMC Software, contacting 2
BMC utilities
  displaying status 768
  running concurrently 82
  terminating 768

BMCDICT table
  considerations 769
  contents 769
  maintaining 770
  when updated 53
  with large number of partitions 110
BMCFORCE data set 352
BMCHIST option 208, 770
BMCHIST table
  backing up 766
  contents 770
  controlling update of 208
  maintenance 772
  querying 768
BMCLGRNX table 772
BMCPWD library 348
BMCRUSER exception 140
BMCSORT
  controlling 580, 583
  data set that contains messages 60, 386
  description 61
  dynamic allocation 187, 188
  installation option 761
  MAXSORTS option 189
  multitasking 587, 720, 730
  SMAX option 723
  SMCORE option 724
  SORTDEVT option 187, 724
  SORTNUM option 188, 725
  version requirement 66
BMCSSTATS keyword of ANALYZE option 202
BMCSSTATS option
  description 259
  example 417
BMCSYNC table
  backing up 766
  considerations 775
  contents 773
  frequency of rows written to, controlling 208
  LOB data considerations 776
  maintaining 776
  running utilities concurrently 82
  serializing 705
  terminating a job 396
  when updated 53
  with large number of partitions 110
  XML data considerations 775
BMCTRANS table 777
BMCTRIG utility, using 138
BMCUUTIL table
  backing up 766
  contents 778
  maintaining 780
  ON FAILURE TERMINATE, effect of 225
  serializing 705
BMCUTIL table (continued)
utility IDs 345
when updated 53
BMCXCOPY table
backing up 766
contents 781
maintaining 785
querying 788
buffers
assigned to DB2 by REORG PLUS 579
controlling usage 575, 579
I/O, maximum virtual storage 589
virtual storage 589
VSAM and the 16-megabyte line 575
BUILD phase, multitasking 587
building REORG PLUS jobs 341, 390
BYCLUSTERKEY keyword of ASSOCIATE option 191
BYTABLE keyword of ASSOCIATE option 191

C
C keyword of DSNUEXIT option 241
C language sample DSNUEXIT user exit 814
CA-ACF2 security product 67
cache for populating document ID column 114, 255, 699
canceling or terminating a job 396
canceling threads
authorizations required 69
FORCE option 248, 693
FORCE_AT option 693
FORCE_RPT option 694
reporting 249, 352, 694
specifying when to cancel 248, 693
CARD exception 140
cardinality
changes, when to run 262
how REORG PLUS determines 601
telling REORG PLUS how to determine 199
CATALOG MANAGER
dynamic data set allocation considerations 91, 296
multiple reorganizations 382
catalog, DB2
reorganizing objects in 111
switching I and J names 106
updating data set allocation 835
updating table statistics 261
using for conditional reorganization 136
CA-Top Secret security product 67
CBUFFS option 678
CCSID, specified at the column level 73, 736
CHANGE MANAGER
dynamic data set allocation considerations 91, 296
multiple reorganizations 382
CHANGE option of SHRLEVEL option 173
changes to product 24
character constants, string rules 291
character data, translation of 281
CHECK pending (CHKP) status
limitations on setting 151
with DELETE 285
with SELECT 283
with UPDATE 287
checks performed on work files 96
cleaning up unused data sets 566
clone objects 171, 678
CLONE option 171, 678
COBOL II option
sample DSNUEXIT user exit 805
specifying user exit language 240
COBOL2 keyword of DSNUEXIT option 240
codes, return
for DSNUEXIT user exit 790
for REXX user exits 832
specifying with ON FAILURE 224
specifying with ON MESSAGE 223, 226
specifying with TIMEOUT 238, 731
column-level security 153
columns
defined with CCSID specification 73, 736
inline LOB 73, 701
limit key number 181
non-key in indexes 28, 701
TIMESTAMP WITH TIME ZONE 281, 734
timestamp, defined with precision 733
XML 739
comma as separator 291
cardinal constant, data translation 281
cardinal options 170
See also keywords
cardinal syntax
alphabetical listing of command options 156
diagrams 160
option descriptions 170
rules 155
common components 61
Common Statistics component 61, 66, 258
common utility tables 110
comparison operators 290
components that REORG PLUS uses 61
See also the individual components
compressed indexes 32, 148, 393
compression
BMCDICT table 769
dictionary 145, 206, 703
how REORG PLUS uses and performs 145
indexes 148, 393
KEEPDICTIONARY option 206, 703
phases that perform 52
restart considerations 393
restrictions with partition rebalancing 146
SORTWK data set 390
concatenation rules
for DSN pattern 319
for Spildsn pattern 334
concurrency issues 82, 646
concurrent sort processes 581
CONDEXEC option
cmd 230
considerations 136
installation 679
interaction between command and installation options 136
condition, specifying for WHERE clause 288
conditional reorganization
considerations 136
limit command options 231, 234
limit installation options 704, 713
REPORTONLY option 236
specifying 679
using the DASD MANAGER PLUS exceptions table 138
using the DB2 catalog 136
console, MVS, using with SHRLEVEL CHANGE 644
constants
description and use 286, 290
floating point 292
rules for 291
specifying for a comparison 290
translation of 281
with UPDATE 286
with WHERE 290
CONTINUE keyword
of LONGLOG option 329, 708
of MGEXTENT option 712
of SHORTMEMORY option 196, 721
of UNLOAD option 204, 735
CONTINUE UTILITY keyword of ON MESSAGE option 222
CONTROL authority 70
control card data sets, LOAD 76
control interval (CI) size, variable 148
controlling BMCHIST updates 849
conventions, documentation 22
copies, image
See also copy data sets
considerations for large number of partitions 598
difference between inline and standard 360
FlashCopy 153
incremental 267
inline. See inline image copies
large number of partitions 598
memory considerations 598
partition-by-growth table spaces 358
partition-level 357, 682
performance considerations 108
registering 267
REORG PLUS 579, 590
specifying full or partial 269
specifying inline option 265
specifying option to enable 264
subset of partitions in single copy 682
copies, incremental 267, 590
copies, snapshot 567
copy data sets
See also incremental copy data sets
allocating 353
buffer usage, controlling 577, 678
considerations when not creating 150
DD DUMMY 97
default DD names example 408
DISP=MOD considerations 97
dynamic allocation. See copy data sets, dynamic allocation of
for partition-by-growth table spaces 358
GDG names 92
incremental 267
inline. See inline image copies
multitasking 586
overriding default ddnames, local 271
overriding default ddnames, remote 273
performance, improving 360
registering 267, 354
registration failure during SHRLEVEL CHANGE 656
specification and usage 352
subset of partitions in single data set 682
temporary 97
copy data sets, dynamic allocation of
See also dynamic allocation of data sets
assigning partitions 269
considerations 355
DDTYPE option 295, 349
expiration date 322
GDG names 92
limiting GDG versions 323
naming 92, 93
retention period 322
COPY option
COPY NO considerations 150
description 264
temporary 269
COPY pending status
temporary 269
COPY registration
BMCXCOPY table 781
REGISTER option 267
COPYDDN option
cmd 271
temporary 269
temporary 417, 492
installation 680
COPYLVL option
cmd 269
temporary 417, 442
installation 680
COPYMAX option 681
COPYSUBSET option 682
CPUs, number of 598
CPYRFAIL option
  actions during failure 656
  description 682
  with DDLIN data set 131
creating additional data sets as needed (REDEFINE option) 566
CURRENT DATE keyword
  of predicate 292
  of UPDATE SET option 286
current date variable
  with DSNPAT option 316, 747
  with SPILDSNP option 726
  with SPIILLDSNPAT option 332
current Julian date variable
  with DSNPAT option 316, 747
  with SPILDSNP option 726
  with SPIILLDSNPAT option 332
CURRENT TIME keyword of UPDATE SET option 286
current time variable
  with DSNPAT option 317, 748
  with SPILDSNP option 726
  with SPIILLDSNPAT option 332
current time variable, short form
  with DSNPAT option 317, 748
  with SPILDSNP option 726
  with SPIILLDSNPAT option 332
CURRENT TIMESTAMP keyword
  of predicate 292
  of UPDATE SET option 286
customer support 3

