REORG PLUS for DB2
Reference Manual

Supporting

Version 11.2 of REORG PLUS for DB2
Version 11.2 of BMC High Speed Utilities for DB2
Version 11.2 of BMC Performance for DB2 Databases
Version 11.2 of BMC Database Performance for DB2

May 2015
Contacting BMC Software

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

United States and Canada

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

Outside United States and Canada

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


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.

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

SAP is a trademark or registered trademark of SAP AG in Germany and in several other countries.

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

Restricted rights legend

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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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

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

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

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

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

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

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

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

Recoverability of the reorganized table space .......................................................... 601
Copy data sets for SHRLEVEL CHANGE ............................................................ 601
Incremental copy data sets for SHRLEVEL CHANGE ........................................... 602
Copy registration failure during a SHRLEVEL CHANGE reorganization ............. 606
Restart considerations for a SHRLEVEL CHANGE reorganization ..................... 607
Performance considerations for SHRLEVEL CHANGE reorganizations .............. 609
Sizing memory for the RID translation map ...................................................... 610
Sizing memory for log records ........................................................................ 611
Sizing the spill data sets ................................................................................... 613
Calculating storage requirements for log data .................................................. 614
Copying nonpartitioned indexes during a partial table space reorganization ......... 615
Making inline copies during a SHRLEVEL CHANGE reorganization ............... 615

Appendix A REORG PLUS installation options 617
Overview of REORG PLUS installation options .................................................. 617
Basic REORG PLUS installation options ............................................................ 618
Descriptions of basic REORG PLUS installation options .................................. 625
Dynamic allocation installation options for REORG PLUS .................. 689
Descriptions of dynamic allocation installation options for REORG PLUS ......... 691
DYNALOC installation option ................................................................. 711

Appendix B Common utility tables 715
Overview of common utility tables ............................................................ 715
Warnings and considerations for common utility tables ................................ 717
Managing common utility tables ................................................................. 719
BMCDICT table ......................................................................................... 720
BMCDICT table considerations ................................................................. 721
Maintaining the BMCDICT table ................................................................. 721
BMCHIST table ......................................................................................... 722
BMCHIST table considerations for COPY PLUS ........................................ 724
BMCHIST table considerations for RECOVER PLUS .................................... 724
Maintaining the BMCHIST table ................................................................. 725
BMCLGRNX table ......................................................................................... 725
BMCSYNC table ......................................................................................... 726
BMCSYNC table considerations ................................................................. 729
Maintaining the BMCSYNC table ................................................................. 731
Cleaning up RECOVER UNLOADKEYS entries .............................................. 731
Shared access levels of BMC utilities ......................................................... 732
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, you can use the following methods to access related publications that support your product or solution:


- View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are available from the following locations:
— Documentation Center

— Support Central (at http://www.bmc.com/support/mainframemonstrations)

— BMC Mainframe YouTube channel (https://www.youtube.com/user/BMCSoftwareMainframe)


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

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

Conventions

This document uses the following special conventions:

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

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

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

Syntax diagrams

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

The following guidelines provide additional information about syntax diagrams:

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

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

**Summary of changes**

This topic summarizes product changes and enhancements by version number and release date.

**Version 11.2.00, May 2015**

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

- REORG PLUS now supports the following features natively:
  - RBA or LRSN conversion to or from extended format (by using the RBALRSN_CONVERSION command option) when you specify SHRLLEVEL CHANGE
    
    **Note**
    
    PTFs BPU7461 and BPU7526 provided this functionality for version 11.1.00. For more information, see the technical bulletin dated February 16, 2015.

  - Partition rebalancing when the following conditions exist:
    - You are reorganizing a table-based partitioned object.
    - You are running on DB2 Version 11 new-function-mode.
    
    **Note**
    
    PTFs BPU7042 and BPJ0839 provided this functionality for version 11.1.00. For more information, see the technical bulletin dated October 13, 2014.

This enhancement resulted in a change in limit key processing in REORG PLUS. Due to this change:
—You no longer need ALTER INDEX and ALTER TABLE privileges for most reorganization jobs.

For more information about required authorizations, see “Setting REORG PLUS authorizations” on page 53.

—The ALTRFAIL installation option is now obsolete. For information about what happens when a failure occurs, see “General rebalancing considerations” on page 116.

—The CPYRFFAIL installation option is now always set to COPYPEND when you are rebalancing partitions during your reorganization.

—REORG PLUS ignores the DB2 subsystem parameter PREVENT_ALTERTB_LIMITKEY for index-controlled partitioned objects.

—In most cases, materializing pending definition changes during the reorganization. Due to this change, the CPYRFFAIL installation option is now always set to COPYPEND when REORG PLUS materializes pending changes.

For more information, see “Reorganizing objects with pending definition changes” on page 98.

■ You can now use the MAPTEXIT user exit with DB2 Version 11. (Version 11.1 of REORG PLUS supported this user exit only with DB2 Versions 9 and 10.)

This user exit provides an automated method for supplying a mapping table when invoking DSNUTILB. For more information, see “MAPTEXIT” on page 231.

■ REORG PLUS no longer requires or supports the individual installation options for each feature that requires invoking DSNUTILB. However, you must still specify DSNUTILB=YES.

This enhancement makes the following installation options obsolete:

—HASHAX
—IXONEX
—IXRANDOM
—LOB
—PENDDDL
—UTILB_COLCCSID
—UTILB_NULLIX

If your installation options module contains any of these options, the module compiles successfully but completes with return code 4 and a message indicating that the option is obsolete.

■ The second parameter of the WBUFFS installation option is now obsolete. REORG PLUS uses only the first parameter for all SYSUT1 data sets.

■ REORG PLUS now supports read-or-replication-only (RREPL) status. Note the following restrictions:
— REORG PLUS invokes DSNUTILB when either of the following conditions exists:

— SHRLLEVEL NONE or SHRLLEVEL CHANGE is in effect.

— Some, but not all, logical partitions of a nonpartitioned index are in RREPL status.

— REORG PLUS terminates when you specify SELECT, DELETE, or UPDATE for a partition that is in RREPL status.

REORG PLUS now supports indexes in which updated entries remain in their original position on the page.

Starting with this release, REORG PLUS does not support DB2 Version 9. Earlier releases will continue to support Version 9.

In addition, REORG PLUS supports DB2 Version 10 only in new-function mode.

This version of REORG PLUS has the following changes to minimum requirements. For full requirement information, see “System setup for REORG PLUS” on page 51.

— DB2 Solution Common Code (SCC) version 11.1.00 with PTF BPJ0944
— DB2 Utilities Common Code (D2U) version 11.2.00
— BMC Common Statistics component version 11.2.00

**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 SHRLLEVEL CHANGE.
- 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:

- Reorganizations of table spaces that contain indexes that are defined with EXCLUDE NULL KEYS (supported by a new installation option, UTILB_NULLIX)

  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)

- Table space reorganizations on DB2 Version 11 when you specify REBALANCE and the table space is defined with table-controlled partitioning

**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.

- 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.
### 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 enhanced 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 the "System setup for REORG PLUS" topic.
  
  — 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*

The following enhancements provide native support 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.

---

**What's new in Version 10.2.00**

- 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.
  - 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.
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.

- 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.

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.

■ 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.

■ 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.

This option requires additional DB2 authorization. For more information, see the REORG PLUS authorization requirements topic.

You can also specify a new DD statement, BMCFORCE DD, to contain the thread cancelation report.

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.

— 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.

— 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.

— 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.

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.

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 description.

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.

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.

— 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. 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
Introduction to REORG PLUS

This chapter describes the features and advantages of REORG PLUS and provides an architectural overview of the product.

Overview of REORG PLUS

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

- **Delivers improved data availability** to meet growing 24 x 7 requirements

- **Operates in a nondestructive manner,** which allows you to easily make the objects available without having to recover

**BMC Software solution integration**

REORG PLUS is also a component of the following BMC solutions. For more information, see the release notes for these solutions.

**BMC Performance for DB2 Databases solution**

The Performance for DB2 Databases solution helps DBAs determine the maintenance tasks that are required on their DB2 objects and, optionally, automate execution of
those tasks. The following REORG PLUS features are available only when you install the Performance for DB2 Databases solution:

- 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.

- 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.

This solution includes the following products and primary technology components:

- **DASD MANAGER PLUS for DB2**
- **REORG PLUS for DB2**
- **SNAPSHOT UPGRADE FEATURE for DB2**

**BMC High Speed Utilities for DB2 solution**

The High Speed Utilities for DB2 solution delivers high availability of objects during data and database maintenance. The solution also significantly reduces runtime costs by minimizing CPU consumption.

This solution includes the following products and primary technology components:

- **CHECK PLUS for DB2**
- **BMC Common Statistics**
- **COPY PLUS for DB2**
- **LOADPLUS for DB2**
- **RECOVER PLUS for DB2**
- **REORG PLUS for DB2**
- **SNAPSHOT UPGRADE FEATURE for DB2**
- **UNLOAD PLUS for DB2**
BMC Database Performance for DB2 solution

The Performance for DB2 Databases solution helps DBAs determine the maintenance tasks that are required on their DB2 objects and, optionally, automate execution of those tasks. The following REORG PLUS features are available only when you install the Database Performance for DB2 solution:

- 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.

- 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.

This solution includes the following products:

- DASD MANAGER PLUS for DB2
- REORG PLUS for DB2
- SNAPSHOT UPGRADE FEATURE for DB2

Tasks that REORG PLUS performs

REORG PLUS accomplishes the standard reorganization tasks and also offers the functional enhancements that this topic describes.

Table 1: REORG PLUS functional enhancements

<table>
<thead>
<tr>
<th>Functional area</th>
<th>Enhanced tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Analyzes resources for the specific reorganization</td>
</tr>
<tr>
<td></td>
<td>■ Provides data about resource requirements</td>
</tr>
<tr>
<td></td>
<td>■ Optionally offloads eligible processing to an IBM z Integrated Information Processor (zIIP)</td>
</tr>
<tr>
<td></td>
<td>■ Optionally allocates sort work files, index work files, unload data files, archive files, and copy files dynamically</td>
</tr>
</tbody>
</table>
### Functional area

<table>
<thead>
<tr>
<th>Enhanced tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Partitions</strong></td>
</tr>
<tr>
<td>- Allows select and delete operations during a partial or full reorganization</td>
</tr>
<tr>
<td>- Rebalances partitions for all access levels, including SHRLEVEL CHANGE</td>
</tr>
<tr>
<td><strong>Conditional reorganizations</strong></td>
</tr>
<tr>
<td>- Uses the traditional values that are stored in the DB2 catalog</td>
</tr>
<tr>
<td>- Optionally uses an expanded set of exceptions and user controls from the DASD MANAGER PLUS exceptions table if you are using the Performance for DB2 Databases or Database Performance for DB2 solution</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
</tr>
<tr>
<td>- Provides data for fine-tuning reorganization performance</td>
</tr>
<tr>
<td>- For an additional performance gain, offers an optional single processing phase</td>
</tr>
<tr>
<td>- Allows multitasking that is not limited by the number of CPUs</td>
</tr>
<tr>
<td><strong>Accessibility</strong></td>
</tr>
<tr>
<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>- Offers a nondestructive reorganization process, meaning that the objects that you are reorganizing remain intact throughout the reorganization</td>
</tr>
<tr>
<td>- Reclaims space that dropped tables and deleted rows occupied</td>
</tr>
<tr>
<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>- 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>Functional area</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>Rebuilding objects and data</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Copies</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Statistics and reports</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
### Differences between REORG PLUS and the IBM DB2 REORG utility

This topic describes the most important functional and operational differences between REORG PLUS and the IBM DB2 REORG utility.

Table 2 on page 36 summarizes these differences for basic REORG PLUS jobs. Table 3 on page 37 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>
<thead>
<tr>
<th>Functional area</th>
<th>Enhanced tasks</th>
</tr>
</thead>
</table>
| **Restart and recovery**| ■ Provides restart capabilities  
                         ■ Optionally recovers from an abnormal termination  
                         ■ 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 |
| **User exits**          | Provides user exit points that allow you to  
                         ■ Provide user-defined variables that you can use to construct data set name patterns for dynamically allocated files  
                         ■ Automatically resize DB2 VSAM objects  
                         ■ Reorder your storage group volumes or use a subset of them  
                         ■ Selectively specify REDEFINE NO for an object  
                         ■ Bypass the insert into BMCHIST and give you dynamic control over updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS at termination  
                         ■ Create (and drop after successful DSNUTILB completion) mapping objects for use when invoking DSNUTILB |
Table 2: Functional and operational differences between REORG PLUS and IBM DB2 REORG

<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. Note: 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>
<tr>
<td>Start/stop status</td>
<td>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.</td>
</tr>
<tr>
<td>Functional or operational area</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Multiple volumes</td>
<td>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.</td>
</tr>
<tr>
<td>EDITPROCs</td>
<td>REORG PLUS uses EDITPROCs to extract keys and update columns.</td>
</tr>
<tr>
<td>Recoverability and restartability</td>
<td>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.</td>
</tr>
<tr>
<td>Partition rebalancing</td>
<td>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.</td>
</tr>
</tbody>
</table>

Table 3: SHRLEVEL CHANGE differences between REORG PLUS and IBM DB2 REORG

<table>
<thead>
<tr>
<th>Functional or operational area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>■ 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.</td>
</tr>
<tr>
<td>Log apply process</td>
<td>■ 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.</td>
</tr>
<tr>
<td>Functional or operational area</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 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. |
| Altering execution            | You can dynamically alter execution by using the Utility Monitor function of the XBM ISPF interface. |
| Displaying status             | 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. |

**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

REORG PLUS processing phases

This section describes and illustrates the processing phases of REORG PLUS.

Table 4 on page 39 describes the REORG PLUS processing phases. The figures in this section 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 42</td>
</tr>
<tr>
<td>Two-phase index</td>
<td>Figure 2 on page 43</td>
</tr>
<tr>
<td>Single-phase table space</td>
<td>Figure 3 on page 44</td>
</tr>
<tr>
<td>Single-phase index space</td>
<td>Figure 4 on page 45</td>
</tr>
<tr>
<td>Invoking DSNUTILB</td>
<td>Figure 5 on page 46</td>
</tr>
</tbody>
</table>

The figures for the various types of SHRLEVEL CHANGE reorganizations are located in the separate chapter that describes SHRLEVEL CHANGE reorganizations.

Table 4: REORG PLUS processing phases

<table>
<thead>
<tr>
<th>Phase name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILINIT</td>
<td>During this phase, REORG PLUS completes the following processes:</td>
</tr>
<tr>
<td></td>
<td>■ Initializes the job</td>
</tr>
<tr>
<td></td>
<td>■ Performs DB2 catalog lookup</td>
</tr>
<tr>
<td></td>
<td>■ Reads, parses, and verifies the REORG command and the IDCAMS control statements in the SYSIDCIN data set</td>
</tr>
<tr>
<td>DSNUTILB</td>
<td>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.</td>
</tr>
<tr>
<td>Phase name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 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  
<p>|           | This phase runs only if you specify SHRLEVEL CHANGE. For more information about the LOGAPPLY phase, see the chapter that describes SHRLEVEL CHANGE reorganizations. |</p>
<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 the chapter that describes SHRLEVEL CHANGE reorganizations. |
| **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. |
Figure 1: Two-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 UNLOAD phase updates the BMCDICT table. The UTILTERM phase updates the BMCHIST table.
Figure 2: Two-phase index reorganization

How REORG PLUS works

Chapter 1 Introduction to REORG PLUS 43
Figure 3: Single-phase table space reorganization

How REORG PLUS works

All phases use the SYSPRINT data set and update the BMGUTIL 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.

The SYSREC data set is optional for a single-phase table space reorganization, but is required for restarting the job. REORG PLUS does not use the SYSREC data set for a single-phase SHRLEVEL REFERENCE or SHRLEVEL CHANGE table space reorganization.
Figure 4: Single-phase index reorganization

How REORG PLUS works

Chapter 1  Introduction to REORG PLUS  45
Figure 5: Processing phases when invoking DSNUTILB

This topic provides a brief overview of the data sets that REORG PLUS uses. The following table 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 detailed information about each data set, see the references listed in this table.
### Table 5: REORG PLUS data sets

<table>
<thead>
<tr>
<th>Data set or ddname</th>
<th>Description</th>
<th>Reference</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>“ASUSRPRT data sets” on page 335</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>“BMCFORCE data sets” on page 336</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>“DDLIN data sets in REORG PLUS” on page 344</td>
</tr>
<tr>
<td>DDLOUT</td>
<td>The DDLOUT output data set contains the SQL ALTER statements for the limit-key updates that REORG PLUS makes during partition rebalancing. REORG PLUS opens this optional data set only if altering limit keys.</td>
<td>“DDLOUT data sets in REORG PLUS” on page 349</td>
</tr>
<tr>
<td>Full copy data sets</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>“Copy data sets in REORG PLUS” on page 336</td>
</tr>
<tr>
<td>Incremental copy data sets</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>“Copy data sets in REORG PLUS” on page 336, “Copy data sets for SHRLEVEL CHANGE” on page 601</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>“SORTWK data sets in REORG PLUS” on page 349</td>
</tr>
<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>“SYSARC data sets in REORG PLUS” on page 353</td>
</tr>
</tbody>
</table>
### Data set or ddname Description Reference

<table>
<thead>
<tr>
<th>Data set or ddname</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSERR a</td>
<td>This data set holds diagnostic messages that REORG PLUS might produce in error situations when running a SHRLEVEL CHANGE reorganization.</td>
<td>“SYSERR data sets in REORG PLUS” on page 356</td>
</tr>
<tr>
<td>SYSEXEC</td>
<td>SYSEXEC specifies the library concatenation where REXX exits reside.</td>
<td>“SYSEXEC data sets in REORG PLUS” on page 357</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>“SYSIDCIN data sets in REORG PLUS” on page 357</td>
</tr>
<tr>
<td>SYSIN</td>
<td>SYSIN is the input data set that contains the REORG command.</td>
<td>“SYSIN data sets” on page 360</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>“SYSPRIN2 data sets” on page 361</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>The SYSPRINT output data set contains REORG PLUS messages.</td>
<td>“SYSPRINT data sets” on page 360</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>“SYSPUNCH data sets in REORG PLUS” on page 361</td>
</tr>
<tr>
<td>SYSRECnn a</td>
<td>The SYSREC data set contains the table space’s unloaded rows. If you use multiple data sets, the nn 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>“SYSREC data sets in REORG PLUS” on page 362</td>
</tr>
<tr>
<td>SYSTSPRT</td>
<td>REXX routes all output from the REXX 'SAY' statements to the SYSTSPRT data set.</td>
<td>“SYSTSPRT data sets in REORG PLUS” on page 366</td>
</tr>
<tr>
<td>SYSUT1nn a</td>
<td>The SYSUT1 data set contains the unloaded index keys. The nn 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>“SYSUT1 data sets in REORG PLUS” on page 366</td>
</tr>
</tbody>
</table>
Data set or ddname | Description | Reference
--- | --- | ---
UTPRINT | The UTILITY data set indicates that sort messages should be reported. However, the actual messages for each sort appear in separate SYSnnnn data sets, where nnnnn is a system-assigned sequential number. | “UTPRINT data sets” on page 369

a 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 that this topic describes.

### 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. 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. 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. 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. This component is installed during REORG PLUS installation, but is maintained separately from REORG PLUS.</td>
</tr>
</tbody>
</table>
## Component

| 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 in REORG PLUS

This chapter provides procedures and information to consider when using REORG PLUS.

System setup for REORG PLUS

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

DB2 support in REORG PLUS

This version of REORG PLUS supports IBM DB2 Versions 10 (new-function mode only) and 11.

REORG PLUS for DB2 system and software requirements

The REORG PLUS product has the following requirements:

- This version of REORG PLUS has the following system requirements:
  - IBM System z10 processor or a later 64-bit z/Architecture processor
  - z/OS Version 1.10 or later

- REORG PLUS requires a minimum of the following versions of BMC common components:
  - Version 2.4.01 of BMCSORT
  - Version 11.2.00 of the DB2 Utilities Common Code (D2U)
  - Version 11.1.00 with PTF BPJ0944 of the DB2 Solution Common Code (SCC)
To update DB2 catalog statistics or the DASD MANAGER PLUS for DB2 database statistics tables, REORG PLUS requires a minimum of version 11.2.00 of the BMC Common Statistics component.

To use the Instant Snapshot technology, REORG PLUS requires version 6.1.00 or later of either 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 an IBM z Integrated Information Processor (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, you must be licensed to use the IBM DB2 REORG utility.

**Required authorization for REORG PLUS**

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 for the Utility products**

The products use one of the following mechanisms to verify authorization:

- If the DB2 DSNX@XAC authorization exit is available for your system, the product 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 the CA-ACF2 and CA-Top Secret products.

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, the product uses the standard DB2 method to check security.

**Setting REORG PLUS authorizations**

REORG PLUS does not run as part of the DB2 subsystem. Therefore, users must have system and data set authorizations that are equivalent to the authorizations that DB2 requires. Use the following procedures to set the necessary authorizations.

**To set DB2 authorizations**

1. For all reorganization jobs, grant the following 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

   *Note*
   REORG PLUS does not check for the DELETE privilege when the SELECT/DELETE option is used. REORG PLUS does not check for the UPDATE privilege when the UPDATE option is used.

2. To enable running a SHRLEVEL CHANGE reorganization, also grant the following additional authorities:
   - TRACE authority
   - MONITOR2 authority
   - DISPLAY authority (if not already granted to PUBLIC)

   *Note*
   These privileges might be implicit in the authority that the users have.

3. To enable reorganizing base table spaces that contain XML columns, also grant SELECT privileges on the following DB2 tables:
4 To enable reorganizing user-defined XML indexes, also grant SELECT privileges on the SYSIBM.SYXMLRELS DB2 table.

5 To enable using the DSRSEXIT user exit to update the DB2 catalog (in other words, the DSRSEXIT user exit has a default of YES for the BMC_ALTER_DB2_CATALOG variable), also complete the following steps:

a For the ALTER TABLESPACE statement, grant one of the following privileges if the user is not the owner of the table space:

- DBADM authority for the database that contains the table
- SYSADM or SYSCTRL authority
- System DBADM

b For the ALTER INDEX or ALTER TABLE statement, grant one of the following privileges if the user is not the owner 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

6 To enable using the MAPTEXIT user exit, also grant the authority to create and drop objects on the DSNDB04 database.

7 To enable the use of the FORCE option to cancel DB2 threads that might prevent a successful drain during a reorganization job, also grant the following authorizations:

- DISPLAY privileges
- One of the following authorities:
Note
These authorizations might be implicit in the authority that the users have.

8 To enable use of the EXTENDED BUFFER MANAGER (XBM) product or SNAPSHOT UPGRADE FEATURE (SUF) component of XBM, ensure that you have the appropriate authorizations for XBM or SUF.

For information about security levels and authorizations for XBM, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

To enable data set access using the DB2 RACF ID

1 Specify OPNDB2ID=YES in your installation options.

This option tells REORG PLUS to use the DB2 RACF ID for data set access.

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.

To enable data set access when not using the DB2 RACF ID

1 Specify OPNDB2ID=NO in your installation options.

This option tells REORG PLUS not to use the DB2 RACF ID for data set access.

2 If using 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, grant a minimum of the following levels of authorization:

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

The following steps illustrate 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 data set authorizations to ARUUMAIN.

3 *To enable using rename or FASTSWITCH processing*, if you establish authority at a node lower than the highest node, grant the same privileges as described in Step 2 on page 55 for the following data sets:

- For STAGEDSN=BMC:
  - VCAT.BMCDBD.database.object.I0001
  - VCAT.BMCDBC.database.object.I0001
  - VCAT.OLDDBD.database.object.I0001
  - VCAT.OLDDBC.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.BMCDBD.database.object.I0001
  - VCAT.BMCDBC.database.object.I0001
  - VCAT.BMCDBD.database.object.J0001
  - VCAT.BMCDBC.database.object.J0001
  - VCAT.DSNDBD.database.object.S0001
  - VCAT.DSNDBC.database.object.S0001

**Overriding the MEMLIMIT system parameter**

The utility 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.

**To override the default MEMLIMIT value**

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

- *(recommended)* Specify REGION=0M in the JCL.

- Specify the one of the following values for the MEMLIMIT parameter in the JCL, or in an SMF IEFUSI exit:
  - Specify NOLIMIT to allow unlimited above-the-bar memory.
—If you are unable to specify NOLIMIT, specify at least 4 GB. If you are operating on 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.

**REORG PLUS features that require DSNUTILB**

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

*Note*

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 base table spaces or LOB table spaces when pending definition changes exist on them
- 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 “Reorganizing LOB data” on page 104.

## 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 certain pending definition changes
  For most pending definition changes, REORG PLUS materializes the change and reorganizes the object natively. For more information, see “Reorganizing objects with pending definition changes” on page 98.

- 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 137.
  - Indexes that contain keys with random ordering
  - Indexes that are defined with EXCLUDE NULL KEYS
  - Nonpartitioned indexes in which some, but not all, logical partitions are in RREPL status.

### Additional types of indexes

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

- Indexes that contain certain pending definition changes
  For most pending definition changes, REORG PLUS materializes the change and reorganizes the object natively. For more information, see “Reorganizing objects with pending definition changes” on page 98.

- 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 137.

- Indexes that contain keys with random ordering

- Nonpartitioned indexes in which some, but not all, logical partitions are in RREPL status.

### Additional types of reorganizations

REORG PLUS invokes DSNUTILB when any of the following conditions exists:
An object is in RREPL status and SHRLEVEL NONE or SHRLEVEL CHANGE is in effect.

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 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

This task describes how to enable REORG PLUS to invoke DSNUTILB when required.

To enable REORG PLUS to invoke DSNUTILB

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

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Required setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUTILB</td>
<td>YES</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>YES for the following DDTYPES:</td>
</tr>
<tr>
<td></td>
<td>■ All work file DDTYPES that the reorganization job requires</td>
</tr>
<tr>
<td></td>
<td>■ (if you specify COPY YES) The LOCPFCPY DDTYPE and other copy DDTYPES if needed</td>
</tr>
<tr>
<td>SORTNUM</td>
<td>2 or greater</td>
</tr>
</tbody>
</table>
For a SHRLEVEL CHANGE table space reorganization, ensure that a mapping table or database is specified and available by completing one of the following actions:

- Specify an existing mapping table with the MAPPINGTABLE command option.
- Provide a user exit that will create and drop the mapping objects for you; then, supply the user exit name in the MAPTEXIT installation or command option. For more information about the MAPTEXIT user exit, see “Using MAPTEXIT to create mapping objects” on page 808.
- (DB2 Version 11 or later only) Specify an existing mapping database with the MAPPINGDATABASE command option.

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.

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, certain options will cause your job to fail when invoking DSNUTILB.

General restrictions for DSNUTILB reorganizations

The following general restrictions apply 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.

REORG PLUS terminates if you are using a DDLIN data set to rebalance partitions.

## Data set allocation for DSNUTILB reorganizations

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

For all reorganizations, you must enable dynamic allocation for the following data sets:

- 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 the following table, 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 7: Dynamic allocation options considerations for a DSNUTILB reorganization

<table>
<thead>
<tr>
<th>Command option</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>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.</td>
</tr>
<tr>
<td>Command option</td>
<td>Considerations</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>
| ALLOC          | Ignores this option  
DSNUTILB dynamically allocates SORTWK DD names. |
| AVGVOLSP       | Ignores this option |
| DSNPAT         | Converts symbolic variables when an equivalent exists  
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 54 on page 299.  
Also, be aware that user-defined variables are not valid for a DSNUTILB reorganization. |
| DSNTYPE        | Ignores this option |
| IFALLOC        | Ignores this option  
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. |
| MAXEXTSZ       | Translates this option to the MAXPRIME keyword of the TEMPLATE control statement |
| SIZEPCT        | Translates this option to the PCTPRIME keyword of the TEMPLATE control statement  
If you specify a value greater than 100, REORG PLUS converts it to 100. |
| THRESHLD       | For copy data sets, translates this option to a LIMIT value in the TEMPLATE control statement  
REORG PLUS also builds the secondary TEMPLATE control statement to which the DB2 REORG utility will switch when the LIMIT value is exceeded.  
For all other data sets, ignores this option |

### Considerations for other REORG PLUS options when running DSNUTILB reorganizations

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

When an option is not valid for the IBM DB2 REORG utility, REORG PLUS responds in one of the following ways:

- Ignores the option
- Terminates with a message about the option
- Translates the option to a valid DB2 REORG option and passes the translated option

The following table 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 62.

Table 8: Options that are incompatible or translated for a DSNUTILB reorganization

<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>DSNEXIT</td>
<td>Ignores this option</td>
</tr>
<tr>
<td>DSPLOCKS</td>
<td>Ignores this option</td>
</tr>
<tr>
<td>DSREXIT</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>IDCACHE</td>
<td>Ignores this option</td>
</tr>
<tr>
<td>IDCDDN</td>
<td>Ignores this option</td>
</tr>
<tr>
<td>ICTYPE INCREMENTAL</td>
<td>If COPY YES, converts to ICTYPE UPDATE</td>
</tr>
<tr>
<td>ICDDN</td>
<td>Ignores this option</td>
</tr>
</tbody>
</table>

Reorganization jobs that invoke DSNUTILB
<table>
<thead>
<tr>
<th>Command option</th>
<th>REORG PLUS response if you include the option</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPLICIT_TZ</td>
<td>Ignores this option</td>
</tr>
<tr>
<td>INLINE NO</td>
<td>Ignores this option</td>
</tr>
<tr>
<td>KEEPDICIONARY</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 SHRLEVEL CHANGE, ignores this option</td>
</tr>
<tr>
<td></td>
<td>• Otherwise, passes as KEEPDICIONARY</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>
<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>For a LOB table space, ignores this option</td>
</tr>
<tr>
<td></td>
<td>For all other objects:</td>
</tr>
<tr>
<td></td>
<td>• No: Ignores this option</td>
</tr>
<tr>
<td></td>
<td>• Yes: Passes as PREFORMAT</td>
</tr>
<tr>
<td>REBALANCE</td>
<td>For partition-by-growth and nonpartitioned table spaces, ignores this option</td>
</tr>
<tr>
<td></td>
<td>For all other table spaces, passes as it is</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>
</tbody>
</table>
### REORG PLUS 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.

<table>
<thead>
<tr>
<th>Command option</th>
<th>REORG PLUS response if you include the option</th>
</tr>
</thead>
<tbody>
<tr>
<td>REDEFINE YES</td>
<td>Passes as REUSE</td>
</tr>
<tr>
<td>REGISTER</td>
<td>Ignores this option</td>
</tr>
<tr>
<td>REUSE</td>
<td>For a LOB table space, ignores this option</td>
</tr>
<tr>
<td></td>
<td>For all other objects, passes as it is</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 UNLOADONLY</td>
<td>Issues message BMC50178E and terminates</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>SPILLDSNPAT</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>
<tr>
<td>XBMID</td>
<td>Ignores this option</td>
</tr>
<tr>
<td>ZIIP</td>
<td>Ignores this option</td>
</tr>
</tbody>
</table>
Executing BMC utilities concurrently

All BMC Software utility products use the BMCSYNC and BMCUTIL tables to control concurrent access to DB2 objects.

The 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 736.

**Note**
The setting of the LOCKROW installation option determines whether the utility uses MVS enqueues or SQL LOCK TABLE statements to serialize the BMCSYNC and BMCUTIL tables. For information about the LOCKROW option, see the installation options descriptions for the utility.

Shared access levels of BMC utilities

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.

**Note**
All BMC 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 736.

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 the following table. For additional serialization and concurrency issues for each utility, see that utility’s reference manual.

The "Access level" column in the following table refers to the value of the **SHRLEVEL** column name in the BMCSYNC table (“BMCSYNC table” on page 726). 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: Shared access levels of BMC utilities

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

**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.

**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.
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 10: 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>
<tr>
<td>All others</td>
<td>One of the following statuses:</td>
</tr>
<tr>
<td></td>
<td>• RW</td>
</tr>
<tr>
<td></td>
<td>• Read-only (RO)</td>
</tr>
</tbody>
</table>

The objects that you are reorganizing can be in any status except those that the following table 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.

**Note**

If an object is in RREPL status, REORG PLUS invokes DSNUTILB.

**Table 11: Restrictive statuses that are not permitted for SHRLEVEL NONE**

<table>
<thead>
<tr>
<th>Status that is not permitted</th>
<th>Full table space reorganization fails</th>
<th>Partial table space reorganization fails</th>
<th>Index reorganization fails</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREST (advisory restart pending)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DBETE (DBET error)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GRECP (group RECOVER pending)</td>
<td>X&lt;sup&gt;a&lt;/sup&gt;</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&lt;sup&gt;b&lt;/sup&gt;</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PSRBD (page set REBUILD pending)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Status that is not permitted</td>
<td>Full table space reorganization fails</td>
<td>Partial table space reorganization fails</td>
<td>Index reorganization fails</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------------</td>
<td>-----------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>RBDP (REBUILD pending)</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RBDP* (logical part REBUILD pending)</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RECP (RECOVER pending)</td>
<td>X&lt;sup&gt;a&lt;/sup&gt;</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>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>
<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>

<sup>a</sup> The table space cannot be in this status, but the associated indexes can be.

<sup>b</sup> The reorganization terminates if one or more of the partitions being reorganized is in this status.

### Status changes for SHRLEVEL NONE

Table 12 on page 71 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.

### Table 12: Status changes during a SHRLEVEL NONE 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>Phase</td>
<td>Status</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Beginning of UNLOAD (two-phase reorganization) or beginning of REORG (single-phase reorganization) | - For an index reorganization, REORG PLUS stops the index space.  
  - For a table reorganization, REORG PLUS stops the table space and all associated index spaces.  
  STOP status ensures that REORG PLUS has exclusive use of the space that it is reorganizing.  
  REORG PLUS sets RECOVER pending status before reloading and building the DB2 objects that are involved in the reorganization. |
| End of RELOAD (two-phase reorganization) or end of REORG (single-phase reorganization) | - REORG PLUS resets the RECOVER pending status for each object as the object’s reload/build task completes. REORG PLUS then starts the table space and index spaces in the status they had when reorganization began.  
  - If you specify COPY YES, REORG PLUS resets the COPY pending status of the table space. If you specify COPY NO, REORG PLUS sets the COPY pending status for the table space.  
  - REORG PLUS resets REORG pending status for each object that was in REORG pending status before the reorganization. |

Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE

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 13: Database initial status requirements for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE

<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>
<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:</td>
</tr>
<tr>
<td></td>
<td>■ RW</td>
</tr>
<tr>
<td></td>
<td>■ Read-only (RO)</td>
</tr>
</tbody>
</table>

The objects that you are reorganizing can be in any status except those that Table 14 on page 74 describes. If the object is in a status that REORG PLUS does not permit, REORG PLUS terminates. An X in a column in this table 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.

**Note**

If an object is in RREPL status, REORG PLUS functions as follows:

- If you specify SHRLEVEL CHANGE, REORG PLUS invokes DSNUTILB.