D

D2U (DB2 Utilities Common Code) 61, 66
DASD MANAGER PLUS
  BMCSTATS keyword of ANALYZE option 202
  BMCSTATS option 259
  BMCTRIG utility 138
  conditional reorganizations 138
  dynamic data set allocation considerations 91, 296
  exceptions table 138
  multiple reorganizations 382
  statistics retrieved by REORG PLUS 572
  updating statistics 259
DASD vendor considerations for Instant Snapshot 125
data availability
  during reorganization 98
  maximizing 599
  performance information 591
data capture flag
  abnormal termination 657
  REORG PLUS resetting 621
  REORG PLUS setting 619
  setting with DATACAP option 683
data classes, SMS 325, 745
data integrity, checks for 98
DATA PACKER for DB2, AMEND function 214
data set allocation. See allocating data sets
data set name pattern
  concatenation 319
  specifying 315, 745
  user exit for 789
data set name pattern user exit
  assembler example 791
  C example 814
  COBOL II example 805
description 789
  LE_C example 823
  LE_COBOL example 805
  specifying 688
  user-defined variables 791
data set redefine user exit
  considerations for nonpartitioned objects 835
description 834
  example 844, 852
  invoking the user exit 837
  ordering STOGROUP volumes 836
  resizing data sets 835
  selectively redefining 835
  specifying 690
  variables passed 838
data set rules 591
data set type, specifying for dynamic allocation 309, 750
data sets, DB2
  changing size of 835
data sets, REORG PLUS
  See also individual data set types and data sets, DB2
data sets, REORG PLUS
  allocating. See allocating data sets
  ASUSRPT 258, 351
  attributes, checks for 96
  BMCFORCE 352
calculation sizes 386
data set rules 591
data set type, specifying for dynamic allocation 309, 750
data sets, DB2
  changing size of 835
data sets, REORG PLUS
  See also individual data set types and data sets, DB2
data sets, REORG PLUS
  allocating. See allocating data sets
  ASUSRPT 258, 351
  attributes, checks for 96
  BMCFORCE 352
calculation sizes 386
data set rules 591
data set type, specifying for dynamic allocation 309, 750
  attributes, checks for 96
  BMCFORCE 352
calculation sizes 386
data set rules 591
  data capture flag
  abnormal termination 657
  REORG PLUS resetting 621
  REORG PLUS setting 619
  setting with DATACAP option 683
data classes, SMS 325, 745
data integrity, checks for 98
  DATA PACKER for DB2, AMEND function 214
data set allocation. See allocating data sets
data set name pattern
  concatenation 319
  specifying 315, 745
  user exit for 789
data set name pattern user exit
  assembler example 791
  C example 814
  COBOL II example 805
description 789
  LE_C example 823
  LE_COBOL example 805
  specifying 688
  user-defined variables 791
data set redefine user exit
  considerations for nonpartitioned objects 835
description 834
  example 844, 852
  invoking the user exit 837
  ordering STOGROUP volumes 836
  resizing data sets 835
  selectively redefining 835
  specifying 690
  variables passed 838
  data set rules 591
data set type, specifying for dynamic allocation 309, 750
data sets, REORG PLUS (continued)
dynamic allocation 744, 753, 760
   See also dynamic allocation of data sets
EAV 40
extended sequential
   allocating 310, 750
   SORTWK restriction 369
   SYSREC restriction 382
FASTSWITCH process 105
large format 309, 750
LOAD control card 76, 294, 378
log spill records 707, 726
pattern for names 567
prefixes, specifying 93
rename and delete process for staging data sets 103, 105
resizing 842
specifying DD statements 348, 386
staging 101
striped 369
SYSPUNCH 76, 294, 378
data sharing
group name variable for DSNPAT option 316
group name variable for SPIILDSNPAT option 332
restart considerations 394
using XBM 253
when DB2 versions are not the same 344
when using table space compression 344
data space
   allocating for log records 337
   allocating for RID maps 336
   estimating size for log records 662
   restricting size for RID maps 660
data, deleting during reorganization 284
data base name variable
   with DSNPAT option 316, 747
   with SPIILDSNP option 726
   with SPIILDSNPAT option 332
Database Performance for DB2 solution 45, 202, 230
database status 84
database, mapping 340
database, specifying table space to be reorganized 170
DATACAP option 683
DATACLAS option 325, 745
DATACLAS value in DSNZPARMs 212
data-sorting indexes, definition 190
date rules for constants 291
DATE SQL function 148, 149
&DATE variable
   with DSNPAT option 316, 747
   with SPIILDSNP option 726
   with SPIILDSNPAT option 332
date variable, Julian
   with DSNPAT option 316, 747
   with SPIILDSNP option 726
   with SPIILDSNPAT option 332
&DATEJ variable
   with DSNPAT option 316, 747
   with SPIILDSNP option 726
   with SPIILDSNPAT option 332
DAY SQL function 148, 149
&DB variable
   with DSNPAT option 316, 747
   with SPIILDSNP option 726
   with SPIILDSNPAT option 332
DB2
   changing the size of VSAM data sets 835
   DSNUTILB program 72
   REORG utility 66
   REORG utility compared to REORG PLUS 48
   subsystem identifier (SSID) parameter of EXEC statement 344
   system setup 65
   TEMPLATE control statements 78
   user exits 151, 584
   versions supported 65
   versions, when different on data sharing group 344
   DB2 authority to execute REORG PLUS 67
DB2 catalog
   reorganizing objects in 111
   switching I and J names 106
   updating data set allocation 835
   updating table statistics 261
   using for conditional reorganizations 136
DB2 data sets. See data sets, DB2
DB2 directory 112
DB2 keyword of ARCHFORMAT option 186, 676
DB2 Solution Common Code (SCC). See SCC
DB2 Utilities Common Code. See D2U
DBCNTL installation data set 404
DBET error status 84
DBETE status 84
DD DUMMY 97, 258, 385
DD statements
$ORTPARM 386
ASUSRPRNT 351
BMCFORCE 352
copy data sets 655
DDLIN 360
DDLOUT 365
description 348, 386
DUMMY 97, 385
for REXX exits 837, 850, 859
SORTWK 366
SYSARC 370
SYSEXEC 373
SYSIDCIN 374
DD statements (continued)
SYIN 376
SYSPRINT2 377
SYSPRINT 376
SYSREC 379
SYSTRM 382
SYSTSPRT 382
SYSUT1 382
using with dynamic allocation 297, 752
UTPRINT 386
DDL
pending changes 73, 715
to rebalance partitions 131
DDLDDN option 187, 683
DDLIN data set
considerations 128
contents 131
description 59
example 529
failure during UTILTERM 400, 674
other requirements 131
sample statements and messages 363, 364
specification and usage 360
supported data types in limit keys 129
unsupported data types in limit keys 130
DDLOUT data set 365
&DDNAME variable 316, 747
ddname variable 316, 747
ddnames, overriding
archive data set 185, 675
DDLIN data set 187, 683
IDCAMS input data set 214
incremental local copy data sets 275, 696
incremental remote copy data sets 277, 716
index work data set 183, 737
local copy data sets 271, 680
output data set 182, 734
remote copy data sets 273, 716
SYSARC 185, 675
SYSREC 182, 734
SYST 183, 737
DDTYPE option
command 294
examples 442, 507
installation 741
deactivating dynamic allocation 296, 742
DEADLINE option
command 327
example 529
installation 683
delete and rename process for staging data sets
backing out after ALTER failure 674
backing out after failure 402
canceling during 398, 658
description 103
failure during 400
falling back from FASTSWITCH process 220, 695
object status following 89, 90
threshold for 221, 566, 695
DELETE keyword
of ORIGDISP option 714
of ORIGINALDISP option 217
of TAPEDISP option 729
DELETE option
authorization information 67
description 284
referential integrity issues 151
SHRLEVEL CHANGE considerations 649
DELETEFILES option
changing the default value 684
data sets deleted 215
description 215
restart considerations 395
when processing performed 215, 684
with dynamic allocation 94
with ON FAILURE TERMINATE 216, 225
with TIMEOUT TERM 216
deleting data during reorganization 284
deleting dynamically allocated work files 94, 215, 684
DELFILES option 684
delimited tokens in command syntax 156
DESCCDE option 685
device type, specifying 187
diagrams, syntax 160
dictionaries, compression
BMCDICT table 769
for partition-by-growth table spaces 147
KEEPDICTIONARY option 206, 703
restrictions with partition rebalancing 146
using existing 147
using with table space compression 145
directory, DB2 112
DISABLED keyword of ZIIP option 253, 739
disabling zIIP processing 253, 739
discard data sets. See SYSARC data sets
discarding rows
  to a discard data set, with DSNUTILB 76, 372
  to an archive data set 370
  with DELETE 284
  with SELECT 282
DISP (data set disposition) 96, 97
DISPLAY option
   description 640
   log apply control 634
   sample output, terse 641
   sample output, verbose 642
DISPLAY privileges 67, 69
displaying drain locks 247, 689
displaying status of BMC utilities 768
document ID column
   cache for 114, 255, 699
   restrictions on 287
document ID indexes 392
drain
   action on timeout 238, 731
   canceling threads for 248, 693
DELETEFILES option, processing 216
displaying locks 247, 689
read claimers, canceling 248, 250, 693
specifying the number of retry attempts 246, 686
specifying the time between retry attempts 246, 686
specifying the timeout value 245, 687
specifying type for SHRLEVEL CHANGE 340, 686
write claimers, canceling 250, 693
DRAIN keyword of LONGLOG option 708
DRAIN option 340
DRAIN_WAIT option 245
DRAINTYP option 686
DRNDELAY option
   description 686
   performance tuning 592
DRNFAIL keyword of DSPLOCKS option 247, 689
DRNRETRY option
   description 686
   performance tuning 592
DRNWAIT option
   description 687
   performance tuning 592
DSECT, assembler parameter block 792
DSEXTENT exception 140
DSN pattern 315, 745
DSN=NULLFILE 97, 385
DSN1COPY, making as part of reorganization 265
DSNDB01 database 112
DSNDB06 database, reorganizing objects in 111
DSNHDECP
   settings for changing decimal points to comma 291
   SSD from 344
DSNPAT option
   command 315
   examples 500, 507
   installation 745
DSNTYPE option 309, 750
DSNUEXIT option 688
DSNUEXIT user exit
   assembler example 791
   C example 814
   COBOL II example 805
   description 789
   LE_C example 823
   LE_COBOL example 805
   specifying 688
   user-defined variables 791
DSNUTILB
   specifying 75
   when invoked 72
DSNUTILB option 237, 689
DSNUTILB processing phase 52
DSNUTILB reorganization
   considerations 72
   data set name pattern, symbolic variables 315, 746
   diagram of phases 58
   dynamic data set allocation 78, 91
   example 506
   how to run 75
   multi-table table spaces 78
   object status 84, 86
   processing phases 58
   recovery 397
   requirements for 75
   restrictions 77
   software requirements 75
   specifying 237, 689
   statistics 143, 262
   symbolic variables for data set name pattern 315, 746
   threads used 71
   user-defined variables 316
DSNUTILS stored procedure 36
DSNZPARMs
   SMS DATACLAS value 212
   utility timeout value 246, 687
DSPLOCKS option
   command 247
   installation 689
   performance tuning 592
DSRSEXIT option 690
DSRSEXIT user exit
   considerations for nonpartitioned objects 835
   description 834
   example 844, 852
   invoking 837
   ordering STOGROUP volumes 836
   resizing data sets 835
   selectively redefining data sets 835
   specifying 690
   variables passed 838
DSSIZE, estimating copy data set size 355, 654
dump, system
   excluding abend codes 690
   generating 720
duration, labeled, specifying as part of a condition 292
DYNAALLOC installation option 188, 725, 761
dynamic allocation of data sets
   activating and deactivating 91, 296, 742
   allocation method 299, 744
dynamic allocation of data sets (continued)
ANALYZE option 199, 387
ARCHDDN option 185
archive (SYSARC) 185, 187, 370
BMCSORT 188
CATALOG MANAGER considerations 91, 296
CHANGE MANAGER considerations 91, 296
command options 293
copy, full image 59, 295, 349
COPYDDN considerations 272
DASD MANAGER PLUS considerations 91, 296
data set type, specifying 309, 750
DD statements with 297, 752
DD type specification 294, 741
DDLDDN considerations 187, 683
deleting 94, 215, 684
description 90
discard data sets (SYSARC) for DSNUTILB jobs 370
DSNUTILB reorganization 78
enabling 91
example 416
expiration date 322
extended sequential 310, 750
GDG names 92
ICDDN considerations 276
incremental image copy 59, 295, 349
index work files (SYSUT) 184, 294, 382
installation options 740
large format 309, 750
limiting size 307, 753
LOAD control card data sets (SYSPUNCH) for DSNUTILB jobs 378
local full copy data sets 272
local incremental copy data sets 276
merging with DD statements 297, 752
method 299, 744
name pattern user exit 788
naming
using DSNPAT 92, 315, 745
using DSNUEXIT 791
options 293
pattern for data set names 315, 745, 791
percentages of allocation size 311, 755
performance 579
prefix (data set) considerations 93, 182, 184
RECOVERYDDN considerations 274
RECOVERYICDDN considerations 278
remote full copy data sets 274
remote incremental copy data sets 278
restarting a job 95, 391, 396
retention period 322
SMS classes 324, 745
sort work files (SORTWK) 294, 350, 366
SPACE option 312
tape devices, number of 293
terminating job 397
threshold for secondary unit/class 757
threshold for secondary values 312, 757
unit count 302
unit names 302, 758
UNLDDN considerations 182
unload work files (SYSREC) 294, 350, 379
user exits 240, 688, 690
using 91
volume count 304
WORKDDN considerations 184
worklist environment 296
dynamic allocation of tape units 95
dynamic allocation options 293
dynamically altering a SHRLEVEL CHANGE reorganization 640
DYNAMNBR parameter 109, 342

E
EAV data sets 40
EDITPROCs
and UXSTATE value 151
DB2 REORG comparison 49
invoking with an AMENDed table 214
libraries that contain 348
row format conversion 150
rows compressed by 146
ENABLE ARCHIVE, tables defined with 279
ENABLED keyword of ZIIP option 252, 739
enabling dynamic allocation 91
enabling zIIP processing 153, 252, 739
END keyword of LONGNAMETRUNC option 178, 708
ENDING AT clause of SQL ALTER statement 33
enqueues, specifying 705
ERP environments, REORG PLUS recommendations 566, 591
estimating data set sizes, with ANALYZE 570
evaluation of predicate 288
example REORG PLUS jobs 403
exceptions, generating a reorganization 138
EXCLDUMP option 690
EXEC statement
description 342
utility parameters 343
executing REORG PLUS jobs 390
execution phases of REORG PLUS
compared to DB2 48
DSNUTILB 52
primary functions 51
SHRLEVEL CHANGE 618, 623, 628
tuning considerations 600
exits, DB2 151
exits, user
accessing the source 788
authorizations for 68, 837
controlling BMCHIST update 849
controlling statistics updates 849
exits, user (continued)
creating mapping objects 857
data set name pattern
   assembler example 791
   C example 814
   COBOL II example 805
description 789
   invoking 688
   LE COBOL example 805
   LE_C example 823
data set redefine
   description 834
   example 844
   invoking 690
dropping mapping objects 857
DSNUEXIT
   assembler example 791
   C example 814
   COBOL II example 805
description 789
   invoking 688
   LE_C example 823
   LE_COBOL example 805
DSRSEXIT
   description 834
   example 844
   invoking 690
mapping table
   description 857
   example 861
   invoking 243, 709
MAPTEXIT
   description 857
   example 861
   invoking 243, 709
REXX
   requirements 858
   restrictions 832
   setting up 837, 850, 859
sample library 788
TERMEXIT
   description 849
   example 882
   invoking 730
variables
   for DSN patterns 316
   for SPILDSN patterns 332
EXPDT option 322, 751
expiration date for certain data sets 322, 751
expression, indexes created on 73, 74, 148, 702
extended address volume (EAV) data sets 40
EXTENDED BUFFER MANAGER. See XBM
extended sequential data sets
   allocating 310, 750
   SORTWK restriction 369
   SYSREC restriction 382
   extending data sets 213, 712
EXTENTS exception 140
extents, secondary 712
EXTPEND keyword of DSNTYPE option 310, 750
EXTREQ keyword of DSNTYPE option 310, 750

F
FAIL keyword
   of FILECHK option 692
   of SHORTMEMORY option 196, 721
failure
   cannot restart in UTILTERM 400
   during ALTER statement processing 402
   during FASTSWITCH processing 107, 220, 695
   during index reorganization 399
   during rename processing 402
   handling for Instant Snapshot 123
   inadequate space 395
   restarting from 391
   specifying how to handle 224
   while setting restrictive statuses 402
falling back from FASTSWITCH to rename process 220,
   695
FARIND exception 141
FAROFF exception 141
FASTSWITCH option 218, 691
FASTSWITCH process
   backing out after failure in UTILTERM 402, 674
   command option 218, 691
   completing manually after failure in UTILTERM 400
   failure during 107
   falling back to rename process 220, 695
   installation option 691
   naming conventions 105
   object status following 106
   restarting in UTILTERM 398
   threshold for 221, 566, 695
   with ORIGINALDISP 217, 714
features of REORG PLUS 46
FIELDPROCs
   in WHERE clause 154, 289
   libraries that contain 348
   statistics gathered 260, 261
FILECHK option 97, 692
five-byte partition variable 317, 748
fixes applied, generating a report for 347
FlashCopy image copies 153
floating point
   columns unsupported in WHERE clauses 154
   decimal 287, 292, 361
   rules for constants 292
FORCE option 248, 693
FORCE_AT option
   command, of FORCE option 248, 250
   installation 693
FORCE_DELAY option 249, 250
FORCE_RPT option
   command, of FORCE option 249, 250
   installation 694
FORCEROLLUP Common Statistics component option 261
FORMAT BMC option of LOADPLUS 186, 675
   format of archive data sets 185, 675
   format of UNLOAD data 48, 186, 676
   formats, syntax diagrams 23
FREE keyword of IFALLOC option 298, 753
   free space in nonleaf index pages, controlling 210
   FSFALLBACK option 220, 695
   FSTHRESHOLD option 221, 695
FULL keyword of COPYLVL option 270, 681
   functions, SQL, indexes created on expressions that contain 148

G
GDG. See generation data group
GDGEMPTY option 751
GDGLIMIT option 323, 752
GDGSCRATCH option 752
generation data group (GDG)
   creating data set name 92
   limiting number of generations 323, 752
   removing from the VTOC 752
   uncataloging 751
granting privileges to PUBLIC 67
   graphic strings rules for constants 291
GRECP (group RECOVER pending) status 85, 87
   group name variable
      with DSNPAT option 316, 747
      with SPIILDSNP option 726
      with SPIILDSNPAT option 332
   group RECOVER pending status (GRECP) 85, 87
   &GRPNM variable
      with DSNPAT option 316, 747
      with SPIILDSNP option 726
      with SPIILDSNPAT option 332

H
hardware compression
   See compression
HASHAX option 696
hash-organized tables 73, 256, 696
   hexadecimal strings rules for constants 291
   hierarchy of log apply control options 636
   high-used RBA
      ANALYZE HURBA option 571
      average row length, determining 571
      cardinality estimate 202, 571
      recovering objects 47
REDEFINE NO option 212
   resetting 212
HISTORY Common Statistics component option 261
   history table. See BMCHIST table
HURBA keyword of ANALYZE option 202
I
I/O
   performance for partitioned objects 575
   performance improvement 384, 576, 588
   performance messages 592
   processing, to maximize 584, 589
   queueing, avoiding 589
ICDDN option 275, 696
ICF (Integrated Catalog Facility) 70
ICTYPE option
   command 267
   examples 542, 553
   installation 697
   performance 590
ID
   subsystem variable
      with DSNPAT option 317, 748
      with SPIILDSNP option 726
      with SPIILDSNPAT option 332
   user variable
      with DSNPAT option 318, 748
      with SPIILDSNP option 727
      with SPIILDSNPAT option 332
   utility (parameter of EXEC statement) 344
   utility prefix variable
      with DSNPAT option 318, 749
      with SPIILDSNP option 727
      with SPIILDSNPAT option 332
   utility suffix variable
      with DSNPAT option 318, 749
      with SPIILDSNP option 727
      with SPIILDSNPAT option 333
   utility variable
      with DSNPAT option 318, 748
      with SPIILDSNP option 727
      with SPIILDSNPAT option 332
   XBM subsystem 253, 738
IDCACHE option 255, 699
IDCAMS, specifying commands for data set redefinition 374, 407, 631
IDCDDN option 214
   identity column restrictions 287
IFALLOC option
   command 297
   example 507
   installation 752
   image copies
      considerations for large number of partitions 598
      difference between inline and standard 360
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

image copies (continued)
FlashCopy 153
incremental 267
inline. See inline image copies
large number of partitions 598
memory considerations 598
partition-level 357, 682
performance considerations 108
registering 267
REORG PLUS 590
specifying full or partial 269
specifying inline option 265
specifying option to enable 264
subset of partitions in single copy 682
image copy data sets
See also incremental copy data sets
allocating 353
buffer usage, controlling 577, 678
considerations when not creating 150
DD DUMMY 97
default DD names example 408
DISP=MOD considerations 97
dynamic allocation. See image copy data sets,
dynamic allocation of
for partition-by-growth table spaces 358
GDG names 92
incremental 267
inline. See inline image copies
multitasking 586
overriding default ddnames, local 271
overriding default ddnames, remote 273
performance, improving 360
registering 267, 354
registration failure during SHRLEVEL CHANGE 656
specification and usage 352
subset of partitions in single data set 682
temporary 97
image copy data sets, dynamic allocation of
assigning partitions 269
considerations 355
DDTYPE option 295, 741
expiration date 322
GDG names 92
limiting GDG versions 323
naming 92, 93
retention period 322
implicit time zone 281
IMPLICIT_TIMEZONE DSNHDECP value 281
IMPLICIT_TZ option 281
inadequate space failure 395
incremental copy data sets
buffer usage, controlling 578
corresponding image copy types 654
ddnames required 655
improving performance 656
overriding default ddnames 275, 277
performance, improving 656
registering 653
specifying and using 653
specifying with ICTYPE 267
incremental image copies 267, 590
INCREMENTAL keyword of ICTYPE option 268, 698
INDEX option 171
index reorganization
See also indexes
ANALYZE option 571
element 463
failure during 399
falling back from FASTSWITCH to rename process 220
INDEX option 171
performing conditionally 136
processing phases
RELOAD 608
REORG phase 612
single-phase reorganization 57
two-phase reorganization 55
two-phase reorganization with SHRLEVEL CHANGE 625
index space name variable
with DSNPAT option 317, 748
with SPIILDSNP option 726
with SPIILDSNAT option 332
index work data sets (SYSUT1)
allocating 351
controlling buffer usage 576
DD DUMMY 97, 385
deleting automatically after job 215
description 60
DSN=NULLFILE 97, 385
dynamic allocation
DDTYPE option 294, 349
description 90
syntax 293
XML table space reorganization 201
multiple 588
overriding default ddname 737
performance considerations 588
tuning considerations 588
XML table space reorganization 115
indexes
catalog, reorganizing 111
compressed 32, 148, 393
indexes (continued)
created on expression 73, 74, 148, 702
data-sorting definition 190
document ID 392
LOB 118
multiple 591
node ID 201
non-key columns in 28, 701
nonpartitioned 121, 122
null keys 74, 736
on a clone table 171, 678
on hash-organized tables 73, 696
on inline LOB columns 73, 702
random keys 74, 506, 702
secondary, REDEFINE 211
SHRLEVEL CHANGE access to objects 90
using Instant Snapshot 122
versioned 32, 111
with pending DDL changes 74
XML 68, 114
INDREFLIMIT option 233
INDREFLM option 699
initial status of objects 84
inline image copies
  performance considerations 590
  restart considerations 396
  specifying 265, 700
  with ICTYPE UPDATE 268, 698
inline LOB columns 73, 149, 701
INLINE option of COPY YES option
description 265
performance information 590
INLINECP option
description 700
performance information 590
INLOB option 701
input data set 60, 376
installation options
  $ARUDYNA macro 740
  $ARUOPTS macro 668
  basic REORG PLUS 668
dynamic data set allocation 740
  list of 668, 740
  multiple modules 347, 668
Installation System 667
instance number of base table space 678
Instant Snapshot
  DASD vendor considerations 125
  handling failures 123
  impact of REDEFINE option 123
  specifying 251, 723
  with multiple data set objects 111
  with nonpartitioned indexes 122
  integer rules for constants 291
  integrity checks performed on work files 96
interface
  with AR/CTL 651
  with XBM or SUF 639
invoking DSNUTILB
  See also DSNUTILB reorganization
considerations 72
  DSNUTILB option 237, 689
  example 506
invoking REORG PLUS 391
IRLMWAIT, used with drain timeout 246, 687
IXINCLCOL option 701
IXONEX option 702
IXRANDOM option 506, 702

J
JCL
  example REORG PLUS jobs 403
  in DBCNTL installation data set 404
&JDATE variable
  with DSNPAT option 316, 747
  with SPILDSNP option 726
  with SPILLDSNPAT option 332
JES3 limitation 386
JOB name variable
  with DSNPAT option 316, 747
  with SPILDSNP option 726
  with SPILLDSNPAT option 332
JOB statement 342
&JOBNAME variable
  with DSNPAT option 316, 747
  with SPILDSNP option 726
  with SPILLDSNPAT option 332
jobs, REORG PLUS
  building 341, 390
  example 403
  executing 390
  for installation options 667
  restarting and recovering 391
  restarting during UTILTERM 398, 658
  terminating or canceling 396, 658

K
KEEPDICTIONARY option
  command 206
  how REORG PLUS uses 145
  installation 703
  performance considerations 566
key compression 205
KEYCARD Common Statistics component option 261
keys
  derived from an expression 73, 148, 702
  limit 129, 131
null 74, 736
random 74, 506, 702

keys (continued)