- If you specify SHRLEVEL REFERENCE, REORG PLUS functions normally in most cases. However, if you specify SELECT, UPDATE, or DELETE for a partition that is in RREPL status, REORG PLUS terminates.
### Statuses that are not permitted for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE

<table>
<thead>
<tr>
<th>Status that is not permitted</th>
<th>SHRLEVEL REFERENCE UNLOADONLY reorganization fails</th>
<th>SHRLEVEL REFERENCE reorganization fails</th>
<th>SHRLEVEL CHANGE reorganization fails</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></td>
<td>X</td>
</tr>
<tr>
<td>PSRBD (page set REBUILD pending)</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RBDP (REBUILD pending)</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RBDP* (logical part REBUILD pending)</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RECP (RECOVER pending)</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>REFP (refresh pending)</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RESTP (restart pending)</td>
<td>X</td>
<td></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></td>
<td>X</td>
</tr>
<tr>
<td>UTOR (utility restrictive state, read-only access allowed)</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>UTRW (utility restrictive state, read/write access allowed)</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>UTUT (utility restrictive state, utility exclusive control)</td>
<td></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 15 on page 75 through Table 17 on page 76 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 15: 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>
<tr>
<td>Beginning of RELOAD</td>
<td>- REORG PLUS stops the table space and associated indexes (for a table space reorganization) or the index space (for an index reorganization).</td>
</tr>
<tr>
<td></td>
<td>- Processing continues in the RELOAD and UTILTERM phases as with SHRLEVEL NONE.</td>
</tr>
</tbody>
</table>

Table 16: 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>
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.

REORG PLUS resets REORG pending status for each object that was in REORG pending status before the reorganization.

If materializing pending definition changes during the reorganization, REORG PLUS resets advisory REORG-pending (AREOR) status.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Status</th>
</tr>
</thead>
</table>
| 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.
REORG PLUS resets REORG pending status for each object that was in REORG pending status before the reorganization.
If materializing pending definition changes during the reorganization, REORG PLUS resets advisory REORG-pending (AREOR) status. |

**Table 17: 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>
</tbody>
</table>
| Beginning of LOGFINAL  | REORG PLUS prevents updates to the affected table and index spaces while applying the last of the log records. ^a  
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. |
### Phase | Status
--- | ---
**UTILTERM** | - REORG PLUS prevents any access to the objects.  
- 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).  
- 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.  
- REORG PLUS resets REORG pending status for each object that was in REORG pending status before the reorganization.  
- If materializing pending definition changes during the reorganization, REORG PLUS resets advisory REORG-pending (AREOR) status.

---

**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.

---

**Dynamic data set allocation in REORG PLUS**

When dynamic allocation is active, REORG PLUS calculates the optimal size and number of data sets and allocates them for you. REORG PLUS also optionally deletes the work files.

Dynamic allocation reduces or eliminates the need to specify DD statements for these files in your JCL. 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.
You activate data set allocation through command syntax or installation option defaults.

**Note**

When REORG PLUS invokes DSNUTILB, you must enable dynamic allocation for the required data sets. You can use some of the options described in this section to control dynamic allocation of these data sets. However, DSNUTILB handles the allocation, so the allocation process is different than this section describes.

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 332.

**Enabling dynamic allocation in REORG PLUS**

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 for REORG PLUS” on page 277.
Using dynamic allocation in a worklist environment

When running in a worklist environment, the utility ignores the ACTIVE option in your installation options module. The utility 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 in REORG PLUS

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 “REORG PLUS user exits” on page 747 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.)

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 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.

### Specifying ddname prefixes

If you specify more than one ddname prefix for dynamic allocation, the prefix for
each ddname must be different enough for the product 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:

<table>
<thead>
<tr>
<th>BMCRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCRD11</td>
</tr>
</tbody>
</table>

The prefixes that you specify must allow the utility 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, the utility pads the data set or partition number with leading zeros. The following table shows an example of how the prefixes that you specify resolve to the generated ddnames.

Table 18: Example of specifying ddname prefixes

<table>
<thead>
<tr>
<th>Prefix specified for SYSUT1 data sets</th>
<th>Number of data sets</th>
<th>Generated ddnames</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSOUT1</td>
<td>9</td>
<td>SYSOUT1...SYSOUT9</td>
</tr>
<tr>
<td>SYSOUT1</td>
<td>10</td>
<td>None (invalid length)</td>
</tr>
<tr>
<td>WORK</td>
<td>10</td>
<td>WORK0001...WORK0010</td>
</tr>
</tbody>
</table>

Deleting dynamically allocated data sets in REORG PLUS

To delete dynamically allocated data sets, specify DELETEFILES YES on your REORG command. You can also specify this preference with the DELFILES installation option.

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.

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 the utility 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 the utility 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, see the THRESHLD option description.

## 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, you can use the SMSUNIT option in the utility to affect that use.

The value of the SMSUNIT option affects the UNIT value as follows:

- When you specify SMSUNIT YES, the utility 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, the utility does not pass the UNIT option.

## Reaching the MAXTAPE limit during dynamic allocation

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

1. The utility 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, the utility 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 the utility 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, the utility 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 in REORG PLUS

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 in REORG PLUS

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 in REORG PLUS
The SHRLEVEL option specifies the level of access that DB2 has to the target spaces during REORG PLUS processing.

REORG PLUS provides the following SHRLEVEL options:

- 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 162.

**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 considerations**

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 considerations**

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 72.

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

- Single-phase reorganization
- DSNUTILB reorganization
- Reorganization of catalog objects
SHRLEVEL REFERENCE considerations

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 72.

SHRLEVEL REFERENCE uses staging data sets as described in “Staging data sets” on page 87.

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

SHRLEVEL CHANGE considerations

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 most of the 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 72.

SHRLEVEL CHANGE uses staging data sets as described in “Staging data sets” on page 87.

For more information about running a SHRLEVEL CHANGE reorganization, see “Online reorganization” on page 565.

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.

- In the FASTSWITCH process, REORG PLUS bypasses the VSAM rename process and changes the DB2 catalog entry to point to the staging data sets.
  
  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 207.

- 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 208.

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 the following table 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 BMCDDBD. The following table 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 DSNDBC (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. The following table illustrates this naming convention.

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

- If STAGEDSN=BMC, REORG PLUS replaces DSN in the DSNDBC 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

- 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.

   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 92 illustrates this 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 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.

The following table illustrates this naming convention (which is equivalent to specifying STAGEDSN=DSN).
Table 22: Naming convention for the FASTSWITCH process

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

**Data set FASTSWITCH process**

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 94 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.
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 “Managing a reorganization that does not complete in the UTILTERM phase” on page 384.

Reorganizing a 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.

- 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.

- Consider specifying REDEFINE NO, either in the installation options or on the REORG command. This option minimizes the time that REORG PLUS requires to delete and redefine the existing VSAM data sets for the table space or indexes.

- Ensure that the value of your LOCKROW installation option is YES.

- Consider using VOLCNT (AUTO,AUTO) rather than the default of 25 to avoid data set allocation limitations when 25 is excessive.
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.
Reorganizing DEFINE NO objects

You can reorganize an object that was defined with the DEFINE NO attribute. However, REORG PLUS does not support certain conditions.

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.

Reorganizing multi-data-set DB2 objects

You can reorganize multi-data-set DB2 objects. However, the following requirements and considerations apply when reorganizing these 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 112.
You can use the DSRSEXIT user exit to redefine a multi-data-set object. For more information, see “Using DSRSEXIT to manage VSAM data set redefinition” on page 787.

**Reorganizing objects with pending definition changes**

REORG PLUS natively materializes pending definition changes during the reorganization in most cases.

REORG PLUS invokes DSNUTILB when any of the following pending changes exists on the object:

- A pending DROP COLUMN request
- A pending change that would convert the table space type (for example from a simple table space to a partition-by-growth table space)
  For information about pending table space type conversions that the IBM DB2 REORG utility supports, see the IBM documentation.
- Any pending changes on a LOB table space or on a base table space that contains a LOB column
- A pending limit key change when all of the following conditions exist:
  — You are reorganizing more than one but not all partitions of a table space.
  — The pending limit key change is the only pending change.
- A pending MAXPARTITIONS change when either of the following conditions exists:
  — Pending changes existed on the object before the MAXPARTITIONS change was requested.
  — The MAXPARTITIONS change is not the first change in a particular request.

To avoid invoking DSNUTILB, BMC recommends that you alter MAXPARTITIONS before any other alters on the object that would result in pending changes.
**Requirements**

If the last partition in the table space is included in a pending ALTER of a limit key, you must include a SYSARC data set in your job. For more information about this data set, see “SYSARC data sets in REORG PLUS” on page 353.

**Restrictions**

The following restrictions apply to pending definition changes:

- REORG PLUS does not materialize pending definition changes but reorganizes the objects when any of the following conditions exists:
  - You are running a partial reorganization.
    - If the only pending change on the table space is a limit key change and you are reorganizing more than one partition, REORG PLUS invokes DSNUTILB.
  - You specify SHRLEVEL NONE or FASTSWITCH NO.
  - You are reorganizing an index that has pending definition changes, but pending definition changes also exist on the associated table space.

- REORG PLUS terminates when any of the following conditions exists:
  - The values for the pending DSSIZE or SEGSIZE definitions are insufficient for the existing user data.
  - The pending DSSIZE value is greater than 4 GB, but the device is not EA-enabled.
  - An alter occurs during the reorganization that creates a pending definition change.
    - In this case, you cannot restart the reorganization. Objects are left in their original status.
  - A pending alter exists on a limit key, and ORDER NO and UNLOAD RELOAD are in effect.

- REORG PLUS changes or ignores the following command installation options:

<table>
<thead>
<tr>
<th>Option</th>
<th>REORG PLUS action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPYRFAIL=TERM</td>
<td>Changes to COPYPEND</td>
</tr>
<tr>
<td>FSFALLBACK</td>
<td>Ignores</td>
</tr>
<tr>
<td>FSTHRESHOLD</td>
<td>Ignores</td>
</tr>
</tbody>
</table>
Option | REORG PLUS action
--- | ---
KEEPDICTIONARY YES | Changes to NO when any of the following definition changes are pending on the table space:
- An alter of the table space buffer pool or segment size
- A table alter that changes the limit key

UPDATEDB2STATS NO | Changes to YES
REORG PLUS always updates DB2 statistics after materializing pending definition changes. This restriction also applies if you have a TERMEXIT user exit that changes UPDATEDB2STATS to NO.

**Additional considerations**

When multiple limit key change requests exist for a reorganization, REORG PLUS materializes only the first changes that it finds based on the following hierarchy:

- Pending limit key alters on the object
- ALTER statements in a DDLIN data set included in your JCL
- Limit key changes as a result of the REBALANCE option specified on your REORG command

**Reorganizing catalog objects**

REORG PLUS lets you reorganize DB2 catalog (database DSNDB06) table spaces and indexes.

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

REORG PLUS reorganizes all other catalog objects 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 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 57.

**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. The following table describes how REORG PLUS responds if these options are in effect.

<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</td>
<td>Changes this option to PREFORMAT NO</td>
</tr>
<tr>
<td>PREFORMAT YES</td>
<td></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.
Reorganizing XML data

You can use REORG PLUS to reorganize either a base table space that contains an XML column or an XML table space.

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.SYSSEQUENCES 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.SYSSEQUENCES 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.

Authorizations

For authorization requirements that apply when reorganizing tables that contain an
XML column, see “Setting REORG PLUS authorizations” on page 53

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 “Setting REORG PLUS authorizations”
on page 53.

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 “Reorganizing partition-by-growth table spaces” on page 113 for additional considerations.

**Reorganizing 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.
However, 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.
- Pending definition changes exist on the 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 57.

   If you do not supply the correct options or environment for invoking DSNUTILB, REORG PLUS issues BMC50178E and terminates.

2. For either of the following conditions, specify AUX YES on the REORG command:
   - An inline LOB column was altered before the reorganization
   - You want to reorganize the base table space and LOB table space in the same job

   For more information about this option, see “AUX” on page 244.

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

1. 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 106.
Considerations when reorganizing the base table space for LOB data

When reorganizing the base table space for LOB data, REORG PLUS does not support certain conditions.

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 as follows:

- For SHRLEVEL REFERENCE, REORG PLUS natively reorganizes the LOB table space and updates the auxiliary index.
- REORG PLUS invokes DSNUTILB when any of the following conditions exists:
  - You specify SHRLEVEL NONE or SHRLEVEL CHANGE.
  - Pending definition changes exist on the table space.
  - You are reorganizing both the LOB table space and base table space in the same job step.

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 57.

If you do not supply the correct options or environment for invoking DSNUTILB, REORG PLUS issues BMC50178E and terminates.

2 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 “AUX” on page 244.

To reorganize LOB table spaces when SHRLEVEL REFERENCE is in effect

1 Create your job as usual, making any adjustments as needed based on the information in “Considerations when reorganizing LOB table spaces (SHRLEVEL REFERENCE)” on page 107.

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. The following table 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></td>
</tr>
<tr>
<td>ANALYZE SAMPLE</td>
<td></td>
</tr>
<tr>
<td>ANALYZE SCAN</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>CONDEXEC BMC</td>
<td>Issues message BMC50056E and terminates</td>
</tr>
<tr>
<td>CONDEXEC YES</td>
<td></td>
</tr>
<tr>
<td>DELETE</td>
<td>Issues message and terminates</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 and terminates</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Issues message and terminates</td>
</tr>
</tbody>
</table>

Floating-point data in REORG PLUS

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

The following general considerations apply 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 time stamp 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.

- When pending definition changes exist on the object, REORG PLUS functions as follows:
  - For pending limit key changes when both of the following conditions exist, REORG PLUS invokes DSNUTILB:
    - You are reorganizing more than one but not all partitions of a table space.
    - The pending limit key change is the only pending change.
  - For all other pending changes, REORG PLUS completes the reorganization but does not materialize the changes.
SHRLEVEL NONE considerations for a partial reorganization

In addition to the general considerations for a partial reorganization, the following considerations apply 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 for a partial reorganization, the following considerations apply when you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE:

- 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.
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 a supported intelligent storage device. For a list of supported devices, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

The following table 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 112.

For information about the XBM and SUF requirements, see “XBM and SUF considerations for REORG PLUS” on page 140.

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 “REORG PLUS for DB2 system and software requirements” on page 51.

2. Ensure that you have a supported intelligent storage device (as documented in the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide).

3. Specify AUTO or YES on the SIXSNAP command or installation option.

4. Specify YES on the REDEFINE command or installation option.

If the value for REDEFINE is NO, REORG PLUS operates as follows:
If SIXSNAP is AUTO, REORG PLUS changes SIXSNAP to NO and produces a software-based copy.

If SIXSNAP is YES, REORG PLUS terminates.

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

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

Considerations for SIXSNAP use

The following considerations apply when using the SIXSNAP option.

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.

Reorganizing partition-by-growth table spaces
You can use REORG PLUS to perform both full and partial reorganizations of partition-by-growth table spaces.

General considerations for partition-by-growth table spaces
The following general considerations apply to reorganizing 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.
If an online SQL UPDATE occurs during a SHRLEVEL CHANGE reorganization and that update causes the updated row to be relocated to another partition, REORG PLUS issues message BMC53099E and terminates.

In addition to the information in the rest of this section, see the following sections for additional partition-by-growth considerations:

- “Building a dictionary” on page 135
- “Copy data sets in REORG PLUS” on page 336
- “SYSREC data sets in REORG PLUS” on page 362
- “Additional restart considerations and restrictions” on page 375

**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.
- The value of the DB2 MAXPARTITIONS parameter is greater than the number of partitions in the table space.
- The value of the MAXNEWPARTS option 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 342.
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 in most cases. However, REORG PLUS terminates when a partition is added when either of the following conditions exists:

- The number of partitions that DB2 adds exceeds MAXNEWPARTS. REORG PLUS issues message BMC53025E.
- You are reorganizing a LOB base table space (with AUX NO). REORG PLUS issues message BMC51287E.

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 342.

Partition rebalancing in REORG PLUS

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 restrictions and considerations” on page 120.
  - 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 169.
Rebalancing partitions in REORG PLUS

Use the following procedure to rebalance partitions in REORG PLUS.

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 for the limit-key updates that REORG PLUS makes.

4. Specify COPY YES on the REORG command.

5. 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.

6. 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 multiple limit key change requests exist for a reorganization, REORG PLUS materializes only the first changes that it finds based on the following hierarchy:

- Pending limit key alters on the object
- ALTER statements in a DDLIN data set included in your JCL
- Limit key changes as a result of the REBALANCE option specified on your REORG command

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 in REORG PLUS” on page 349.

REORG PLUS changes the following options:

- CPYRFAIL=TERM to COPYPEND
- FSFALLBACK YES to NO

When a failure occurs while processing limit key updates, REORG PLUS always terminates and DB2 rolls back the limit key updates. Depending on other conditions of the job, REORG PLUS completes one of the following processes:

- Backs out any other changes made by the reorganization
- Abends, leaving the object in RECP status

**Note**

The ALTRFAIL installation option is obsolete as of REORG PLUS version 11.2.

---

**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 102)

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

When rebalancing partitions, REORG PLUS treats each partition that is participating in the reorganization as if it were in REORP status.
The following considerations apply when rebalancing partitions and objects are in REORP status before the reorganization:

- 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 and issues message BMC50138I. If the image copies cannot be registered, REORG PLUS terminates. The value change ensures that REORG PLUS can return the table space to its original state.

- REORG PLUS resets REORP for each object that was in REORP status before the reorganization.

**DDLIN data set restrictions and considerations**

When you use a DDLIN data set to rebalance partitions, 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.

The following restrictions and considerations apply to using a DDLIN data set to rebalance partitions:

**Restrictions**

- REORG PLUS terminates when any of the following conditions exists:
  
  — You are reorganizing a table space that contains an archive-enabled table.
  
  — You are running a DSNUTILB reorganization.

- 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 sets in REORG PLUS” on page 344.
REORG PLUS ignores the DB2 subsystem parameter PREVENT_ALTERTB_LIMITKEY for index-controlled partitioned objects.

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.

**Additional considerations**

- 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.

- 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 347 and “Sample ALTER TABLE statements and resulting messages” on page 348.

- You can use a previously created DDLOUT data set as DDLIN input. For more information about this data set, see “DDLOUT data sets in REORG PLUS” on page 349.

**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 considerations**

The following DSNUTILB considerations apply to the REBALANCE option:

- REORG PLUS invokes DSNUTILB when you specify REBALANCE and all of the following conditions exist:
— You are reorganizing a table space that contains a LOB column.

— You specify SHRLEVEL REFERENCE.

— You do not specify the AUX option on the REORG command.

**Note**
In this case, the DB2 REORG utility defaults to AUX YES.

- When invoking DSNUTILB, REORG PLUS ignores the REBALANCE option for partition-by-growth or nonpartitioned table spaces.

For more information about running a DSNUTILB reorganization, see “Reorganization jobs that invoke DSNUTILB” on page 57.

**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 partition number</th>
<th>Physical partition number</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>Logical partition number</th>
<th>Physical partition number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<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:
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>Logical partition number</th>
<th>Physical partition number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>5</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:

```
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:

```
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 Performance for DB2 Databases or Database Performance for DB2 solution. For more information about the BMC value, see “Conditional reorganizations using the DASD MANAGER PLUS exceptions table” on page 127.

**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 LEAFDSLML.

**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.

The following table 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.

**Table 26: Using the CONDEXEC command and installation options**

<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 27 on page 126</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>
</tbody>
</table>
**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.

The following table shows how the limit command options interact with the limit installation options and the resulting REORG PLUS action.

**Table 27: Using the limit command and installation options**

<table>
<thead>
<tr>
<th>Limit command option</th>
<th>Limit installation option value</th>
<th>REORG PLUS action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specified with no value</td>
<td>integer or NONE</td>
<td>Uses the value of the corresponding installation option. REORG PLUS considers only the limit options that you specify in the command. If the installation option value is NONE, REORG PLUS does not consider the limit option.</td>
</tr>
<tr>
<td>Specified with a value</td>
<td>integer or NONE</td>
<td>Uses the command option value and does not consider any limit option that is not on the command.</td>
</tr>
<tr>
<td>(Nothing specified)</td>
<td>integer</td>
<td>Honors the limit installation option and uses its value.</td>
</tr>
<tr>
<td>(Nothing specified)</td>
<td>NONE</td>
<td>Honors the values on the other limit installation options but does not consider the option with NONE specified.</td>
</tr>
</tbody>
</table>

For details about the CONDEXEC command, see “CONDEXEC” on page 217. The limit option descriptions follow the CONDEX command description. For a
Conditional reorganizations using the DASD MANAGER PLUS exceptions table

When you specify a threshold for an exception in the BMCTRIG function of DASD MANAGER PLUS and then specify one or more objects, BMCTRIG evaluates the objects against the exception criteria. If you tell it to save exceptions, BMCTRIG 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.

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

Considerations when using DASD MANAGER PLUS exceptions

The following table describes the general actions that REORG PLUS takes based on the type of exception found in the DASD MANAGER PLUS exceptions table.
### Table 28: Reorganization run and exceptions reset by type of DASD MANAGER PLUS exception

<table>
<thead>
<tr>
<th>Type of exception</th>
<th>Type of reorganization run</th>
<th>Exceptions reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table space</td>
<td>Full table space</td>
<td>Table space exceptions and exceptions for indexes built during the reorganization</td>
</tr>
<tr>
<td>Table space for participating partitions</td>
<td>Partial table space</td>
<td>Table space exceptions for participating partitions and exceptions for index partitions</td>
</tr>
<tr>
<td>Index</td>
<td>Full index</td>
<td>Exceptions for the specified index</td>
</tr>
<tr>
<td>Index for participating partitions of index</td>
<td>Partial index</td>
<td>Exceptions for index partitions</td>
</tr>
<tr>
<td>Table</td>
<td>Full table space or index</td>
<td><em>(table space reorganization)</em> Table space exceptions and exceptions for indexes built during the reorganization <em>(index reorganization)</em> Exceptions for the specified index</td>
</tr>
<tr>
<td>Table for participating partitions</td>
<td>Partial table space or index</td>
<td><em>(table space reorganization)</em> Table space exceptions for participating partitions and exceptions for index partitions <em>(index reorganization)</em> Exceptions for index partitions</td>
</tr>
</tbody>
</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 29 on page 129.
Exceptions that trigger a reorganization

Table 29 on page 129 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>
<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 a</td>
</tr>
<tr>
<td></td>
<td>DSEXTENT</td>
<td>Dsextents</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 a</td>
</tr>
<tr>
<td></td>
<td>REORMDEL</td>
<td>Mass del reorg</td>
<td>Mass deletion a</td>
</tr>
<tr>
<td></td>
<td>REORGMODS</td>
<td>Mods since reorg</td>
<td>Initiates a reorganization after a large number of rows have been modified a</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 a</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>Space</td>
<td>Percentage of change in tracks that an object uses a</td>
</tr>
<tr>
<td></td>
<td>BMCRUSER</td>
<td>BMCRUSER</td>
<td>User-defined exception in DASD MANAGER PLUS Note: For information about setting up a user-defined exception and naming it BMCRUSER, see the <em>DASD MANAGER PLUS for DB2 User Guide.</em></td>
</tr>
<tr>
<td>Reorganization type</td>
<td>Value in the exceptions table</td>
<td>DASD MANAGER PLUS field name</td>
<td>Brief description of the exception</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------</td>
<td>-------------------------------</td>
<td>-----------------------------------</td>
</tr>
<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>
<tr>
<td></td>
<td>FAROFF</td>
<td>Faroff</td>
<td>Percentage of rows that are more than 16 pages from the optimal position. This exception is evaluated only for clustering indexes. A high Faroff percentage indicates that clustering might be degrading.</td>
</tr>
<tr>
<td></td>
<td>PACTHI</td>
<td>PctActivHi</td>
<td>Table spaces in which the percentage of active pages is greater than or equal to the specified value</td>
</tr>
<tr>
<td></td>
<td>PACTLO</td>
<td>PctActivLo</td>
<td>Table spaces in which the percentage of active pages is less than or equal to the specified value</td>
</tr>
<tr>
<td></td>
<td>PCTCLUS</td>
<td>PctClus</td>
<td>Cluster ratio of the object</td>
</tr>
<tr>
<td></td>
<td>PCTDROP</td>
<td>Pct Dropped Rows</td>
<td>Percentage of space that is occupied by dropped rows</td>
</tr>
<tr>
<td></td>
<td>REORPEND</td>
<td>Reorg Pend</td>
<td>Indication that the partition or object is in REORG pending status</td>
</tr>
<tr>
<td></td>
<td>TOTALIND</td>
<td>Totalind</td>
<td>Percentage of rows in a table that are not in their optimal positions</td>
</tr>
<tr>
<td></td>
<td>TOTALOFF</td>
<td>Totaloff</td>
<td>Percentage of table space rows that are not in optimal position by the index key</td>
</tr>
<tr>
<td></td>
<td>UNCLUST</td>
<td>Unclust inserts</td>
<td>Number of unclustered inserts</td>
</tr>
</tbody>
</table>
### DB2 real-time statistics in REORG PLUS

REORG PLUS resets the real-time statistics and updates timestamp values (in both the DSNRTSDB and DB2 memory) during the UTILTERM phase.

Note the following considerations about real-time statistics in REORG PLUS:

- When REORG PLUS stops the object, DB2 updates real-time statistics values with the most current accumulated data. REORG PLUS updates those values again during the UTILTERM phase to reflect the effect of the reorganization.

<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>AREOPEND</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>LEAFTOFF</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 partition.</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.
- 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.

The following table lists the columns in tables SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS that REORG PLUS resets for each partition and the updated value:

### Table 30: Real-time statistics updates

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Updated value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIBM.SYSTABLESPACESTATS</td>
<td>UPDATESTATSTIME</td>
<td>Timestamp of the update</td>
</tr>
<tr>
<td></td>
<td>NACTIVE</td>
<td>Actual value</td>
</tr>
<tr>
<td></td>
<td>NPAGES</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>REORGINserts</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>REORGDELETES</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>REORGUIPDATES</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>REORGNUCLUSTINS</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>REORGDISORGLOB</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>REORGMASSDELETE</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>REORGNEARINDREF</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>REORGFARINDREF</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>STATSLASTTIME</td>
<td>(UPDATEDB2STATS YES only) 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>COPYUPDATEDPAGES</td>
<td>Zero when a copy is taken</td>
</tr>
<tr>
<td></td>
<td>COPYCHANGES</td>
<td>Zero when a copy is taken</td>
</tr>
<tr>
<td></td>
<td>COPYUPDATELRSN</td>
<td>Null when a copy is taken</td>
</tr>
<tr>
<td></td>
<td>COPYUPDATETIME</td>
<td>Null when a copy is taken</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>Actual value</td>
</tr>
<tr>
<td></td>
<td>TOTALROWS</td>
<td>Actual value</td>
</tr>
<tr>
<td></td>
<td>DATASIZE</td>
<td>Actual value</td>
</tr>
<tr>
<td>Table</td>
<td>Column</td>
<td>Updated value</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>SYSIBM.SYSINDEXSPACESTATS</td>
<td>UPDATESTATSTIME</td>
<td>Timestamp of the reset</td>
</tr>
<tr>
<td>Note: For nonpartitioned indexes in partial reorganizations, REORG PLUS updates only the SPACE and EXTENTS columns of this table.</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>REORGINSERTS</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>REORGPSEUDODELETES</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>REORGMASSDELETE</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>REORGLEAFNEAR</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) 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>

## Single-phase reorganization

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.

- 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 in REORG PLUS” on page 362
- “SYSUT1 data sets in REORG PLUS” on page 366

For information about restarting the job during a single-phase reorganization, see “Restarting REORG PLUS” on page 374.

### Table space compression in REORG PLUS

This section describes how REORG PLUS handles table space compression.

The KEEPDICTIONARY installation option 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 “Partition rebalancing in REORG PLUS” on page 115.)

■ One of the following definition changes are pending on the table space:
  — An alter of the table space buffer pool or segment size
  — A table alter that changes the limit key

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 “BMCDICT table” on page 720.

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 in REORG PLUS

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 “Additional restart considerations and restrictions” on page 375.

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 137, the utility 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, the utility invokes DSNUTILB.

Restrictions on indexes on expression

When any of the following conditions exists, the utility 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 columns.

- 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.

---

## Reordered row format in REORG PLUS

REORG PLUS supports reordered row format (RRF), as follows:

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

### Row format 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 HONOR_KEEPDICTIONARY 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.

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 in REORG PLUS

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 with REORG PLUS

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.
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.

**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 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 “REORG PLUS for DB2 system and software requirements” on page 51.

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.

4 Ensure that the value of the ZIIP installation or command option is YES.

## 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 “Instant Snapshot with nonpartitioned indexes” on page 110.

## 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 581.
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
  - 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 describes the options that you can use with the REORG command.

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.
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

This topic provides a quick reference to the REORG PLUS command options.

The following table 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 31: REORG PLUS command options**