ABEND keyword of TIMEOUT option 239
ACTIVE option 296, 742
ALL keyword
  of DRAIN option 340
  of DRAINTYP option 686
  of FORCE option 250, 693
  of REGISTER option 267
ALLOC option 299, 744
ALTRFAIL option 674
AMENDED option 214
ANALMAX option 674
ANALYZE option 199
ANY keyword of ALLOC option 299, 744
ARC option 674
ARCHDDN option 185, 675
ARCHFORMAT option 185, 675
ARCHIVE keyword of DDTYPE option 294, 742
ARCROWS option 229
ASM keyword of DSNUEXIT option 240
ASSOCIATE option of ORDER YES option 191
AUTO keyword
  of ICTYPE option 268
  of SIXSNAP option 251
  of VOLCNT option 304, 760
AUTOESTSPACE option 256
AUX option 256
AUXREORG option 676
AVAILPAGEPCT option 193, 676
AVGVLSP option 306, 744
BASIC keyword of DSNTYPE option 309, 750
BEGINNING keyword of LONGNAMETRUNC option 178, 708
BILDMAX option 677
BMCHIST option 208, 678
BMCHIST option 208, 678
BMISTAT keywords of ANALYZE option 202
BMCSSTATS option 259
BYCLUSTERKEY keyword of ASSOCIATE option 191
BYTE option of ASSOCIATE option 191
C keyword of DSNUEXIT option 241
CBUFFS option 678
CHANGE option of SHRLEVEL option 173
CLONE option 171, 678
COBOL2 keyword of DSNUEXIT option 240
CONDEXEC option 230, 679
CONTINUE keyword
  of LONGLOG option 329, 708
  of MGEXTENT option 712
  of SHORTMEMORY option 196, 721
  of UNLOAD option 204, 735
CONTINUE UTILITY keyword of ON MESSAGE option 222
COPY option 264
COPYDDN option 271, 680
COPYLVL option 269, 680
COPYMAX option 681
COPYSUBSET option 682
CPRFAIL option 682
CURRENT DATE keyword
  of predicate 292
  of UPDATE set option 286
CURRENT TIME keyword of UPDATE SET option 286
CURRENT TIME keyword of UPDATE SET option 286
CURRENT TIMESTAMP keyword
  of predicate 292
  of UPDATE SET option 286
DATACAP option 683
DATACLAS option 325, 745
DB2 keyword of ARCHFORMAT option 186, 676
DDLDDN option 187
DDTYPE option 294, 741
DEADLINE option 327, 683
DEFER keyword of MAXRO option 327, 710
DELAY option 330, 684
DELETE keyword
  of ORIGDISP option 714
  of ORIGINALDISP option 217
  of TAPEDISP option 729
DELETE option 284
DELETEFILES option 215
DELET.FILES option 684
DESCCDE option 685
description of
  command options 170
  installation options 673
DISABLED keyword of ZIIP option 253, 739
DISPL option 640
DRAIN keyword of LONGLOG option 708
DRAIN option 340
DRAIN_WAIT option 245
DRAINTYP option 686
DRNDELAY option 686
DRNFAIL keyword of DSPLOCKS option 247, 689
DRNRETRY option 686
DRNWAIT option 687
DSNPAT option 315, 745
DSNTYPE option 309, 750
DSNUEXIT option 240, 688
DSNUTLB option 237, 689
DSPLOCKS option 247, 689
DSREXIT option 241, 690
ENABLED keyword of ZIIP option 252, 739
END keyword of LONGNAMETRUNC option 178, 708
keywords (continued)
EXCLDUMP option 690
EXPDT option 322, 751
EXTREF keyword of DSNTYPE option 310, 750
EXTREQ keyword of DSNTYPE option 310, 750
FAIL keyword
  of FILECHK option 692
  of SHORTMEMORY option 196, 721
FASTSWITCH option 218, 691
FILECHK option 692
FORCE option 248, 693
FORCE_AT option
  command, of FORCE option 248, 250
  installation 693
FORCE_DELAY option of FORCE option 249, 250
FORCE_RPT option
  command, of FORCE option 249, 250
  installation 694
FREE keyword of IFALLOC option 298, 753
FSFALLBACK option 220, 695
FSTHRESHOLD option 221, 695
FULL keyword of COPYLVL option 270, 681
GDGEMPTY option 751
GDGLIMIT option 323, 752
GDGSCRATCH option 752
HASHAX option 696
HURBA keyword of ANALYZE option 202
ICDDN option 275, 696
ICTYPE option 267, 697
IDCACHE option 255, 699
IDCDDN option 214
IFALLOC option 297, 752
IMPLICIT_TZ option 281
INCREMENTAL keyword of ICTYPE option 268, 698
INDEX option 171
INDREFLIMIT option 233
INDREFLM option 699
INLINE option of COPY YES option 265
INLINECP option 700
INLOB option 701
IXINCLCOL option 701
IXONEX option 702
IXRANDOM option 702
KEEPDICTIONARY option 206, 703
LARGE keyword of DSNTYPE option 309, 750
LAST keyword of PART option 176
LASTRETRY keyword of FORCE_AT option 249, 693
LE_C keyword of DSNEXIT option 241
LE_COBOL keyword of DSNEXIT option 241
LEAFDISTLIMIT option 234
LEAFDSLM option 704
LOB option 704
LOCBFCPY keyword of DDTYPE option 295, 742
LOCBCICPY keyword of DDTYPE option 295, 742
LOCKROW option 705
LOCPFCPY keyword of DDTYPE option 295, 742
LOCPICPY keyword of DDTYPE option 295, 742
LOG NO option 209
LOGFINAL option 338, 706
LOGMEM option 337, 706
LOGSPIL option 707
LOGSPILL option 338
LOGTHRESHLD option 326
LOGTHRH option 707
LONGLOG option 329, 708
LONGNAMETRUNC option 178, 708
MAPPINGDATABASE option 340
MAPPINGTABLE option 340
MAPTEXIT option 243, 709
MAXEXTSZ option 307, 753
MAXNEWPARTS option 177, 709
MAXRO option 326, 710
MAXSORTMEMORY option 197, 711
MAXSORTS option 189
MAXTAPE option 293, 711
MGEXTENT option 712
MGMTCLAS option 325, 754
MIDDLE keyword of LONGNAMETRUNC option 178, 708
MINIMAL keyword of ALLOC option 299, 744
MINSORTMEMORY option 197, 712
MSGLEVEL option 712
NLPCFTFREE option 210
NONE keyword
  of DATACLAS option 745
  of DEADLINE option 328, 683
  of DRAIN_WAIT option 245
  of DRNWAIT option 687
  of DSNTYPE option 309, 750
  of DSNEXIT option 240, 688
  of DSPLOCKS option 247, 689
  of DSRSEXIT option 242, 690
  of FORCE option 248, 693
  of INDREFLM option 699
  of LEAFDSLM option 704
  of LOGFINAL option 339, 706
  of MAPTEXIT option 244, 709
  of MGMTCLAS option 754
  of OFFPOSLSM option 713
  of REGISTER option 267
  of SHRLEVEL option 172
  of SPIILLSTORCLAS option 331
  of SPIILLUNIT option 330
  of SPIILSCLS option 727
  of SPIILUNIT option 728
  of STORCLAS option 756
  of TERMEXIT option 243, 730
NOSYSREC option 205
NULL keyword
  of predicate 292
  of UPDATE SET option 286
OFFPOS LIMIT option 231
OFFPOSLSM option 713
ON FAILURE option 224
keywords (continued)
ON MESSAGE option 222
ON n COLUMNS keyword of REBALANCE option 181
ONLY keyword of ANALYZE option 201
OPND2ID option 713
OPTIMIZED keyword of ALLOC option 299, 744
ORDER option 190
ORIGDISP option 714
ORIGINALDISP option 217
PART keyword of COPYLVL option 269, 680
PART option 175
PAUSE keyword
of ANALYZE option 201
of UNLOAD option 205
PENDDDL option 715
PLAN option 715
PREFORMAT option 227, 715
RCVICDDN option 716
RCVRPEND keyword of ALTRFAIL option 674
RCVYDDN option 716
READERS keyword of FORCE option 248, 693
REBALANCE option 179
RECOVERYDDN option 277
RECOVERYICDDN option 277
REDEFINE option 210, 717
REFERENCE keyword of SHRLEVEL option 173
REFERENCE UNLOADONLY keyword of SHRLEVEL option 172
REGISTER keyword of COPY YES option 267
RELOAD keyword of UNLOAD option 204, 735
REMBFCPY keyword of DDTYPE option 295, 742
REMBICPY keyword of DDTYPE option 295, 742
REMPFCPY keyword of DDTYPE option 295, 742
REMPICPY keyword of DDTYPE option 295, 742
RENAME keyword
of ORIGDISP option 714
of ORIGINALDISP option 217
RENUMAX option 717
REORG option 170
REPORT keyword of BMCSTATS option 260
REPORTONLY keyword of FORCE option 250
REPORTONLY option 236
RESET keyword of MGEXTENT option 712
RETCODE keyword
of ON FAILURE TERMINATE UTILITY option 226
of ON MESSAGE CONTINUE UTILITY option 223, 226
RETPD option 322, 754
RETRY keyword
of DSPLOCKS option 247, 689
of FORCE_AT option 249, 693
RETRY option 246
RETRY_DELAY option 246
REUSE option 213

REXX keyword
of DSRSEXIT option 242, 690
of MAPTEXIT option 244, 709
of TERMEXIT option 243, 730
RIDMAPMEM option 336
RIDMDSZ option 718
RIDMMAXD option 718
RMAPMEM option 718
RORGMAX option 719
ROUTCDE option 720
SAMPLE keyword of ANALYZE option 201
SCAN keyword of ANALYZE option 201
SCPYMAX option 720
SDUMP option 720
SELECT option 282
SHORTMEMORY option 195, 721
SHRLEVEL option 172
SIXSNAP option 251, 723
SIZEPCT option 311, 755
SMAX option 723
SMCONE option 724
SMS option 300, 755
SMSUNIT option 301, 756
SORTDATA option 190
SORTDEVT option 187, 724
SORTKEYS option 174
SORTNUM option 188, 725
SORTWORK keyword of DDTYPE option 294, 742
SPACE option 312
SPIILDSNP option 726
SPIILDSNPAT option 331
SPIILSTORCLAS option 331
SPIILLUNIT option 330
SPIILSCLS option 727
SPIILUNIT option 728
SQL keyword
of DRAIN_WAIT option 246
of DRNWAIT option 687
SQLDELAY option 728
SQLRETRY option 728
STAGEDSN option 728
STANDARD keyword of NLPCTFREE option 210
START keyword of FORCE_AT option 249, 693
STOP UTILITY keyword
of ON FAILURE option 226
of ON MESSAGE option 222
STOP@CMT option 729
STOPDELAY option 729
STORRETRY option 729
STORCLAS option 325, 756
SUFSTART keyword of WTOMSG option 254
SYNC option 208
SYSPUNCH keyword of WTOMSG option 254
SYNC option 208
SYSPUNCH keyword of WTOMSG option 254
TABLESPACE option 170
TAPE_DISP option 729
TASKMAX option 730

Index 887
keywords (continued)
TEMPRALDATA option 730
TERM keyword
   of ALTRFAIL option 674
   of CPYRFAIL option 682
   of LONGLOG option 329, 708
   of TIMEOUT option 239, 731
TERMEXIT option 242, 730
TERMINATE UTILITY keyword of ON FAILURE option 225
THRESHLD option 312, 757
TIMEOUT option 238, 731
TOTALPAGEPCT option 194, 732
TSPREC option 733
TSSAMPLEPCT option 263, 733
TSTZ option 734
UBUFFS option 734
UNIT option 302, 758
UNITCNT option 302, 759
UNLDDN option 182, 734
UNLDMAX option 735
UNLOAD keyword of DDTYPE option 294, 742
UNLOAD option 204, 735
UNLOADONLY keyword of SHRLEVEL REFERENCE option 172
UPDATE keyword of ICTYPE option 268, 698
UPDATE option 286
UPATEDB2STATS option 261
USE keyword of IFALLOC option 297, 752
UTIL keyword
   of DRAIN_WAIT option 246
   of DRNWAIT option 687
UTILB_COLCCSID option 736
UTILB_NULLIX option 736
UTSMEM option 737
UXSTATE option 737
VERBOSE keyword of DISPLAY option 640
VOLCNT option 304, 760
WARN keyword of FILECHK option 692
WBUFSS option 737
WHERE option 288
WITH TIME ZONE keyword
   of CURRENT_TIMESTAMP keyword of predicate
      labeled duration, specifying as part of a condition 292
   of UPDATE SET CURRENT_TIMESTAMP option 286
WORK keyword of DDTYPE option 294, 742
WORKDDN option 183, 737
WORKUNIT option 738
WRITERS keyword
   of DRAIN option 340
   of DRAINTYP option 686
WTOMSG option 254
XBIMID option 253, 738
XML option 739
ZIIP option 252, 739

L
labeled duration, specifying as part of a condition 292
large format sequential data sets 309, 750
LARGE keyword of DSNTYPE option 309, 750
large number of partitions, considerations for 108
LAST keyword of PART option 176
LASTTRY keyword of FORCE_AT option 249, 693
LE C language sample DSNUEXIT user exit 823
LE COBOL language sample DSNUEXIT user exit 805
LE_C keyword of DSNUEXIT option 241
LE_COBOL keyword of DSNUEXIT option 241
LEAFDIST exception 142
LEAFDISTLIMIT option 234
LEAFDSLM option 704
LEAFFOFF exception 142
LEAFTOFF exception 142
length parameter, in UPER or LOWER SQL functions 149
LEVELINC exception 142
LEVELMIN exception 142
LEVELS exception 142
limit keys
   changing 128, 131
   number of columns for 181
   report of 366
   supported data types 129
limit options
   command 231, 234
   determining which values to use 137
   installation 704, 713
LIMIT value of TEMPLATE control statement 313, 758
LOAD control cards. See SYSPUNCH data sets
LOADPLUS utility, loading from SYSPARC data set 186, 371
LOB data
   BMCSYNC table considerations 776
   considerations 119
   how to reorganize 118, 704
   inline 73, 149, 701
   restart considerations 392
LOB indexes 118
LOB option 704
LOB table spaces
   considerations 119
   how to reorganize 118, 704
   in catalog, reorganizing 111
   reorganizing when reorganizing the base table space 72, 256, 676
   restarting 392
locale, in UPER or LOWER SQL functions 149
LOCBPFCPY keyword of DDTYPE option 295, 742
LOCBICPY keyword of DDTYPE option 295, 742
LOCKROW option 705
locks, displaying 247, 689
LOCIPFCPY keyword of DDTYPE option 295, 742
LOCPICPY keyword of DDTYPE option 295, 742
log apply control options
  displaying values 640
  hierarchy diagram 636
  summary 638
  using the MVS console to change 644
  using Utility Monitor to view/change 639
log control task 619
LOG NO option 209
log range table 772
log records
  application 620
  defining 337, 338
  performance considerations 567, 662
  specifying logfinal start time 338
  spill data set for 707
LOG YES option 49
LOGAPPLY phase
  description 620
  primary function 52
  restarting 399
  specifying ON FAILURE 225
LOGFINAL option 338, 706
LOGFINAL phase
  criteria for entering 635
  defining time to end 327
  defining time to start 338
  restarting 399
  specifying drain type 340, 686
logical page list status 85, 87
logical part REBUILD pending status 85, 87
LOGMEM option
  command 337
  installation 706
  performance considerations 662
LOGSPI option 707
LOGSPI option 338
LOGTHRESHLD option 326
LOGTHRSBH option 707
long object names
  syntax rules 156
  truncating in messages 178, 708
longlog condition
  benefits due to REORG PLUS technology 630
  how determined 629
LONGLOG option
  command 329
  example 542
  installation 708
LONGLNAME_TRUNC option 178, 708
LOWER SQL function 148, 149
LPL status 85, 87

M

macros for installation options 668, 740, 741
MAINT parameter of EXEC statement 347
maintaining common utility tables 766
management classes, SMS 325, 754
mapping table
  specifying 75, 340
  user exit for 857
mapping table user exit
description 857
example 861
  restrictions 243, 709
  specifying 243, 709
  variables for 859
MAPPINGDATABASE option 340
MAPPINGTABLE option 340
MAPEXIT option 243, 709
MAPEXIT user exit
description 857
example 861
  restrictions 243, 709
  variables for 859
  variables passed 859
Mass del reorg exception 140
MACCC, specifying in SYSIDCIN 375, 376
MAXEXTSZ option 307, 753
maximum amount of sort memory 711
MAXNEWPARTS option
  command 177
  example 517
  installation 709
MAXPRIME keyword of TEMPLATE control statement 308, 754
MAXRO option
  command 326
  example 529
  installation 710
MAXSORTMEM option 197, 711
MAXSORTS option 189
MAXTAPE option
  command 293
  installation 711
  reaching the limit 95
MAXVALUE keyword of SQL ALTER statement 33
MEMBER CLUSTER, universal table spaces defined as 29, 737
MEMLIMIT system parameter 71, 590
memory
  above the bar 71
  above the line 193, 582, 676
  below the line 582, 583
message level (MSGLEVEL) parameter
  installation option 712
message level (MSGLEVEL) parameter of EXEC statement
EXEC statement 347
messages
  allocation-related from ANALYZE phase 572
  BMCSORT, data set for 60, 386
  changing output displayed 347
messages (continued)
  continuing after receiving certain messages 222
  level displayed 347
  long names in 178, 708
  stopping on particular messages 222
  viewing in worklist environment 377
method of dynamic allocation 299
MGEXTENT option 712
MGMTCLAS option 325, 754
MIDDLE keyword of LONGNAMETRUNC option 178, 708
MINIMAL keyword of ALLOC option 299
minimal method of dynamic allocation 299
minimum amount of sort memory 712
MINSORTMEMORY option 197, 712
MINVALUE keyword of SQL ALTER statement 33
Mods since reorg exception 140
MONITOR2 authority 67
MONTH SQL function 148, 149
MSGLEVEL
  installation option 712
  parameter of EXEC statement 347
multi-data-set objects
  and REDEFINE 102, 213
  and SIXSNAP 123
  considerations for using 111
  DD statements for 355, 655
  STOGROUP-defined 111
multiple data sets
  allocating 584
  index (SYSUT1) 588
  performance tuning 384
  SYSREC data sets 60, 575, 588
  unload data sets 182
  work data sets 183, 576
multiple indexes 591
multiple installation options modules 347, 668
multiple partitions
  specifying 175
  subset of partitions in single image copy 682
multiple reorganizations, considerations 48, 382
multi-table table spaces
  ordering rows within 191
  with DSNUTILB reorganization 78
multitasking
  enabling 584
  I/O bound phases 586
  installation options 720, 730
  multiple index data sets 588
  phases that invoke sort 587
  specifying 585
multivolume restrictions 369
MVS console, using with SHRLEVEL CHANGE 644
MVS enqueues, specifying 705

N
names of common utility tables, determining 767
names of objects, truncating in messages 178, 708
naming data sets
  copy, partition level 357
  dynamically allocated 92
  GDG names 92
  original 104, 106
  patterns for dynamic allocation 92
  prefixes for ddnames 93
  staging 103, 105
  user exit 789
nested functions in an expression 149
NEW parameter of EXEC statement 345
NEW/RESTART parameter of EXEC statement 346
NEW/RESTART(PHASE) parameter of EXEC statement 346
NLPCTFREE option 210
NOEXEC option of the REXX compiler 858
node ID indexes 201
NOLIMIT value of MEMLIMIT system parameter 71, 343, 590, 659
NONE keyword
  of DATAACLAS option 745
  of DEADLINE option 328, 683
  of DRAIN_WAIT option 245
  of DRNWAIT option 687
  of DSNTYPE option 309, 750
  of DSNUEXIT option 240, 688
  of DSPLOCKS option 247, 689
  of DSRSEXIT option 242, 690
  of FORCE option 248, 693
  of INDREFLM option 699
  of LEAFDSLM option 704
  of LOGFINAL option 339, 706
  of MAPTEXIT option 244, 709
  of MGMTCLAS option 754
  of OFFPOSLM option 713
  of REGISTER option 267
  of SHRLEVEL option 172
  of SPILLSTORCLAS option 331
  of SPILLUNIT option 330
  of SPILSCLS option 727
  of SPILOUT option 728
  of STORCLAS option 756
  of TERMEXIT option 243, 730
non-key columns, indexes containing 28, 701
nonleaf index page, controlling the amount of free space 210
nonpartitioned indexes, using Instant Snapshot with 122
nonpartitioned objects, considerations when redefining 835
NOSYSREC option 205
null keys 74, 736
NULL keyword
  of predicate 292
  of UPDATE SET option 286
NULLFILE DSN 97, 385
number of CPUs 598
numeric variables 316, 335
NUNIFORM exception 142

O

object availability, SHRLEVEL option 98, 172, 599
object names, long
  syntax rules 156
  truncating in messages 178, 708
object status 84, 86
OFFPOSLIMIT option 231
OFFPOSLM option 713
ON FAILURE option
  description 224
  example 499
  with DELETEFILES option 225
ON MESSAGE option 222
ON n COLUMNS keyword of REBALANCE option 181
online reorganization. See SHRLEVEL CHANGE
ONLY keyword of ANALYZE option 201
operating system requirements 65
operators, comparison 290
operators, with predicate 290
OPNDB2ID
  effects on RACF authorizations 69
  installation option 713
optimal dynamic allocation method 299
OPTIMIZED keyword of ALLOC option 299
options
  command 170
  copy 264
  dynamic allocation 293
  installation 667
  processing of REORG PLUS 170
  selective unload and update 279, 292
  SHRLEVEL CHANGE 326, 340
  statistics 258
options module. See installation options 668
ORDER option
  description 190
  example 481
  memory usage 597
  performance considerations 597
ordering rows. See ORDER option
ordering STOGROUP volumes 836
ORIGDISP option 714
original data set
  deleting and renaming 217
  FASTSWITCH processing 218, 691
ORIGINALDISP option 217

overriding default ddnames
  BCPY 271
  BCPZ 271
  BICY 275
  BICZ 275
  BIRY 277
  BIRZ 277
  BRCY 273
  BRCZ 273
  SYSARC 185
  SYSIDCIN 214
  SYSREC 182
  SYSUT1 183