<table>
<thead>
<tr>
<th>Command option</th>
<th>Reference</th>
<th>Valid for REORG INDEX?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>“ACTIVE” on page 280</td>
<td>Yes</td>
</tr>
<tr>
<td>ALLOC</td>
<td>“ALLOC” on page 283</td>
<td>Yes</td>
</tr>
<tr>
<td>AMENDED</td>
<td>“AMENDED” on page 202</td>
<td>No</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>“ANALYZE” on page 187</td>
<td>Yes</td>
</tr>
<tr>
<td>ARCHDDN</td>
<td>“ARCHDDN” on page 174</td>
<td>No</td>
</tr>
<tr>
<td>ARCHFORMAT</td>
<td>“ARCHFORMAT” on page 175</td>
<td>No</td>
</tr>
<tr>
<td>ARCROWS</td>
<td>“ARCROWS” on page 216</td>
<td>No</td>
</tr>
<tr>
<td>ASSOCIATE</td>
<td>“ASSOCIATE” on page 180</td>
<td>No</td>
</tr>
<tr>
<td>AUTOESTSPACE</td>
<td>“AUTOESTSPACE” on page 243</td>
<td>No</td>
</tr>
<tr>
<td>AUX</td>
<td>“AUX” on page 244</td>
<td>No</td>
</tr>
<tr>
<td>AVAILPAGEPCT</td>
<td>“AVAILPAGEPCT” on page 182</td>
<td>Yes</td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>“AVGVOLSP” on page 289</td>
<td>Yes</td>
</tr>
<tr>
<td>BMCHIST</td>
<td>“BMCHIST” on page 196</td>
<td>Yes</td>
</tr>
<tr>
<td>BMCSTATS</td>
<td>“BMCSTATS” on page 246</td>
<td>Yes</td>
</tr>
<tr>
<td>CLONE</td>
<td>“CLONE” on page 161</td>
<td>Yes</td>
</tr>
<tr>
<td>CONDEXEC</td>
<td>“CONDEXEC” on page 217</td>
<td>Yes</td>
</tr>
<tr>
<td>COPY</td>
<td>“COPY” on page 251</td>
<td>No</td>
</tr>
<tr>
<td>COPYDDN</td>
<td>“COPYDDN” on page 259</td>
<td>No</td>
</tr>
<tr>
<td>COPYLVL</td>
<td>“COPYLVL” on page 257</td>
<td>No</td>
</tr>
<tr>
<td>Command option</td>
<td>Reference</td>
<td>Valid for REORG INDEX?</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>“DATACLAS” on page 307</td>
<td>Yes</td>
</tr>
<tr>
<td>DDLDDN</td>
<td>“DDLDDN” on page 176</td>
<td>No</td>
</tr>
<tr>
<td>DDTYPE</td>
<td>“DDTYPE” on page 278</td>
<td>Yes</td>
</tr>
<tr>
<td>DEADLINE</td>
<td>“DEADLINE” on page 309</td>
<td>Yes</td>
</tr>
<tr>
<td>DELAY</td>
<td>“DELAY” on page 312</td>
<td>Yes</td>
</tr>
<tr>
<td>DELETE</td>
<td>“DELETE FROM” on page 268</td>
<td>No</td>
</tr>
<tr>
<td>DELETEFILES</td>
<td>“DELETEFILES” on page 202</td>
<td>Yes</td>
</tr>
<tr>
<td>DRAIN</td>
<td>“DRAIN” on page 322</td>
<td>Yes</td>
</tr>
<tr>
<td>DRAIN_WAIT</td>
<td>“DRAIN_WAIT” on page 232</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNPAT</td>
<td>“DSNPAT” on page 297</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTYPE</td>
<td>“DSNTYPE” on page 292</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNUEXIT</td>
<td>“DSNUEXIT” on page 226</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNUTILB</td>
<td>“DSNUTILB” on page 223</td>
<td>Yes</td>
</tr>
<tr>
<td>DSPLOCKS</td>
<td>“DSPLOCKS” on page 234</td>
<td>Yes</td>
</tr>
<tr>
<td>DSRSEXIT</td>
<td>“DSRSEXIT” on page 228</td>
<td>Yes</td>
</tr>
<tr>
<td>EXPDT</td>
<td>“EXPDT” on page 304</td>
<td>No</td>
</tr>
<tr>
<td>FASTSWITCH</td>
<td>“FASTSWITCH” on page 205</td>
<td>Yes</td>
</tr>
<tr>
<td>FORCE</td>
<td>“FORCE” on page 235</td>
<td>Yes</td>
</tr>
<tr>
<td>FORCE_AT</td>
<td>“FORCE_AT” on page 236</td>
<td>Yes</td>
</tr>
<tr>
<td>FORCE_DELAY</td>
<td>“FORCE_DELAY” on page 237</td>
<td>Yes</td>
</tr>
<tr>
<td>FORCE_RPT</td>
<td>“FORCE_RPT” on page 237</td>
<td>Yes</td>
</tr>
<tr>
<td>FSFALLBACK</td>
<td>“FSFALLBACK” on page 207</td>
<td>Yes</td>
</tr>
<tr>
<td>FSTHRESHOLD</td>
<td>“FSTHRESHOLD” on page 208</td>
<td>Yes</td>
</tr>
<tr>
<td>GDGLIMIT</td>
<td>“GDGLIMIT” on page 305</td>
<td>No</td>
</tr>
<tr>
<td>ICDDN</td>
<td>“ICDDN” on page 262</td>
<td>No</td>
</tr>
<tr>
<td>ICTYPE</td>
<td>“ICTYPE” on page 255</td>
<td>No</td>
</tr>
<tr>
<td>IDCACHE</td>
<td>“IDCACHE” on page 242</td>
<td>No</td>
</tr>
<tr>
<td>IDCDDN</td>
<td>“IDCDDN” on page 201</td>
<td>Yes</td>
</tr>
<tr>
<td>IFALLOC</td>
<td>“IFALLOC” on page 282</td>
<td>Yes</td>
</tr>
<tr>
<td>IMPLICIT_TZ</td>
<td>“IMPLICIT_TZ” on page 266</td>
<td>No</td>
</tr>
<tr>
<td>INDREFLIMIT</td>
<td>“INDREFLIMIT” on page 220</td>
<td>No</td>
</tr>
<tr>
<td>Command option</td>
<td>Reference</td>
<td>Valid for REORG INDEX?</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>INLINE</td>
<td>“INLINE” on page 253</td>
<td>No</td>
</tr>
<tr>
<td>KEEPDICIONARY</td>
<td>“KEEPDICIONARY” on page 194</td>
<td>No</td>
</tr>
<tr>
<td>LEAFDISTLIMIT</td>
<td>“LEAFDISTLIMIT” on page 221</td>
<td>Yes</td>
</tr>
<tr>
<td>LOG</td>
<td>“LOG NO” on page 197</td>
<td>No</td>
</tr>
<tr>
<td>LOGFINAL</td>
<td>“LOGFINAL” on page 320</td>
<td>Yes</td>
</tr>
<tr>
<td>LOGMEM</td>
<td>“LOGMEM” on page 319</td>
<td>Yes</td>
</tr>
<tr>
<td>LOGSPILL</td>
<td>“LOGSPILL” on page 320</td>
<td>Yes</td>
</tr>
<tr>
<td>LOGTHRESHLD</td>
<td>“LOGTHRESHLD” on page 308</td>
<td>Yes</td>
</tr>
<tr>
<td>LONLOG</td>
<td>“LONLOG” on page 311</td>
<td>Yes</td>
</tr>
<tr>
<td>LONGLAMETRUNC</td>
<td>“LONGLAMETRUNC” on page 168</td>
<td>Yes</td>
</tr>
<tr>
<td>MAPPINGDATABASE</td>
<td>“MAPPINGDATABASE” on page 323</td>
<td>No</td>
</tr>
<tr>
<td>MAPPINGTABLE</td>
<td>“MAPPINGTABLE” on page 323</td>
<td>No</td>
</tr>
<tr>
<td>MAPTEXIT</td>
<td>“MAPTEXIT” on page 231</td>
<td>No</td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td>“MAXEXTSZ” on page 290</td>
<td>Yes</td>
</tr>
<tr>
<td>MAXNEWPARTS</td>
<td>“MAXNEWPARTS” on page 167</td>
<td>No</td>
</tr>
<tr>
<td>MAXRO</td>
<td>“MAXRO” on page 308</td>
<td>Yes</td>
</tr>
<tr>
<td>MAXSORTMEMORY</td>
<td>“MAXSORTMEMORY” on page 186</td>
<td>Yes</td>
</tr>
<tr>
<td>MAXSORTS</td>
<td>“MAXSORTS” on page 178</td>
<td>Yes</td>
</tr>
<tr>
<td>MAXTAPE</td>
<td>“MAXTAPE” on page 278</td>
<td>Yes</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>“MGMTCLAS” on page 307</td>
<td>Yes</td>
</tr>
<tr>
<td>MINSORTMEMORY</td>
<td>“MINSORTMEMORY” on page 186</td>
<td>Yes</td>
</tr>
<tr>
<td>NLPCTFREE</td>
<td>“NLPCTFREE” on page 197</td>
<td>Yes</td>
</tr>
<tr>
<td>NOSYSREC</td>
<td>“UNLOAD” on page 192</td>
<td>No</td>
</tr>
<tr>
<td>OFFPOSLIMIT</td>
<td>“OFFPOSLIMIT” on page 219</td>
<td>No</td>
</tr>
<tr>
<td>ON FAILURE</td>
<td>“ON FAILURE” on page 210</td>
<td>Yes</td>
</tr>
<tr>
<td>ON MESSAGE</td>
<td>“ON MESSAGE” on page 209</td>
<td>No</td>
</tr>
<tr>
<td>ORDER</td>
<td>“ORDER” on page 179</td>
<td>Yes</td>
</tr>
<tr>
<td>ORIGINALDISP</td>
<td>“ORIGINALDISP” on page 204</td>
<td>Yes</td>
</tr>
<tr>
<td>Command option</td>
<td>Reference</td>
<td>Valid for REORG INDEX?</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>PART</td>
<td>“PART” on page 165</td>
<td>Yes</td>
</tr>
<tr>
<td>PREFORMAT</td>
<td>“PREFORMAT” on page 213</td>
<td>Yes</td>
</tr>
<tr>
<td>RBALRSN_CONVERSION</td>
<td>“RBALRSN_CONVERSION” on page 214</td>
<td>Yes</td>
</tr>
<tr>
<td>REBALANCE</td>
<td>“REBALANCE” on page 169</td>
<td>No</td>
</tr>
<tr>
<td>RECOVERYDDN</td>
<td>“RECOVERYDDN” on page 260</td>
<td>No</td>
</tr>
<tr>
<td>RECOVERYICDDN</td>
<td>“RECOVERYICDDN” on page 263</td>
<td>No</td>
</tr>
<tr>
<td>REDEFINE</td>
<td>“REDEFINE” on page 198</td>
<td>Yes</td>
</tr>
<tr>
<td>REGISTER</td>
<td>“REGISTER” on page 255</td>
<td>No</td>
</tr>
<tr>
<td>REORG INDEX</td>
<td>“INDEX” on page 161</td>
<td>Yes</td>
</tr>
<tr>
<td>REORG TABLESPACE</td>
<td>“TABLESPACE” on page 160</td>
<td>No</td>
</tr>
<tr>
<td>REPORTONLY</td>
<td>“REPORTONLY” on page 222</td>
<td>Yes</td>
</tr>
<tr>
<td>RETPD</td>
<td>“RETPD” on page 305</td>
<td>No</td>
</tr>
<tr>
<td>RETRY</td>
<td>“RETRY” on page 233</td>
<td>Yes</td>
</tr>
<tr>
<td>RETRY_DELAY</td>
<td>“RETRY_DELAY” on page 234</td>
<td>Yes</td>
</tr>
<tr>
<td>REUSE</td>
<td>“REUSE” on page 201</td>
<td>Yes</td>
</tr>
<tr>
<td>RIDMAPMEM</td>
<td>“RIDMAPMEM” on page 318</td>
<td>Yes</td>
</tr>
<tr>
<td>SELECT</td>
<td>“SELECT * FROM” on page 266</td>
<td>No</td>
</tr>
<tr>
<td>SHORTMEMORY</td>
<td>“SHORTMEMORY” on page 184</td>
<td>Yes</td>
</tr>
<tr>
<td>SHRLEVEL</td>
<td>“SHRLEVEL” on page 162</td>
<td>Yes</td>
</tr>
<tr>
<td>SIXSNAP</td>
<td>“SIXSNAP” on page 238</td>
<td>No</td>
</tr>
<tr>
<td>SIZEPCT</td>
<td>“SIZEPCT” on page 293</td>
<td>Yes</td>
</tr>
<tr>
<td>SMS</td>
<td>“SMS” on page 284</td>
<td>Yes</td>
</tr>
<tr>
<td>SMSUNIT</td>
<td>“SMSUNIT” on page 285</td>
<td>Yes</td>
</tr>
<tr>
<td>SORTDATA</td>
<td>“SORTDATA” on page 179</td>
<td>No</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>“SORTDEVT” on page 176</td>
<td>Yes</td>
</tr>
<tr>
<td>SORTKEYS</td>
<td>“SORTKEYS” on page 165</td>
<td>No</td>
</tr>
<tr>
<td>SORTNUM</td>
<td>“SORTNUM” on page 177</td>
<td>Yes</td>
</tr>
<tr>
<td>SPACE</td>
<td>“SPACE” on page 294</td>
<td>Yes</td>
</tr>
<tr>
<td>SPILLDSNPAT</td>
<td>“SPILLDSNPAT” on page 314</td>
<td>Yes</td>
</tr>
<tr>
<td>SPILLSTORCLAS</td>
<td>“SPILLSTORCLAS” on page 313</td>
<td>Yes</td>
</tr>
<tr>
<td>Command option</td>
<td>Reference</td>
<td>Valid for REORG INDEX?</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>SPILLUNIT</td>
<td>“SPILLUNIT” on page 313</td>
<td>Yes</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>“STORCLAS” on page 307</td>
<td>Yes</td>
</tr>
<tr>
<td>SYNC</td>
<td>“SYNC” on page 196</td>
<td>Yes</td>
</tr>
<tr>
<td>TERMEXIT</td>
<td>“TERMEXIT” on page 229</td>
<td>Yes</td>
</tr>
<tr>
<td>THRESHLD</td>
<td>“THRESHLD” on page 295</td>
<td>Yes</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>“TIMEOUT” on page 224</td>
<td>Yes</td>
</tr>
<tr>
<td>TOTALPAGEPCT</td>
<td>“TOTALPAGEPCT” on page 183</td>
<td>Yes</td>
</tr>
<tr>
<td>TSSAMPLEPCT</td>
<td>“TSSAMPLEPCT” on page 250</td>
<td>No</td>
</tr>
<tr>
<td>UNIT</td>
<td>“UNIT” on page 285</td>
<td>Yes</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>“UNITCNT” on page 286</td>
<td>Yes</td>
</tr>
<tr>
<td>UNLDDN</td>
<td>“UNLDDN” on page 171</td>
<td>No</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>“UNLOAD” on page 192</td>
<td>Yes</td>
</tr>
<tr>
<td>UPDATE</td>
<td>“UPDATE tableName SET” on page 270</td>
<td>No</td>
</tr>
<tr>
<td>UPDATEDB2STATS</td>
<td>“UPDATEDB2STATS” on page 248</td>
<td>Yes</td>
</tr>
<tr>
<td>VOLCNT</td>
<td>“VOLCNT” on page 287</td>
<td>Yes</td>
</tr>
<tr>
<td>WHERE condition</td>
<td>“WHERE” on page 273</td>
<td>No</td>
</tr>
<tr>
<td>WORKDDN</td>
<td>“WORKDDN” on page 173</td>
<td>Yes</td>
</tr>
<tr>
<td>WTOMSG</td>
<td>“WTOMSG” on page 242</td>
<td>Yes</td>
</tr>
<tr>
<td>XBMID</td>
<td>“XBMID” on page 241</td>
<td>Yes</td>
</tr>
<tr>
<td>ZIIP</td>
<td>“ZIIP” on page 240</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Command syntax diagrams for REORG PLUS**

This topic diagrams the options allowed on the REORG command.
Figure 8: REORG PLUS command syntax diagram (part 1 of 6)

- REORG
  - TABLESPACE databaseName
  - INDEX
    - tableName
    - indexName
  - CLONE
- SHRLEVEL
  - NONE
  - REFERENCE UNLOADONLY
  - REFERENCE
  - CHANGE
    - CHANGE block
    - See CHANGE block detail
- PART
  - partitionNumber
  - partitionNumber: partitionNumber
  - LAST
    - partitionNumber: LAST
- MAXNEWPARTS integer
- LONGNAMETRUNC
- MIDDLE
  - BEGINNING
  - END
- REBALANCE
  - ON n COLUMNS
- UNLDDN
- SYSREC
  - ddname
- WORKDDN
  - (SYSUT1
    - ddname1
    - SORTOUT
      - ddname2
  - )
- ARCHDDN
- SYSARC
  - ddname
- ARCHFORMAT
  - BMC
  - DB2
- DDLDDN
  - DDLIN
    - ddname
- SORTDEVT deviceType
  - SORTDATA
  - SORTNUM
  - 32
  - MAXSORTS
  - 0

* Option has a corresponding installation option
Figure 10: REORG PLUS command syntax diagram (part 3 of 6)
Figure 11: REORG PLUS command syntax diagram (part 4 of 6)

```
DSNUTILB^a  YES  NO

TIMEOUT^a  TERM  ABEND

DSNUEXIT^a  (  exitName  programLanguage

DSRSEXIT^a  NONE

TERMEXIT^a

MAPEXIT^a

DRAIN_WAIT^a  NONE  UTIL  SQL

RETRY^a  255  integer

RETRY_DELAY^a  1  integer

DSPLOCKS^a

DRNFAIL  NONE  RETRY

FORCE^a  NONE  READERS  ALL  REPORTONLY

FORCE_AT^a  START  RETRY  LASTRETRY

FORCE_DELAY^a  0  integer

FORCE_RPT^a  NO  YES
```

^a Option has a corresponding installation option
Figure 12: REORG PLUS command syntax diagram (part 5 of 6)

Statistics options

Copy options

ICTYPE\(^a\)

\(^a\) Option has a corresponding installation option
Figure 13: REORG PLUS command syntax diagram (part 6 of 6)

Copy options continued

Selective unload and update options

Dynamic data set allocation options

*a Option has a corresponding installation option*
The following diagrams show 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 14: Detail syntax diagrams (part 1 of 5)

**select block detail**

```
SELECT * FROM creatorName. tableName
```

**delete block detail**

```
DELETE * FROM creatorName. tableName
```

**update block detail**

```
UPDATE creatorName. tableName
```

```
SET columnName = constant

NULL
CURRENT DATE
CURRENT TIME
CURRENT TIMESTAMP

WITH TIME ZONE
```

* Option has a corresponding installation option
Figure 15: Detail syntax diagrams (part 2 of 5)

condition block detail

predicate block

`columnName` < constant

- <= NULL
- < > CURRENT DATE
- = CURRENT TIMESTAMP
- `=`
- `!=` WITH TIME ZONE
- `>`
- `>=`

IN

NOT

See predicate block detail

Option has a corresponding installation option
Figure 16: Detail syntax diagrams (part 3 of 5)

**DD type block detail**

```
DDTYPE
   UNLOAD
   WORK
   SORTWORK
   ARCHIVE
   SYSPUNCH
   LOCPFCPY
   LOCPIPY
   LOCBFCPY
   LOCBI PY
   REMPFCPY
   REMPI CPY
   REMBFCPY
   REMBICPY

IFALLOC
   USE
   FREE

ALLOC
   OPTIMIZED
   MINIMAL
   ANY

SMS
   YES
   NO

SMSUNIT
   NO
   YES

UNIT
   (SYSALLDA
      unitName1
      SYSALLDA
      unitName2
   )

See SMS class block details

UNITCNT
   (0
      integer1
      integer2
      0
   )
```

*a Option has a corresponding installation option*
Figure 17: Detail syntax diagrams (part 4 of 5)

DD type block detail continued

- VOLCNT\textsuperscript{a} - (integer1, AUTO, 25)
- AVGVCNTLSP\textsuperscript{a} : (integer1, TRK, CYL, K, 30000)
- MAXEXTSZ\textsuperscript{a} - (integer1, TRK, CYL, K, 0)
- DSTYPE\textsuperscript{b} - (NONE, LARGE, BASIC, EXTREQ, EXTPREF)
- SIZEPCT\textsuperscript{a} - (primary, 100, secondary)
- SPACE (primary, secondary, CYL)
- TRK

\textsuperscript{a} Option has a corresponding installation option

158  \textit{REORG PLUS for DB2 Reference Manual}
Figure 18: Detail syntax diagrams (part 5 of 5)

**DD type block detail continued**

- DSNPAT\(^a\)  
  - pattern
  - ‘NONE’
  - ‘.’

- EXPDT date\(^a\)
- RETPD integer\(^a\)
- GDGLIMIT\(^a\)
- integer

**SMS class block detail**

- DATACLAS\(^a\)
- MGMTCLAS\(^a\)
- STORCLAS\(^a\)
- (NONE)
  - class1
  - NONE
  - class2

**change block detail**

- LOGTHRESHLD integer\(^a\)
- MAXRO\(^a\)
- integer
  - DEFER

- DEADLINE\(^a\)
  - NONE
  - time
  - time\(^a\)

- LONGLOG\(^a\)
  - CONTINUE
  - TERM
  - DRAIN

- DELAY integer\(^a\)

- SPILLUNIT\(^a\)
- unitName
  - NONE

- SPILLSTORCLAS\(^a\)
  - class
  - NONE

- SPILLSNPAT pattern\(^a\)
  - RIDMAPMEM integer\(^a\)
  - LOGMEM integer\(^a\)
  - LOGFINAL\(^a\)
  - NONE
  - time\(^a\)

- LOGSPILL\(^a\)
  - (primary)
  - secondary

- DRAIN\(^a\)
  - ALL
  - WRITERS

- MAPPINGTABLE tableName
- MAPPINGDATABASE 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.

The options are grouped according to the following 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 REORG PLUS 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**

The TABLESPACE option names the table space to be reorganized. All indexes that are associated with the table space are also reorganized.

```
REORG TABLESPACE databaseName. tableSpaceName
```

This option does not apply to an index reorganization.
**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 “REORG PLUS serialization and concurrency” on page 66.

**INDEX**

The INDEX option names the index (and thus the corresponding index space) to be reorganized.

*This option does not apply to a table space reorganization.*

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 and does not materialize pending definition changes on that 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 in REORG PLUS” on page 85.

**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.
Restrictions

The following restrictions apply to SHRLEVEL NONE:

- SHRLEVEL NONE is not valid when reorganizing catalog objects. REORG PLUS issues a message and terminates.

- REORG PLUS invokes DSNUTILB when either of the following conditions exists:
  - You are reorganizing a LOB table space.
  - You are reorganizing an object that is in RREPL status.

- When SHRLEVEL NONE is in effect, REORG PLUS does not materialize any pending definition changes.

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 considerations” on page 87.

**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.

- REORG PLUS invokes DSNUTILB when any of the following conditions exists:
  
  — You are reorganizing a LOB table space.
  
  — You are reorganizing an object that is in RREPL status.
  
  — All of the following conditions exist:
    
    — You are reorganizing a table space that contains a LOB column.
—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 for REORG PLUS” on page 307.

—For information about how the function works, see “Online reorganization” on page 565.

**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 109.

**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 in REORG PLUS” on page 115.

REORG PLUS invokes DSNUTILB 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 “Reorganizing a large number of partitions” on page 94.

**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.

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

The MAXNEWPARTS option specifies limits to the number of partitions that are added during a reorganization.

This option applies to partition-by-growth table spaces only.

MAXNEWPARTS limits the number of partitions as follows:

- 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 “Copy data sets in REORG PLUS” on page 336 and “SYSREC data sets in REORG PLUS” on page 362.

**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. 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 specifies where to truncate names that are longer than the area that is available in report-style messages.

**Restriction**

When invoking DSNUTILB, the utility 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. The utility 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 specifies to truncate outward from the middle of the name. You can shorten this value to M. The symbols >> indicate the truncated section, as shown in the following example:

BMC50521I CREATOR          NAME
BMC50522I LONG_CREATOR_NAME >> VERY_LONG_TABLE_NAME >>

BEGINNING

BEGINNING, or B, specifies to truncate from the beginning of the name. The symbols << indicate the truncated section, as shown in the following example:

BMC50521I CREATOR          NAME
BMC50522I <G_CREATOR_NAME << TABLE_NAME_GREATER_THAN_THIRTY_CHARACTERS

END

END, or E, specifies to truncate from the end of the name. The symbols >> indicate the truncated section, as shown in the following example:

BMC50521I CREATOR          NAME
BMC50522I LONG_CREATOR_NAME >> VERY_LONG_TABLE_NAME_GREATER_THAN_THIRTY_C>>

REBALANCE

For index-controlled and table-controlled partitioned table spaces, you can use the REBALANCE command option to rebalance partitions.

For index-controlled and table-controlled partitioned table spaces, you can use the REBALANCE command option to rebalance partitions.

This option does not apply to index-only or nonpartitioned table space reorganizations.

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 in REORG PLUS” on page 115.

- When invoking DSNUTILB, REORG PLUS handles this option as follows:
  - For partition-by-growth and nonpartitioned table spaces, REORG PLUS ignores this option.
  - For all other objects, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

- 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 sets in REORG PLUS” on page 344.

- When you restart a reorganization, you cannot change the REBALANCE specification.

- REORG PLUS invokes DSNUTILB when all of the following conditions exist:
— You are reorganizing a table space that contains a LOB column.
— 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 122.

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

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.

This option applies to a table space reorganization only.

For information about specifying and using the SYSREC data set, see “SYSREC data sets in REORG PLUS” on page 362.
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

- **Not acceptable set:**
  - BMCRD
  - BMCRD11

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.

**Specifying the default**

You can specify the default for the UNLDDN command option in your installation options module by using the UNLDDN installation option. 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 in REORG PLUS” on page 366.

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:
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.

**Specifying the default**

You can specify the default for the WORKDDN command option in your installation options module by using the WORKDDN installation option. 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**

ARCHDDN allows you to override the default ddname of the archive data set (SYSARC).

This option applies to a table space reorganization only.

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 sets in REORG PLUS” on page 353.

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. 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**

ARCHFORMAT allows you to specify the format of your archive data set (SYSARC).

\[
\text{ARCHFORMAT } \quad \text{BMC} \\
\quad \text{DB2}
\]

This option applies to a table space reorganization only.

For more information about the SYSARC data set, see “SYSARC data sets in REORG PLUS” on page 353.

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. 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.

**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.

---

**DDLDDN**

DDLDDN allows you to override the default ddname of the DDLIN data set.

This option applies to a table space reorganization only.

For information about using a DDLIN data set in your REORG PLUS jobs, see “DDLIN data sets in REORG PLUS” on page 344.

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. 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. 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 certain 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. The following table 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 32: SORTNUM 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 350. For more information about the parameter itself, see “DYNALOC installation option” on page 711.</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 “SORTWK data sets in REORG PLUS” on page 349.

**Specifying the default**

You can specify the default for the SORTNUM command option in your installation options module by using the SORTNUM installation option. 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.

The following table describes the effects that MAXSORTS and its relationship with the SMAX installation option have on REORG PLUS processing:
Table 33: 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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</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>

For more information about the multitasking options, see the following references:

- Table 101 on page 536
- “Performance considerations for sort processing options” on page 530
- “Multitasking processes that invoke BMCSORT” on page 537
- The description of the SMAX installation option

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

SORTDATA

SORTDATA is available only for compatibility with the IBM DB2 REORG utility command, and REORG PLUS treats this option as a comment.

This option applies to a table space reorganization only.

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 548.

**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.

AVAILPAGEPCT

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.

**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 cannot perform an optimal sort due to an insufficient number of available or total pages that it is enabled to allocate, the SHORTMEMORY 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 AVAILPAGEPCT command option in your installation options module by using the AVAILPAGEPCT installation option. 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.

**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 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. 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.

SHORTMEMORY

The SHORTMEMORY option controls the action that REORG PLUS takes when certain memory shortages exists during sort processing.

The memory shortages that trigger an action are as follows:

- 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.

The following table 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
Table 34: 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 CONTINUE</th>
<th>SHORTMEMORY FAIL</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>
<td>REORG PLUS fails.</td>
</tr>
<tr>
<td></td>
<td>Insufficient as specified by MINSORTMEMORY</td>
<td></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 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>
<td>REORG PLUS fails.</td>
</tr>
<tr>
<td></td>
<td>Insufficient as specified by MINSORTMEMORY or less than 1024 KB, whichever is greater</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Insufficient as specified by the first parameter of SMCORE</td>
<td></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. 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 34 on page 185.
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.

```
MINSORTMEMORY integer
```

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.

Note

BMC recommends that you use a value of 0.

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. 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 or command option and 2097152.

**Note**

BMC recommends that you use a value of 0.

For information about how this option interacts with the SMCORE installation option, see “Performance considerations for sort processing options” on page 530.

**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. 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.

**ANALYZE**

ANALYZE gathers information about the objects that you are reorganizing and provides estimated data set sizes.

ANALYZE provides estimates for the following data sets:

- Unload (SYSREC)
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.

The following table shows the combinations of ANALYZE options and the effects that they have on how REORG PLUS determines cardinality and average row length.
### Table 35: 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)</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</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANALYZE PAUSE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANALYZE ONLY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANALYZE SAMPLE</td>
<td>REORG PLUS samples the table space or index space.</td>
<td></td>
</tr>
<tr>
<td>ANALYZE PAUSE SAMPLE</td>
<td>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 ONLY SAMPLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANALYZE 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></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</td>
<td>REORG PLUS changes the keywords to ANALYZE PAUSE or ANALYZE ONLY.</td>
<td></td>
</tr>
<tr>
<td>ANALYZE ONLY HURBA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANALYZE 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>
<tr>
<td>ANALYZE PAUSE BMCSTATS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANALYZE ONLY BMCSTATS</td>
<td></td>
<td></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 Performance for DB2 Databases or 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 controls the processing phases that REORG PLUS uses.

Based on this option, REORG PLUS either continues processing after the UNLOAD phase, suspends execution after the UNLOAD phase, or combines the UNLOAD and RELOAD phases into a single processing phase.

Specifying NOSYSREC is the same as specifying UNLOAD RELOAD.
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 559.

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 “REDEFINE” on page 198 and “SYSIDCIN data sets in REORG PLUS” on page 357.

- If you specify UNLOAD RELOAD with ORDER NO, see “ORDER” on page 179 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.

**Restrictions**

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.

**KEEPDICTIONARY**

The KEEPDICTIONARY option tells REORG PLUS whether to keep the existing compression dictionary.

![KEEPDICTIONARY Diagram](image)

*This option applies to a table space reorganization only.*

If you specify KEEPDICTIONARY without a value, REORG PLUS assumes KEEPDICTIONARY YES. For more information about using compression, see “Table space compression in REORG PLUS” on page 134.

**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 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. 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 KEEPDICITIONARY.

**Restrictions**

REORG PLUS ignores a value of YES and treats the option as if you specified KEEPDICITIONARY 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 KEEPDICITIONARY 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.

![BMCHIST Diagram]

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 229 and “Using TERMEXIT to control BMCHIST and statistics updates” on page 801.

**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. 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**

The SYNC option controls how frequently REORG PLUS writes to the BMCSYNC table.

![SYNC Diagram]

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.
Table 36: NLPCTFREE option keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STANDARD</td>
<td>(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.</td>
</tr>
<tr>
<td>BMC</td>
<td>Tells REORG PLUS to honor the DB2 PCTFREE value in the DB2 catalog, even if the value is greater than 10</td>
</tr>
<tr>
<td>integer</td>
<td>Identifies the percentage of each nonleaf index page to reserve as free space. You can specify any integer from 0 through 99.</td>
</tr>
</tbody>
</table>

**REDEFINE**

The REDEFINE 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 DSVCI 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.
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 228 and “Using DSRSEXIT to manage VSAM data set redefinition” on page 787.

Specifying the default

You can specify the default for the REDEFINE command option in your installation options module by using the REDEFINE installation option. 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.

- 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 sets in REORG PLUS” on page 357.

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 DSNZPAPRMx for the SMS DATACLASS 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 228 and “Using DSRSEXIT to manage VSAM data set redefinition” on page 787.

- 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.

**NO**

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**

The REUSE option is provided 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 handles this option as follows:
  - For a LOB table space, REORG PLUS ignores this option.
  - For all other objects, 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 sets in REORG PLUS” on page 357.

**AMENDED**

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.

![AMENDED option](image)

This option applies to a table space reorganization only.

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.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>(default) Prevents REORG PLUS from re-encoding the rows in the table space.</td>
</tr>
<tr>
<td>YES</td>
<td>Causes REORG PLUS to invoke any table’s EDITPROC to both decode and then re-encode the rows.</td>
</tr>
</tbody>
</table>

**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.

![DELETEFILES option](image)
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. 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 “ON FAILURE” on page 210.

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**

The ORIGINALDISP option tells REORG PLUS how to handle the original data sets when using staging data sets.
This option applies to SHRLEVEL REFERENCE and SHRLEVEL CHANGE only.

With this option, you can 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. 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.

Table 38: ORIGINALDISP option keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE</td>
<td>If you specify this option, REORG PLUS deletes the original data sets.</td>
</tr>
<tr>
<td>RENAME</td>
<td>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 87.</td>
</tr>
</tbody>
</table>

**FASTSWITCH**

The FASTSWITCH option determines the action that REORG PLUS takes in the UTILTERM phase regarding the staging data sets.

This option applies to SHRLEVEL REFERENCE or SHRLEVEL CHANGE.

For more information, see “Staging data sets and the FASTSWITCH process” on page 92.
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 and FSTHRESHOLD 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. 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.

When FASTSWITCH NO is in effect, REORG PLUS does not materialize any pending definition changes.

FSFALLBACK

The FSFALLBACK option tells REORG PLUS whether to perform rename processing of your staging data sets when FASTSWITCH processing fails.

This option applies to SHRLEVEL REFERENCE or SHRLEVEL CHANGE.

Specifying the default

You can specify the default for the FSFALLBACK command option in your installation options module by using the FSFALLBACK installation option. 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.
— Pending definition changes exist on the object that you are reorganizing.
— You are rebalancing partitions.
— REORG PLUS is invoking DSNUTILB.
— REORG PLUS is converting objects from basic row format to reordered row format.
— REORG PLUS is converting RBA or LRSN format.

**FSTHRESHOLD**

The FSTHRESHOLD option provides a threshold above which REORG PLUS performs FASTSWITCH processing.

![FSTHRESHOLD Diagram]

*This option applies to SHRLEVEL REFERENCE or SHRLEVEL CHANGE.*

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.

**Restrictions**

REORG PLUS ignores this option when either of the following conditions exists:

■ FASTSWITCH NO is in effect.
■ Pending definition changes exist on the object that you are reorganizing.

**Specifying the default**
You can specify the default for the FSTHRESHOLD command option in your installation options module by using the FSTHRESHOLD installation option. 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**

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.

This option applies to a table space reorganization only.

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.

The following table describes processing in each phase when REORG PLUS abnormally terminates and ON FAILURE TERMINATE UTILITY is in effect.

**Table 39: 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>
<tr>
<td>Abend in phase</td>
<td>If you specify TERMINATE</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------</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. |
| **UTILITYTERM** | 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

**PREFORMAT**

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 sets in REORG PLUS” on page 357.

**Specifying the default**

You can specify the default for the PREFORMAT command option in your installation options module by using the PREFORMAT installation option. 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 handles this option as follows:

- For a LOB table space, REORG PLUS ignores this option.
- For all other objects, 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:

- 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.

- When reorganizing a base table space that contains a LOB column, REORG PLUS 
  operates as follows:

  — When you do not specify AUX, or you specify AUX NO, REORG PLUS 
    converts the base table space only. To convert the LOB table space in this case, 
    you must run a separate reorganization on the LOB table space, specifying the 
    RBALRSN_CONVERSION option.

  Note
  
  If you do not specify AUX and other conditions of your reorganization cause 
  REORG PLUS to invoke DSNUTILB, the IBM REORG utility defaults to 
  AUX YES and converts both the base table space and the LOB table space.

  — When you specify AUX YES, REORG PLUS invokes DSNUTILB to convert both 
    the base table space and the LOB table space.

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.

ARCROWS

ARCROWS is an estimate of the number of rows that REORG PLUS will discard to a dynamically allocated archive (SYSARC) data set.

This option applies to a table space reorganization only.

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.
CONDEXEC

The CONDEXEC option specifies whether REORG PLUS should consider performing a conditional reorganization.

For more information about conditional reorganizations, see “Conditional reorganization” on page 124.

Specifying the default

You can specify the default for the CONDEXEC command option in your installation options module by using the CONDEXEC installation option. 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 more information, see the limit command options (OFFPOSLIMIT, INDREFLIMIT, and LEAFDISTLIMIT) or installation options (OFFPOSLM, INDREFLM, and LEAFDSLM).

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 127
- “Conditional reorganization” on page 124

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 for DB2 or Performance for DB2 Databases 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

The OFFPOSLIMIT option allows you to set conditions under which REORG PLUS reorganizes a table space.

```
OFFPOSLIMIT integer
```

This option applies to a table space reorganization only.

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.

```
(NEAROFFPOSF + FAROFFPOSF) * 100 / CARDF=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 124.

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>
</tbody>
</table>
Return code | Description
--- | ---
2 | 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.

**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. 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**

The INDREFLIMIT option allows you to set conditions under which REORG PLUS reorganizes a table space.

\[
\text{(NEARINDREF + FARINDREF) \times 100 / CARDF=} \text{integer}
\]

This option applies to a table space reorganization only.

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.

\[
(\text{NEARINDREF + FARINDREF}) \times 100 / \text{CARDF=} \text{integer}
\]

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 INDREFLM installation option. If you specify CONDEXEC NO or BMC on the REORG command, REORG PLUS
ignores the INDFULIMIT command option. For more information, see “Conditional reorganization” on page 124.

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 INDFULIMIT 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 INDFULIMIT value. REORG PLUS does not perform a reorganization.</td>
</tr>
<tr>
<td>2</td>
<td>A calculated value exceeded the INDFULIMIT 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 INDFULIMIT command option in your installation options module by using the INDFULIM installation option. 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

The LEAFDISTLIMIT option allows you to set conditions under which REORG PLUS reorganizes an index.

This option applies only to an index-only reorganization.
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 124.

**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. 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 Performance for DB2 Databases or 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 124.

**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 57.

**Specifying the default**

You can specify the default for the DSNUTILB command option in your installation options module by using the DSNUTILB installation option. 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 57.

**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. The following table describes the return code information.

### Table 40: 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. 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.

**DSNUEXIT**

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 SPILLDSNPAT option to create patterns for names for dynamically allocated data sets. For more information about these options, see “DSNPAT” on page 297 and “SPILLDSNPAT” on page 314. For more information about creating a DSNUEXIT user exit and user-defined variables, see “Using DSNUEXIT to construct data set name patterns in REORG PLUS” on page 748.

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. 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 41: Program language keywords for DSNUEXIT

<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>
<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.

**DSRSEXIT**

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 sets in REORG PLUS” on page 357.

For details about this user exit, see “REORG PLUS user exits” on page 747.

**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. 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 “Using TERMEXIT to control BMCHIST and statistics updates” on page 801.

**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.
- REORG PLUS always updates DB2 statistics after materializing pending definition changes. If you have a TERMEXIT user exit that changes UPDATEDB2STATS to NO in this case, REORG PLUS changes that specification to UPDATEDB2STATS 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. 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**

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 “Using MAPTEXIT to create mapping objects” on page 808.

**Restrictions**

The following restrictions apply to the MAPTEXIT option and the MAPTEXIT user exit:

- REORG PLUS terminates if you also specify the MAPPINGDATABASE option.
- 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 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. 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 MAPTEXIT 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.
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. 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.

<table>
<thead>
<tr>
<th>Table 42: DRAIN_WAIT option keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Keyword</strong></td>
</tr>
<tr>
<td>NONE</td>
</tr>
<tr>
<td>UTIL</td>
</tr>
<tr>
<td>SQL</td>
</tr>
</tbody>
</table>
| 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 the utility 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. The utility 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 the utility 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. The utility 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 the utility what action to take regarding displaying claims and locks if a drain attempt times out.

You can specify one of the following keywords:
Table 43: DSPLOCKS option keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRNFAIL</td>
<td>DRNFAIL displays the claims and locks once, after the final attempt to obtain the drain times out.</td>
</tr>
<tr>
<td>NONE</td>
<td>NONE displays no claims or locks.</td>
</tr>
<tr>
<td>RETRY</td>
<td>RETRY displays claims and locks after each drain timeout.</td>
</tr>
</tbody>
</table>

**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. The utility was shipped with a default value of DRNFAIL for this option. The command overrides the default that is in the installation options module.

**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. 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 cancels read claimers at the point specified by the FORCE_AT option.

**FORCE_AT**

The FORCE_AT option specifies when 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. 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.)
Table 44: FORCE_AT option keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>FORCE_AT START starts canceling threads when the drain request begins.</td>
</tr>
<tr>
<td>RETRY</td>
<td>FORCE_AT RETRY starts canceling threads the first time the drain process times out and REORG PLUS attempts to retry the drain.</td>
</tr>
<tr>
<td>LASTRETRY</td>
<td>FORCE_AT LASTRETRY starts 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 or DRNRETRY installation option.</td>
</tr>
</tbody>
</table>

**FORCE_DELAY**

The FORCE_DELAY option specifies 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 starts 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. 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 specifies whether to display a report of the canceled threads.

Table 45: FORCE_RPT option keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>NO does not display the report.</td>
</tr>
<tr>
<td>YES</td>
<td>YES displays 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 SYSPRINT.</td>
</tr>
</tbody>
</table>

*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. REORG PLUS was shipped with a default value of NO for this option.
FORCE_RPT command option overrides the default that is in the installation options module.

**ALL**

FORCE ALL cancels 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 236.

**FORCE_DELAY**

For information about this option, see “FORCE_DELAY” on page 237.

**FORCE_RPT**

For information about this option, see “FORCE_RPT” on page 237.

**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 SYSPRINT.

**FORCE_AT**

For information about this option, see “FORCE_AT” on page 236.

**FORCE_DELAY**

For information about this option, see “FORCE_DELAY” on page 237.

**FORCE_RPT**

REORG PLUS ignores FORCE_RPT when you specify FORCE REPORTONLY.

**SIXSNAP**

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.
This option applies only to a SHRLEVEL REFERENCE or SHRLEVEL CHANGE partial table space reorganization.

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 110. For a list of 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. 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.

Table 46: SIXSNAP option keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>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.</td>
</tr>
<tr>
<td>YES</td>
<td>YES tells REORG PLUS to use only Instant Snapshot technology. If the Instant Snapshot copy fails, REORG PLUS terminates.</td>
</tr>
<tr>
<td>NO</td>
<td>NO tells REORG PLUS to use the software-based copy method to copy each nonpartitioned index.</td>
</tr>
</tbody>
</table>
**ZIIP**

The ZIIP option tells REORG PLUS whether to attempt to use IBM 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. 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.

- 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 for REORG PLUS” on page 140.

For the steps to enable zIIP processing, see “Using XBM or SUF to enable zIIP processing” on page 140.

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

For information about how to determine zIIP usage, view the Quick Course "EXTENDED BUFFER MANAGER for DB2 - Measuring zIIP Usage."

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.