P

packages, REORG PLUS 715
PACTHI exception 141
PACTLO exception 141
page set REBUILD pending status 85, 87
parallel sort processes 581
parameters, EXEC statement 343
PART keyword of COPYLVL option 269, 680
PART option
  considerations 121
  description 175
  example 442, 468
&PART variable 317, 747
&PART5 variable 317, 748
partial reorganization
  assigning copy data sets 682
  compressed indexes 393
document ID indexes 392
  falling back from FASTSWITCH to rename process 220
  general considerations 121
  REDEFINE 211
  secondary indexes 211
  SHRLEVEL CHANGE 90, 122, 619
  SHRLEVEL NONE 85, 121
  SHRLEVEL REFERENCE 88, 122
  SHRLEVEL REFERENCE UNLOADONLY 88
  specifying copy data sets 682
partition number variable 317, 747
partition rebalancing
  considerations 128
  DDLIN data set 131
  requirements 128
  using REBALANCE option 179
partition-by-growth table spaces
  allocated as DSSIZE 355, 654
  building a dictionary 147
copies for 358
defined as MEMBER CLUSTER 29, 737
example 517
partition-by-growth table spaces (continued)
   extension by DB2 127
   extension by REORG PLUS 126
   inline image copies 266
   invalid with REBALANCE 180
   LAST command option 176
   MAXNEWPARTS option 177, 709
   partition-level SYSRECs 380
   partitions added by REORG PLUS 126
   partitions copied 265
   rebalancing considerations 130
   restart considerations 393
   when partitions are added 605, 609, 621
partitioned objects, reorganizing
   See also partition-by-growth table spaces
   examples 416, 441, 468
   I/O performance 575
   large number of partitions 108
   PART option 175
   table space memory considerations 598
partition-level copies 357, 682
partitioned objects, large number of 108
pattern, data set name
   concatenation 319
   description 315
   installation option 745
   user exit for 688, 789
   variables for 791
pattern, spill data set name
   concatenation 334
   description 332
   user exit for 688, 789
   variables for 791
PAUSE keyword
   of ANALYZE option 201
   of UNLOAD option 205
pausing after analysis 201
pausing after unloading data 205
Pet Dropped Rows exception 141
PetActivHi exception 141
PetActivLo exception 141
PCTCLUS exception 141
PCTDROP exception 141
PENDDDL option 715
pending DDL changes 73, 715
pending statuses
   for SHRLEVEL NONE 84
   for SHRLEVEL REFERENCE, UNLOADONLY, and
   CHANGE 87
percent of table space pages to sample for statistics 263,
733
percentages of data set allocation size 311, 755
performance considerations
   command options 236
   dynamic allocation 579
   ERP environments 591
   full copy data sets 360
genereal 565, 589
I/O 586, 589
incremental copy data sets 656
index work files 588
inline image copies 590
KEEPDICTIONARY option 566
log records 662
multiple data sets 584
multitasking 584
ORDER option 597
processing phases 600, 610
sort 587
spill data sets 664
SYSUT1 data sets 588
persistent read only (PRO) status 85, 87
phases of REORG PLUS
   compared to DB2 REORG 48
   primary functions 51
   SHRLEVEL CHANGE 618, 623, 628
   tuning considerations 600
PLAN option 715
plan, specifying product plan 715
precision, defined on timestamp columns 733
predicate
   evaluation 288
   option (with WHERE) 290
prefix, variable, utility ID
   with DSNPAT option 318, 749
   with SPIILDSN option 727
   with SPIILDSN option 332
prefix, specifying ddname
   COPYDDN 272
   general considerations 93
   ICDDN 276
   RECOVERYDDN 274
   RECOVERYICDDN 278
   UNLDDN 182
   WORKDDN 184
PREFORMAT option
   command 227
   example 554
   installation 715
PRO (persistent read only) status 85, 87
processing options of REORG PLUS 170
processing phases of REORG PLUS
   compared to DB2 REORG 48
   primary functions 51
   SHRLEVEL CHANGE 618
   tuning considerations 600
processor requirements 65
product changes 24
product plan, specifying 715
product support 3
PSEUDODL exception 142
PSRBD status 85, 87
PUBLIC, granting to 67
publications, related 21
queueing, I/O, avoiding 589

RACF (IBM Resource Access Control Facility)
authority required for REORG PLUS 69
OPNDB2ID impact 69
security exit 67
random keys in indexes 74, 506, 702
range-partitioned table spaces
defined as MEMBER CLUSTER 29, 737
rebalancing 171, 257
RBA, high-used
    ANALYZE HURBA option 571
    average row length, determining 571
    cardinality estimate 202, 571
    recovering objects 47
    REDEFINE NO option 212
    resetting 212
RBDP (REBUILD pending) status 85, 87, 674
RBDP* status 85, 87
RCVICDDN option 716
RCVRPEND keyword of ALTRFAIL option 674
RCVYDDN option 716
read claimers, canceling 248, 250, 693
read/write status and SHRLEVEL CHANGE 90, 619
READERS keyword of FORCE option 248, 693
read-only status
    and SHRLEVEL CHANGE 90
    and SHRLEVEL REFERENCE 49
    and SHRLEVEL REFERENCE UNLOADONLY 49
read-or-replication-only (RREPL) status 85, 87
reallocation of data sets, by user 395
real-time statistics 142
REBALANCE option
    command 179
    initial status requirements 84, 86
rebalancing partitions
    DDLIN data set 131, 360
    initial status requirements 84, 86
    REBALANCE option 133, 179
    requirements and considerations 127
REBUILD pending (RBDP) status 85, 87, 674
recopied data sets 394
RECOVER pending (RECP) status
    after ALTER failure 674
    restrictions and usage with failures 399
    with SHRLEVEL NONE 85
    with SHRLEVEL REFERENCE, UNLOADONLY, and
    CHANGE 87
recovery
    after terminating a DB2 object 397
    considerations when not making copies 150
    during SHRLEVEL CHANGE 657
    during UTILTERM phase 400
    from an abnormal termination 224
    of reorganized table space 150
RECOVERYDDN option
    command 273
    example 492
RECOVERYICDDN option 277
RECP (RECOVER pending) status 85, 87, 674
REDEFINE option
    command 210
    example 442
    installation 717
    performance considerations 566, 568
    requirements 102
    with data set redefine exit 837
    with SIXSNAP option 123
redefining data sets
    ordering STOGROUP volumes 836
    pausing a job after analysis 201
    pausing a job after unloading data 205
    providing IDCAMS commands in SYSIDCIN 374, 376
    REDEFINE option, specifying 210, 717
    requirements 102
    resizing 835
    selectively redefining 835
    STOGROUP-defined data sets 210
    SYSIDCIN 374, 376
    VCAT-defined data sets 210, 374, 376
    with user exit 837
REFERENCE keyword of SHRLEVEL option 173
REFERENCE UNLOADONLY keyword of SHRLEVEL option 172
referential integrity considerations
    deleting data during reorganization 285
    general issues 151
    selecting data for unload/reload 283
    updating value in a foreign key column 287
REFP status 85, 87
refresh pending status 85, 87
REGION parameter 343, 589, 593
REGISTER keyword of COPY YES option
    corresponding image copy types (incremental) 654
    description 267
    example 492
registering copy data sets 267, 354
related publications 21
RELOAD keyword of UNLOAD option 204, 735
RELOAD phase
    check for available resources 606
    functions with SHRLEVEL CHANGE 620
    primary functions 52
    specifying ON FAILURE 225
    tuning considerations 588, 605
REMBFCPY keyword of DDTYPE option 295, 742
REMBICPY keyword of DDTYPE option 295, 742
REMPFCPY keyword of DDTYPE option 295, 742
REMPICPY keyword of DDTYPE option 295, 742
rename and delete process for staging data sets
backing out after failure in UTILTERM 402, 674
canceling during 398, 658
completing manually after failure in UTILTERM 400
description 103, 105
failure during 400
falling back from FASTSWITCH process 220, 695
multitasking 586
object status following 89, 90
threshold for 221, 566, 695
RENAME keyword
of ORIGDISP option 714
of ORIGINALDISP option 217
rename process for original data sets
description 104
multitasking 586
RENNMAX option 717
reordered row format 149
REORG option 170
REORG pending (REORP) status 131, 257
REORG phase
general considerations 144
multitasking 587
primary functions 52
restarting 399
tuning considerations 609
REORG type variable
with DSNPAT option 317, 748
with SPILDSNP option 726
with SPILLDSNPAT option 332
REORG utility of IBM
comparison with 48
invoking DSNUTILILB to use 72, 237, 689
restarting 399
report
of applied fixes 347
of canceled threads 249, 352, 694
of SQL ALTER statements 366
statistics 258, 260
REPORT keyword of BMCSTATS option 260
REPORTONLY keyword of FORCE option 250
REPORTONLY option 236
requirements, system and software 65
RESET keyword of MGEXTENT option 712
resetting a column value to a constant 286
resetting DASD MANAGER PLUS exceptions 138
resizing data sets 835, 842
Resource Access Control Facility. See RACF
resources, check for in RELOAD phase 606
RESTART parameter of EXEC statement 345
restarting
ACTIVE option considerations 96
after inadequate space failure 395
compressed indexes considerations 393
considerations 392
data sharing considerations 394
DELETEFILES option considerations 395
document ID indexes considerations 392
during UTILTERM phase 398, 658
dynamic allocation considerations 95, 391, 396
from failure 391
inline image copies considerations 396
LOB table spaces considerations 392
paused job, example 474
prior to UTILTERM for SHRLEVEL CHANGE 657
recopied data sets considerations 394
RESTART parameters 346
restart parameters of EXEC statement 345
restrictions 392
statistics considerations 395, 658
work files considerations 396
XML table spaces considerations 392
RESTP status 85, 87
restrictive statuses
for SHRLEVEL NONE 84
for SHRLEVEL REFERENCE, UNLOADONLY, and
CHANGE 87
RETCODE keyword
of ON FAILURE TERMINATE UTILITY option 226
of ON MESSAGE CONTINUE UTILITY option 223,
226
retention period for certain data sets 322, 754
RETPD option 322, 754
RETRY keyword
of DSPLOCKS option 247, 689
of FORCE_AT option 249, 693
RETRY option 246
RETRY parameter of DYNALOC option 763
RETRY_DELAY option 246
retrieving drain 246, 686
retrieving dynamic allocation of SORTWK 763
return codes
for DSNUEXIT user exit 790
retrieving SQL -911 728
REXX exits 833
specifying with ON FAILURE 224, 226
specifying with ON MESSAGE 223, 226
specifying with TIMEOUT 238, 731
REUSE option 213
REXX exits
accessing the source 788
common variables 832
REXX exits (continued)
  data set redefine
    description 834
    example 844, 852
  installation option 690
DSRSEXIT
  description 834
  example 844, 852
  installation option 690
mapping table creation 857
MAPTEXIT
  description 857
  example 861
  invoking 243, 709
restrictions 832
return codes 832
sample library 788
setting up 837, 850, 859
TERMEXIT 849
REXX keyword
  of DSRSEXIT option 242, 690
  of MAPTEXIT option 244, 709
  of TERMEXIT option 243, 730
RID translation map
  benefits due to REORG PLUS technology 630
  defining memory for 336
  description 619
  restricting memory for 660
RIDMAPMEM option
  command 336
  restricting size 718
RIDMDSSZ option 718
RIDMMAXD option 718
RMAPMEM option 718
RORGMAX option 587, 719
ROUTCDE option 720
row change time stamp column 121, 287
row format 149
row length, determining average 199
row-level security 153
ROWS/KEY exception 142
RREPL (read-or-replication-only) status 85, 87
&RTYPE variable
  with DSNPAT option 317, 748
  with SPIILDSNP option 726
  with SPIILDSNPAT option 332
rules for constants 291
running BMC utilities concurrently 82
RUNSTATS cardinality changes 262

Sampling
  cardinality 201, 571
  for statistics 263, 733
SAS/C runtime library 348
SCAN keyword of ANALYZE option 201
scanning, controlling 201
SCC (DB2 Solution Common Code)
  description 61
  load library 348
  version requirements 66
SCPYMAX option 720
SDUMP option 720
secondary extents 712
secondary indexes
  multitasking 586
  REDEFINE 211
secondary templates, switching to 313, 758
security
  mechanisms 67
  requirements 66
  row- and column-level 153
SELECT option
  authorization information 67
  example 492
  referential integrity issues 151
  SHRLEVEL CHANGE considerations 649
  specification information 282
SELECT privileges 68
selecting data, specifying conditions 282
selective unload and update options 279, 292
selectively redefining VSAM data sets 835
Send New command of XBM 640
sequential data sets, extended
  allocating 310, 750
  SORTWK restriction 369
  SYSREC restriction 382
serialization issues 82, 646
shadow data sets. See staging data sets
SHORTMEMORY option 195, 721
SHRLEVEL CHANGE
  altering dynamically 640
  batch applications using AR/CTL 650, 651, 674
  benefits 44
  change block options 326, 340
  command option 173
  concurrency issues and restrictions 646
  conditional reorganization 136
  considerations when using SELECT or DELETE 649
  controlling the log apply process 633, 638
  copy data sets 652
  criteria for ending 635
  criteria for entering LOGFINAL 635
  data capture flag 619, 621
  DB2 resources used 629
  description 100
SHRLEVEL CHANGE (continued)
diagram of log apply control options 636
diagrams of phases 623, 628
displaying job information 640
examples 529, 542, 553
evaluation phases, description 618, 622
failure during 674
flow diagram 617
incompatible options 648
incremental copy data sets 653
information displayed by XBM 639
initial object status 87
options that control 635
overview 617
recovery 657
restart 657
restrictive statuses 87
specifying drain type 340, 686
staging data sets 101
status changes during reorganization 90
using XBM to control execution 639, 644
SHRLEVEL column of BMCSYNC table 82, 774
SHRLEVEL NONE
command option 172
initial object status 84
restrictive statuses 84
status changes during reorganization 86, 88, 90
SHRLEVEL option
considerations for use 98, 599
description 172
SHRLEVEL REFERENCE
command option 173
description 100
examples 416, 441
failure during 674
initial object status 87
restrictive statuses 87
staging data sets 101
status changes during reorganization 88
terminating job 398
SHRLEVEL REFERENCE UNLOADONLY
command option 172
description 99
initial object status 87
restrictive statuses 87
status changes during reorganization 88
single-phase reorganization
diagrams 56
issues 144
performance considerations 601, 609
SHRLEVEL CHANGE diagrams 626, 628
specifying 204, 735
SYSREC considerations 379, 575
SYSUT1 considerations 576
SIXSNAP option
command 251
installation 723
with REDEFINE option 123
SIXSNAP processing
DASD vendor considerations 125
handling failures 123
with multiple data set objects 111
SIZEPCT option 311, 755
sliding-scale calculation for secondary extents 712
SMAX option
controlling BMCSORT 583
description 723
performance considerations 598
used with MAXSORTS 189
SMCORE option
controlling BMCSORT 581
description 724
SMS (IBM Storage Management Subsystem)
classes 324, 745
dynamic data set allocation 300, 755
SMS extended sequential data sets
allocating 310, 750
SORTWK restriction 369
SYSREC restriction 382
SMS option 300, 755
SMSUNIT option 301, 756
snapshot copy 567
SNAPSHOT UPGRADE FEATURE (SUF)
description 152
for nonpartitioning indexes 152
requirements 122
using to enable zIIP processing 153, 252, 739
using with SHRLEVEL CHANGE option 619
software compression. See compression
software requirements 66
Solution Common Code. See SCC
sort processing
concurrent processes 581
controlling 580
multitasking 587
sort processing options
AVAILPAGEPCT 193, 676
MAXSORTMEMORY 197, 711
MAXSORTS 189
MINSORTMEMORY 197, 712
SHORTMEMORY 195, 721
SMAX 723
SMCORE 724
SORTDATA 190
SORTDEV 724
SORTKEYS 174
SORTNUM 188, 598, 725
sort work data set. See SORTWK data set
| 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 |

**SORTDATA option**
- description 190
- example 499

**SORTDEVT option** 187, 724
- sorting data, memory usage 597

**SORTKEYS option**
- command 188
- example 507
- installation 725
- turning on BMC ASD 98

**SORTWK data sets**
- allocating 367
- compression and ANALYZE option 390
- deleting 215
- description 59
- dynamic allocation
  - DDTYPE option 294
  - description 90
  - SORTDEVT option 187
  - syntax 293
  - with SORTNUM 188, 725
- performance information 580
- size estimate provided by ANALYZE 388
- specification and usage 366

**SORTWORK keyword of DDTYPE option** 294, 742
- space
  - allocations 356
  - failure due to inadequate 395
  - key compression of nonleaf pages in unload phase 205
  - specifying for allocation 312
- SPACE exception 140
- SPACE option 312
- specifying installation options module 347

**SPIILDSNP option** 567, 726
- spill data sets
  - calculating size 664
  - description of options 649
  - naming 331
  - pattern
    - concatenation 334
    - description 332
  - variables for 791

**SPIILDSNPAT option** 331, 569
**SPIILSTORCLAS option** 331, 569
**SPIILUNIT option** 330, 569
**SPIILSCLS option** 567, 727
**SPIILUNIT option** 567, 728
**SQL statements**
- altering data set sizes 835
- altering limit keys 127, 360
- altering objects in DDLIN data set 360
- deleting rows from the BMCDICT table 770
- deleting rows from the BMCASYNCTable 772
- deleting rows from the BMCSYNT table 776, 780
- deleting rows from the BMCXCOPY table 786
- displaying BMC utilities 768
- querying BMCHST table 768
- querying BMMCCOPY table 768
- reporting ALTER statements in DDLOUT data set 366
- terminating BMC utilities 768

**SQLDELAY option** 728
**SQLRETRY option** 728
**SSID (DB2 subsystem identifier) parameter of EXEC statement** 344

**&SSID variable**
- with DSNSPAT option 317, 748
- with SPIILDSNP option 726
- with SPIILDSNPAT option 332

**STAGEDSN option**
- and rename process 103
- authority needed 70
- data set rules 591
- description 728
- performance consideration 566
- staging data sets
  - authorization 599
  - canceling or terminating job during UTILTERM 398, 658
  - defining with IDCAMS commands in SYSIDCIN 374, 376
  - description 101
  - example 416, 441
  - failure during switching of data sets 400
  - falling back from FASTSWITCH to rename process 220, 695
  - FASTSWITCH process 105
  - naming conventions 103, 105
  - preallocation requirements 102, 213
  - rename and delete process 103, 105
  - SHRLEVEL CHANGE used in RELOAD phase 620
  - threshold for determining switch process 221, 566, 695

**STANDARD keyword of NLPCTFREE option** 210
**START keyword of FORCE_AT option** 249, 693
**statistics**
- bypassing 572
- DASD MANAGER PLUS 202, 259, 572
- DB2 catalog updates 261
- options 258
- output data set, optional 351
- real-time 142
- report output 258, 260
- restart considerations 395, 658

**SPACE exception** 140
**SPACE option** 312
**specifying installation options module** 347
**spill data sets**
- calculating size 664
- description of options 649
- naming 331
- pattern
  - concatenation 334
  - description 332
- variables for 791

**STAGEDSN option**
- and rename process 103
- authority needed 70
- data set rules 591
- description 728
- performance consideration 566
- staging data sets
  - authorization 599
  - canceling or terminating job during UTILTERM 398, 658
  - defining with IDCAMS commands in SYSIDCIN 374, 376
  - description 101
  - example 416, 441
  - failure during switching of data sets 400
  - falling back from FASTSWITCH to rename process 220, 695
  - FASTSWITCH process 105
  - naming conventions 103, 105
  - preallocation requirements 102, 213
  - rename and delete process 103, 105
  - SHRLEVEL CHANGE used in RELOAD phase 620
  - threshold for determining switch process 221, 566, 695

**STANDARD keyword of NLPCTFREE option** 210
**START keyword of FORCE_AT option** 249, 693
**statistics**
- bypassing 572
- DASD MANAGER PLUS 202, 259, 572
- DB2 catalog updates 261
- options 258
- output data set, optional 351
- real-time 142
- report output 258, 260
- restart considerations 395, 658
statistics (continued)
  sampling 263, 733
  with DSNUTILB reorganization 143, 262
STATISTICS option of IBM DB2 REORG utility 262
status requirements of objects for reorganization 84, 86
status, BMC utilities 768
STEP name variable
  with DSNPAT option 317, 748
  with SPILDSNP option 726
  with SPILLDSNPAT option 332
STEPLIB DD statement 348, 858
&STEPNAME variable
  with DSNPAT option 317, 748
  with SPILDSNP option 726
  with SPILLDSNPAT option 332
STOGROUP-defined data sets
  ordering through user exit 836
  redefining data sets during reorganization 210
  requirements 102
  using DSNZPARM values 212
STOP status 49
STOP UTILITY keyword
  of ON FAILURE option 226
  of ON MESSAGE option 222
STOP@CMT option 729
STOPDELAY option 729
STOPRETRY option 729
storage above the bar 71
storage classes, SMS 325, 756
Storage Management Subsystem. See SMS
storage, virtual 71
STORCLAS option 325, 756
stored procedure, DSNUTILS 36
striped data sets 369
subset of partitions in single image copy 682
SUBSTR SQL function 148
subsystem ID parameter of EXEC statement 344
subsystem ID variable
  with DSNPAT option 317, 748
  with SPILDSNP option 726
  with SPILLDSNPAT option 332
subsystem ID, XBM 253, 738
SUF. See SNAPSHOT UPGRADE FEATURE
suffix variable, utility ID
  with DSNPAT option 318, 749
  with SPILDSNP option 727
  with SPILLDSNPAT option 333
SUFSTART keyword of WTOMSG option 254
summary of changes 24
support, customer 3
supported data types in limit keys 129
supported DB2 versions 65
suspend and resume interface, with AR/CTL 650
switch process for staging data sets
  description 105
  falling back after failure 220
  FASTSWITCH option 218, 691
  FSFALLBACK option 220, 695
  FSTHRESHOLD option 221, 695
  threshold for 221, 566, 695
switching, template 313, 758
symbolic variables
  DSN pattern 316, 747
  SPIILDSNP pattern 726
  SPIILDSNPAT pattern 332
SYNC option 208
syntax
  alphabetical listing of command options 156
  command option descriptions 170
  diagrams 160
  rules 155
syntax, format for diagrams 23
SYSCALLDA unit name 758
SYSARC data sets
  changing the default ddname 675
  description 370
  determining size with ARROWS 229
  dynamic allocation
    DDTYPE option 294
    description 90
    limiting GDG versions 323
    specifying expiration date 322
    specifying retention period 322
    using GDG names 92
  example 492
  format of, specifying 185, 675
  overriding default ddname 185
  specification and usage 370
  with partition rebalancing 128
SYSCOPY table
  duplicate entries 267
  registering copies in 112, 267
  registration information 354, 653
SYSCOPY table space, reorganizing 111
SYSERR data sets, specification and usage 373
SYSEXEC data sets, specification and usage 373
SYSIBM.SYSCOLUMNS table 262
SYSIBM.SYS_COPY table
  duplicate entries 267
  registering copies in 112, 267
  registration information 354, 653
SYSIBM.SYSFIELDS table 260, 261
SYSIBM.SYSEXECSPACESTATS real-time statistics table 144
SYSIBM.SYSSERIES table 68, 255, 699
SYSIBM.SYSSERIESDEP table 68
SYSIBM.SYSTABLESPACESTATS real-time statistics table 143
SYSIBM.SYSSXMLRELS table 68
SYSIDCIN data sets
  description 60
  example 407
SYSIDCIN data sets (continued)
overriding default ddname 214
specification and usage 374
when required 102
SYSIDCIN DD statement 374
SYsin data sets
description 60
specification and usage 376
SYSPRIN2 data sets
changing message levels 347
description 60
specification and usage 377
worklist environment considerations 377
SYSPRINT data sets
changing message levels 347
description 60
example REORIG PLUS job output 403
specification and usage 376
SYSPUNCH data sets
allocating 294
description 60
specification and usage 378
SYSPUNCH keyword of DDTYPE option 294, 742
SYsREC data sets
controlling buffer usage 575
DD DUMMY 97
deleting automatically after job 215
description 60
DSN=NULLFILE 97
dynamic allocation
DDTYPE option 294
description 90
syntax 293
multiple 60, 575, 588
overriding default ddname 183, 737
performance considerations 588
size estimate provided by ANALYZE 388
SMS extended sequential data sets 385
specification and usage 382
tuning considerations 588
worklist environment considerations 385
XML table space reorganization 201
SYSTUT1 data sets
allocating 351
controlling buffer usage 576
DD DUMMY 97, 385
deleting automatically after job 215
description 60
DSN=NULLFILE 97, 385
dynamic allocation
DDTYPE option 294, 349
description 90
syntax 293
XML table space reorganization 201
multiple 588
overriding default ddname 183, 737
performance considerations 588
size estimate provided by ANALYZE 388
SMS extended sequential data sets 385
specification and usage 382
tuning considerations 588
worklist environment considerations 385
XML table space reorganization 115