```
XBMID ssid
```

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 for REORG PLUS” on page 140. 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. The command option overrides the default that is in the installation options module.

**WTOMSG**

Specify WTOMSG SUFSTART to write message BMC50008I to the MVS system log to indicate that the snapshot-processing initialization has successfully completed.

\[
\text{WTOMSG} - \text{SUFSTART 'text'}
\]

*This option applies to SHRLEVEL CHANGE only.*

You can use the text of this message to trigger the submission of jobs that you want to run concurrently with the utility 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. The utility truncates strings that are greater than 50 characters. Quotes cannot appear within the text string.

**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.

\[
\text{IDCACHE} - 10000, \text{integer}
\]
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 2147483647.

REORG PLUS reserves this cache of numbers in SYSIBM.SYSEQUSENCES. 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. 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**

The AUTOESTSPACE option specifies whether to calculate the optimal hash space size for the table space.

This option applies to a DSNUTILB table space reorganization only.

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.

**Table 47: AUTOESTSPACE option keywords**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>YES (the default) indicates that real-time statistics values are to be used to calculate the hash space.</td>
</tr>
<tr>
<td>NO</td>
<td>NO indicates that the HASH SPACE value that is defined on the table is to be used.</td>
</tr>
</tbody>
</table>
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.

**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.

**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 DSNUTILB installation or command option 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 participating partition is in REORP status.

REORG PLUS invokes DSNUTILB when the table space contains an inline LOB column whose length definition was altered before the reorganization.

Statistics options

The statistics options allow you to update statistical information in the DASD MANAGER PLUS database statistics tables and in the DB2 catalog.

LOADPLUS and REORG PLUS use the BMC Common Statistics component to update these statistics.

Note

The utility automatically updates the DB2 real-time statistics tables.

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 utility SYSPRINT.
  To use this method, specify an ASUSRPRT DD statement in your JCL.

- Send the output to the utility SYSPRINT.
  To use this method, do not specify an ASUSRPRT DD statement in your JCL. The utility automatically sends the report to your SYSPRINT.
The Common Statistics component does not send this output to a SYSPRINT2 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.

- **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 BMCPWD 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 801.
- You can control the amount of table space sampling that REORG PLUS does for these statistics. For more information, see “TSSAMPLEPCT” on page 250.
- 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 246.

**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 250.

- 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.

- UPDATEDB2STATS NO, which is the default, tells REORG PLUS not to update statistics in the DB2 catalog.

When materializing pending definition changes during the reorganization, REORG PLUS changes UPDATEDB2STATS NO to YES.
**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.

**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 affect updating DB2 statistics in the following cases:

— You cannot change UPDATEDB2STATS NO to YES.

— When the object that you are reorganizing has pending definition changes, you cannot change UPDATEDB2STATS YES to NO.

For information about the TERMEXIT user exit, see “Using TERMEXIT to control BMCHIST and statistics updates” on page 801.

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 250.

**TSSAMPLEPCT**

The TSSAMPLEPCT option enables you to specify a percentage of table space pages that you want REORG PLUS to sample when gathering statistics.

\[
\text{TSSAMPLEPCT} \rightarrow 100 \leftarrow \text{integer}
\]

*This option applies to a table space reorganization only.*

The following values are valid for this option:

- 1 through 50 samples the specified percentage of the table space pages.
- 100 reads 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. 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 reloaded the table space*, or a standard image copy or DSN1COPY-type copy *after it reloaded the table space*. Additional options control other aspects of the copy process, including the ability to override the default dnames 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 “Reorganizing partition-by-growth table spaces” on page 113.
REORG PLUS creates all 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 in REORG PLUS” on page 336.

- 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 option specification. For information about SHRLEVEL CHANGE and the incremental copy data sets, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 602.

- 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. 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.

Specify one of the following options:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>REGISTER ALL registers all copies requested in the JCL by ddname.</td>
</tr>
<tr>
<td>NONE</td>
<td>REGISTER NONE does not register any of the copies with DB2 and does not leave the object in COPY pending status.</td>
</tr>
<tr>
<td>ddname</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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. 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**

ICTYPE AUTO 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 ICTYPE 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 ICTYPE 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

COPYLVL tells REORG PLUS how to assign the dynamically allocated image copy data sets in certain cases.

REORG PLUS uses the COPYLVL option only when dynamic allocation is active for copy data sets and only 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 in REORG PLUS” on page 336. For more information regarding incremental copy data sets, see “Incremental copy data sets for SHRLEVEL CHANGE” on page 602.

Specifying the default
You can specify the default for the COPYLVL command option in your installation options module by using the COPYLVL installation option. 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 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 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 in REORG PLUS” on page 336.

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.

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 BCPY, and specify BCPY157 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. 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**

RECOVERYDDN allows you to override the default ddnames or ddname prefixes that were specified in your installation options for remote copy data sets.

![Diagram](image)

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 in REORG PLUS" on page 336.
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.

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. 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

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.

```
ICDDN - ( BICY ddname1 , BICZ ddname2 )
```

This option applies to SHRLEVEL CHANGE only.

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, `ddname1` will be the DB2 local primary and `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 602.

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.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

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.

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.

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. 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**

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.

```
RECOVERYICDDN ( BIRY ddname1, BIRZ ddname2 )
```

*This option applies to SHRLEVEL CHANGE only.*

When you register the copies, `ddname1` will be the DB2 remote primary and `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 602.

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.

**Restriction**

When invoking DSNUTILB, REORG PLUS ignores this option.

**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 \texttt{COPYLVL PART}), \texttt{REORG PLUS} appends the partition number to the ddname prefix. Specify a prefix that results in eight characters or less after \texttt{REORG PLUS} appends the highest partition number.

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.

\textit{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 \texttt{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

\textit{Specifying the default}

You can specify the default for the \texttt{RECOVERYICDDN} command option in your installation options module by using the \texttt{RCVICDDN} installation option. \texttt{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.

\textbf{Selective unload and update options for \texttt{REORG TABLESPACE}}

The \texttt{SELECT} and \texttt{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 \texttt{UPDATE} option allows you to reset column values to a constant.

\textit{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
- Reorganizations of objects that are in RREPL status

### 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**

The IMPLICIT_TZ option enables you to specify the time zone that you want REORG PLUS to use in certain cases.

```
IMPLICIT_TZ 'timeZoneString'
```

This option does not apply to an index-only reorganization.

REORG PLUS uses this time zone 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.

**SELECT * FROM**

The SELECT * FROM option tells REORG PLUS to select rows that match the specified WHERE clause conditions from the named table.

```
SELECT * FROM tableName [creatorName] WHERE condition block
```

This option does not apply to an index-only reorganization.
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
- Reorganizations of objects that are in RREPL status

For restrictions on the WHERE clause of your SELECT statement, see “WHERE” on page 273.

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 sets in REORG PLUS” on page 353.

- If you are running a SHRLEVEL CHANGE reorganization, see “SHRLEVEL CHANGE considerations for using SELECT or DELETE” on page 599.
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 more information, see “WHERE” on page 273.

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

The DELETE FROM option tells REORG PLUS to delete the rows that match the specified WHERE clause conditions from the named table.

\[
\text{DELETE * FROM} \quad \text{tableName} \quad \text{WHERE condition block}
\]

This option does not apply to an index-only reorganization.

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
- Reorganizations of objects that are in RREPL status

For restrictions on the WHERE clause of your DELETE statement, see “WHERE” on page 273.

**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 sets in REORG PLUS” on page 353.

- If you are running a SHRLEVEL CHANGE reorganization, see “SHRLEVEL CHANGE considerations for using SELECT or DELETE” on page 599.

**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 more information, see “WHERE” on page 273.
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**

`UPDATE tableName SET` enables you to reset column values to a constant.

You cannot use this option for an index-only or SHRLEVEL CHANGE reorganization.

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 the following table.
Table 49: 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. Observe the rules for constants in Table 51 on page 275. 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).</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</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
  - 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
  - Reorganizations of objects that are in RREPL status

- 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 273.

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 more information, see “WHERE” on page 273.

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.
WHERE

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. The following table shows the result of the WHERE clause when you use the AND, OR, or NOT operator. For more information about predicates, see “predicate” on page 274.

Table 50: Truth table for AND, OR, NOT

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p AND q</th>
<th>p OR q</th>
<th>NOT p</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</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

**Predicate**

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:

- `=`
- `!=`
- `>`
- `>=`
- `<=`
- `<`
- `!=`
- `IN`
<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>

**constant**

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 51 on page 275. 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 51: Rules for constants**

<table>
<thead>
<tr>
<th>Type of constant</th>
<th>Additional details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>Identical to DB2 SQL integer constants</td>
</tr>
<tr>
<td></td>
<td>The constant must be within the range of the column’s data type.</td>
</tr>
<tr>
<td>Type of constant</td>
<td>Additional details</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Decimal          | Identical to DB2 SQL decimal constants  <br>The constant must be within the range of the column’s data type.  <br>Note:  <br>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.  <br>Example of using the comma as a decimal point:  
`SELECT * FROM owner.table`  
`WHERE SALARY > 100.00`  
Example of using the comma as both a decimal point and a separator:  
`SELECT * FROM owner.table`  
`WHERE SALARY IN (100,00, 200,00, 300,00)` |
| Character string | Identical to DB2 SQL strings  <br>Use ‘’ to denote an empty string.  <br>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  <br>Note:  <br>‣ 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  <br>Exceptions:  <br>‣ 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.  
<br>Note: If you specify a time zone on a timestamp constant, this value overrides any value that you specify for the IMPLICIT_TZ option. |
| Floating point   | Not supported                                                                                                                                 |
| Decimal floating point | Not supported                                                                                                                                 |
| Binary string    | Not supported                                                                                                                                 |
| LOB              | Not supported                                                                                                                                 |
### 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 in “IMPLICIT_TZ” on page 266) 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 for REORG PLUS

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 in REORG PLUS” on page 77.

**MAXTAPE**

The MAXTAPE option allows you to specify a maximum number of tape devices that REORG PLUS will use when dynamically allocating data sets.

```
MAXTAPE integer
```

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 82.

**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. 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.

**DDTYPE**

The DDTYPE 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. The following table lists the keywords that you can specify for the data set types that you want to allocate dynamically.

### Table 52: DDTYPE keywords

<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></td>
</tr>
<tr>
<td>LOCPFCPY</td>
<td></td>
</tr>
<tr>
<td>LOPICPY</td>
<td></td>
</tr>
<tr>
<td>LOCBFCPY</td>
<td></td>
</tr>
<tr>
<td>LOCICPY</td>
<td></td>
</tr>
<tr>
<td>REMPFCPY</td>
<td></td>
</tr>
<tr>
<td>REMPICPY</td>
<td></td>
</tr>
<tr>
<td>REMBFCPY</td>
<td></td>
</tr>
<tr>
<td>REMBICPY</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** REORG PLUS ignores this DDTYPE option when invoking DSNUTILB to reorganize a LOB table space.
Keyword | Applies specifications for dynamic allocation to
---|---
SYSPUNCH | LOAD control statement data set for DSNUTILB jobs (SYSPUNCH)
| **Note**: REORG PLUS ignores this DDTYPE option when invoking DSNUTILB to reorganize a LOB table space.
LOCPFCPY | Local primary full copy data set (BCPY)
LOCPICPY | Local primary incremental copy data set (BICY)
LOCBFCPY | Local backup full copy data set (BCPZ)
LOCBICPY | Local backup incremental copy data set (BICZ)
REMPFCPY | Remote primary full copy data set (BRCY)
REMPICPY | Remote primary incremental copy data set (BIRY)
REMBFCPY | Remote backup full copy data set (BRCZ)
REMBICPY | Remote backup incremental copy data set (BIRZ)

**Specifying the default**

You can also specify the DD type in your installation options module by using the DDTYPE installation option. Any dynamic data set allocation options that you specify with the DDTYPE command option override the defaults established at installation for the same DDTYPE.

**ACTIVE**

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. 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. 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.

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.

ALLOC

The ALLOC option allows you to specify the method that you want REORG PLUS to use when dynamically allocating your sort work files.

This option applies only to sort work data sets.

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. 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 “Allocating SORTWK data sets in your JCL” on page 352.
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.

SMS

The SMS specifies whether to pass the SMS classes and extended data set types (EXTREQ and EXTPREF) to SMS during dynamic allocation.

---

Whether the utility 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. The utility 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 passes SMS classes and extended data set types during dynamic allocation. The following considerations apply to SMS YES:

- When you specify SMS YES, the utility 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, the utility includes the SMS classes in the TEMPLATE control statement that it builds for the IBM utility.
NO

NO does not pass SMS classes and extended data set types during dynamic allocation. When you specify SMS NO, the utility always passes the UNIT value during dynamic allocation.

SMSUNIT

The SMSUNIT option specifies whether to pass the UNIT value in the SMS allocation parameter list to SMS during dynamic allocation.

- SMSUNIT
  - NO
  - YES

If you do not specify SMS YES, the utility ignores the SMSUNIT option. The utility 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. The utility 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 does not pass the value of the UNIT option.

YES

YES passes the value of the UNIT option.

When invoking DSNUTILB, the utility includes the UNIT option values in the TEMPLATE control statements that it builds for the IBM utility.

UNIT

The UNIT option allows you specify unit names for use during dynamic allocation.
How the utility uses these values depends on the type of data sets that you are allocating:

- For non-SMS-managed data sets, the UNIT option allows you to specify the primary and secondary unit names to use 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 SMSSUNIT YES, this option supplies the unit names to pass 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 controls which of the specified unit names the utility selects.

**DSNUTILB reorganization jobs**

When invoking DSNUTILB, the utility includes the first parameter in the TEMPLATE control statement that it builds for the IBM utility.

If building a secondary TEMPLATE control statement for your copy data sets, the utility includes the second UNIT parameter in that template. For information about building a secondary template, see the description of the THRESHLD option.

**Specifying the default**

You can specify the default for the UNIT command option in your installation options module by using the UNIT installation option. The utility 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 the utility to use the system default.

If you specify a second value (integer2), the utility uses this number when the value for THRESHLD 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, the utility includes the first parameter in the TEMPLATE control statement that it builds for the IBM utility.

If building a secondary TEMPLATE control statement for your copy data sets, the utility includes the second UNITCNT parameter in that template. For information about building a secondary template, see the description of the THRESHLD option.

**Additional considerations**

The following additional considerations apply to the UNITCNT option:

- The utility 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. The utility 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 for this option are as follows:

- 0 to not specify a volume count for dynamic allocation.
- Integer values 1 through 255 to specify the number of volumes.
- AUTO 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).

The utility computes the volume count by dividing the size estimate by the value specified for the AVGVOLSP installation or command option.

If you specify a second value (integer2), the utility uses that value when the value for the THRESHLD installation or command option 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, the utility includes the first parameter in the TEMPLATE control statement that it builds for the IBM utility.

If building a secondary TEMPLATE control statement for your copy data sets, the utility includes the second VOLCNT parameter in that template. For information about building a secondary template, see the description for the THRESHLD option.

**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, the value is changed to 1.
- To avoid performance problems, specify only the number of volumes that you need.
- For 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.
Specifying the default

You can specify the default for the VOLCNT command option in your installation options module by using the VOLCNT installation option. The utility was shipped with a default value of 25 for this option. The command option overrides the default that is in the installation options module.

AVGVOLSP

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

The utility 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), the utility 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, the utility 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, the utility computes a value for VOLCNT that is too large. However, if you specify a value for this option that is too large, the utility 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. The utility 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, the MAXEXTSZ 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, 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 (ensuring backward compatibility with previous product versions).

If you specify a single value and a comma as follows, REORG PLUS takes the missing value from the MAXEXTSZ installation option:

\[
\text{MAXEXTSZ (} (\text{integer1}, K),) \\
\text{or} \\
\text{MAXEXTSZ (}, (\text{integer2}, K))
\]

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. 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 the description for the THRESHLD option.

**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. 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 to allocate.

If you specify a second value, the utility uses that value when the value that is in effect for the THRESHLD option is exceeded.

**Restrictions**

The following restrictions apply to the DSNTYPE option:

- The DSNTYPE option is not valid for tape data sets.
- When invoking DSNUTILB, the utility 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. The utility 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.

You can specify one of the following options:
Table 53: DSNTYPE option keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>Specifies not use any extended attributes for this data set allocation</td>
</tr>
<tr>
<td>LARGE</td>
<td>Allocates this data set as a large format sequential data set. This option enables data sets larger than 65,535 tracks.</td>
</tr>
<tr>
<td>BASIC</td>
<td>Allocates this data set as a basic sequential data set. This data set will be limited to 65,535 tracks.</td>
</tr>
</tbody>
</table>
| EXTREQ  | Allocates 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 | Specifies that you prefer that this data set be allocated 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.

\[
\text{SIZEPCT} \rightarrow (\text{primary} \cdot \frac{100}{100}, \text{secondary})
\]

The values that you specify must be greater than 0.

- \textit{primary} indicates the percentage of the primary data set size calculated by REORG PLUS that you actually want allocated.

- \textit{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.

**Specify the default**

You can specify the default for the SIZEPCT command option in your installation options module by using the SIZEPCT installation option. 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.

**SPACE**

The SPACE option enables you to override the DSNUTILB space calculation for the dynamically allocated data set.

This option is meaningful only for a DSNUTILB reorganization.

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, the utility ignores this option.
**THRESHLD**

The THRESHLD option allows you to specify a threshold value, in kilobytes, above which the utility applies secondary values to allocated data sets.

The utility tests this threshold for each data set to be allocated. If the size for a particular data set is greater than the threshold, the utility performs the following tasks:

- When SMS is NO, the utility 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, the utility 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, the utility 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, the utility handles this option differently, depending on the type of data set that you are allocating:

- For copy data sets, the utility translates this option to a LIMIT value in the TEMPLATE control statement that it builds for the corresponding IBM utility. The utility also builds the secondary TEMPLATE control statement to which the IBM utility will switch when the LIMIT is exceeded.

  The utility assumes that the value that you supply for THRESHLD is in kilobytes and translates that value to the appropriate value and unit of measure for the LIMIT keyword. Note the following additional information about this value:
  - The utility rounds down to the nearest whole value.
  - If you specify a value that would cause the utility to translate to a value less than 1 cylinder, the utility builds the template with a LIMIT value of 1 CYL.

- For all other data sets, the utility 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 19: THRESHLD example 1**

```plaintext
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 20: THRESHLD example 2**

```plaintext
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 21: THRESHLD example 3**

```plaintext
SMS YES
STORCLAS(STORCLS1,STORCLS2)
MGMTCLAS(MGMTCLS1, MGMTCLS2)
DATACLAS(DATACLS1, DATACLS2)
THRESHLD 720000
```

In example 4, the utility 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 22: 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. The utility was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.

**DSNPAT**

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 54 on page 299) 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. 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.
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 54 on page 299 to construct your pattern. In addition, you can use text or provide user-defined variables from a user exit. For information about the DSNUEXIT command option, see “DSNUEXIT” on page 226. User exits are further discussed in “REORG PLUS user exits” on page 747.

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 “DSNUEXIT user-defined variables for REORG PLUS” on page 750.
### Table 54: Symbolic variables for the DSNPAT command option

<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 <em>MMDDYY</em>)</td>
<td>6 bytes</td>
<td>Variable passed</td>
</tr>
<tr>
<td>&amp;DATEJ</td>
<td>Current Julian date (in the form <em>YYYYDDD</em>)</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>The 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 <em>YYDDD</em>)</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>
<tr>
<td>&amp;PART</td>
<td>Partition for this data set allocation</td>
<td></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 000 or 0000.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbolic variable</td>
<td>Definition</td>
<td>Length of result</td>
<td>DSNUTILB reorganization</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------</td>
<td>-----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>&amp;PART5</td>
<td>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. 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 DTYPE 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.TSNANE.P04096</td>
<td>5 bytes for table spaces with 4096 partitions or less</td>
<td>Variable passed</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 in the JCL. REORG PLUS ignores PROC names.</td>
<td>8 bytes maximum</td>
<td>Variable passed</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 You must have a security package to use this variable.</td>
<td>8 bytes maximum</td>
<td>Variable passed</td>
</tr>
<tr>
<td>&amp;UTIL</td>
<td>BMC utility ID REORG PLUS truncates longer utility IDs to eight characters.</td>
<td>8 bytes maximum</td>
<td>Variable translated to the IBM &amp;UTILID variable</td>
</tr>
<tr>
<td>&amp;UTILPFX</td>
<td>BMC utility ID prefix</td>
<td>8 bytes maximum</td>
<td>Value passed</td>
</tr>
</tbody>
</table>
## Symbolic variable Definition Length of result DSUNIXLB reorganization

<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;UTILSFX</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 part that you are reorganizing</td>
<td>8 bytes</td>
<td>Job terminated</td>
</tr>
</tbody>
</table>

---

### Utility ID variable

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 might be included in &UTILSFX. Depending on the utility ID that is specified for this reorganization job, &UTILSFX might be blank.

For example, if the utility ID is USER1/WORK1, the variables contain the following values:

```
&UTIL = USER1/WO
&UTILPFX = USER1
&UTILSFX = WORK1
```

If the utility ID is USER1//WORK1, the variables contain the following values:

```
&UTIL = USER1//W
&UTILPFX = USER1
&UTILSFX = /WORK1
```

The following special delimiting characters tell REORG PLUS to split the 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>
</tbody>
</table>

---

\( ^{a} \) REORG PLUS removes any trailing blanks in the result.

---

Table 55: 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>
</tbody>
</table>
### 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. The following table 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.

### Table 56: Variable concatenation examples

<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>RDABDEV</td>
</tr>
<tr>
<td>Make two nodes from the values of two variables.</td>
<td>&amp;UID._DEPT</td>
<td>RDAB.DEV</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>
<tr>
<td>Task</td>
<td>Code</td>
<td>Result (where &amp;UID=RDAB and _DEPT=DEV)</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------------------------------------</td>
</tr>
<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>

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 in REORG PLUS” on page 79.

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.

```
EXPDT date
```

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. 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 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.

```
RETPD integer
```

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. 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 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.

```
GDGLIMIT integer
```
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. 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.

SMS class block

These options allow you to specify the classes to use for SMS allocations.

You can repeat this block to specify more than one type of SMS class.

You must enclose the class names for each of the class types in parentheses and you must separate them with 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 the utility 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).

DSNUTILB reorganization jobs

When invoking DSNUTILB, the utility includes these classes in the TEMPLATE control statement that it builds for the IBM utility.
If building a secondary TEMPLATE control statement for your copy data sets, the utility includes the secondary classes in that template. For information about building a secondary template, see the THRESHLD option.

**Specifying the default**

You can specify the default for the SMS class block in your installation options module by using the DATACLAS, MGMTCLAS, or STORCLAS installation options. The utility 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 to use 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 to use 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 to use 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 for REORG PLUS**

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 specific to SHRLEVEL CHANGE. For more information, see “Online reorganization” on page 565.
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 in REORG PLUS” on page 589.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process in REORG PLUS” on page 582.

Specifying the default

You can specify the default for the LOGTHRESHLD command option in your installation options module by using the LOGTHRSH installation option. 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.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process in REORG PLUS” on page 582.

You can 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 in REORG PLUS” on page 589.
Specifying the default

You can specify the default for the MAXRO command option in your installation options module by using the MAXRO installation option. 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.

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.

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 in REORG PLUS” on page 589.

The DEADLINE option interacts with other options to control the log apply process. For information, see “Control of the log apply process in REORG PLUS” on page 582.

**Specifying the default**

You can specify the default for the DEADLINE command option in your installation options module by using the DEADLINE installation option. 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 “MVS console” on page 593) and continues to apply the log records until the delay interval specified on the DELAY option expires. If the longlog condition still exists, REORG PLUS takes the action that you specify with one of the following options.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process in REORG PLUS” on page 582.

You can 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 in REORG PLUS” on page 589.

**Specifying the default**

You can specify the default for the LONGLOG command option in your installation options module by using the LONGLOG installation option. 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 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.

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.

DELAY integer

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 in REORG PLUS” on page 589.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process in REORG PLUS” on page 582.

Specifying the default

You can specify the default for the DELAY command option in your installation options module by using the DELAY installation option. 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 597 and “Performance considerations for SHRLEVEL CHANGE reorganizations” on page 609.

**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. 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 597 and “Performance considerations for SHRLEVEL CHANGE reorganizations” on page 609.

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 SPIILCLS installation option. 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.

**SPILDSNPAT**

The SPILDSNPAT option tells REORG PLUS to use a particular pattern of variables and text to create a prefix for the spill data set names.

```
  SPIILDSNPAT 'pattern'
```

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 SPILDSNPAT command option in your installation options module by using the SPIILDSNP installation option. 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 the following table to construct your pattern. In addition, you can use text or provide user-defined variables from a
user exit. For information about the DSNUEXIT option, see “DSNUEXIT” on page 226. User exits are further discussed in “REORG PLUS user exits” on page 747.

Table 57: Symbolic variables for the SPILLDSNPAT command option

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of result a</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>Note: 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 in the JCL</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td></td>
<td>Note: 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 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></td>
<td>Note: You must have a security package to use the job user ID variable.</td>
<td></td>
</tr>
<tr>
<td>&amp;UTIL</td>
<td>BMC utility ID</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td></td>
<td>Note: REORG PLUS truncates longer utility IDs to eight characters.</td>
<td></td>
</tr>
<tr>
<td>&amp;UTILPFX</td>
<td>BMC utility ID prefix</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;UTILSFX</td>
<td>BMC utility ID suffix</td>
<td>8 bytes maximum</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>
</tr>
</tbody>
</table>

a REORG PLUS removes any trailing blanks in the result.
User-defined variables must begin with an underscore character, as in _DEPT. For more information, see “DSNUEXIT user-defined variables for REORG PLUS” on page 750.

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>
<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>
Any other special characters in the utility ID might cause REORG PLUS to generate invalid data set names.

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:

```
SPILLDSNPAT 'UID.TSIX'
```

The following example combines actual text with symbolic variables to generate a data set name prefix:

```
SPILLDSNPAT '&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. The following table 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>
<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>RDABDEV</td>
</tr>
<tr>
<td>Make two nodes from the values of two variables.</td>
<td>&amp;UID:_DEPT</td>
<td>RDAB.DEV</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>
<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:

```
SPILLDSNPAT '&DB.&DATE'
```

The following statement is correct:

```
SPILLDSNPAT '&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.

**RIDMAPMEM**

The RIDMAPMEM option specifies the amount of data space memory (in kilobytes) that REORG PLUS can use to store the RID translation maps.

```
RIDMAPMEM integer
```

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 611. 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 597

Calculating the data space size, see “Sizing memory for the RID translation map” on page 610

RIDMMAXD and RIDMDSSZ installation options, see their descriptions in “Descriptions of basic REORG PLUS installation options” on page 625

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. 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.

```
LOGMEM integer
```

The value that you specify 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 612.

- 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 597. For performance information, see “Sizing memory for log records” on page 611.

**Specifying the default**

You can specify the default for the LOGMEM command option in your installation options module by using the LOGMEM installation option. 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.

```
LOGSPILL - ( - primary [ - secondary ] )
```

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 597.

**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 LOGSPILL installation option. 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 in REORG PLUS” on page 589.

The LOGFINAL option interacts with other options to control the log apply process. For information, see “Control of the log apply process in REORG PLUS” on page 582.

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. 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 in REORG PLUS” on page 585.
**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.

Specify one of the following options:

- **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. 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**

The MAPPINGTABLE option specifies the mapping table that REORG PLUS provides to the IBM DB2 REORG utility when invoking DSNUTILB for SHRLEVEL CHANGE table space reorganizations.

```
MAPPINGTABLE tableName
```

The following restrictions apply to the MAPPINGTABLE option:

- This option is meaningful only when REORG PLUS invokes DSNUTILB. Otherwise, REORG PLUS treats this option as a comment.

- If you also specify the MAPTEXIT option, REORG PLUS ignores MAPTEXIT.

- REORG PLUS terminates if you also specify the MAPPINGDATABASE option.

**MAPPINGDATABASE**

*(DB2 Version 11 or later)* The MAPPINGDATABASE 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.
The following restrictions apply to the MAPPINGDATABASE option:

- 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.

- REORG PLUS terminates if you also specify either of the following options:
  
  — MAPPINGTABLE
  
  — MAPTEIXT
Building and executing REORG PLUS jobs

This chapter describes the elements of a REORG PLUS job, including the data sets REORG PLUS uses. This chapter also describes how to run jobs and what to do if a job terminates or is canceled.

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 ("JOB statement" on page 325)
- An EXEC statement with the appropriate utility parameters ("REORG PLUS EXEC statement" on page 326)
- STEPLIB or JOBLIB DD statements as needed ("REORG PLUS STEPLIB DD statement" on page 331)
- DD statements as needed for the appropriate number and size of data sets for the data structure that you are reorganizing ("REORG PLUS DD statements" on page 332)
- REORG PLUS control statements using the appropriate command syntax ("Syntax of the REORG command" on page 143)

For examples of REORG PLUS JCL and SYSPRINT output, see "Examples of REORG PLUS jobs" on page 389.

JOB statement

Include a 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 326.

REORG PLUS EXEC statement

The REORG PLUS EXEC statement specifies the module (ARUUMAIN) to be run for the REORG PLUS utility.

The EXEC statement also specifies the utility parameters, which are described in “Utility parameters on the REORG PLUS EXEC statement” on page 327.

You can include the REGION parameter on either your EXEC statement or your JOB statement. For recommendations, see “REGION parameter” on page 326.

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 utility 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
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.

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

Utility parameters on the REORG PLUS 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:

```sql
  //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, the utility uses the DB2 installation default from the DSNHDECP module. The utility depends on the application defaults module being named DSNHDECP. If you do not specify an SSID and the utility cannot find a module named DSNHDECP in your LINKLIST or STEPLIB, the utility terminates.
Note the following considerations when running the utility in a data sharing environment:

- The utility supports the DB2 group attachment name capability. When you supply a group attachment name as the SSID, the utility uses it to connect all plans. The utility 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, the utility 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, the utility uses the default, `userID.jobName`. Each BMC utility job should have a unique ID.

Utility IDs that include special characters might cause the utility to generate invalid data set names when using dynamic allocation. For more information, see the DSNPAT or DSNNAME option description.

**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 374.

*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.

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.

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 the utility 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 SYSPRINT2 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 the description of this option in “Descriptions of basic REORG PLUS installation options” on page 625.

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 “REORG PLUS installation options” on page 617. In addition, the Installation System documentation explains how to create multiple installation options modules.

REORG PLUS 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 the following table 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 60: Data set type descriptions and quick command reference**

<table>
<thead>
<tr>
<th>Data set type</th>
<th>Description reference</th>
<th>Default ddname</th>
<th>DTYPE option keyword</th>
<th>ddname or prefix installation option</th>
<th>ddname or prefix command option</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALTER statement output</strong></td>
<td>“DDLOUT data sets in REORG PLUS” on page 349</td>
<td>DDLOUT</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Archive</strong></td>
<td>“SYSARC data sets in REORG PLUS” on page 353</td>
<td>SYSARC</td>
<td>ARCHIVE</td>
<td>ARCHDDN</td>
<td>ARCHDDN</td>
</tr>
<tr>
<td><strong>Command input</strong></td>
<td>“SYSIN data sets” on page 360</td>
<td>SYSIN</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Copy, full</strong></td>
<td>“Copy data sets in REORG PLUS” on page 336</td>
<td></td>
<td>LOCPFCPY</td>
<td>COPYDDN (primary copies)</td>
<td>COPYDDN (primary copies)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LOCBFCPY</td>
<td>RCVYDDN (remote copies)</td>
<td>RECOVERY DDN (remote copies)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>REMPFCPY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>REMBFCPY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data set type</td>
<td>Description reference</td>
<td>Default ddname</td>
<td>DTYPE option keyword</td>
<td>dname or prefix installation option</td>
<td>dname or prefix command option</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>----------------------</td>
<td>------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Copy, incremental</td>
<td>“Incremental copy data sets for SHRLEVEL CHANGE” on page 602</td>
<td>BICY</td>
<td>LOCPICPY</td>
<td>ICDDN</td>
<td>ICDDN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BICZ</td>
<td>LOCBICPY</td>
<td>RCVICDDN</td>
<td>RECOVERYI CDDN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIRY</td>
<td>REMPICPY</td>
<td>ICDDN</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIRZ</td>
<td>REMBICPY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discard (DSNUTILB only)</td>
<td>“SYSARC data sets in REORG PLUS” on page 353</td>
<td>SYSARC</td>
<td>ARCHIVE</td>
<td>ARCHDDN</td>
<td>ARCHDDN</td>
</tr>
<tr>
<td>Error</td>
<td>“SYSSERR data sets in REORG PLUS” on page 356</td>
<td>SYSSERR</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>REXX exits</td>
<td>“SYSEXEC data sets in REORG PLUS” on page 357</td>
<td>SYSEXEC</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>IDCAMS input</td>
<td>“SYSIDCIN data sets in REORG PLUS” on page 357</td>
<td>SYSIDCIN</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>IDCDDN</td>
</tr>
<tr>
<td>Index work files</td>
<td>“SYSUT1 data sets in REORG PLUS” on page 366</td>
<td>SYSUT1</td>
<td>WORK</td>
<td>WORKDDN</td>
<td>WORKDDN</td>
</tr>
<tr>
<td>Input</td>
<td>“DDLIN data sets in REORG PLUS” on page 344</td>
<td>DDLIN</td>
<td>Not applicable</td>
<td>DDLDDN</td>
<td>DDLDDN</td>
</tr>
<tr>
<td>LOAD statements (DSNUTILB only)</td>
<td>“SYSPUNCH data sets in REORG PLUS” on page 361</td>
<td>SYSPUNCH</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Data set type</td>
<td>Description reference</td>
<td>Default ddname</td>
<td>DDTYPE option keyword</td>
<td>dname or prefix installation option</td>
<td>dname or prefix command option</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------</td>
<td>----------------</td>
<td>-----------------------</td>
<td>------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Message output</td>
<td>“SYSTERM data sets in REORG PLUS” on page 366</td>
<td>SYSTERM</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Message output</td>
<td>“SYSPRINT data sets” on page 360</td>
<td>SYSPRINT</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Message output</td>
<td>“SYSPRIN2 data sets” on page 361</td>
<td>SYSPRIN2</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Other</td>
<td>“Other data sets” on page 370</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>“SYSTSPRT data sets in REORG PLUS” on page 366</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>“UTPRINT data sets” on page 369</td>
<td>UTPRINT</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Sort work files</td>
<td>“SORTWK data sets in REORG PLUS” on page 349</td>
<td>SORTWK</td>
<td>SORTWORK</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Statistics report output</td>
<td>“ASUSRPRT data sets” on page 335</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>“BMCFORCE data sets” on page 336</td>
<td>BMCFORCE</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Unload</td>
<td>“SYSREC data sets in REORG PLUS” on page 362</td>
<td>SYSREC</td>
<td>UNLOAD</td>
<td>UNLDDN</td>
<td>UNLDDN</td>
</tr>
</tbody>
</table>

a 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.

b This value is the default ddname that REORG PLUS passes to the IBM DB2 REORG utility for the &DDNAME variable of the data set name pattern.

c 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.
  