T

T table names, determining 767
table space compression
description 145
KEEPDICTIONARY option 206, 703
restrictions with partition rebalancing 146
table space reorganization
performing conditionally 136
status requirements 84, 86
table spaces
base, instance number of 678
catalog, reorganizing 111
defined as MEMBER CLUSTER 29, 737
LOB 116, 392
multi-table 78
name variable
with DSNPAT option 317, 748
with SPILDSNP option 726
with SPILLDSNPAT option 332
partition-by-growth 130
partitioned, large number of partitions 598
range-partitioned 29, 171, 737
recoverability 150
SHRLEVEL CHANGE access to objects 90
specifying for reorganization 171
universal 29, 737
XML 114
table-controlled partitioning 370, 553
tables
archive enabled 279
cloned 171, 678
tables (continued)
  mapping 75, 243, 340
  organized by hash 73, 256, 696
  temporal 27, 279, 730

tables, BMC
  backing up 766
  BMCDICT 769
  BMCHIST 770
  BMCLGRNX 772
  BMCSYNC 53, 773
  BMCTRANS 777
  BMCTILL 53, 778
  BMXCOPY 777, 781
  common utility 110
  considerations 766
  determining names 767
  querying 768
  warnings 766
TABLESPACE option 170

tape
  changing work file disposition 729
  devices, number of 95, 293, 711
  with MAXTAPE 95
TAPEDISP option 729
TASKMAX option 730
  tasks
    performed by REORG PLUS 46
    specifying maximum for sort 189
  technical support 3
TEMPLATE control statements 78, 313, 758
  template switching 313, 758
  temporal tables 27, 279, 730
  temporary data sets
    definition 96
    how REORG PLUS handles 96, 97
TEMPRALDATA option 730
TERM keyword
  of ALTRFAIL option 674
  of CPYRFAIL option 682
  of LONGLONG option 329, 708
  of TIMEOUT option 239, 731
TERM parameter 396
  TERM parameter of EXEC statement 346
TERMEXIT option 730
TERMEXIT user exit
  description 849
  invoking 850
  specifying 730
  using 849
  variables passed 851
TERMINATE UTILITY keyword of ON FAILURE option
  DELETEFILES option processing 216
  description 225

terminating
  DB2 object, recovery after 397
  or canceling a job 201, 396
  SHRLEVEL CHANGE reorganization 635
  terminating BMC utilities 768
  TERSE keyword of DISPLAY option 640
  threads, canceling
    authorizations required 69
    FORCE option 248, 693
    FORCE_AT option 693
    FORCE_RPT option 694
    reporting 249, 352, 694
    specifying when to cancel 248, 693
  threads, number used 71
  THRESHOLD option 94, 312, 757
  threshold for determining switch process
    performance recommendation 566
    specifying 221, 695
  threshold for secondary values 94, 312, 757
  time
    specifying with DEADLINE 328
    specifying with LOGFINAL 339
  time rules for constants 291
  &TIME variable
    with DSNPADD option 317, 748
    with SPIILDSNP option 726
    with SPIILDSNP option 332
  time variable, current
    with DSNPADD option 317, 748
    with SPIILDSNP option 726
    with SPIILDSNP option 332
  time variable, short form
    with DSNPADD option 317, 748
    with SPIILDSNP option 726
    with SPIILDSNP option 332
  time zone, timestamp
    data type 129, 734
    implicit time zone 281
    with CURRENT TIMESTAMP keyword of predicate 292
    with DEADLINE 328
    with LOGFINAL 339
    with UPDATE SET CURRENT TIMESTAMP 286
  &TIME4 variable
    with DSNPADD option 317, 748
    with SPIILDSNP option 726
    with SPIILDSNP option 332
  timeout
    displaying locks 247, 689
    specifying the number of retry attempts 246, 686
    specifying the time between retry attempts 246, 686
    specifying the value for 245, 687
TIMEOUT option
  command 238
  installation 731
  with DELETEFILES processing 216
  timeout, transactions retry after 592
  timestamp
    columns with precision 733
    columns with time zone 734
    specifying with DEADLINE 328
timestamp (continued)
specifying with LOGFINAL 339
with time zone
data type 129
implicit time zone 281
with CURRENT TIMESTAMP keyword of
predicate 292
with DEADLINE 328
with LOGFINAL 339
with UPDATE SET CURRENT TIMESTAMP 286
TIMESTAMP WITH TIME ZONE, defined on timestamp
columns 129, 734
tokens in syntax, rules for 156
TOTALIND exception 141
TOTALOFF exception 141
TOTALPAGEPCT option 194, 732
TRACE authority 67
translation of command constants 281
triggers
advisory pending 142
append inserts 142
generating a reorganization 138
truncating long names in messages 178, 708
&TSIX variable
with DSNPAT option 317, 748
with SPIILDSNP option 726
with SPIILDSNPAT option 332
TSPREC option 733
TSSAMPLEPCT option 263, 733
TSTZ option 734
tuning REORG PLUS
dynamic allocation of data sets 579
I/O processing 584, 589
improving memory use 597
installation options 566
memory use 597
multitasking 584
phases 600
two-phase reorganization
diagrams 54
example 492
SHRLEVEL CHANGE diagrams 623, 625
specifying 204, 735
unit count 302
unit names, for dynamic allocation 302, 758
UNIT option
command 302
installation 758
with SMSUNIT option 95, 301
UNITCNOPT option 302, 759
universal table spaces 29, 737
UNLDDN option 182, 734
UNLDMAX option
description 735
multitasking considerations 587
unload data sets. See SYSREC data sets
UNLOAD keyword of DDTYPE option 294, 742
UNLOAD option
command 204
CONTINUE example 492
installation 735
PAUSE example 468
with REDEFINE 204
UNLOAD phase
functions with SHRLEVEL CHANGE 619
multitasking 587
ORDER option 597
partial reorganization 211
pausing and continuing after unloading data 204
primary functions 52
restarting 398
specifying ON FAILURE 225
tuning considerations 580, 602
unload work files. See SYSREC data sets
UNLOADONLY keyword of SHRLEVEL REFERENCE
option 172
unsupported data types in limit keys 130
unsupported features of DB2 153
unused pages 193, 676
UPDATE keyword of ICTYPE option 268, 698
UPDATE option
authorization information 67
description 286
example 417
options 292
referential integrity issues 151
UPDATEDB2 Common Statistics component option 261
UPDATEDB2STATS option
description 261
example 407, 508
with DSNUMILB reorganization 262
UPPER SQL function 148, 149
USE keyword of IFALLOC option 297, 752
user exits
accessing the source 788
authorizations for 68, 837
controlling BMCHIST update 852
controlling statistics updates 852
user exits (continued)

creating mapping objects 857
data set name pattern
  assembler example 791
  C example 814
  COBOL II example 805
description 789
  invoking 688
  LE COBOL example 805
  LE_C example 823
variables 316, 332
data set redefine
description 834
  example 844
  invoking 690

DB2 151, 584
dropping mapping objects 857

DSNUEXIT
  assembler example 791
  C example 814
  COBOL II example 805
description 789
  invoking 688
  LE_C example 823
  LE_COBOL example 805

DSRSEXIT
description 834
  example 844
  invoking 690

mapping table
description 857
  example 861
  invoking 243, 709

MAPTEXIT
description 857
  example 861
  invoking 243, 709

REXX
requirements 858
restrictions 832
setting up 837, 850, 859

sample library 788

TERMEXIT
controlling BMCHIST update 849
controlling statistics updates 849
description 849
  example 852

user ID variable
  with DSNPAT option 318, 748
  with SPIILDSNP option 727
  with SPIILDSNPAT option 332
user-defined variables 316, 791

USERID (user identifier) parameter of EXEC statement 347

&USERID variable
  with DSNPAT option 318, 748
  with SPIILDSNP option 727
  with SPIILDSNPAT option 332

UTILTERM phase
backing out after failure 401, 674
completing manually after failure 400
failure involving DDLIN processing 400, 674
for SHRLEVEL CHANGE 622
primary functions 53, 208, 678
UTILTERM phase (continued)
  recovery during 400
  restart during 399, 658
  specifying ON FAILURE 225
  unable to restart 400
UTPRINT data set
  description 60
  specification and usage 386
UTRO (utility restrictive state, read-only access allowed) 85, 87
UTRW (utility restrictive state, read/write access allowed) 85, 87
UTSMEM option 737
UTUT (utility restrictive state, utility exclusive control) 85, 88
UXSTATE option 584, 737

V

VALIDPROCs
  libraries that contain 348
  row format conversion 150
variable control interval (CI) size 148
variables
  for DSN pattern
    concatenation 319
    description 316, 745
    numeric 316
    user-defined 791
  for SPILLDSN pattern
    concatenation 334
    description 332, 726
    numeric 335
    user-defined 791
  passed to DSRSEXIT user exit 838
  passed to MAPTEXIT user exit 859
  passed to TERMEXIT user exit 851
  REXX user exits, common for 832
&VCAT variable
  with DSNPAT option 318, 749
  with SPIILDSNP option 727
  with SPIILDSNPAT option 333
VCAT-defined data sets
  redefining during reorganization 210, 374, 376
  requirements 102
VCAT-defined table space, example 407
VCATNAME variable
  with DSNPAT option 318, 749
  with SPIILDSNP option 727
  with SPIILDSNPAT option 333
VERBOSE keyword of DISPLAY option 640
versioned indexes
  compressed 32
  in SYSTSIPIT table space 111
  versions of components, required 66
versions of DB2
  supported 65
  when different on data sharing group 344
versions, XML 28, 739
VIO data sets, restrictions 369
virtual storage
  above the bar 71
  above the line 589
  and MEMLIMIT 71
  below the line 589
  providing maximum 589
VOLCNT option 304, 760
volume count 304
VSAM buffers and the 16-megabyte line 575
VSAM data sets
  FASTSWITCH process 105
  ordering STOGROUP volumes 836
  redefining during reorganization 211
  redefining through a user exit 834
  rename and delete process 103
  resizing 835
  selectively redefining 835

W

WARN keyword of FILECHK option 692
WBUFFS option 737
WEPR status 85, 88
WHERE option
  DELETE usage 285
  description 288
  example 492
  SELECT usage 283
  UPDATE usage 287
WITH TIME ZONE keyword
  of CURRENT TIMESTAMP keyword of predicate 292
  of UPDATE SET CURRENT TIMESTAMP option 286
work files
  See also dynamic allocation of data sets
  deleting 94, 215
  dynamic allocation 187, 683
  multiple 584
  restart considerations 396
  sort (SORTWK) 367
WORK keyword of DDTYPE option 294, 742
WORKDDN option 183, 737
worklist environment
  dynamic data set allocation 91, 296
  multiple reorganization considerations 382
  restriction when deleting files 685
  SMS extended sequential data sets 382
  viewing utility output 377
WORKUNIT option 738
write claimers, canceling 250, 693
write page error status 85, 88
WRITERS keyword
  of DRAIN option 340
  of DRAINTYP option 686
WTO messages, specifying routing code 720
WTOMSG option 254

X

XBM
  data sharing 253
  description for SHRLEVEL CHANGE 152
  for nonpartitioning indexes 152
  information displayed for SHRLEVEL CHANGE 639
  initialization 619
  Instant Snapshot function with nonpartitioned
  indexes 122
  requirements 122
  Send New command 640
  using the MVS console 644
  using the Utility Monitor 639
  using to control a SHRLEVEL CHANGE
  reorganization 639, 644
  using to enable zIIP processing 153, 252, 739
XBMGROUP name 253
XBMID option
  command 253
  example 530
  installation 738

XML data
  authorizations 68
  BMCSYNC table considerations 775
  considerations when reorganizing 113
  document ID column 114
  restrictions on partition rebalancing 113
XML indexes 68, 114
XML option 739
XML table spaces
  ANALYZE option 115, 200
  considerations when reorganizing 114
  restrictions on ANALYZE SCAN 202
  SYSUT1 data sets for reorganization 201
XML versions 28, 739

Y

YEAR SQL function 148, 149

Z

ZIIP option 252, 739
zIIP processing
  enabling 153
  option to enable or disable 252, 739
  requirements 66