  **Note**
  Dynamic allocation of the work and copy data sets is required for a DSNUTILB reorganization.

- Copy
- SORTWK

  **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
- SYSUT1

  For more information about dynamic allocation, see “Dynamic data set allocation in REORG PLUS” on page 77.

- 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 “Using the ANALYZE option in REORG PLUS to estimate data set allocation” on page 370.

- Calculate the space allocation based on your knowledge of the table space that you are reorganizing and the information in this section.

ASUSRPRT data sets

You can specify an ASUSRPRT 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 SYSPRINT, regardless of whether you specify this DD statement in your JCL.

Alternatively, you can use one of the following options for your statistics output:
Send it to SYSPRINT.

To use this method, do not specify an ASUSRPRT DD statement in your JCL. The utility automatically sends the report to your SYSPRINT.

**Note**

The Common Statistics component does not send this output to a SYSPRINT2 data set.

Suppress the output.

To suppress the output, specify the following DD statement in your JCL:

```
/ASUSRPRT DD DUMMY
```

**BMCFORCE data sets**

You can specify a BMCFORCE DD statement in your JCL to allocate a data set to contain the thread cancelation report output that is generated when you specify one of the following options:

- FORCE REPORTONLY
- FORCE_RPT YES with FORCE READERS or FORCE ALL

If you do not specify a BMCFORCE DD statement in your JCL, the utility sends the report to your SYSPRINT.

**Copy data sets in REORG PLUS**

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 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
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 61: 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>
<tr>
<td>BICYnn</td>
<td>Local primary incremental</td>
</tr>
<tr>
<td>BICZnn</td>
<td>Local backup incremental</td>
</tr>
<tr>
<td>BRCYnn</td>
<td>Remote primary incremental</td>
</tr>
<tr>
<td>BRCZnn</td>
<td>Remote backup incremental</td>
</tr>
</tbody>
</table>

When you are reorganizing either all partitions or a single subset of contiguous partitions of a partitioned table space, you can use the COPYLVL option to specify how REORG PLUS assigns the dynamically allocated full and incremental copy data sets.

**Requirements**

REORG PLUS requires the following copy data sets when you specify COPY YES:

- For all reorganization jobs, BCPY is required when you specify COPY YES.
- 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 602.

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 60 on page 332.
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 335 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 62 on page 339, 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 342.

If you specify DELETEFILES YES and the job terminates after a failure, REORG PLUS deletes any unregistered full and incremental copy data sets.

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 information about partition-level copies, see “Partition-level copies” on page 341 on page 357 and “Partition-by-growth table spaces” on page 342.

For more information about dynamic allocation, see “Dynamic data set allocation in REORG PLUS” on page 77.

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 the following table. Do not specify separate data sets for nonpartitioned, multi-data-set table spaces.

Table 62: 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>
<tr>
<td>Table space being copied</td>
<td>Type of reorganization</td>
<td>Copy DD statements required</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------</td>
<td>----------------------------</td>
</tr>
</tbody>
</table>
| Partitioned, including partition-by-growth | Full (entire table space) | Make one of the following specifications:  
  - Specify one DD statement for each copy type that you want REORG PLUS to make. Do not use nn in the ddname.  
  - 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 342. |
| Partial (selected partitions using PART option) | Make one of the following specifications:  
  - 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.  
  - 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 342. |

The value of the COPYSUBSET installation option must be YES.

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 the following table to determine the value to use for your data set allocation.
Table 63: Calculations for allocating copy data sets

<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>

**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 in REORG PLUS” on page 83.

- 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 \textit{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.

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 80.

- 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 62 on page 339, 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).

**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 251.

**DDLIN data sets in REORG PLUS**

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 updates 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 additional considerations, see “Rebalancing partitions in REORG PLUS” on page 116.

**Guidelines for using the DDLIN data set**

REORG PLUS processes the DDLIN data set according to the following rules:

- REORG PLUS ignores the DB2 subsystem parameter PREVENT_ALTERTB_LIMITKEY. Therefore, this parameter has no effect on the processing of the statements in your DDLIN data set.

- 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 ALTERTABLE 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:

    ```sql
    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
— 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 one of the following types of table spaces:

— 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

— Table spaces that contain archive-enabled tables

■ 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

■ You are running a DSNUTILB 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:

```sql
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:

```sql
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')
BMC51232I ALTER STATEMENT 1 WILL BE PROCESSED
BMC51232I ALTER STATEMENT 3 WILL BE PROCESSED
BMC51293I 2 ALTER STATEMENT(S) WILL BE PROCESSED FROM THE DDLIN FILE
```
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');
BMC50102I ALTER TABLE F509085D.LART003
BMC50102I ALTER PARTITION 6 ENDING AT (X'6F');
BMC50102I ALTER TABLE F509085D.LART003
BMC50102I ALTER PARTITION 7 ENDING AT (X'7F');
BMC50102I ALTER TABLE F509085D.LART003
BMC50102I ALTER PARTITION 8 ENDING AT (X'8F');
BMC50102I ALTER TABLE F509085D.LART003
BMC50102I ALTER PARTITION 9 ENDING AT (X'9F');
BMC50102I ALTER TABLE F509085D.LART003
BMC50102I ALTER PARTITION 10 ENDING AT (X'FF');
BMC51232I ALTER STATEMENT 1 WILL BE PROCESSED
BMC51232I ALTER STATEMENT 2 WILL BE PROCESSED
BMC51232I ALTER STATEMENT 3 WILL BE PROCESSED
BMC51232I ALTER STATEMENT 4 WILL BE PROCESSED
BMC51232I ALTER STATEMENT 5 WILL BE PROCESSED
BMC51232I ALTER STATEMENT 6 WILL BE PROCESSED
BMC51232I ALTER STATEMENT 7 WILL BE PROCESSED
BMC51232I ALTER STATEMENT 8 WILL BE PROCESSED
BMC51232I ALTER STATEMENT 9 WILL BE PROCESSED
```
**DDLOUT data sets in REORG PLUS**

The DDLOUT output data set contains the SQL ALTER statements for the limit-key updates that REORG PLUS makes during partition rebalancing.

REORG PLUS opens this optional data set only if it updates limit keys.

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 in REORG PLUS**

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 352.

**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 DDTYPETYPE 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.

**BMCSORT allocation**

If BMCSORT allocation is active, 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

- Special cases when you have allocated sort work data sets in your JCL (see “Allocating SORTWK data sets in your JCL” on page 352)
The following table describes how BMCSORT allocates sort work files based on values for the SORTDEVT and SORTNUM options and the BMCSORT DYNALOC installation option.

**Table 64: 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 SORTNUM n specified (where n is greater than 0)</td>
<td>ON or OFF If the value is OFF, specifying a value greater than 0 for SORTNUM or specifying SORTDEVT changes this value to ON.</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>SORTDEVT not specified and SORTNUM 0</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></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 “Using the ANALYZE option in REORG PLUS to estimate data set allocation” on page 370.

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 sets in REORG PLUS**

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
— Materialization of a pending definition change that alters the limit key of the last partition
— 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 either of the following conditions exist:
  - A pending definition change exists that alters the limit key of the last partition.
  - 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
- Materializing a pending definition change that alters the limit key of the last partition
- 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.
- 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.

**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 in REORG PLUS” on page 83.
- 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.

**SYSERR data sets in REORG PLUS**

The SYSERR data set contains diagnostic messages that REORG PLUS might produce in error situations when running a SHRLEVEL CHANGE reorganization.
You should not depend on the format of this proprietary data set; this format is subject to change without notice.

**SYSEXEC data sets in REORG PLUS**

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

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 809.

This DD statement specifies the data set that contains these exits.

**SYSIDCIN data sets in REORG PLUS**

SYSIDCIN is the input data set that contains 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 in REORG PLUS” on page 382.

**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 374.

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 358, 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 358, 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 87.
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 sets**

The SYSIN data set is an input data set that contains the utility command. The utility always requires the SYSIN data set.

The UTILINIT phase reads, parses, and verifies the utility 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). The utility uses only columns 1 through 72.

**SYSPRINT data sets**

The SYSPRINT data set is the output data set that contains the utility’s messages. The utility always requires the SYSPRINT data set.

*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.

The utility 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 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 DSNUTILB reorganizations:

- Do not specify FREE=CLOSE.
- You cannot use a PDS or PDSE.

For information about the level of messages that the utility displays and how to change the message level, see the description for the MSGLEVEL utility parameter.
SYSPRIN2 data sets

The SYSPRIN2 data set is an optional output data set that contains the utility’s 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.

The utility 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 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, the utility ignores the SYSPRIN2 DD statement.

For information about the level of messages that the utility displays and how to change the message level, see the MSGLEVEL parameter description.

SYSPUNCH data sets in REORG PLUS

The SYSPUNCH data set contains LOAD control statements for the rows that are discarded and written to the discard file during a DSNUTILB reorganization.

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.
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.

**SYSREC data sets in REORG PLUS**

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. The following table lists the circumstances under which REORG PLUS requires a SYSREC data set.

**Table 65: 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>
</tbody>
</table>
Type of reorganization | SYSREC requirement
------------------------|-----------------------------------
Single-phase table space reorganization when you specify SHRLEVEL NONE | Optional
**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 374.

For information about performance improvements for this data set, see “SYSREC data set” on page 525.

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 335 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 the following table to determine how many SYSREC data sets to allocate. Partition-by-growth table spaces have special requirements as described after the table.

Table 66: 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:

  ```
  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 “Reorganizing partition-by-growth table spaces” on page 113.

**Naming conventions for partition-level SYSREC data sets**

JCL rules limit ddnames to eight characters. When you use multiple SYSRECnn data sets, the dname has two parts: the dname prefix (minimum of one character) and the nn (one to seven characters). The dname in the installation options module or on the REORG command becomes the prefix.

The dname 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 dname (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 to rebalance partitions.

■ REORG PLUS deletes the SYSREC data sets when you specify DELETEFILES YES and the reorganization ends successfully.

■ 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.

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 in REORG PLUS” on page 83.

■ 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.
**SYSTERM data sets in REORG PLUS**

The SYSTERM data set is the output data set that contains DASD MANAGER PLUS SQL and C runtime error messages. Use this data set if you are using REORG PLUS as part of the Database Performance or Performance for DB2 Databases solution.

**SYSTSPRT data sets in REORG PLUS**

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 in REORG PLUS**

SYSUT1 represents the index work data set that contains the unloaded index keys and serves as input for BMCSORT.  

**Note**

You should not depend on the format of this proprietary data set; this format is subject to change without notice.

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. The following table lists the circumstances under which REORG PLUS requires the SYSUT1 data set.

Table 67: SYSUT1 usage

<table>
<thead>
<tr>
<th>Reorganization type</th>
<th>SYSUT1 requirement</th>
</tr>
</thead>
</table>
| Single-phase index reorganization when you specify SHRLEVEL Reference or SHRLEVEL Change | Not used  
References to single-phase reorganizations in this section are for SHRLEVEL NONE only. |
| Single-phase index reorganization when you specify SHRLEVEL NONE | Required if you want to be able to restart the job WARNING: If you do not specify a SYSUT1 data set when performing a single-phase reorganization with SHRLEVEL NONE, you might not be able to restart the job. For more information, see “Restarting REORG PLUS” on page 374. |
| Two-phase index reorganization | Required |
Reorganization type | SYSUT1 requirement
--- | ---
LOB table space reorganization | Not used
Table space reorganization where the following types of indexes exist:  
- Nonpartitioned index  
- Clustering index using ORDER NO  
- Data-partitioned secondary index | Required

Methods for allocating SYSUT1 data sets

Use one of the methods described in “Methods for allocating copy and work data sets” on page 335 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 the following table 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 526.

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.

- 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.

**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 in REORG PLUS” on page 83.

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 sets**

The utility always requires the UTPRINT data set if sorting is necessary.

The presence of this data set tells the utility to report sort messages. However, the actual messages for each sort process appear in separate SYSnnnn 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. 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 the utility is multitasking its sort activity. If you attempt to use an unsupported DD, you risk S1FB abends when concurrent sort tasks are running.

---

**Chapter 4  Building and executing REORG PLUS jobs**
Other data sets

Because it uses BMCSORT, the utility ignores any traditional sort routine DD statements (such as $ORT Parm and DFSPARM) that you specify.

Using the ANALYZE option in REORG PLUS to estimate 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)
- Full image copy (BCPY, BCPZ, BRCY, and BRCZ)
- Incremental image copy (BICY, BICZ, BIRY, and BIRZ)

Estimates provided by the ANALYZE option

Table 69 on page 371 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.

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 69: Estimates provided by the ANALYZE option

<table>
<thead>
<tr>
<th>Data sets for which estimates are provided</th>
<th>Table space reorganization</th>
<th>Index reorganization</th>
<th>Information provided</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single SYSREC data set</td>
<td>Yes</td>
<td>Not applicable</td>
<td>Provides an estimate for all table space reorganizations except for a single-phase reorganization of a partitioned table space</td>
</tr>
<tr>
<td>Multiple SYSREC data sets</td>
<td>Yes</td>
<td>Not applicable</td>
<td>Provides an estimate for each partition that you are reorganizing in a partitioned table space only</td>
</tr>
<tr>
<td>Single SYSUT1 data set</td>
<td>Yes</td>
<td>Yes</td>
<td>For a table space reorganization, provides an estimate for all non-data-sorting indexes and includes any indexes being created. When you specify ORDER NO, the estimate includes the clustering index. For an index reorganization, the estimate is for the index that you are reorganizing.</td>
</tr>
<tr>
<td>Multiple SYSUT1 data sets</td>
<td>Yes</td>
<td>Not applicable</td>
<td>Provides an estimate for each non-data-sorting index, including a non-data-sorting index being created. If you specified ORDER NO, ANALYZE provides an additional value for the clustering index, including a clustering index being created.</td>
</tr>
<tr>
<td>SORTWK data sets</td>
<td>Yes</td>
<td>Yes</td>
<td>Provides two estimates, an optimum value and a minimum value. Each estimate is the total for all SORTWK data sets. Divide this value by the number of SORTWK data sets to get the individual data set sizes. ANALYZE provides the estimates only when a sort will be performed.</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>
<td>Provides an estimate for single copy data sets when you are performing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ Any full table space reorganization</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ A partial reorganization in which the partitions are specified as a contiguous subset and COPYSUBSET is YES</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>
<td>Provides an estimate for each partition that you are reorganizing in a partitioned table space only</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

**Considerations**

The following considerations apply to the estimates that ANALYZE provides:

- REORG PLUS cannot take into account rows bypassed with SELECT or DELETE.
- 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.
- 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 523.
- 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 191.
- 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.
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.

The following table describes whether REORG PLUS uses compressed or expanded rows when KEEPDICTIONARY is in effect.

Table 70: Record size for SYSREC data sets

<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 332. Ensure that all required libraries are available and APF-authorized as described in “REORG PLUS STEPLIB DD statement” on page 331.

**Restarting REORG PLUS**

Restart capabilities in REORG PLUS depend on the type of reorganization that you are running, as described in this section.

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 607.

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, including creating a pending definition change, 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.

Specify RESTART(PHASE) to restart REORG PLUS at the beginning of the last incomplete phase. 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 “Recovering from a failure in REORG PLUS” on page 382.
**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:
    - SHRLEVEL 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:

- If MAXPARTITIONS is altered before you restart a job, 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 SHRLEVEL 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:
  - 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:
    - 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 the following table 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>
<thead>
<tr>
<th>Type of reorganization</th>
<th>Phase for restart</th>
<th>Data sets recopied when RESTART</th>
<th>Data sets recopied when RESTART(PHASE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-phase</td>
<td>REORG phase</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Objects with pending definition changes

The following considerations apply when restarting a reorganization on an object with pending definition changes:

- You can restart the reorganization only if the FASTSWITCH or rename process completed.
  For more information, see “Managing a reorganization that does not complete in the UTILTERM phase” on page 384.

- REORG PLUS ignores any pending definition changes that occurred before you restarted the reorganization.

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 reorganization**

This section describes how to terminate or cancel a reorganization.

**To cancel a reorganization**

Use this procedure if you want to end the utility immediately and want the ability to restart your job.

1. Cancel the job by using the MVS or TSO CANCEL command.

**To terminate a reorganization**

Use this procedure if you want to end the job and do not intend to restart.

1. Terminate the reorganization job by performing one of the following actions:
   - 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.

   If the job is currently running, it terminates at the next sync point.

**Recovering the DB2 object after terminating or canceling a reorganization**

After you terminate or cancel a reorganization job, you might need to perform steps to recover your DB2 objects.

The following table 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 72: Recovering objects after terminating or canceling jobs

<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>UTILINIT or UNLOAD</td>
<td>Any</td>
<td>All objects are usable and no additional steps are required. <strong>Note:</strong> If you have not specified DELETEFILES=YES and you cancel a SHRLEVEL CHANGE reorganization, data sets and staging data sets might remain allocated. If you do not plan to restart the utility, manually delete the data sets and work files, if necessary.</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. <strong>a</strong></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, reset the STATUS column in the BMCUTIL table to S (for Stopped), then restart the job.</td>
</tr>
<tr>
<td></td>
<td>SHRLEVEL REFERENCE or SHRLEVEL CHANGE</td>
<td>If you do not plan to restart the job, recover any unusable objects. <strong>a</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Restart the objects in their original statuses to make them usable.</td>
</tr>
</tbody>
</table>
Recovering from a failure in REORG PLUS

This section provides instructions for recovering after a reorganization job fails.

The first column in the following table 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 73: Recovering from a reorganization failure

<table>
<thead>
<tr>
<th>Phase in which job fails</th>
<th>Recovery steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILINIT</td>
<td>1 Resubmit the job.</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>1 Correct the problem that caused the failure and restart the job with RESTART.</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>1 Correct the problem that caused the failure.</td>
</tr>
<tr>
<td></td>
<td>2 Restart or resubmit the job:</td>
</tr>
<tr>
<td></td>
<td>■ For SHRLEVEL CHANGE, resubmit the job.</td>
</tr>
<tr>
<td></td>
<td>■ For SHRLEVEL NONE or SHRLEVEL REFERENCE, restart the job with RESTART.</td>
</tr>
<tr>
<td>Phase in which job fails</td>
<td>Recovery steps</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>
| RELOAD - redefinition of VSAM data sets, reload and index build processing | 1. Correct the problem that caused the failure.  
2. If the IDCAMS DEFINE failed, manually define the data set.  
3. Restart or resubmit the job:  
  ■ For SHRLEVEL CHANGE, resubmit the job.  
  ■ For SHRLEVEL NONE or SHRLEVEL REFERENCE, restart the job. If you want to start your job at the beginning of the RELOAD phase, specify RESTART(PHASE), otherwise restart the job with RESTART. |
| RELOAD - copy processing | 1. Correct the problem that caused the failure.  
2. Restart or resubmit the job:  
  ■ For SHRLEVEL CHANGE, resubmit the job.  
  ■ For SHRLEVEL NONE or SHRLEVEL REFERENCE, restart the job with RESTART. |
| REORG | 1. Correct the problem that caused the failure.  
2. Restart or resubmit the job:  
  ■ For SHRLEVEL CHANGE, resubmit the job.  
  ■ For SHRLEVEL REFERENCE, 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, 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. |
### Managing a reorganization that does not complete 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 the reorganization” on page 384
- “Backing out the reorganization” on page 386

#### Completing the reorganization

If 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.

To complete the reorganization if the FASTSWITCH or rename process completed

1. Issue a DB2 START command with ACCESS (UT) for the objects that you are reorganizing.
2. Remove the appropriate pending statuses, 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.

To complete the reorganization if the FASTSWITCH or rename process did not complete

In certain cases, you cannot complete the reorganization if FASTSWITCH or rename processing did not complete. In these cases, either REORG PLUS automatically backs out the reorganization, or you must back it out manually. For more information, see “Backing out the reorganization” on page 386.

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 90.
   - 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 92.
3. Issue a DB2 START command with ACCESS (UT) for the objects that you are reorganizing.
4. Remove the appropriate pending statuses, 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**

How you back out a reorganization depends on whether you are using the rename or FASTSWITCH process.

Use the information in this section when any of the following conditions exists:

- You want to back out the reorganization.
- FASTSWITCH or rename processing did not complete and any of the following processes did not complete:
  - Limit-key updates
  - Row format conversion (BRF or RRF)
  - RBA or LRSN conversion

**Note**

REORG PLUS backs out the reorganization for you in the following cases:

- If pending definition changes exist on the object
- 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

**To back out the reorganization 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 90.

2 Issue a DB2 START command with ACCESS (FORCE) to start your object with its original status.
To back out the reorganization when the FASTSWITCH process has completed

The presence of message BMC50895I in your SYSPRINT indicates that the FASTSWITCH process completed.

1. Issue a DB2 STOP command for the objects that you are going to rename.

2. Change the staging data set names to the original data set names, and the original data set names to the staging data set names.

3. Issue a DB2 START command with ACCESS (FORCE) for the objects that you are reorganizing.

To back out the reorganization when the FASTSWITCH process did not complete

If message BMC50895I does not appear in your SYSPRINT, the FASTSWITCH process did not complete.

1. Issue a DB2 START command with ACCESS (FORCE) for the objects that you are reorganizing.
Examples of REORG PLUS jobs

This chapter presents several examples of jobs that were run by using the REORG PLUS product.

Overview of REORG PLUS examples

Each example in this chapter 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 “Syntax of the REORG command” on page 143. For information about JCL statement requirements, see “Building the REORG PLUS job” on page 325.

Use the following table to locate an example with a specific reorganization, process, or object type, or one that uses a specific option.

Table 74: Cross-reference of examples by function

<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, 10-15</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, 12-15</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>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</td>
</tr>
<tr>
<td>ASUSRPRT DD DUMMY</td>
<td>2, 6</td>
</tr>
<tr>
<td>DDLIN data set</td>
<td>12, 14</td>
</tr>
<tr>
<td>SYSIDCIN data set</td>
<td>1</td>
</tr>
<tr>
<td><strong>Command option</strong></td>
<td></td>
</tr>
<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>Function</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ANALYZE ONLY</td>
<td>7</td>
</tr>
<tr>
<td>ANALYZE SAMPLE</td>
<td>11, 14-15</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</td>
</tr>
<tr>
<td>COPY YES</td>
<td>1, 2, 3, 8-15</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>DDTYPE</td>
<td>1-6, 9-15</td>
</tr>
<tr>
<td>DEADLINE</td>
<td>12</td>
</tr>
<tr>
<td>DELAY</td>
<td>12-13</td>
</tr>
<tr>
<td>DELETEFILES NO</td>
<td>1</td>
</tr>
<tr>
<td>DSNPAT</td>
<td>2-6, 9-15</td>
</tr>
<tr>
<td>FASTSWITCH</td>
<td>12-15</td>
</tr>
<tr>
<td>IFALLOC</td>
<td>10</td>
</tr>
<tr>
<td>INLINE YES</td>
<td>10</td>
</tr>
<tr>
<td>KEEPDICIONARY NO</td>
<td>11</td>
</tr>
<tr>
<td>LOGMEM</td>
<td>13</td>
</tr>
<tr>
<td>LOGSPILL</td>
<td>12-13</td>
</tr>
<tr>
<td>LONGLOG</td>
<td>13</td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td>10</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, 9-15</td>
</tr>
<tr>
<td>PART</td>
<td>3, 5, 6</td>
</tr>
<tr>
<td>PREFORMAT YES</td>
<td>10, 11</td>
</tr>
<tr>
<td>RBALRSN_CONVERSION</td>
<td>13</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>Function</td>
<td>Examples</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------</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>12-15</td>
</tr>
<tr>
<td>SIZEPCT</td>
<td>12-13</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>12-13</td>
</tr>
<tr>
<td>SPILLSTORCLAS</td>
<td>12-13</td>
</tr>
<tr>
<td>SPILLUNIT</td>
<td>12-13</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>10-15</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</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>12-15</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.

The following table 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>
<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>
</tbody>
</table>
The following figure shows JCL for example 1:

**Figure 23: JCL for example 1**

```plaintext
// JOB
//*
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/* SEGMENTED TABLESPACE, VCAT DEFINED, INDEX CREATED WITH DEFER *
/* YES. REORG TABLESPACE WITH REDDEFINE YES AND SYSIDCIN IS USED, *
/* 4 COPIES (TWO LOCALS & TWO REMOTES), *
/* REGISTER ALL COPIES (DEFAULT), UPDATE DB2 CATALOG WITH STATS *
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
// 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 -
   (DEHJCAT.DSNDBC.ARUDB001.TS00111.I0001.A001) -
   PURGE -
   CLUSTER -
   CATALOG(DEHJCAT)
DELETE -
   (DEHJCAT.DSNDBC.ARUDB001.TS001.I0001.A001) -
   PURGE -
   CLUSTER -
   CATALOG(DEHJCAT)
DEFINE CLUSTER( -
   NAME(DEHJCAT.DSNDBC.ARUDB001.TS001.I0001.A001) -
   LINEAR -
   REUSE -
   VOLUMES(TM0053 ARU4CD ARU112) -
   CYLINDERS (1 1) -
   SHAREOPTIONS(3 3) -
   ) -
   DATA ( -
   NAME(DEHJCAT.DSNDBD.ARUDB001.TS001.I0001.A001) -
   ) -
   CATALOG(DEHJCAT)
DEFINE CLUSTER( -
   NAME(DEHJCAT.DSNDBC.ARUDB001.TS00111.I0001.A001) -
   LINEAR -
   REUSE -
   VOLUMES(TM0055 ARU4CD ARU112) -
   CYLINDERS (10 5) -
   SHAREOPTIONS(3 3) -
   ) -
   DATA ( -
   NAME(DEHJCAT.DSNDBD.ARUDB001.TS00111.I0001.A001) -
   ) -
   CATALOG(DEHJCAT)
/*
//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)
```
The following figure shows the SYSPRINT output for example 1:
Example 1: VCAT-defined segmented table space with SYSIDCIN

```
BMC50470I VOLCNT  = (25,25) (25,25) N/A
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK)) ((30000,TRK),(30000,TRK)) N/A
BMC50470I DSMTYPE  = (NONE,NONE) (NONE,NONE) N/A
BMC50470I DATACLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I MFTCLAS  = (NONE,NONE) (NONE,NONE) N/A
BMC50470I STORCLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I THRESHLD = 0 0 0
BMC50470I MAKEITSZ = (0,K),(0,K) (0,K),(0,K) N/A
BMC50470I EXPDT    = N/A
BMC50470I RETPD    = N/A
BMC50470I DDTYPE   = LOCPFCPY LOCPICPY REMPFCPY
BMC50470I GDGSCRAT = NO NO NO
BMC50470I GDGEMPTY = N/A N/A N/A
BMC50470I DATACLAS = (NONE,NONE) (NONE,NONE) (NONE,NONE)
BMC50470I DSNTYPE  = (NONE,NONE) (NONE,NONE) (NONE,NONE)
BMC50470I VOLCNT   = (25,25) (25,25) N/A
BMC50470I STORCLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I MGMTCLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I DATACLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK)) ((30000,TRK),(30000,TRK)) N/A
BMC50470I UNITCNT  = (0,0) (0,0) N/A
BMC50470I VOLCNT   = (25,25) (25,25) N/A
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK)) ((30000,TRK),(30000,TRK)) N/A
BMC50470I DSMTYPE  = (NONE,NONE) (NONE,NONE) N/A
BMC50470I DATACLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I MFTCLAS  = (NONE,NONE) (NONE,NONE) N/A
BMC50470I STORCLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I THRESHLD = 0 0 0
BMC50470I MAKEITSZ = (0,K),(0,K) (0,K),(0,K) N/A
BMC50470I EXPDT    = N/A
BMC50470I RETPD    = N/A
BMC50470I DDTYPE   = LOCPFCPY LOCPICPY REMPFCPY
BMC50470I GDGSCRAT = NO NO NO
BMC50470I GDGEMPTY = N/A N/A N/A
BMC50470I DATACLAS = (NONE,NONE) (NONE,NONE) (NONE,NONE)
BMC50470I DSNTYPE  = (NONE,NONE) (NONE,NONE) (NONE,NONE)
BMC50470I VOLCNT   = (25,25) (25,25) N/A
BMC50470I STORCLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I MGMTCLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I DATACLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK)) ((30000,TRK),(30000,TRK)) N/A
BMC504700I UNITCNT  = (0,0) (0,0) N/A
BMC50470I VOLCNT   = (25,25) (25,25) N/A
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK)) ((30000,TRK),(30000,TRK)) N/A
BMC50470I DSMTYPE  = (NONE,NONE) (NONE,NONE) N/A
BMC50470I DATACLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I MFTCLAS  = (NONE,NONE) (NONE,NONE) N/A
BMC50470I STORCLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I THRESHLD = 0 0 0
BMC50470I MAKEITSZ = (0,K),(0,K) (0,K),(0,K) N/A
BMC50470I EXPDT    = N/A
BMC50470I RETPD    = N/A
BMC50470I DDTYPE   = LOCPFCPY LOCPICPY REMPFCPY
BMC50470I GDGSCRAT = NO NO NO
BMC50470I GDGEMPTY = N/A N/A N/A
BMC50470I DATACLAS = (NONE,NONE) (NONE,NONE) (NONE,NONE)
BMC50470I DSNTYPE  = (NONE,NONE) (NONE,NONE) (NONE,NONE)
BMC50470I VOLCNT   = (25,25) (25,25) N/A
BMC50470I STORCLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I MGMTCLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I DATACLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK)) ((30000,TRK),(30000,TRK)) N/A
BMC50470I UNITCNT  = (0,0) (0,0) N/A
BMC50470I VOLCNT   = (25,25) (25,25) N/A
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK)) ((30000,TRK),(30000,TRK)) N/A
BMC50470I DSMTYPE  = (NONE,NONE) (NONE,NONE) N/A
BMC50470I DATACLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I MFTCLAS  = (NONE,NONE) (NONE,NONE) N/A
BMC50470I STORCLAS = (NONE,NONE) (NONE,NONE) N/A
BMC50470I THRESHLD = 0 0 0
BMC50470I MAKEITSZ = (0,K),(0,K) (0,K),(0,K) N/A
```
Example 1: VCAT-defined segmented table space with SYSIDCIN

Chapter 5 Examples of REORG PLUS jobs 397
Example 1: VCAT-defined segmented table space with SYSIDCIN

```
--- REORG PLUS for DB2 Reference Manual ---
```
Example 2: Single-phase reorganization of a partitioned table space using 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 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.

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.
The following table describes the key command options and DD statements for this job:

**Table 76: Key command options and DD statements used in example 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>
</tbody>
</table>
| UPDATE                                        | Resets a specific value to a constant  
In this case, the EMPDEPT column of table ARU.TS0071 is set to the constant DP where EMPNO is equal to 011110009. |
| COPY YES                                      | Creates a DB2 image copy of the table space  
Because the default for the INLINECP installation option is YES, this copy is an inline image copy.  
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. |
| COPYDDN (FULLCP)                              | Specifies a ddname of FULLCP for the data set that receives a full image copy of the table space that you are reorganizing  
This option overrides the default name of BMCCPY (used in case 1 only). |
| COPYLVL FULL                                  | In case 1, COPYLVL FULL overrides the default of COPYLVL PART, allocating a single image copy data set for all partitions being reorganized.  
In case 2, COPYLVL PART allocates an image copy data set for each partition that you are reorganizing. |
| BMCSTATS YES                                  | Tells REORG PLUS to save the BMC statistics in the DASD MANAGER PLUS database statistics tables  
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. |
| SHRLEVEL NONE                                 | 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.  
In case 2, SHRLEVEL REFERENCE is specified. The objects that you are reorganizing are available in read-only (RO) status during the job. |
| ANALYZE SCAN                                  | Calculates optimal data set sizes for dynamic data set allocation (used in case 2 only) |
| //SYSREC01... //SYSREC16                     | Data sets that contain the rows that you are reorganizing  
This example specifies multiple SYSRECnn data sets, one for each partition where nn corresponds to the partition number.  
In case 1, the unload data sets are not required for single-phase reorganization. However, to ensure restartability, they are included in this example.  
In case 2, the unload data sets are not specified in the JCL because dynamic data set allocation is active. |
Command options and DD statements used in JCL | Description
--- | ---
//SYSUT101 //SYSUT102 | 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.
//FULLCP | The ddname 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).

The following figure shows the JCL for case 1 of example 2:

**Figure 24: JCL for example 2, case 1**

```
// 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),             *
//* 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='db2ssid.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=*                                         *
// SYSUDUMP DD SYSOUT=*                                         *
// SYSTEM DD SYSOUT=*                                         *
// ASUSRPRT DD DUMMY                                          *
//*
// SYSCRC01 DD DSN=ARU.EXMPL02A.SYSCRC01,                           *
//  UNIT=WORK,SPACE=(CYL,(20,10)),                                 *
//  DISP=(MOD,CATLG,CATLG)                                        *
// SYSCRC02 DD DSN=ARU.EXMPL02A.SYSCRC02,                           *
//  UNIT=WORK,SPACE=(CYL,(20,10)),                                 *
//  DISP=(MOD,CATLG,CATLG)                                        *
// SYSCRC03 DD DSN=ARU.EXMPL02A.SYSCRC03,                           *
//  UNIT=WORK,SPACE=(CYL,(20,10)),                                 *
//  DISP=(MOD,CATLG,CATLG)                                        *
// SYSCRC04 DD DSN=ARU.EXMPL02A.SYSCRC04,                           *
//  UNIT=WORK,SPACE=(CYL,(20,10)),                                 *
//  DISP=(MOD,CATLG,CATLG)                                        *
// SYSCRC05 DD DSN=ARU.EXMPL02A.SYSCRC05,                           *
//  UNIT=WORK,SPACE=(CYL,(20,10)),                                 *
//  DISP=(MOD,CATLG,CATLG)                                        *
// SYSCRC06 DD DSN=ARU.EXMPL02A.SYSCRC06,                           *
//  UNIT=WORK,SPACE=(CYL,(20,10)),                                 *
//  DISP=(MOD,CATLG,CATLG)                                        *
// SYSCRC07 DD DSN=ARU.EXMPL02A.SYSCRC07,                           *
```
The following figure shows the JCL for case 2 of example 2:

**Figure 25: JCL for example 2, case 2**

```sql
// JOBJ
//
//  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.
```
Example 2: Single-phase reorganization of a partitioned table space using dynamic data set allocation

**SYSPRINT for example 2, case 1**

The following figure shows the SYSPRINT output for case 1 of example 2:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDGLIMIT</td>
<td>5</td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>AVGVOLSP</td>
<td>((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>VOLCNT</td>
<td>(25,25)</td>
</tr>
<tr>
<td>UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>SIZEPCT</td>
<td>(100,100)</td>
</tr>
<tr>
<td>SMSUNIT</td>
<td>NO</td>
</tr>
<tr>
<td>SMS</td>
<td>NO</td>
</tr>
<tr>
<td>ALLOC</td>
<td>NO</td>
</tr>
<tr>
<td>IFALLOC</td>
<td>USE</td>
</tr>
<tr>
<td>DDTYPE</td>
<td>SYSPUNCH</td>
</tr>
<tr>
<td>GDGEMPTY</td>
<td>NO</td>
</tr>
<tr>
<td>EXPDT</td>
<td>N/A</td>
</tr>
<tr>
<td>PLAN</td>
<td>ARUQA</td>
</tr>
<tr>
<td>FSFALLBACK</td>
<td>RORGMAX=300%</td>
</tr>
<tr>
<td>FORCE</td>
<td>NONE</td>
</tr>
<tr>
<td>FORCE_RPT</td>
<td>NO</td>
</tr>
<tr>
<td>FILECHK</td>
<td>WARN</td>
</tr>
<tr>
<td>EXCLDUMP</td>
<td>(X37,X22,X06)</td>
</tr>
<tr>
<td>DSRSEXIT</td>
<td>(NONE,REXX)</td>
</tr>
</tbody>
</table>
| REORG PLUS for DB2 Reference Manual

Example 2: Single-phase reorganization of a partitioned table space using dynamic data set allocation
Example 2: Single-phase reorganization of a partitioned table space using dynamic data set allocation
Example 2: Single-phase reorganization of a partitioned table space using dynamic data set allocation
The following figure shows the SYSPRINT output for case 2 of example 2:
Example 2: Single-phase reorganization of a partitioned table space using dynamic data set allocation

Chapter 5  Examples of REORG PLUS jobs  409
Example 2: Single-phase reorganization of a partitioned table space using dynamic data set allocation

REOR X PLUS for DB2 Reference Manual
Example 2: Single-phase reorganization of a partitioned table space using dynamic data set allocation
Example 2: Single-phase reorganization of a partitioned table space using dynamic data set allocation
REORG PLUS for DB2 Reference Manual
Example 2: Single-phase reorganization of a partitioned table space using dynamic data set allocation
Example 3: Partial reorganization with 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.

The following table describes the key command options for this job:

Table 77: Key command options used in example 3

<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 (case 1)</td>
<td>Reorganizes partitions 1 through 5, 8, and 16 (case 1)</td>
</tr>
<tr>
<td>PART 1:5 (case 2)</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 (case 1)</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>REDEFINE YES (default) (case 2)</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. This example shows a portion of this ASUSRPRT data set for case 2.</td>
</tr>
</tbody>
</table>
Command options used in JCL | Description
--- | ---
SHRLEVEL NONE (case 1) | In case 1, the default SHRLEVEL NONE is used. The objects that you are reorganizing are stopped and unavailable during the job.
SHRLEVEL REFERENCE (case 2) | In case 2, SHRLEVEL REFERENCE is specified. The objects that you are reorganizing are available in read-only status during the job.
DDTYPE LOCBFCPY ACTIVE YES | Activates dynamic allocation for the local backup copy and the remote copies, overriding the installation defaults
DDTYPE REMPFCPY ACTIVE YES
DDTYPE REMBFCPY ACTIVE YES

The following figure shows the JCL for case 1 of example 3:

**Figure 26: JCL for example 3, case 1**

```sql
// 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="db2ssid,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=*
// SYSDUMP DD SYSOUT=*
// SYSTERM DD SYSOUT=*
// ASUSRPT DD SYSOUT=*
// */ SYsin DD *
REORG TABLESPACE ARUDB007.TS007 PART 1:5, 8, 16
COPY YES
REDEFINE NO
BMCSTAS YES
DDTYPE UNLOAD
DSNPAT 'ARU.EXMPL03A.&DDNAME'
DDTYPE WORK
DSNPAT 'ARU.EXMPL03A.&DDNAME'
DDTYPE LOCBFCPY
DSNPAT 'ARU.EXMPL03A.&DDNAME'
DDTYPE REMPFCPY ACTIVE YES
DSNPAT 'ARU.EXMPL03A.&DDNAME'
DDTYPE REMBFCPY ACTIVE YES
```
The following figure shows the JCL for case 2 of example 3:

**Figure 27: 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, SHLEVEL REFERENCE                              |
// | ONE SYSREC PER PART.                                            |
// | * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
// BMCREORG EXEC PGM=ARUUMAIN,
// PARM=’db2ssid,EXMPL03B,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=*  
// SYSDUMP DD SYSOUT=* 
// SYSTEM DD SYSOUT=*  
// ASUSRPRNT DD SYSOUT=* */
// SYSIN DD *
REORG TABLESPACE ARUDB007.TS007 PART 1:5
SHLEVEL REFERENCE
COPY YES
COPYLVL FULL
DDMSTATS 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’
/*
```

**SYSPRINT for example 3, case 1**

The following figure shows the SYSPRINT output for case 1 of example 3:

```
***** B M C R E O R G   P L U S   F O R   D B 2   V11R2.00  *****
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 3/05/2015 13:40:51
BMC50002I UTILITY ID = 'EXMPL03A'. DB2 SUBSYSTEM ID = 'DEHJ'. OPTION MODULE = 'ARU$OPTS'.
BMC50471I z/OS 2.1.0,PID=HBB7790,DFSMS FOR Z/OS=2.1.0,DB2=11.1.0
BMC50471I REGION=0M,BELOW 16M=8836K,ABOVE 16M=1305696K,IEFUSI=NO,CPUS=3
BMC50471I REORG PLUS FOR DB2--V11.02.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V11.02.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V11.01.00
BMC50471I REORG PLUS FOR DB2 - VII.02.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE - VII.02.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE - VII.01.00
```

Chapter 5 Examples of REORG PLUS jobs 419
Example 3: Partial reorganization with SHRLEVEL REFERENCE and a single copy of a subset of partitions
Example 3: Partial reorganization with SHRLEVEL REFERENCE and a single copy of a subset of partitions

SYSPRINT for example 3, case 2

The following figure shows the SYSPRINT output for case 2 of example 3:

```
```

424  REORG PLUS for DB2 Reference Manual
Example 3: Partial reorganization with SHRLLEVEL REFERENCE and a single copy of a subset of partitions

Chapter 5  Examples of REORG PLUS jobs  425
Example 3: Partial reorganization with SHRLEVEL REFERENCE and a single copy of a subset of partitions

REORG PLUS for DB2 Reference Manual
Example 3: Partial reorganization with SHRLEVEL REFERENCE and a single copy of a subset of partitions
ASUSRPR for example 3, case 2

The following figure shows a portion of the statistics report sent to ASUSRPR 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.

Example 3: Partial reorganization with SHRLEVEL REFERENCE and a single copy of a subset of partitions

The following figure shows a portion of the statistics report sent to ASUSRPR for example 3, case 2. The figure shows one set of statistics for each object type. Ellipses (...) at the beginning of a line indicate the omitted statistics sections.
**Example 3: Partial reorganization with SHRLEVEL REFERENCE and a single copy of a subset of partitions**

```
<table>
<thead>
<tr>
<th>TABLE</th>
<th>ARU.TS007</th>
<th>HISTORY</th>
<th>SAVESTATS</th>
<th>Y</th>
<th>UPDATEDBZ</th>
<th>N</th>
<th>UTILCODE</th>
<th>R</th>
<th>LOCATION</th>
<th>(NULL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLING</td>
<td>N</td>
<td>ATTRIBUTES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POTY</td>
<td>1</td>
<td>STORETY</td>
<td>1</td>
<td>N</td>
<td>STORETY</td>
<td>19</td>
<td>FREPAGE</td>
<td>15</td>
<td>FREPAGE</td>
<td>(NULL)</td>
</tr>
<tr>
<td>ALLCOUNT</td>
<td>1</td>
<td>ALLSTORTE</td>
<td>1</td>
<td>N</td>
<td>ALLSTORTE</td>
<td>19</td>
<td>CATNAME</td>
<td>15</td>
<td>CATNAME</td>
<td>(NULL)</td>
</tr>
<tr>
<td>RBACLUSTER</td>
<td>1</td>
<td>RBACLUSTER</td>
<td>AUTOLOAD</td>
<td>-</td>
<td>AUTOLOAD</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>ROWMINFOUND</td>
<td>STATISTICS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ROWAVG</td>
<td></td>
<td>109</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NACTIVE</td>
<td>116</td>
<td>REORGSPACE</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-470</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REORGSPACE_KB</td>
<td></td>
<td>1488</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROWMAXFOUND</td>
<td></td>
<td>165</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXROWS</td>
<td>255</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TYPE</td>
<td>(BLANK)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ENCODING</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSSIZE</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PARTITIONS</td>
<td></td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>OSTYPE</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXGROUP</td>
<td>265</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXPARTITIONS</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAXPARTITIONS</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NACTIVE</td>
<td>116</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROWMINFOUND</td>
<td></td>
<td>107</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROWAVG</td>
<td>109</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NACTIVE</td>
<td>116</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>REORGSPACE</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-470</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REORGSPACE_KB</td>
<td></td>
<td>1488</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(NULL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROWMAXFOUND</td>
<td></td>
<td>165</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COLUMN**

<table>
<thead>
<tr>
<th>ASSOCIATIONS</th>
<th>EMPHIREDATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLESPACE</td>
<td>ARU007.TS007</td>
</tr>
<tr>
<td>TABLE</td>
<td>ARU.TS007</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COLUMN</th>
<th>EMPHIREDATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLCARD</td>
<td>89</td>
</tr>
<tr>
<td>NULLS</td>
<td>0</td>
</tr>
</tbody>
</table>

**STATISTICS**

| BMESTATS V11.02.0D REPORT FOR DEHJ V1115 (111) TIME 2015-03-05-13.41.58.272985 |
|---------------------------------------|---------------------------------------|
| COLMAX                                | 5                                    |
| SCALING                               | 0                                    |
| NULLS                                 | 0                                    |

**CURRENT TABLESPACE**

ARUDB007.TS007

**CURRENT DATABASE**

DEHJ

**REORG PLUS for DB2 Reference Manual**

430
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 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.

The following table 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 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>

The following figure shows the JCL for example 4:

```
// ** Segment Tablespace, One Index (Nonunique, Nonclustered) *
```
**SYSPRINT for example 4**

The following figure shows the SYSPRINT output for example 4:

```
REORG PLUS for DB2 Reference Manual

Example 4: Index-only reorganization
```

```
**** BMC REORG PLUS FOR DB2 V11R2.00 ****

(C) COPYRIGHT 1988 - 2015 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

The following figure shows the SYSPRINT output for example 4:

```
```
Example 4: Index-only reorganization
Example 4: Index-only reorganization

REORG PLUS for DB2 Reference Manual
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.

The following table 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 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>BMCSTATS YES</td>
<td>Tells REORG PLUS to update the statistics in the DASD MANAGER PLUS database statistics tables and the DB2 catalog</td>
</tr>
<tr>
<td>UPDATEDB2STATS YES</td>
<td>REORG PLUS does not save the statistics in this case, because the job is paused.</td>
</tr>
</tbody>
</table>

The following figure shows the JCL for example 5:

**Figure 29: JCL for example 5**

```sql
// JOB
//*/
//** PARTITIONED TABLESPACE, ONE CLUSTERING INDEX, TWO SECONDARY INDEXES REORG UNLOAD PAUSE PART 01 OF PARTITIONED TABLESPACE. */
//** BMCREORG EXEC PGM=ARUUMAIN,
//** PARM=\"db2ssid,EXMPL05,NEW,,MSGLEVEL(1),ARU$OPTS\"
//** STEPLIB DD DISP=SHR,DSN=productLibraries
//** DD DISP=SHR,DSN=DB2.DSNEXIT
```
### SYSPRINT for example 5

The following figure shows the SYSPRINT output for example 5:

```
----- BMC REORG PLUS FOR DB2 V11.02.00 -----

ROCK PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762

(C) COPYRIGHT 1988 - 2015 BMC SOFTWARE, INC.

***** BMC REORG PLUS FOR DB2 V11.02.00 *****

Example 5: Partial reorganization with UNLOAD PAUSE

```

```sql
//SYSIN
/*
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 5: Partial reorganization with UNLOAD PAUSE
Example 6: Restart of a paused reorganization job

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.

The following table 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 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>The JCL for this example also includes the DD statement `/ASUSRPT DD DUMMY. This DD statement tells REORG PLUS not to produce a report of the statistics.</td>
</tr>
</tbody>
</table>
The following figure shows the JCL for example 6:

**Figure 30: JCL for example 6**

```plaintext
// JOB
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// PARTITIONED TABLESPACE, ONE CLUSTERING INDEX, TWO SECONDARY INDEXES.
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// 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=\"db2ssid,EXMPL05.RESTART,,MSGLEVEL(1),ARU$OPTS\",
// STEPLIB DD DISP=SHR,DSN=productLibraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
// * *
// SYSPRINT DD SYOUT=* 
// SYSPRINT DD SYOUT=* 
// UTPRINT DD SYOUT=* 
// SYSDUMP DD SYOUT=* 
// SYSRM DD SYOUT=* 
// ASUSRPR DD DUMMY
// *
// SYSLIB DD *
// REORG TABLESPACE ARUDB032.TS032 PART 01
// UNLOAD PAUSE
// BMCREORG EXEC PGM=ARUUMAIN,
// PARM=\"db2ssid,EXMPL05.RESTART,,MSGLEVEL(1),ARU$OPTS\",
// STEPLIB DD DISP=SHR,DSN=productLibraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
// *
// SYSPRINT DD SYOUT=* 
// SYSPRINT DD SYOUT=* 
// UTPRINT DD SYOUT=* 
// SYSDUMP DD SYOUT=* 
// SYSRM DD SYOUT=* 
// ASUSRPR DD DUMMY
// *
// SYSLIB DD *
// REORG TABLESPACE ARUDB032.TS032 PART 01
// UNLOAD PAUSE
```

**SYSPRINT for example 6**

The following figure shows the SYSPRINT output for example 6:

```
*****  B M C   R E O R G   P L U S   F O R   D B 2    V11R2.00  *****
(C) COPYRIGHT 1988 - 2015 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
BMC50001 I UTILITY EXECUTION STARTING 2/28/2015  17:18:47 ...
BMC50001 I UTILITY ID = 'EXMPL05'.  DB2 SUBSYSTEM ID = 'DEHJ'.  OPTION MODULE = 'ARU$OPTS'.
BMC50001 I REORG PLUS FOR DB2--V11.02.00
BMC50001 I   NO MAINTENANCE TO REPORT
BMC50001 I BMCSORT ENGINE--V02.04.01
BMC50001 I   MAINT: BPJ0691  BPJ0718  BPJ0881  BPJ0922
BMC50001 I BMC STATS API--V11.02.00
BMC50001 I   NO MAINTENANCE TO REPORT
BMC50001 I EXTENDED BUFFER MANAGER--V06.01.00
BMC50001 I   MAINT: BPJ0601  BPJ0667  BPJ0671  BPJ0674  BPJ0675  BPJ0676  BPJ0682  BPJ0686  BPJ0689  BPJ0697  BPJ0700  BPJ0702
BMC50001 I          BPJ0703  BPJ0706  BPJ0712  BPJ0715  BPJ0729  BPJ0773  BPJ0786  BPJ0794  BPJ0806  BPJ0830  BPJ0833  BPJ0839
BMC50001 I          BPJ0845  BPJ0866  BPJ0892  BPJ0913  BPJ0916  BPJ0930  BPJ0944  BPJ0946
BMC50001 I BMCREORG EXEC PGM=ARUUMAIN,
BMC50001 I   PARM=\"db2ssid,EXMPL05.RESTART,,MSGLEVEL(1),ARU$OPTS\",
BMC50001 I   STEPLIB DD DISP=SHR,DSN=productLibraries
BMC50001 I   DD DISP=SHR,DSN=DB2.DSNEXIT
BMC50001 I   DD DISP=SHR,DSN=DB2.DSNLOAD
BMC50001 I   SYSPRINT DD SYOUT=* 
BMC50001 I   SYSPRINT DD SYOUT=* 
BMC50001 I   UTPRINT DD SYOUT=* 
BMC50001 I   SYSDUMP DD SYOUT=* 
BMC50001 I   SYSRM DD SYOUT=* 
BMC50001 I   ASUSRPR DD DUMMY
BMC50001 I   SYSLIB DD *
BMC50001 I   REORG TABLESPACE ARUDB032.TS032 PART 01
BMC50001 I   UNLOAD PAUSE
BMC50001 I   BMCREORG EXEC PGM=ARUUMAIN,
BMC50001 I   PARM=\"db2ssid,EXMPL05.RESTART,,MSGLEVEL(1),ARU$OPTS\",
BMC50001 I   STEPLIB DD DISP=SHR,DSN=productLibraries
BMC50001 I   DD DISP=SHR,DSN=DB2.DSNEXIT
BMC50001 I   DD DISP=SHR,DSN=DB2.DSNLOAD
BMC50001 I   SYSPRINT DD SYOUT=* 
BMC50001 I   SYSPRINT DD SYOUT=* 
BMC50001 I   UTPRINT DD SYOUT=* 
BMC50001 I   SYSDUMP DD SYOUT=* 
BMC50001 I   SYSRM DD SYOUT=* 
BMC50001 I   ASUSRPR DD DUMMY
BMC50001 I   SYSLIB DD *
BMC50001 I   REORG TABLESPACE ARUDB032.TS032 PART 01
BMC50001 I   UNLOAD PAUSE
```

Example 6: Restart of a paused reorganization job
Example 6: Restart of a paused reorganization job

```
REORG PLUS for DB2 Reference Manual

Example 6: Restart of a paused reorganization job

RMC50470 DSNUNIT = NO
RMC50470 SIZEPC = (100,100)
RMC50470 UNIT = (SYSLDA,SYSLDA)
RMC50470 UNITCNT = (0,0)
RMC50470 VOLCNT = (25,25)
RMC50470 AVGVOLSP = (130000,TRK),(130000,TRK)
RMC50470 DATECLAS = (NONE,NONE)
RMC50470 UNLOAD = ((10,20),(0,0))
RMC50470 TIMEOUT = (25,25)
RMC50470 THRESHLD = 0
RMC50470 MAKEITZ = ((0,0),(0,0))
RMC50470 EXPDT =
RMC50470 DDTYPE = REMPLCPY
RMC50470 DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
RMC50470 REMPLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
RMC50470 REMPFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
RMC50470 REMPICPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
RMC50470 REMBFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
RMC50470 REMBLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
RMC50470 LOCPLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
RMC50470 LOCBLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
RMC50470 LOCBICPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
RMC50470 LOCBFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
RMC50470 LOCPFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
RMC50470 SYSPUNCH DSNPAT=&UID.&UTILPFX.&DDNAME
RMC50470 ARCHIVE DSNPAT=&UID.&UTILPFX.&DDNAME
RMC50470 SORTWORK DSNPAT=&UID.&UTILPFX.&DDNAME
RMC50470 WORK DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50470 UNLOAD DSNPAT=&UID..BMC.&TSIX.&DDNAME

BMC50483I GDGSCRAT = NO
BMC50483I GDGLIMIT = 5
BMC50483I RETPD =
BMC50483I EXPDT =
BMC50483I MAXEXTSZ = ((0,K),(0,K))
BMC50483I THRESHLD = 0
BMC50483I STORCLAS = (NONE,NONE)
BMC50483I MGMTCLAS = (NONE,NONE)
BMC50483I DATACLAS = (NONE,NONE)
BMC50483I DSNTYPE = (NONE,NONE)
BMC50483I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50483I VOLCNT = (25,25)
BMC50483I UNITCNT = (0,0)
BMC50483I UNIT = (SYSLDA,SYSLDA)
BMC50483I VOLCNT = (25,25)
BMC50483I AVGVOLSP = (130000,TRK),(130000,TRK)
BMC50483I DATECLAS = (NONE,NONE)
BMC50483I UNLOAD = ((10,20),(0,0))
BMC50483I TIMEOUT = (25,25)
BMC50483I THRESHLD = 0
BMC50483I MAKEITZ = ((0,0),(0,0))
BMC50483I EXPDT =
BMC50483I DDTYPE = LOCPLCPY
BMC50483I DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50483I LOCPLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50483I LOCBLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50483I LOCBICPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50483I LOCBFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50483I LOCPFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50483I SYSPUNCH DSNPAT=&UID.&UTILPFX.&DDNAME
BMC50483I ARCHIVE DSNPAT=&UID.&UTILPFX.&DDNAME
BMC50483I SORTWORK DSNPAT=&UID.&UTILPFX.&DDNAME
BMC50483I WORK DSNPAT=&UID..BMC.&TSIX.&DDNAME
BMC50483I UNLOAD DSNPAT=&UID..BMC.&TSIX.&DDNAME

BMC50470 DDTYPE = REMPLCPY
BMC50470 DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50470 REMPLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50470 REMPFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50470 REMPICPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50470 REMBFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50470 REMBLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50470 LOCPLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50470 LOCBLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50470 LOCBICPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50470 LOCBFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50470 LOCPFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50470 SYSPUNCH DSNPAT=&UID.&UTILPFX.&DDNAME
BMC50470 ARCHIVE DSNPAT=&UID.&UTILPFX.&DDNAME
BMC50470 SORTWORK DSNPAT=&UID.&UTILPFX.&DDNAME
BMC50470 WORK DSNPAT=&UID..BMC.&TSIX.&DDNAME
BMC50470 UNLOAD DSNPAT=&UID..BMC.&TSIX.&DDNAME

RMC50483I UNLOAD DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50483I WORK DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50483I SORTWORK DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50483I ARCHIVE DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50483I SYSPUNCH DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50483I UNLOAD DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50483I WORK DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50483I SORTWORK DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50483I ARCHIVE DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50483I SYSPUNCH DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50471I DDTYPE = LOCPLCPY
RMC50471I DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
RMC50471I LOCPLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
RMC50471I LOCBLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
RMC50471I LOCBICPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
RMC50471I LOCBFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
RMC50471I LOCPFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
RMC50471I SYSPUNCH DSNPAT=&UID.&UTILPFX.&DDNAME
RMC50471I ARCHIVE DSNPAT=&UID.&UTILPFX.&DDNAME
RMC50471I SORTWORK DSNPAT=&UID.&UTILPFX.&DDNAME
RMC50471I WORK DSNPAT=&UID..BMC.&TSIX.&DDNAME
RMC50471I UNLOAD DSNPAT=&UID..BMC.&TSIX.&DDNAME

BMC50471I DDTYPE = UNLOAD
BMC50471I DSNPAT='ARU.EXMPLO.ADDNAME'
BMC50471I DDTYPE = WORK
```
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.

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.

The following table describes the key command options for these jobs:

### Table 81: Key command options used in example 7

<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 (case 1)</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. In case 2, with ORDER YES specified, the clustering index is embedded in the SYSREC.</td>
</tr>
<tr>
<td>ORDER YES (case 2)</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. 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>

The following figure shows the JCL for case 1 of example 7:

**Figure 31: JCL for example 7, case 1**

```
// JOB
//**
//** SEGMENTED TABLESPACE, 3 INDEXES (ONE CLUSTERED, TWO NONCLUSTERED) *
//** CASE 1: REORG TABLESPACE, ORDER NO, ANALYZE ONLY. *
//**
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='db2ssid,EXMPL07A,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB DD  DISP=SHR,DSN=productLibraries
// DD  DISP=SHR,DSN=DB2.DSNEXIT
// DD  DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSIN DD  *
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//UPRINT DD SYSOUT=* 
//SYSDUMP DD SYSOUT=* 
//SYSTEM DD SYSOUT=* 
//*
//SYSSIN DD *
REORG TABLESPACE ARUDB250.TS250
UNLOAD CONTINUE 
ORDER NO 
ANALYZE ONLY 
/*
```
The following figure shows the JCL for case 2 of example 7:

**Figure 32: JCL for example 7, case 2**

```
//BMCREORG EXEC PGM=ARUUMAIN,
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//* CASE 2: REORG TABLESPACE, ORDER YES, ANALYZE ONLY.                *
//*                                                                   *
//* SEGMENTED TABLESPACE, 3 INDEXES (ONE CLUSTERED, TWO NONCLUSTERED) *
//BMCREORG EXEC PGM=ARUUMAIN,
//* PARM='db2ssid,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=*   
//SYSUDUMP DD SYSOUT=*   
//SYSTERM DD SYSOUT=*   
//SYSIN DD *           
// REORG TABLESPACE ARUDB250.TS250 ORDER YES ANALYZE ONLY
```

**SYSPRINT for example 7, case 1**

The following figure shows the SYSPRINT output for case 1 of example 7:

```
****** BMC REORG PLUS FOR DB2 V11R2.00 ******
REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
(C) COPYRIGHT 1988 - 2015 BMC SOFTWARE, INC.
***** BMC REORG PLUS FOR DB2 V11R2.00 ******

Example 7: ANALYZE ONLY to generate space estimates

Chapter 5 Examples of REORG PLUS jobs 445
Example 7: ANALYZE ONLY to generate space estimates

BMC50483I UNLOAD DSNPAT=UID..BMC..ATSIX.&DDNAME
BMC50483I WORK DSNPAT=UID..BMC..ATSIX.&DDNAME
BMC50483I SORTWK DSNPAT=UID..BMC..ATSIX.&DDNAME
BMC50483I ARCHIV DSNPAT=UID..BMC..ATSIX.&DDNAME
BMC50483I SYSUNCH DSNPAT=UID..BMC..ATSIX.&DDNAME
BMC50483I LOCICOPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I LOCICOPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I LOCICOPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I LOCICOPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I LOCICOPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I LOCICOPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I ARCHIVE DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I SORTWORK DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I WORK DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50471I DB2 GUNDEEP MODULE SETTINGS:
BMC50471I VERSION = 1
BMC50471I SUBSYS DEFAULT = SYSTEM
BMC50471I DEFAULT DEFAULT = SYSTEM
BMC50471I CHARACTER SET = ALPHANUM
BMC50471I DATE FORMAT = USA
BMC50471I TIME FORMAT = USA
BMC50471I LOCAL DATE LENGTH = 0
BMC50471I LOCAL TIME LENGTH = 0
BMC50471I DECIMAL ARITHMETIC = INT
BMC50471I DOLLAR SYMBOL = DOLLAREC
BMC50471I SQL ENCODING SCHEME = EBCDIC
BMC50471I APPL ENCODING SCHEME = EBCDIC
BMC50471I MIXED = NO
BMC50471I EBCDIC CSEID = (37,69534,69534)
BMC50471I ASCII CSEID = (38,69534,69534)
BMC50471I UNCODE CSEID = (387,1208,1208)
BMC50471I SIMPLIFIED TIME ZONE = CURRENT (-06:00)
BMC50482I DB2 MODE = NRM
BMC50471I BICUT1 = "BMCUTIL.CMN_BICUT1"
BMC50471I BICTEN = "BMCUTIL.CMN_BICTEN"
BMC50471I BICINT = "BMCUTIL.CMN_BICINT"
BMC50471I BICDOC = "BMCUTIL.CMN_BICDOC"
BMC50471I BMCECOPY = "BMCUTIL.CMN_BMCECOPY"
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I BMCXCOPY = 'BMCUTIL.CMN_BMCXCOPY'
BMC50471I BMCHIST = 'BMCUTIL.CMN_BMCHIST'
BMC50471I BMCSYNC = 'BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCUTIL = 'BMCUTIL.CMN_BMCUTIL'
BMC50483I REMBLCPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I REMPLCPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I REMBFCPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I REMPICPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I REMPFCPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I LOCBLCPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I LOCLCPLCOPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I LOCFLCOPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I LOCPLCOPY DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I ARCHIVE DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I SORTWORK DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I WORK DSNPAT=UID..BMC..ATSIX..FAPART..TATIME
BMC50483I UNLOAD DSNPAT=UID..BMC..ATSIX..FAPART..TATIME

Chapter 5  Examples of REORG PLUS jobs  447
Example 7: ANALYZE ONLY to generate space estimates

The following figure shows the SYSPRINT output for case 2 of example 7:

```sql
Example 7: ANALYZE ONLY to generate space estimates

The following figure shows the SYSPRINT output for case 2 of example 7:

```
Example 7: ANALYZE ONLY to generate space estimates
Example 7: ANALYZE ONLY to generate space estimates

BMC50483I UNLOAD DSNPAT=UID..BMC.&TSIX.&DDNAME
BMC50483I WORK DSNPAT=UID..BMC.&TSIX.&DDNAME
BMC50483I SORTWORK DSNPAT=UID..BMC.&TSIX.&DDNAME
BMC50483I ARCHIVE DSNPAT=UID..BMC.&TSIX.&DDNAME
BMC50483I GDGLIMIT 5
BMC50483I GDGEMPTY NO
BMC50483I GDGSCRAT NO
BMC50483I UNLOAD DSNPAT=&UID..BMC.&TSIX.&DDNAME
BMC50483I WORK DSNPAT=&UID..BMC.&TSIX.&DDNAME
BMC50483I SORTWORK DSNPAT=&UID.&UTILPFX.&DDNAME
BMC50483I ARCHIVE DSNPAT=&UID.&UTILPFX.&DDNAME
BMC50483I SYSPUNCH DSNPAT=&UID.&UTILPFX.&DDNAME
BMC50483I LOCPFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50483I LOCBFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50483I LOCBICPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50483I LOCPLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50483I LOCBLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50483I REMPFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50483I REMPICPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50483I REMBFCPY DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
BMC50483I REMBLCPY DSNPAT=&UID.&DDNAME.&TSIX..T&TIME
BMC50471I DB2 GSDHDECP MODULE SETTINGS:
BMC50471I VERSION 110
BMC50471I SUBSYSTEM DEFAULT DEHJ
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 (-06:00)
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 BMCXCOPY 'BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50102I REORG TABLESPACE ARUDB250.TS250
BMC50102I ORDER YES
BMC50102I ANALYZE ONLY
BMC50102I ORDER YES
BMC50102I ANALYZE ONLY
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.

The following table 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>REGISTER (ARUCPY1, ARUCPY3)</td>
<td>REORG PLUS makes four copies of the table space after reorganizing it.</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>
</tbody>
</table>

Chapter 5 Examples of REORG PLUS jobs 451
<table>
<thead>
<tr>
<th>Command options and DD statements used in JCL</th>
<th>Description</th>
</tr>
</thead>
</table>
| RECOVERYDDN (ARUCPY3, ARUCPY4)                | ■ 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  
■ Overrides the default names of BMCRCY and BMCRCZ |
| ARCHDDN (ARUARC)                              | ■ Specifies a ddname of ARUARC for the archive data set  
■ Overrides the default name of SYSARC |
| REDEFINE NO                                  | Tells REORG PLUS not to delete and redefine the VSAM data sets containing the DB2 objects  
Instead, REORG PLUS will reset the HURBA. |
| ORDER NO                                     | Performs no ordering at all, and the rows retain the order of the table before reorganization |
| SELECT * FROM                                | Specifies the table name that contains the rows and that only rows that meet the specified WHERE clause are to be unloaded and reloaded |
| WHERE                                        | Specifies the conditions that must be true for rows to be unloaded and reloaded |
| //SYSREC                                     | Contains the rows that you are reorganizing |
| //ARUCPY1                                     | The ddnames that are used for the data sets that receive full image copies of the table space that you are reorganizing |
| //ARUCPY2                                     |  |
| //ARUCPY3                                     |  |
| //ARUCPY4                                     |  |
| //ARUARC                                     | The ddname that is used for the data set that will contain the rows not selected |
| //SYSUT1                                     | The SYSUT1 data set to be allocated for the nonpartitioned secondary index |

The following figure shows the JCL for example 8:

**Figure 33: JCL for example 8**

```
//        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='db2ssid,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=*  
```
Example 8: Selective unload with discards to archive data set

**SYSPRINT for example 8**

The following figure shows the SYSPRINT output for example 8:

```
****** BMC REORG PLUS FOR DB2 V11R2.00 ******
```

...output details...

```
Chapter 5  Examples of REORG PLUS jobs   453
```
Example 8: Selective unload with discards to archive data set
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

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.

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.

The following table 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>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 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's are used.</td>
</tr>
<tr>
<td>SORTDATA</td>
<td>Enables this job to be compatible if run with the IBM DB2 REORG utility</td>
</tr>
<tr>
<td>ON FAILURE UNLOAD TERMINATE UTILITY RETCODE 31</td>
<td>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 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.</td>
</tr>
<tr>
<td>DDTYPE LOCBFCPY ACTIVE YES</td>
<td>Activates dynamic allocation for the local backup copy and the remote copies, overriding the installation defaults</td>
</tr>
<tr>
<td>DDTYPE REMPFCPY ACTIVE YES</td>
<td>Activates dynamic allocation for the local backup copy and the remote copies, overriding the installation defaults</td>
</tr>
<tr>
<td>DSNPAT</td>
<td>Specifies data set names for dynamic allocation that override the default patterns</td>
</tr>
<tr>
<td>//SYSREC</td>
<td>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.</td>
</tr>
</tbody>
</table>

The following figure shows the JCL for example 9:

**Figure 34: JCL for example 9**

```
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

ON FAILURE TESTING WITH ON FAILURE UNLOAD TERMINATE UTILITY RETCODE SPECIFIED IN REORG, REORG STARTED ALL TABLESPACES AND TERMINED THE UTILITY WITH A RETURN CODE 31. IN THIS CASE, RESTART IS NOT ALLOWED.

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 TERMINED THE UTILITY WITH A RETURN CODE 31. IN THIS CASE, RESTART IS NOT ALLOWED.

BMCREORG EXEC PGM=ARUUMAIN,
PARM='db2ssid,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**
**SYSPRINT DD SYSOUT=***

**SYSOUT DD SYSOUT=***

**UTPRINT DD SYSOUT=***

**SYSDUMP DD SYSOUT=***

**SYSTERM DD SYSOUT=***

**SYSREC DD DSN=ARU.EXMPL09.SYSREC,**
**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
DDSPAT 'ARU.EXMPL09.&DDNAME'
DDTYPE LOCPFCPY
DDSPAT 'ARU.EXMPL09.&DDNAME'
DDTYPE LOCBFCPY
DDSPAT 'ARU.EXMPL09.&DDNAME'
DDTYPE LOCPFCPY ACTIVE YES
DDSPAT 'ARU.EXMPL09.&DDNAME'
DDTYPE REMPFCPY ACTIVE YES
DDSPAT 'ARU.EXMPL09.&DDNAME'
DDTYPE REMPFCPY ACTIVE YES
DDSPAT 'ARU.EXMPL09.&DDNAME'

**SYSPRINT for example 9**

The following figure shows the SYSPRINT output for example 9:
Example 9: ON FAILURE with a user-specified return code

Chapter 5  Examples of REORG PLUS jobs  459
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.

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.

The following table 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>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 |
<table>
<thead>
<tr>
<th>Command options used in JCL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDER YES</td>
<td>Tells REORG PLUS to sort the rows. ORDER YES is passed to DSNUTILB as SORTDATA YES, and DSNUTILB will sort the clustering keys.</td>
</tr>
<tr>
<td>REDEFINE NO</td>
<td>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.</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>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.</td>
</tr>
<tr>
<td>SORTNUM</td>
<td>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.</td>
</tr>
<tr>
<td>DDTYPE LOCBFCPY ACTIVE YES</td>
<td>Activates dynamic data set allocation for the specified data set types.</td>
</tr>
<tr>
<td>DDTYPE REMPFCPY ACTIVE YES</td>
<td></td>
</tr>
<tr>
<td>DDTYPE REMBFCPY ACTIVE YES</td>
<td></td>
</tr>
<tr>
<td>IFALLOC</td>
<td>Because this job invokes DSNUTILB, REORG PLUS ignores this option.</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 patterns. Because this job invokes DSNUTILB, REORG PLUS includes this pattern in the TEMPLATE control statement that it builds for the DB2 REORG utility.</td>
</tr>
<tr>
<td>UNIT</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>MAXEXTSZ</td>
<td></td>
</tr>
<tr>
<td>UNITCNT</td>
<td></td>
</tr>
<tr>
<td>VOLCNT</td>
<td></td>
</tr>
<tr>
<td>THRESHLD</td>
<td></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>

The following figure shows the JCL for example 10:

**Figure 35: JCL for example 10**

```plaintext
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * J O B
```
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='db2ssid,ARUDB070.RG1,NEW,MSGLEVEL(1),ARU$OPTS'
//STEP1B DD DISP=SHR,DSN=productLibraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSPRINT DD SYSOUT=* SYSPRINT DD SYSOUT=* SYSPRINT DD SYSOUT=* SYSTERM DD SYSOUT=* //SYSin DD *
REORG TABLESPACE ARUDB070.LARS$JBA SHRLEVEL NONE UNLOAD CONTINUE COPY YES INLINE YES COPYVL FULL COPYDDN BMCOPY RECOVERYDDN BMRYP PREFERENCES YES REDEFINE NO ORDER YES UPDATEDB2STATS YES SORTDEVT 3390 SORTNUM 12 SORTDATA DDTYPE UNLOAD UNIT (WORK) IFALLOC USE DSNPAT 'ARU.LARREORG.DEHJ.ARUDB070.&DDNAME' MAXEXTSZ 100 UNITCNT (1,15) VOLCNT AUTO DDTYPE WORK UNIT (WORK) IFALLOC USE DSNPAT 'ARU.LARREORG.DEHJ.ARUDB070.&DDNAME' DDTYPE LOCPFCPY IFALLOC USE DSNPAT 'ARU.LARREORG.DEHJ.&DB.&TSIX.&DDNAME' UNIT (WORK) THRESHLD 10000 DDTYPE LOCBFCPY ACTIVE YES IFALLOC USE DSNPAT 'ARU.LARREORG.DEHJ.&DB.&TSIX.&DDNAME' UNIT (WORK) DDTYPE REMPFCPY ACTIVE YES UNIT (WORK) IFALLOC USE DSNPAT 'ARU.LARREORG.DEHJ.&DB.&TSIX.&DDNAME' DDTYPE REMPBCPY ACTIVE YES UNIT (WORK) IFALLOC USE DSNPAT 'ARU.LARREORG.DEHJ.&DB.&TSIX.&DDNAME' /* SYSPRINT for example 10 */

The following figure shows the SYSPRINT output for example 10:

**** BMC R E O R G   P L U S   F O R   D B 2   V11R2.00  *****

---

**SYSPRINT for example 10**

The following figure shows the SYSPRINT output for example 10:
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

Chapter 5  Examples of REORG PLUS jobs  465
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

REOR PLUS for DB2 Reference Manual
Example 10: DSNUTILB reorganization and index that contains keys with random ordering

Chapter 5 Examples of REORG PLUS jobs  467
Example 11: Partition-by-growth table space

In this example, REORG PLUS reorganizes a partition-by-growth table space.

The job 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.

The following table describes the key command options for this job:

Table 85: Key command options used in example 11

<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>
</tbody>
</table>
### Command options used in JCL

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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. 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, UPDATEDB2STATS YES</td>
<td>Tells REORG PLUS to update statistics in the DASD MANAGER PLUS database statistics tables and the DB2 catalog. The SYSPRINT shows the messages that the Common Statistics component displays for the statistics being updated.</td>
</tr>
</tbody>
</table>

The following figure shows the JCL for example 11:

**Figure 36: JCL for example 11**

```bash
//          JOB
//          /*
//          /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//          /* FULL UNIVERSAL (PARTITION BY GROWTH) TABLESPACE REORG WITH       *
//          /* THE USE OF THE MAXNEWPARTS OPTION.                              *
//          /* * PARTITIONS ADDED DURING REORG (SEE REORG MESSAGES BMC50175I). *
//          /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//LARREORG EXEC PGM=ARUUMAIN,REGION=0M,COND=(7,LT),
// PARM='db2ssid,RAJRGD3.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=*    
//UTPRINT   DD  SYSOUT=*  
//SYSTERM   DD  SYSOUT=*  
//*
//SYSIN     DD  *
REORG TABLESPACE ARUJGD3.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'
```
**SYSPRINT for example 11**

The following figure shows the SYSPRINT output for example 11:

```
****** BMC REORG PLUS FOR DB2 ******
(C) COPYRIGHT 1988 - 2015 BMC SOFTWARE, INC.
***  BMC REORG PLUS FOR DB2  ***

Example 11: Partition-by-growth table space

```

**Example 11: Partition-by-growth table space**

The following figure shows the SYSPRINT output for example 11:
<table>
<thead>
<tr>
<th>BMC50470I MAKEITSZ</th>
<th>(0,K),(0,K)</th>
<th>(0,K),(0,K)</th>
<th>(0,K),(0,K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50470I EXPDT</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I RETSO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC50470I GDGLIMIT</td>
<td>S</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 GDSCRAT</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BMC50470I DTYPE</td>
<td>LOCBCPFPY</td>
<td>LOCBCPFPY</td>
<td>REMBFCPY</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 ATTLOC</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 SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I UNIT</td>
<td>(SYSSALDA,SYSSALDA)</td>
<td>(SYSSALDA,SYSSALDA)</td>
<td>(SYSSALDA,SYSSALDA)</td>
</tr>
<tr>
<td>BMC50470I UNITCNT</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>BMC50470I AVOIDSLP</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I DSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STOCLS</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 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I AVOIDSLP</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I DSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STOCLS</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 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I AVOIDSLP</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I DSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STOCLS</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 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I AVOIDSLP</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I DSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STOCLS</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 UNIT</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(SYSALLDA,SYSALLDA)</td>
</tr>
<tr>
<td>BMC50470I SIZEPCT</td>
<td>(100,100)</td>
<td>(100,100)</td>
<td>(100,100)</td>
</tr>
<tr>
<td>BMC50470I AVOIDSLP</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
<td>(((30000,TRK),(30000,TRK))</td>
</tr>
<tr>
<td>BMC50470I DSTYPE</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I DATACLS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I MGMTCLS</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
<td>(NONE,NONE)</td>
</tr>
<tr>
<td>BMC50470I STOCLS</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>
</tbody>
</table>

Example 11: Partition-by-growth table space

472 **REORG PLUS for DB2 Reference Manual**
Chapter 5  Examples of REORG PLUS jobs    473

Example 11: Partition-by-growth table space
Example 11: Partition-by-growth table space

REORG PLUS for DB2 Reference Manual

474
Example 11: Partition-by-growth table space

Chapter 5 Examples of REORG PLUS jobs 475

Example 11: Partition-by-growth table space
<table>
<thead>
<tr>
<th>TABLESPACE</th>
<th>ARIUGD3.LARSZABA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HISTORY</strong></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>PARTITIONS</td>
<td></td>
</tr>
<tr>
<td>OSIZE</td>
<td>1048576</td>
</tr>
<tr>
<td>MAXROWS</td>
<td>10</td>
</tr>
<tr>
<td>LOB</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>NACTIVE</td>
<td>2619042</td>
</tr>
<tr>
<td>NWUNKNOWN</td>
<td>10</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td></td>
</tr>
<tr>
<td>SPACE</td>
<td>18048000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLEPART</th>
<th>ARIUGD3.LARSZABA PART 001</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HISTORY</strong></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>ALLCOVER</td>
<td>C</td>
</tr>
<tr>
<td>RUN_FORMAT</td>
<td>RRF</td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>CARD</td>
<td>130986</td>
</tr>
<tr>
<td>FULL</td>
<td>638</td>
</tr>
<tr>
<td>PCTFREE</td>
<td>50</td>
</tr>
<tr>
<td>PCTPAGES</td>
<td>100</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td>1</td>
</tr>
<tr>
<td>SPACE</td>
<td>2185700</td>
</tr>
</tbody>
</table>

| EXTENTS | 21 VOLCOUNTER 2 DEVTYPE 3390 VOLUME 18048000 NUMDATASETS 1 |

| **HISTORY** |                           |
| SAVESTATS  | Y                          |
| SAMPLING   | N                          |
| PARTITIONS |                            |
| OSIZE      | 1048576                    |
| MAXROWS    | 10                         |
| LOB        |                            |
| N          |
| STATISTICS |                            |
| NACTIVE    | 2619042                    |
| NWUNKNOWN  | 10                         |
| ALLOCATION |                            |
| SPACE      | 18048000                   |

<table>
<thead>
<tr>
<th>TABLEPART</th>
<th>ARIUGD3.LARSZABA PART 002</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HISTORY</strong></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>ALLCOVER</td>
<td>C</td>
</tr>
<tr>
<td>RUN_FORMAT</td>
<td>RRF</td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>CARD</td>
<td>130986</td>
</tr>
<tr>
<td>FULL</td>
<td>638</td>
</tr>
<tr>
<td>PCTFREE</td>
<td>50</td>
</tr>
<tr>
<td>PCTPAGES</td>
<td>100</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td></td>
</tr>
<tr>
<td>SPACE</td>
<td>2185700</td>
</tr>
</tbody>
</table>

| EXTENTS | 19 VOLCOUNTER 3 DEVTYPE 3390 VOLUME 18048000 NUMDATASETS 1 |

| **HISTORY** |                           |
| SAVESTATS  | Y                          |
| SAMPLING   | N                          |
| PARTITIONS |                            |
| OSIZE      | 1048576                    |
| MAXROWS    | 10                         |
| LOB        |                            |
| N          |
| STATISTICS |                            |
| NACTIVE    | 2619042                    |
| NWUNKNOWN  | 10                         |
| ALLOCATION |                            |
| SPACE      | 18048000                   |

<table>
<thead>
<tr>
<th>TABLEPART</th>
<th>ARIUGD3.LARSZABA PART 003</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HISTORY</strong></td>
<td></td>
</tr>
<tr>
<td>SAVESTATS</td>
<td>Y</td>
</tr>
<tr>
<td>SAMPLING</td>
<td>N</td>
</tr>
<tr>
<td>PARTITIONS</td>
<td></td>
</tr>
<tr>
<td>OSIZE</td>
<td>1048576</td>
</tr>
<tr>
<td>MAXROWS</td>
<td>10</td>
</tr>
<tr>
<td>LOB</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>NACTIVE</td>
<td>2619042</td>
</tr>
<tr>
<td>NWUNKNOWN</td>
<td>10</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td></td>
</tr>
<tr>
<td>SPACE</td>
<td>18048000</td>
</tr>
</tbody>
</table>

| EXTENTS | 16 VOLCOUNTER 3 DEVTYPE 3390 VOLUME 18048000 NUMDATASETS 1 |

| **HISTORY** |                           |
| SAVESTATS  | Y                          |
| SAMPLING   | N                          |
| PARTITIONS |                            |
| OSIZE      | 1048576                    |
| MAXROWS    | 10                         |
| LOB        |                            |
| N          |
| STATISTICS |                            |
| NACTIVE    | 2619042                    |
| NWUNKNOWN  | 10                         |
| ALLOCATION |                            |
| SPACE      | 18048000                   |

<table>
<thead>
<tr>
<th>TABLEPART</th>
<th>ARIUGD3.LARSZABA PART 004</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HISTORY</strong></td>
<td></td>
</tr>
<tr>
<td>SAVESTATS</td>
<td>Y</td>
</tr>
<tr>
<td>SAMPLING</td>
<td>N</td>
</tr>
<tr>
<td>PARTITIONS</td>
<td></td>
</tr>
<tr>
<td>OSIZE</td>
<td>1048576</td>
</tr>
<tr>
<td>MAXROWS</td>
<td>10</td>
</tr>
<tr>
<td>LOB</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>NACTIVE</td>
<td>2619042</td>
</tr>
<tr>
<td>NWUNKNOWN</td>
<td>10</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td></td>
</tr>
<tr>
<td>SPACE</td>
<td>18048000</td>
</tr>
</tbody>
</table>

| EXTENTS | 19 VOLCOUNTER 4 DEVTYPE 3390 VOLUME 18048000 NUMDATASETS 1 |

| **HISTORY** |                           |
| SAVESTATS  | Y                          |
| SAMPLING   | N                          |
| PARTITIONS |                            |
| OSIZE      | 1048576                    |
| MAXROWS    | 10                         |
| LOB        |                            |
| N          |
| STATISTICS |                            |
| NACTIVE    | 2619042                    |
| NWUNKNOWN  | 10                         |
| ALLOCATION |                            |
| SPACE      | 18048000                   |

<table>
<thead>
<tr>
<th>TABLEPART</th>
<th>ARIUGD3.LARSZABA PART 005</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HISTORY</strong></td>
<td></td>
</tr>
<tr>
<td>SAVESTATS</td>
<td>Y</td>
</tr>
<tr>
<td>SAMPLING</td>
<td>N</td>
</tr>
<tr>
<td>PARTITIONS</td>
<td></td>
</tr>
<tr>
<td>OSIZE</td>
<td>1048576</td>
</tr>
<tr>
<td>MAXROWS</td>
<td>10</td>
</tr>
<tr>
<td>LOB</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>NACTIVE</td>
<td>2619042</td>
</tr>
<tr>
<td>NWUNKNOWN</td>
<td>10</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td></td>
</tr>
<tr>
<td>SPACE</td>
<td>18048000</td>
</tr>
</tbody>
</table>

| EXTENTS | 26 VOLCOUNTER 2 DEVTYPE 3390 VOLUME 18048000 NUMDATASETS 1 |

| **HISTORY** |                           |
| SAVESTATS  | Y                          |
| SAMPLING   | N                          |
| PARTITIONS |                            |
| OSIZE      | 1048576                    |
| MAXROWS    | 10                         |
| LOB        |                            |
| N          |
| STATISTICS |                            |
| NACTIVE    | 2619042                    |
| NWUNKNOWN  | 10                         |
| ALLOCATION |                            |
| SPACE      | 18048000                   |

<table>
<thead>
<tr>
<th>TABLEPART</th>
<th>ARIUGD3.LARSZABA PART 006</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HISTORY</strong></td>
<td></td>
</tr>
<tr>
<td>SAVESTATS</td>
<td>Y</td>
</tr>
<tr>
<td>SAMPLING</td>
<td>N</td>
</tr>
<tr>
<td>PARTITIONS</td>
<td></td>
</tr>
<tr>
<td>OSIZE</td>
<td>1048576</td>
</tr>
<tr>
<td>MAXROWS</td>
<td>10</td>
</tr>
<tr>
<td>LOB</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>STATISTICS</td>
<td></td>
</tr>
<tr>
<td>NACTIVE</td>
<td>2619042</td>
</tr>
<tr>
<td>NWUNKNOWN</td>
<td>10</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td></td>
</tr>
<tr>
<td>SPACE</td>
<td>18048000</td>
</tr>
</tbody>
</table>

| EXTENTS | 26 VOLCOUNTER 2 DEVTYPE 3390 VOLUME 18048000 NUMDATASETS 1 |
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 limit-key updates 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.

The following table describes the key command options, installation options, and DD statements for this job:

<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>Command options and DD statements used in JCL; installation options</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>-------------</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. 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>
<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>
Command options and DD statements used in JCL; installation options

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//BMCCPY</td>
</tr>
<tr>
<td>//BMCCPZ</td>
</tr>
<tr>
<td>//BMRCY</td>
</tr>
<tr>
<td>//BMRCZ</td>
</tr>
<tr>
<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</td>
</tr>
<tr>
<td>//BMCICZ</td>
</tr>
<tr>
<td>//BMCIRY</td>
</tr>
<tr>
<td>//BMCIRZ</td>
</tr>
<tr>
<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>

The following figure shows the JCL for example 12:

**Figure 37: JCL for example 12**

```sql
// 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='db2ssid,LABCOPB.LARDBXAB.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=* 
// SYSPRINT DD SYSPRINT=* 
// SYSPRINT DD SYSPRINT=* 
// SYSPRINT DD SYSPRINT=* 
// 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 
// BMRCY DD DSN=ARU.EXMPL12.BMRCY, 
// DISP=(,CATLG), 
// SPACE=(CYL,(50,20),RLSE),UNIT=WORK 
// BMRCZ DD DSN=ARU.EXMPL12.BMRCZ, 
// 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 * 
```
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

The following figure shows the SYSPRINT output for example 12:

```
REORG TABLESPACE LARDBXAB.LARXABA
SHRLEVEL CHANGE
UNLOAD CONTINUE
XBMID XBM
FASTSWITCH YES
COPY YES
ORDER YES
DEADLINE 2015-03-05-19.30.00.00
MAXRO 600
DELAY 300
LOGSPELL (1024,1024)
SPIPPDASNPT 'ARU.LARWORK.DEFF'
SPIPLLUNIT NONE
SPIPLSTORCLAS 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 LOCPCPY 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');
```

Chapter 5  Examples of REORG PLUS jobs  481
<table>
<thead>
<tr>
<th>BMC504071</th>
<th>DTYPE</th>
<th>ALLOC</th>
<th>SMSUNIT</th>
<th>UNIT</th>
<th>UNICTY</th>
<th>VOLCNT</th>
<th>AVGDSIZE</th>
<th>GDGEMPTY</th>
<th>GDGLIMIT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UNLOAD</td>
<td>N/A</td>
<td>NO</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(0,0)</td>
<td>(25,25)</td>
<td>((0,0),(0,0))</td>
<td>NO</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>WORK</td>
<td>N/A</td>
<td>NO</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(0,0)</td>
<td>(25,25)</td>
<td>((0,0),(0,0))</td>
<td>NO</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>WORK</td>
<td>N/A</td>
<td>NO</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(0,0)</td>
<td>(25,25)</td>
<td>((0,0),(0,0))</td>
<td>NO</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SORTWORK</td>
<td>N/A</td>
<td>NO</td>
<td>(SYSALLDA,SYSALLDA)</td>
<td>(0,0)</td>
<td>(25,25)</td>
<td>((0,0),(0,0))</td>
<td>NO</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning**

**REORG PLUS for DB2 Reference Manual**
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

BMC54831 UNLOAD DSNPAT=UID..BMC..DDNAME
BMC54831 WORK DSNPAT=UID..BMC..DDNAME
BMC54831 ARCHIVE DSNPAT=UID..BMC..DDNAME
BMC54831 SYSUPCH DSNPAT=UID..BMC..DDNAME
BMC54831 LOCPFCY DSNPAT=UID..BMC..DDNAME
BMC54831 LOCPICY DSNPAT=UID..BMC..DDNAME
BMC54831 REMPFCY DSNPAT=UID..BMC..DDNAME
BMC54831 REMPICY DSNPAT=UID..BMC..DDNAME
BMC54831 LRBPFCY DSNPAT=UID..BMC..DDNAME
BMC54831 LRBPICY DSNPAT=UID..BMC..DDNAME
BMC54831 SYSPUNCH DSNPAT=UID..BMC..DDNAME
BMC54831 ARCHIVE DSNPAT=UID..BMC..DDNAME
BMC54831 SORTWORK DSNPAT=UID..BMC..DDNAME
BMC54831 WORK DSNPAT=UID..BMC..DDNAME
BMC54831 UNLOAD DSNPAT=UID..BMC..DDNAME

Chapter 5  Examples of REORG PLUS jobs    483
Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

484 REORG PLUS for DB2 Reference Manual
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.

The following table describes the key command options for this job:
Table 87: Command options used in example 13

<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 300</td>
<td>Tells REORG PLUS to start the LOGFINAL phase when the estimated time to apply records in the LOGFINAL phase is less than 300 seconds</td>
</tr>
<tr>
<td>RBALRSN_CONVERSION EXTENDED</td>
<td>Tells REORG PLUS to convert to extended (10-byte) format any object whose RBA or LRSN is in basic (6-byte) format</td>
</tr>
<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</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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>Command options used in JCL</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------</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>

The following figure shows the JCL for example 13:

**Figure 38: 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='db2ssid,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
//        SYSIN DD *
//        SYSIN DD *
//        REORG TABLESPACE LARDBXBB.LARSXBBA
//        SHRLEVEL CHANGE
//        XBMID XMB
//        FASTSWITCH YES
//        ORDER YES
//        MAXRO 300
//        RBALRSN_CONVERSION EXTENDED
//        LONGLOG TERM
//        DELAY 300
//        RIDMAPMEM 20480
//        LOGMEM 30720
//        LOGSPILL (1024,1024)
//        SPILLDSNPAT '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'
```
## SYSPRINT for example 13

The following figure shows the SYSPRINT output for example 13:

```
***** BMC REORG PLUS FOR DB2 V112.00 *****
(C) COPYRIGHT 1988 - 2015 BMC SOFTWARE, INC.
BMC50001I UTILITY EXECUTION STARTING 3/09/2015 9:27:47 ...
BMC50001I UTILITY ID = "LARCONS.LARDDB" DB2 SUBSYSTEM ID = "DEUX". OPTION MODULE = "ARUSORTS".
BMC50471I REORG PLUS FOR DB2 V112.00
BMC50471I DB2 UTILITIES COMMON CODE V112.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE V112.00
BMC50471I DB2 REORG PLUS FOR DB2 V112.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE V112.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE V112.00
BMC50471I NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE V112.00

DDTYPE LOCPFCPY UNIT (WORK) SIZEPCT (150,150)
DSNAP "ARU.EXAMPL3..&BB..T5X..&DNAME"

Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

### Example 13 Details

```
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

```

### SYSPRINT Output

```
```

### Description

The SYSPRINT output for example 13 includes detailed information about the REORG PLUS job execution, including statistics and messages related to the SHRLEVEL CHANGE with LONGLOG and DELAY options. The output illustrates the impact of these options on the DB2 environment, highlighting performance metrics and resource usage.

### Additional Information

- **SYSPRINT Details**: The SYSPRINT output provides a comprehensive log of the job execution, including error messages, status updates, and performance statistics.
- **Example 13 Parameters**: SHRLEVEL CHANGE with LONGLOG and DELAY options are specified in the SYSPRINT output, indicating their role in the example.

---

**Chapter 5 Examples of REORG PLUS jobs**

489
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

REORG PLUS for DB2 Reference Manual
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

Chapter 5  Examples of REORG PLUS jobs  491
Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

Chapter 5  Examples of REORG PLUS jobs  493
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 JCL in this example includes both the REBALANCE option and a DD statement for a DDLIN data set. The ALTER statements in the DDLIN data set take precedence over the REBALANCE option to define new partition boundaries. Message 51297I shows that the limit keys have been altered.

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.

The following table 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>ORDER YES</td>
<td>Sorts the data rows</td>
</tr>
<tr>
<td>REBALANCE</td>
<td>Normally defines new partition boundaries and evenly redistributes rows across the reorganized partitions</td>
</tr>
<tr>
<td></td>
<td>Because the JCL also includes a DD statement for a DDLIN data set, the DDLIN data set takes precedence over the REBALANCE option</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 INLINECP installation option is YES, REORG PLUS creates an inline image copy</td>
</tr>
<tr>
<td>Command options used in JCL</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------</td>
</tr>
<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>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>

The following figure shows the JCL for example 14:

**Figure 39: JCL for example 14**

```sql
// JOB
// PARTITIONED TABLESPACE THAT USES TABLE-BASED PARTITIONING. *
// DATA-PARTITIONED SECONDARY INDEX (DPSI) AND NON-PARTITIONED *
// SECONDARY (NPSI) INDEXES DEFINED. *
/// THE TABLESPACE HAS 8 PARTS. THE REORG INCLUDES BOTH DDLIN AND *
// REBALANCE. *
/// LARREORG EXEC PGM=ARUUMAIN,REGION=0M,COND=(7,LT),
// PARM='db2ssid,EXM14B.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=* 
// SYSPRINT DD SYSOUT=* 
// UTPRINT DD SYSOUT=* 
// SYSDUMP DD DSN=ARU.LARWORK.DEFF.EXM14B.LARSA1BA.SYSDUMP,
// DISP=(,CATLG),
// SPACE=(CYL,(1,100),RLSE),UNIT=WORK,FREE=CLOSE
// SYSSERR DD DSN=ARU.LARWORK.DEFF.EXM14B.LARSA1BA.SYSSERR,
// DISP=(,CATLG),
// SPACE=(CYL,(1,100),RLSE),UNIT=WORK,FREE=CLOSE
// SYSSIN DD *
// REORG TABLESPACE EXM14B.LARSA1BA
// XBMID XBMB SHRLEVEL CHANGE
// COPY YES
// ORDER YES
// ANALYZE SAMPLE
// REBALANCE
// FASTSWITCH YES
// DDTYPE UNLOAD
// UNIT (WORK)
DSNPAT 'ARU.LARREORG.DEFF.&DB.&TSIX.&DDNAME'
// DDTYPE WORK
// UNIT (WORK)
DSNPAT 'ARU.LARREORG.DEFF.&DB.&TSIX.&DDNAME'
// DDTYPE LOCPFCPY
// UNIT (WORK)
DSNPAT 'ARU.LARREORG.DEFF.&DB.&TSIX.&DDNAME'
// DDLIN DD *
ALTER TABLE EXM14B.LART001
ALTER PARTITION 1 ENDING AT ('2049-12-31-00.00.00.000000')
//*
```
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

The following figure shows the SYSPRINT output for example 14:
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

REORG PLUS for DB2 Reference Manual
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

REORG PLUS for DB2 Reference Manual
Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

502 REORG PLUS for DB2 Reference Manual
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 has a pending buffer pool ALTER. During the reorganization, REORG PLUS materializes the pending definition change.

Message 50138I shows which options have been changed because pending definition changes exist on the table space.

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.

The following table describes the key command options for this job:

Table 89: Key command options used in example 15

<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>COPY YES</td>
<td>Creates a DB2 image copy of the table space</td>
</tr>
<tr>
<td></td>
<td>Because the INLINECP installation option is YES, REORG PLUS creates an inline image copy.</td>
</tr>
<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>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>

The following figure shows the JCL for example 15:

Figure 40: JCL for example 15

```
// JOB
//******************************
//** PARTITIONED TABLE SPACE (UTS)  
//** WITH PENDING DDL - ALTER BUFFERPOOL ISSUED PRIOR TO REORG  
//******************************
//LARREORG EXEC PGM=ARUUMAIN,REGION=0M,COND=(7,LT),  
// PARM='db2ssid,EXM15B.RG1,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB DD DISP=SHR,DSN=productLibraries  
// DD DISP=SHR,DSN=productLibraries  
// SYSPRINT DD SYSOUT=*  
// DD SYSOUT=*
//UTPRINT DD SYSOUT=*  
//SYSMDUMP DD DSN=ARU.LARWORK.DEHJ.EXM15B.LARSA1BA.SYSMDUMP  
```
Example 15 - Materializing pending definition changes

SYSPRINT for example 15

The following figure shows the SYSPRINT for example 15.

Note

To reduce the length of the output in this book, the figure shows one set of statistics for each object type. Ellipses (...) at the beginning of the line indicate the omitted statistics sections.
### Example 15 - Materializing pending definition changes

<table>
<thead>
<tr>
<th>BMC50471</th>
<th>DSPOOLS=NOBR*TRAIL</th>
<th>PROFFORMAT=NO</th>
<th>UNLOAD=300%</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471</td>
<td>DSRESET=(NONE,NOEX)</td>
<td>RIVICON=DE.cfg</td>
<td>BMCE10 (BMCE2)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>IECOMM=(BMCE2,BMCE2)</td>
<td>BMCE0=BMCE1</td>
<td>BMCE10=BMCE2</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FASTINT=NO</td>
<td>RDEFINE=YES</td>
<td>WRUFF=(20,10)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FILEGR-WARN</td>
<td>REWRITE=NO</td>
<td>WORKDIR=SYS11</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FORCE=START (3)</td>
<td>RMDMDSZ=200152</td>
<td>WORKUNIT=SYS11</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FORCE=STOP</td>
<td>RMDMDSZ=</td>
<td>WORKUNIT=SYS11</td>
</tr>
<tr>
<td>BMC50471</td>
<td>TSFILEBACK=NO</td>
<td>RMDMDSZ=</td>
<td>ZIPENABLED</td>
</tr>
<tr>
<td>BMC50471</td>
<td>PLAN=ANUDA</td>
<td>RMDMDSZ=</td>
<td>ZIPENABLED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BMC50471</th>
<th>DSPOOLS=NOBR*TRAIL</th>
<th>PROFFORMAT=NO</th>
<th>UNLOAD=300%</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC50471</td>
<td>DSRESET=(NONE,NOEX)</td>
<td>RIVICON=DE.cfg</td>
<td>BMCE10 (BMCE2)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>IECOMM=(BMCE2,BMCE2)</td>
<td>BMCE0=BMCE1</td>
<td>BMCE10=BMCE2</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FASTINT=NO</td>
<td>RDEFINE=YES</td>
<td>WRUFF=(20,10)</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FILEGR-WARN</td>
<td>REWRITE=NO</td>
<td>WORKDIR=SYS11</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FORCE=START (3)</td>
<td>RMDMDSZ=200152</td>
<td>WORKUNIT=SYS11</td>
</tr>
<tr>
<td>BMC50471</td>
<td>FORCE=STOP</td>
<td>RMDMDSZ=</td>
<td>WORKUNIT=SYS11</td>
</tr>
<tr>
<td>BMC50471</td>
<td>TSFILEBACK=NO</td>
<td>RMDMDSZ=</td>
<td>ZIPENABLED</td>
</tr>
<tr>
<td>BMC50471</td>
<td>PLAN=ANUDA</td>
<td>RMDMDSZ=</td>
<td>ZIPENABLED</td>
</tr>
</tbody>
</table>

### REORG PLUS for DB2 Reference Manual

506
Example 15 - Materializing pending definition changes

Chapter 5  Examples of REORG PLUS jobs  507
Example 15 - Materializing pending definition changes

REORG PLUS for DB2 Reference Manual
### Example 15 - Materializing pending definition changes

**BMC51941**

<table>
<thead>
<tr>
<th>Table</th>
<th>Exm15b.Lart001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Exm15b.Larsa1ba</td>
</tr>
<tr>
<td>Indexpart</td>
<td>Exm15b.Larxpc1a</td>
</tr>
<tr>
<td>Space</td>
<td>270</td>
</tr>
<tr>
<td>Space_KB</td>
<td>12960</td>
</tr>
<tr>
<td>Allocation</td>
<td></td>
</tr>
<tr>
<td>High2K</td>
<td>X'4460070200000000'</td>
</tr>
<tr>
<td>Low2K</td>
<td>X'1910020200000000'</td>
</tr>
<tr>
<td>NumNonLeaf</td>
<td>3</td>
</tr>
<tr>
<td>NLevels</td>
<td>2</td>
</tr>
<tr>
<td>Clusterratio</td>
<td>97</td>
</tr>
<tr>
<td>NLeaf</td>
<td>10</td>
</tr>
<tr>
<td>Sampling</td>
<td>N</td>
</tr>
<tr>
<td>SaveStats</td>
<td>N</td>
</tr>
<tr>
<td>Updatedb2</td>
<td>Y</td>
</tr>
<tr>
<td>Utilcode</td>
<td>R</td>
</tr>
<tr>
<td>Location</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

**BMC51942**

<table>
<thead>
<tr>
<th>Table</th>
<th>Exm15b.Lart001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Exm15b.Larsa1ba</td>
</tr>
<tr>
<td>Indexpart</td>
<td>Exm15b.Larxpc1a</td>
</tr>
<tr>
<td>Space</td>
<td>270</td>
</tr>
<tr>
<td>Space_KB</td>
<td>12960</td>
</tr>
<tr>
<td>Allocation</td>
<td></td>
</tr>
<tr>
<td>High2K</td>
<td>X'4460070200000000'</td>
</tr>
<tr>
<td>Low2K</td>
<td>X'1910020200000000'</td>
</tr>
<tr>
<td>NumNonLeaf</td>
<td>3</td>
</tr>
<tr>
<td>NLevels</td>
<td>2</td>
</tr>
<tr>
<td>Clusterratio</td>
<td>97</td>
</tr>
<tr>
<td>NLeaf</td>
<td>10</td>
</tr>
<tr>
<td>Sampling</td>
<td>N</td>
</tr>
<tr>
<td>SaveStats</td>
<td>N</td>
</tr>
<tr>
<td>Updatedb2</td>
<td>Y</td>
</tr>
<tr>
<td>Utilcode</td>
<td>R</td>
</tr>
<tr>
<td>Location</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

**BMC51943**

<table>
<thead>
<tr>
<th>Table</th>
<th>Exm15b.Lart001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Exm15b.Larsa1ba</td>
</tr>
<tr>
<td>Indexpart</td>
<td>Exm15b.Larxpc1a</td>
</tr>
<tr>
<td>Space</td>
<td>270</td>
</tr>
<tr>
<td>Space_KB</td>
<td>12960</td>
</tr>
<tr>
<td>Allocation</td>
<td></td>
</tr>
<tr>
<td>High2K</td>
<td>X'4460070200000000'</td>
</tr>
<tr>
<td>Low2K</td>
<td>X'1910020200000000'</td>
</tr>
<tr>
<td>NumNonLeaf</td>
<td>3</td>
</tr>
<tr>
<td>NLevels</td>
<td>2</td>
</tr>
<tr>
<td>Clusterratio</td>
<td>97</td>
</tr>
<tr>
<td>NLeaf</td>
<td>10</td>
</tr>
<tr>
<td>Sampling</td>
<td>N</td>
</tr>
<tr>
<td>SaveStats</td>
<td>N</td>
</tr>
<tr>
<td>Updatedb2</td>
<td>Y</td>
</tr>
<tr>
<td>Utilcode</td>
<td>R</td>
</tr>
<tr>
<td>Location</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

**BMC51944**

<table>
<thead>
<tr>
<th>Table</th>
<th>Exm15b.Lart001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Exm15b.Larsa1ba</td>
</tr>
<tr>
<td>Indexpart</td>
<td>Exm15b.Larxpc1a</td>
</tr>
<tr>
<td>Space</td>
<td>270</td>
</tr>
<tr>
<td>Space_KB</td>
<td>12960</td>
</tr>
<tr>
<td>Allocation</td>
<td></td>
</tr>
<tr>
<td>High2K</td>
<td>X'4460070200000000'</td>
</tr>
<tr>
<td>Low2K</td>
<td>X'1910020200000000'</td>
</tr>
<tr>
<td>NumNonLeaf</td>
<td>3</td>
</tr>
<tr>
<td>NLevels</td>
<td>2</td>
</tr>
<tr>
<td>Clusterratio</td>
<td>97</td>
</tr>
<tr>
<td>NLeaf</td>
<td>10</td>
</tr>
<tr>
<td>Sampling</td>
<td>N</td>
</tr>
<tr>
<td>SaveStats</td>
<td>N</td>
</tr>
<tr>
<td>Updatedb2</td>
<td>Y</td>
</tr>
<tr>
<td>Utilcode</td>
<td>R</td>
</tr>
<tr>
<td>Location</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

---

**Example 15 - Materializing pending definition changes**

**BMC51945**

<table>
<thead>
<tr>
<th>Table</th>
<th>Exm15b.Lart001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Exm15b.Larsa1ba</td>
</tr>
<tr>
<td>Indexpart</td>
<td>Exm15b.Larxpc1a</td>
</tr>
<tr>
<td>Space</td>
<td>270</td>
</tr>
<tr>
<td>Space_KB</td>
<td>12960</td>
</tr>
<tr>
<td>Allocation</td>
<td></td>
</tr>
<tr>
<td>High2K</td>
<td>X'4460070200000000'</td>
</tr>
<tr>
<td>Low2K</td>
<td>X'1910020200000000'</td>
</tr>
<tr>
<td>NumNonLeaf</td>
<td>3</td>
</tr>
<tr>
<td>NLevels</td>
<td>2</td>
</tr>
<tr>
<td>Clusterratio</td>
<td>97</td>
</tr>
<tr>
<td>NLeaf</td>
<td>10</td>
</tr>
<tr>
<td>Sampling</td>
<td>N</td>
</tr>
<tr>
<td>SaveStats</td>
<td>N</td>
</tr>
<tr>
<td>Updatedb2</td>
<td>Y</td>
</tr>
<tr>
<td>Utilcode</td>
<td>R</td>
</tr>
<tr>
<td>Location</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

---

**Example 15 - Materializing pending definition changes**

**BMC51946**

<table>
<thead>
<tr>
<th>Table</th>
<th>Exm15b.Lart001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Exm15b.Larsa1ba</td>
</tr>
<tr>
<td>Indexpart</td>
<td>Exm15b.Larxpc1a</td>
</tr>
<tr>
<td>Space</td>
<td>270</td>
</tr>
<tr>
<td>Space_KB</td>
<td>12960</td>
</tr>
<tr>
<td>Allocation</td>
<td></td>
</tr>
<tr>
<td>High2K</td>
<td>X'4460070200000000'</td>
</tr>
<tr>
<td>Low2K</td>
<td>X'1910020200000000'</td>
</tr>
<tr>
<td>NumNonLeaf</td>
<td>3</td>
</tr>
<tr>
<td>NLevels</td>
<td>2</td>
</tr>
<tr>
<td>Clusterratio</td>
<td>97</td>
</tr>
<tr>
<td>NLeaf</td>
<td>10</td>
</tr>
<tr>
<td>Sampling</td>
<td>N</td>
</tr>
<tr>
<td>SaveStats</td>
<td>N</td>
</tr>
<tr>
<td>Updatedb2</td>
<td>Y</td>
</tr>
<tr>
<td>Utilcode</td>
<td>R</td>
</tr>
<tr>
<td>Location</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

---

**Example 15 - Materializing pending definition changes**

**BMC51947**

<table>
<thead>
<tr>
<th>Table</th>
<th>Exm15b.Lart001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Exm15b.Larsa1ba</td>
</tr>
<tr>
<td>Indexpart</td>
<td>Exm15b.Larxpc1a</td>
</tr>
<tr>
<td>Space</td>
<td>270</td>
</tr>
<tr>
<td>Space_KB</td>
<td>12960</td>
</tr>
<tr>
<td>Allocation</td>
<td></td>
</tr>
<tr>
<td>High2K</td>
<td>X'4460070200000000'</td>
</tr>
<tr>
<td>Low2K</td>
<td>X'1910020200000000'</td>
</tr>
<tr>
<td>NumNonLeaf</td>
<td>3</td>
</tr>
<tr>
<td>NLevels</td>
<td>2</td>
</tr>
<tr>
<td>Clusterratio</td>
<td>97</td>
</tr>
<tr>
<td>NLeaf</td>
<td>10</td>
</tr>
<tr>
<td>Sampling</td>
<td>N</td>
</tr>
<tr>
<td>SaveStats</td>
<td>N</td>
</tr>
<tr>
<td>Updatedb2</td>
<td>Y</td>
</tr>
<tr>
<td>Utilcode</td>
<td>R</td>
</tr>
<tr>
<td>Location</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

---

**Example 15 - Materializing pending definition changes**

**BMC51948**

<table>
<thead>
<tr>
<th>Table</th>
<th>Exm15b.Lart001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Exm15b.Larsa1ba</td>
</tr>
<tr>
<td>Indexpart</td>
<td>Exm15b.Larxpc1a</td>
</tr>
<tr>
<td>Space</td>
<td>270</td>
</tr>
<tr>
<td>Space_KB</td>
<td>12960</td>
</tr>
<tr>
<td>Allocation</td>
<td></td>
</tr>
<tr>
<td>High2K</td>
<td>X'4460070200000000'</td>
</tr>
<tr>
<td>Low2K</td>
<td>X'1910020200000000'</td>
</tr>
<tr>
<td>NumNonLeaf</td>
<td>3</td>
</tr>
<tr>
<td>NLevels</td>
<td>2</td>
</tr>
<tr>
<td>Clusterratio</td>
<td>97</td>
</tr>
<tr>
<td>NLeaf</td>
<td>10</td>
</tr>
<tr>
<td>Sampling</td>
<td>N</td>
</tr>
<tr>
<td>SaveStats</td>
<td>N</td>
</tr>
<tr>
<td>Updatedb2</td>
<td>Y</td>
</tr>
<tr>
<td>Utilcode</td>
<td>R</td>
</tr>
<tr>
<td>Location</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

---

**Example 15 - Materializing pending definition changes**

**BMC51949**

<table>
<thead>
<tr>
<th>Table</th>
<th>Exm15b.Lart001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Exm15b.Larsa1ba</td>
</tr>
<tr>
<td>Indexpart</td>
<td>Exm15b.Larxpc1a</td>
</tr>
<tr>
<td>Space</td>
<td>270</td>
</tr>
<tr>
<td>Space_KB</td>
<td>12960</td>
</tr>
<tr>
<td>Allocation</td>
<td></td>
</tr>
<tr>
<td>High2K</td>
<td>X'4460070200000000'</td>
</tr>
<tr>
<td>Low2K</td>
<td>X'1910020200000000'</td>
</tr>
<tr>
<td>NumNonLeaf</td>
<td>3</td>
</tr>
<tr>
<td>NLevels</td>
<td>2</td>
</tr>
<tr>
<td>Clusterratio</td>
<td>97</td>
</tr>
<tr>
<td>NLeaf</td>
<td>10</td>
</tr>
<tr>
<td>Sampling</td>
<td>N</td>
</tr>
<tr>
<td>SaveStats</td>
<td>N</td>
</tr>
<tr>
<td>Updatedb2</td>
<td>Y</td>
</tr>
<tr>
<td>Utilcode</td>
<td>R</td>
</tr>
<tr>
<td>Location</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>
### Example 15 - Materializing pending definition changes

#### Tablespace: EXM15B.LARSA1BA

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>C001_INTEGER</td>
<td>C</td>
<td>8</td>
<td>EXTENDED, STORNAME ARULARSG</td>
</tr>
</tbody>
</table>

#### Table: EXM15B.LART001

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>C009_TIMESTAMP</td>
<td>TIMESTMP</td>
<td>10</td>
<td>N, NACTIVE</td>
</tr>
</tbody>
</table>

#### Index: EXM15B.LARXPS1B

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'1990040200000000'</td>
<td>(CHAR)</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

#### Most Frequent Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Frequency</th>
<th>COLVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'1920030200000000'</td>
<td>0.003205</td>
<td></td>
</tr>
<tr>
<td>X'1920040200000000'</td>
<td>0.003205</td>
<td></td>
</tr>
<tr>
<td>X'1920060200000000'</td>
<td>0.003205</td>
<td></td>
</tr>
<tr>
<td>X'1920070200000000'</td>
<td>0.003205</td>
<td></td>
</tr>
<tr>
<td>X'1930010200000000'</td>
<td>0.003205</td>
<td></td>
</tr>
</tbody>
</table>

#### Highest Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Type</th>
<th>LowK</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'2290080200000000'</td>
<td>X'1910020200000000'</td>
<td></td>
</tr>
</tbody>
</table>

#### Statistics

- **RBA_FORMAT**: EXTENDED
- **STORNAME**: ARULARSG
- **ALLOCUNIT**: C
- **STORTYPE**: I
- **NLAVGKEYLEN**: 3
- **VCATNAME**: DEHJCAT
- **PQTY**: 6
- **SQTY**: 3
- **PCTFREE**: 5
- **FREEPAGE**: 0
- **LOCATION**: DEHJ

---

### Chapter 5  Examples of REORG PLUS jobs  511
Example 15 - Materializing pending definition changes

REORG PLUS for DB2 Reference Manual
Tuning REORG PLUS jobs

This chapter provides information that enables you to improve the performance or memory use of your jobs.

Tuning for performance in REORG PLUS

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 REORG PLUS

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. The following table describes these recommendations.

For additional installation option changes for SHRLEVEL CHANGE, see “Setting SHRLEVEL CHANGE installation options for optimal performance in REORG PLUS” on page 516.

**Note**
If you modify these options after installation, you must resubmit the installation job for the modifications to take effect.
## Table 90: Installation default option changes for performance

<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>KEEPDICTIONARY</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 DSRSEXIT 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>
<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 in REORG PLUS

In addition to the recommendations that apply to all reorganizations, BMC recommends additional changes to installation options that you can make to enhance the performance of SHRLEVEL CHANGE jobs.

The following table describes these changes.
Table 91: 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>
<tr>
<td>MAXRO</td>
<td>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.</td>
</tr>
<tr>
<td>SIXSNAP</td>
<td>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.</td>
</tr>
<tr>
<td>SPILDSNP</td>
<td>Specify the data set name pattern to use when allocating the LOGSPIL data set. The default value is &amp;&amp;UID. To prevent duplicate names when the same user runs two jobs concurrently, change the default value.</td>
</tr>
<tr>
<td>SPILSCLS</td>
<td>Specify the SMS storage class, if any, from which you want to obtain the LOGSPIL data set volume.</td>
</tr>
<tr>
<td>SPILUNIT</td>
<td>Specify the device type or generic unit type to which you want the LOGSPIL data set allocated.</td>
</tr>
</tbody>
</table>

Specifying command options for optimal performance in REORG PLUS

This section lists REORG command options that you can set to improve performance in all REORG PLUS jobs.

The following table lists REORG command options that correspond to installation options that are listed in Table 90 on page 516. BMC recommends that you specify these command options if they are not already specified in the installation options.

If you are using SHRLEVEL CHANGE, also see “Specifying SHRLEVEL CHANGE command options for optimal performance in REORG PLUS” on page 518 for additional command options.
### Table 92: 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>KEEPDICTIIONARY YES</td>
<td>KEEPDICTIIONARY=YES</td>
</tr>
<tr>
<td>REDEFINE NO</td>
<td>REDEFINE=NO</td>
</tr>
<tr>
<td>UNITCNT ( n )</td>
<td>UNITCNT=( n )</td>
</tr>
<tr>
<td></td>
<td>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>

The following table describes additional recommended command options. (No installation options correspond to these command options.)

### Table 93: 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 in REORG PLUS

This section lists REORG command options that you can set to improve performance in SHRLEVEL CHANGE jobs.

The following table 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 94: 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>Command option</td>
<td>Corresponding installation option</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------</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>LOGTHRSH</td>
</tr>
<tr>
<td>LONGLOG</td>
<td>LONGLOG</td>
</tr>
<tr>
<td>MAXRO</td>
<td>MAXRO</td>
</tr>
<tr>
<td>RIDMAPMEM</td>
<td>RMAPMEM</td>
</tr>
<tr>
<td>Not available</td>
<td>RIDMDSSZ</td>
</tr>
<tr>
<td>Not available</td>
<td>RIDMMAXD</td>
</tr>
<tr>
<td>SIXSNAP</td>
<td>SIXSNAP</td>
</tr>
<tr>
<td>SPILLDSNPAT</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 REORG PLUS 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 the following table to find information about a specific option.

**Table 95: Additional performance information about options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Reference</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>“Performance considerations for ANALYZE command options” on page 520</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>“Performance considerations for sort processing options” on page 530</td>
</tr>
<tr>
<td>CBUFFS</td>
<td>Controls buffers for writing copy data sets</td>
<td>“Performance considerations for buffer installation options” on page 525</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Reference</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DDTYPE</td>
<td>Controls dynamic allocation</td>
<td>“Performance considerations for dynamic allocation options” on page 529</td>
</tr>
<tr>
<td>MAXSORTS</td>
<td>Controls number of sort processes</td>
<td>“Performance considerations for sort processing options” on page 530</td>
</tr>
<tr>
<td>ORDER</td>
<td>Controls whether REORG PLUS sorts the data</td>
<td>“Performance considerations for sort processing options” on page 530</td>
</tr>
<tr>
<td>MAXSORTMEMORY</td>
<td>Maximum amount of memory that REORG PLUS can allocate to each sort task</td>
<td>“Performance considerations for sort processing options” on page 530</td>
</tr>
<tr>
<td>MINSORTMEMORY</td>
<td>Minimum amount of memory that REORG PLUS should allocate to each sort task</td>
<td>“Performance considerations for sort processing options” on page 530</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>“Performance considerations for sort processing options” on page 530</td>
</tr>
<tr>
<td>SMAX</td>
<td>Controls number of sort processes</td>
<td>“Performance considerations for sort processing options” on page 530</td>
</tr>
<tr>
<td>SMCORE</td>
<td>Controls sort memory usage</td>
<td>“Performance considerations for sort processing options” on page 530</td>
</tr>
<tr>
<td>UBUFFS</td>
<td>Controls buffers for reading and writing SYSREC data sets</td>
<td>“Performance considerations for buffer installation options” on page 525</td>
</tr>
<tr>
<td>UXSTATE</td>
<td>Controls DB2 user exit processing</td>
<td>“Performance considerations for the UXSTATE installation option” on page 534</td>
</tr>
<tr>
<td>WBUFFS</td>
<td>Controls buffers for reading and writing SYSUT1 data sets</td>
<td>“Performance considerations for buffer installation options” on page 525</td>
</tr>
</tbody>
</table>

**Performance considerations for ANALYZE command options**

The ANALYZE option that you specify can affect the performance of your REORG PLUS job.

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 the description for the ANALYZE option.

The SCAN, SAMPLE, HURBA, and BMCSTATS options determine how REORG PLUS gathers the information that it needs for analysis. The following table 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 96: 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</td>
<td>Least</td>
<td>Most</td>
</tr>
</tbody>
</table>

BMCSTATS provides the same accuracy as SCAN if the statistics in the DASD MANAGER PLUS tables are current.

a BMCSTATS is available only if you are using the Performance for DB2 Databases or 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**

The following table describes the actions that REORG PLUS takes based on the object that you are reorganizing and the ANALYZE option that you specify.
### Table 97: REORG PLUS actions for ANALYZE option

<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 Performance for DB2 Databases or 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.

**ANALYZE messages**

When you specify ANALYZE PAUSE or ANALYZE ONLY, you can use the
messages that REORG PLUS issues to tune the performance of your REORG PLUS
job.

For more information about using the information from the ANALYZE phase for
allocating your data sets, see “Using the ANALYZE option in REORG PLUS to
estimate data set allocation” on page 370. If you do not want to specify ANALYZE
PAUSE or ANALYZE ONLY, you must either have REORG PLUS dynamically
allocate your data sets, or calculate the data set sizes yourself.

This section lists and describes the messages that REORG PLUS issues when you
specify ANALYZE PAUSE or ANALYZE ONLY.
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

No action is required.

This message provides the count of data pages that REORG PLUS will read during the unload process.

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.

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.

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.

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.

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.
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.

Use this message in conjunction with message BMC51265I or BMC51266I.

**BMC50485I ESTIMATED CARDINALITY OF objectType objectName = n**

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.

Use this message in conjunction with message BMC51265I or BMC51266I.

**Performance considerations for 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 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.

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. BMC recommends a value of 20 for this option.

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.

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 for REORG TABLESPACE” on page 251.
**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.

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 605.

**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 540.
Performance considerations for 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.

- 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.

Performance considerations for 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)
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.

Performance considerations for the ORDER command option

The ORDER option that you specify can affect the performance of your REORG PLUS job.

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.

Performance considerations for 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.

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 533
- 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.

Minimum and maximum amounts of memory per sort that you specify with the MINSORTMEMORY and MAXSORTMEMORY options

Values that you specify for the multitasking installation options described in “Multitasking installation options” on page 535

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 or command option and MAXSORTMEMORY installation or command option.

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 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.

For example, if you specify SMCORE=(0K,700K) and SORTNUM=32, REORG PLUS responds as follows:

- Uses all of the sort optimization installation and command options to calculate the total memory that is required for sort processing
- 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 the MAXSORTS option and Table 101 on page 536, 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” on page 535.

Performance considerations for the UXSTATE installation option

The value that you specify for the UXSTATE option can affect the performance of your REORG PLUS job.

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 in REORG PLUS

During execution, REORG PLUS determines the most effective arrangement of tasks when running in a multiprocessor environment. However, you can also affect the amount of multitasking that REORG PLUS performs.

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**

The following table 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.

<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&lt;sup&gt;a&lt;/sup&gt;</td>
<td>REORG phase</td>
</tr>
<tr>
<td>UNLDMAX&lt;sup&gt;a&lt;/sup&gt;</td>
<td>UNLOAD phase</td>
</tr>
<tr>
<td>BILDMAX&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Index build process</td>
</tr>
<tr>
<td>COPYMAX</td>
<td>Image copy process</td>
</tr>
<tr>
<td>RENMMMAX</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>

* 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. The following table lists the valid values for TASKMAX.

<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><em>n</em></td>
<td>The maximum number of tasks that REORG PLUS can start is <em>n</em> (where <em>n</em> is a positive integer from 1 through 32767).</td>
</tr>
<tr>
<td><em>n</em>%</td>
<td>The maximum number of tasks that REORG PLUS can start is <em>n</em>% of the number of CPUs on the system (where <em>n</em> is a positive integer from 1 through 32768).</td>
</tr>
</tbody>
</table>

The following table lists the valid values for all of the other multitasking options.
Table 100: 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.

The following table 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 537.

Table 101: 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” on page 536.

Multitasking I/O bound processes

The following multitasked processes are I/O bound:
- ANALYZE phase
- Table space copy process
- Nonpartitioned 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 536, 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 SORTWKnn DD statements in your JCL.
- Specify the following:
  - DDTYPE SORTWORK ACTIVE NO on your REORG command, or disable dynamic allocation for the SORTWKnn DDs in your options module
  - The SORTDEV'T installation or command option
  - One SYSRECnn DD per partition
  - One SYSUT1nn DD per index
  - SORTNUM 32

For more information about setting the environment for BMCSORT, see “Performance considerations for sort processing options” on page 530.

**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 “SYSREC data set” on page 525.

**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 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 for REORG PLUS

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 “Performance considerations for buffer installation options” on page 525 and “Performance considerations for sort processing options” on page 530.
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.

Additional performance information for SHRLEVEL CHANGE options

The copy options that you specify can affect your SHRLEVEL CHANGE reorganizations.

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 “Setting REORG PLUS authorizations” on page 53 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.

When the 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`

When transactions and the 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 from REORG PLUS

You can use the information in the messages described in this section to monitor REORG PLUS performance and to fine-tune future runs.
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 REORG PLUS EXEC statement” on page 327.

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.

**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 540 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:

- The 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 “Performance considerations for buffer installation options” on page 525) 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.

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
- The 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.

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.

**BMC50486I** taskNumber: BMCSORT STARTED, nK BELOW 16M, nK 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 in REORG PLUS**

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 in REORG PLUS” on page 534.

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 in REORG PLUS

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 the following table to help determine the appropriate option.

<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.

**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 “Setting REORG PLUS authorizations” on page 53.
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

The following figures show the objects that the ANALYZE phase might use (except ANALYZE HURBA, which uses no objects).

Figure 41: 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.
Figure 42: ANALYZE phase for an index reorganization

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 in REORG PLUS” on page 349)
- 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 568.
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.

The following figures 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 569.

**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.

The following figures show the objects that REORG PLUS might use in the RELOAD phase.
Figure 45: RELOAD phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY table space reorganization

Figure 46: RELOAD phase: SHRLEVEL REFERENCE table space reorganization

During the UTILTERM phase, REORG PLUS performs rename and delete operations that result in the staging data sets replacing the original data sets.
Figure 47: RELOAD phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY index reorganization

Figure 48: 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 in REORG PLUS” on page 349)
- 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 the following table 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 103: Single-phase reorganization considerations for SYSREC or SYSUT1 data set

<table>
<thead>
<tr>
<th>Specified SHRLEVEL</th>
<th>Table space reorganization</th>
<th>Index reorganization</th>
<th>Effect on restartability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHRLEVEL NONE (the default)</td>
<td>You can omit SYSREC data set for an additional performance gain.</td>
<td>You can omit SYSUT1 data set for an additional performance gain.</td>
<td>If you omit the data set, your job might not be restartable.</td>
</tr>
<tr>
<td>SHRLEVEL REFERENCE</td>
<td>REORG PLUS does not use SYSREC data set, even if you specify it.</td>
<td>REORG PLUS does not use SYSUT1 data set, even if you specify it.</td>
<td>The job is restartable due to the nondestructive nature of this type of reorganization.</td>
</tr>
<tr>
<td>SHRLEVEL CHANGE</td>
<td>REORG PLUS does not use SYSREC data set, even if you specify it.</td>
<td>REORG PLUS does not use SYSUT1 data set, even if you specify it.</td>
<td>The job is not restartable until the UTILTERM phase begins, and restarting the job before the UTILTERM phase begins is unnecessary because the original data sets are unaffected.</td>
</tr>
</tbody>
</table>

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 (“UNLOAD phase” on page 552) and the RELOAD phase (“RELOAD phase” on page 555).

The following figures show the objects that REORG PLUS uses in the REORG phase.
Figure 49: REORG phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY table space reorganization

The SYSREC data set is optional for a single-phase table space reorganization, but is required for restarting the job.
Figure 50: REORG phase: SHRLEVEL REFERENCE table space reorganization

During the UTILTERM phase, REORG PLUS performs rename and delete operations that result in the staging data sets replacing the original data sets.
Figure 51: REORG phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY index reorganization

Figure 52: REORG phase: SHRLEVEL REFERENCE index reorganization

During the UTILTERM phase, REORG PLUS performs rename and delete operations that results in the staging data sets replacing the original data sets.
Online reorganization

This chapter describes how to run an online reorganization and provides considerations for online reorganizations.

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

An online reorganization allows read-write (RW) application access to DB2 table space and index space objects during most of the reorganization process.

To invoke an online reorganization, REORG PLUS provides the SHRLEVEL CHANGE keyword. When you specify SHRLEVEL CHANGE, REORG PLUS performs 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
The following figure illustrates the processing flow for a SHRLEVEL CHANGE reorganization.

**Figure 53: Processing flow for SHRLEVEL CHANGE**

**How SHRLEVEL CHANGE works**

This section describes SHRLEVEL CHANGE processing, including the key tasks that are performed during each execution phase.

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.

**Figure 54 on page 573 through Figure 59 on page 578 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:
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 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 555.

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 in REORG PLUS” on page 582.

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 in REORG PLUS” on page 589.

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. 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 607.
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 608.

**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 87.

<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 608 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 restrictions and considerations” on page 120.

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 the following statistics:

■ Real-time statistics
Introduction to online reorganizations

- 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 54 on page 573 through Figure 59 on page 578 show the data sets that each execution phase uses during a SHRLEVEL CHANGE reorganization.
Figure 54: Processing phases: SHRLEVEL CHANGE two-phase table space reorganization (part 1 of 2)
Figure 55: Processing phases: SHRLEVEL CHANGE two-phase table space reorganization (part 2 of 2)

LOGAPPLY phase

LOGFINAL phase

BICYnn

BIZnn

BIRYnn

BIRZnn

Staging table space

Staging indexes

DASD MANAGER PLUS statistics tables

DB2 catalog

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 BMCUITL and BMCSYNC tables.

For partial reorganizations, the nonpartitioned indexes are copied to the staging data sets before the index update operations.

All sort processing uses the UTPRINT data set. The UNLOAD phase updates the BMCDICT table. The UTILTERM phase updates the BMCHIST table.
Figure 56: 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.
Figure 57: Processing phases: SHRLEVEL CHANGE single-phase table space reorganization (part 1 of 2)
Figure 58: Processing phases: SHRLEVEL CHANGE single-phase table space reorganization (part 2 of 2)

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.

LOGAPPLY phase

Staging table space

Staging indexes

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.

UTILTERM phase

DB2 catalog

BICYnn

BICZnn

BiRYnn

BIRZnn

DASD MANAGER PLUS statistics tables

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 operations.

All sort processing uses the UTPRINT data set. The UNLOAD phase updates the BMCDICT table. The UTILTERM phase updates the BMCHIST table.
Figure 59: Processing phases: SHRLEVEL CHANGE single-phase index reorganization

SYSIN → UTILINIT phase → DB2 catalog

DASD MANAGER PLUS exceptions tables → Original index → DASD MANAGER PLUS statistics tables

SYSIDCIN

XBM → REORG phase → SORTWKnn

ANALYZE phase

LOGAPPLY phase

LOGFINAL phase

DASD MANAGER PLUS statistics tables → UTILTERM phase → DB2 catalog

All phases use the SYSPRINT data set and update the BMCUTIL 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 EMCHIST 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 log 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 updates 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 limit-key updates are made in the UTILTERM phase. For more information, see “DDLIN data set restrictions and considerations” on page 120.

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 the REBALANCE option description.

**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.

**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.

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 477
- “Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY” on page 486
- “Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing” on page 494

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 “REORG PLUS for DB2 system and software requirements” on page 51.

Also, ensure that you have the appropriate authorizations as described in “Setting REORG PLUS authorizations” on page 53.
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.

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

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 60.

5 Specify any additional SHRLEVEL CHANGE options as needed (see descriptions in “SHRLEVEL CHANGE options for REORG PLUS” on page 307).

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 in REORG PLUS

You can control the log apply process with certain options in REORG PLUS.

These options enable you to perform the following tasks:

- 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 in REORG PLUS” on page 589.

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 in REORG PLUS

This section describes the options that REORG PLUS provides to allow you to control the log apply process during the reorganization.

You control the log apply process 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. The following table lists these options:

Table 104: Log apply control options

<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>
<tr>
<td>DELAY</td>
<td>DELAY specifies the number of seconds that are to elapse from the time that REORG PLUS detects a longlog condition until it performs the action that is specified on the LONGLOG option. Setting this option to a large number gives you time to make decisions regarding a course of action when a longlog condition is detected. After you decide what value to set for LONGLOG, you can also change DELAY to a smaller number to make that action happen sooner.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LOGFINAL</td>
<td>The LOGFINAL timestamp value specifies when REORG PLUS is to start the LOGFINAL phase. If you specify NONE, the other options control the start of the phase.</td>
</tr>
<tr>
<td>LOGTHRESHLD</td>
<td>LOGTHRESHLD specifies the number of log records that REORG PLUS uses as the threshold to use to begin the LOGFINAL phase. This phase begins when REORG PLUS determines that the number of log records remaining to be applied is less than or equal to the threshold value.</td>
</tr>
<tr>
<td>LONGLOG</td>
<td>LONGLOG tells REORG PLUS what action to take after it detects a longlog condition that lasts for the duration of time that is specified by DELAY. A longlog condition exists when 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.</td>
</tr>
<tr>
<td></td>
<td>The following values are valid for LONGLOG:</td>
</tr>
<tr>
<td></td>
<td>■ TERM tells REORG PLUS to terminate the reorganization.</td>
</tr>
<tr>
<td></td>
<td>■ DRAIN tells REORG PLUS to start the LOGFINAL phase.</td>
</tr>
<tr>
<td></td>
<td>■ CONTINUE tells REORG PLUS to continue the LOGAPPLY phase.</td>
</tr>
<tr>
<td>MAXRO</td>
<td>When REORG PLUS estimates that it can apply the remaining log records within the number of seconds that are specified for MAXRO, it begins the LOGFINAL phase. This estimate does not include the time that was spent producing incremental image copies and running the UTILTERM phase. If you originally set MAXRO to DEFER, REORG PLUS applies the log records indefinitely unless you specify a value in the DEADLINE option that is reached or you change DEFER to a value.</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>This option displays information about the log apply task. You can set the value to TERSE to get a summary of information for the entire process, or you can set it to VERBOSE to get information for each task that is part of the multitasking job. For the syntax and an example of the output from the DISPLAY command, see “Utility Monitor” on page 589.</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 loglog 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 loglog condition does not exist, and either the MAXRO or the LOGTHRESHLD conditions exist.
- The loglog 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 loglog condition exists, the time that you specify in the DELAY option is reached, and you set LONGLOG to TERM.

Hierarchy of log apply control options in REORG PLUS

This section illustrates how REORG PLUS uses the log apply control options to control SHRLEVEL CHANGE processing after the LOGAPPLY phase starts.
The following figure represents one 15-second sample interval. Processing repeats indefinitely, until either the LOGFINAL phase begins or the reorganization ends.

Figure 60: How options determine log apply processing (part 1 of 3)
Figure 61: How options determine log apply processing (part 2 of 3)
Figure 62: How options determine log apply processing (part 3 of 3)

Diagram:

- C
  - Timer already started? (No: Set timer to DELAY value and start timing)
    - Yes
      - End of DELAY interval reached? (Yes: A)
        - No
          - LONGLOG = CONTINUE?
            - Yes
              - LONGLOG = TERMINATE?
                - Yes
                  - Reorganization fails
                - No
                  - LONGLOG = DRAIN
            - No
              - Start LOGFINAL phase
Using XBM to view and dynamically control the log apply process in REORG PLUS

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 in REORG PLUS” on page 583) 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, described in this section, from the XBM Send New command panel to get additional information about processing of the log records.

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:

```plaintext
DISPLAY option value
```

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 593.

**DISPLAY TERSE output**

The following figures show an example of the output from the DISPLAY TERSE command.
**Figure 63: DISPLAY TERSE output**

![Display TERSE output](image)

**DISPLAY VERBOSE output**

The following figures show an example of the output from the DISPLAY VERBOSE command.
**Figure 64: DISPLAY VERBOSE output**

<table>
<thead>
<tr>
<th>Job name</th>
<th>ARUAVR$E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility name</td>
<td>REORG PLUS Online</td>
</tr>
<tr>
<td>Command</td>
<td>display verbose</td>
</tr>
</tbody>
</table>

Following response received return code : 0

BMC50803  LOG APPLY DISPLAY AT 05/30/2005 09:12:10.920206
BMC50822  CURRENT PHASE: LOGAPPLY, STARTED: 05/30/2005 08:58:35.938569
BMC50804  LOG APPLY TASKS : 1 TASKS, 1 STARTED, 0 FINISHED
BMC50805  0 RECORDS QUEUED, 9984 RECORDS APPLIED
BMC50807  TASK 102: STARTED

**Control of the log apply process in REORG PLUS**

592  **REORG PLUS for DB2 Reference Manual**
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:

```
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 DESCDE 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.

### Log apply control option scenarios for REORG PLUS

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 most of the 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 initial required statuses and status changes during the reorganization, see “Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE” on page 72.

Operational considerations for online reorganizations

This section describes certain considerations that apply when running an online reorganization.

In addition to the considerations described in this section, the following sections contain considerations for online reorganizations:

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorizations</td>
<td>“Setting REORG PLUS authorizations” on page 53</td>
</tr>
</tbody>
</table>
### Interacting with applications

You can use the WTOMSG SUFSTART command option 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:

- LOGSPI=20000, 10000
- SPILUNIT=WORK
- SPILSCLS=NONE
- SPILDSNP=&&UID

For information about these options, see “REORG PLUS installation options” on page 617.

You can also override the values through options on the REORG PLUS command (LOGSPI, SPILUNIT, SPILLSTORCLAS, and SPILDSNPAT). For the syntax of these command options, see “SHRLEVEL CHANGE options for REORG PLUS” on page 307.

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 LOGSPI 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 SPILUNIT 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 SPILDSNPAT 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 SPILUNIT, SPILLSTORCLAS, or SPILDSNPAT 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.

Using APPLICATION RESTART Control to suspend application processing

For batch applications that are running the BMC APPLICATION RESTART CONTROL for DB2 product (AR/CTL), an online reorganization can use the AR/CTL suspend-and-resume interface to suspend application processing during the reorganization.

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.

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

For more information about AR/CTL, see the *APPLICATION RESTART Control User Guide*.

**Before you begin**

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.

■ In the subsystem that you use to run the batch DB2 application, the BMC Consolidated Subsystem (BCSS) must be active. The BCSS component that supports the DB2 and VSAM features of the Application Enhancement Series products must be initialized.

**To implement the suspend-and-resume interface with AR/CTL**

1. Ensure that the REORG PLUS ARC installation option is set to YES.

2. Make the CPU authorization password for REORG PLUS available to the BCSS by 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.

3. Set up the reorganization job step.

4. In each batch program that you want AR/CTL to control, implement AR/CTL checkpoint/restart services as described in the AR/CTL documentation.

5. 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 602.

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.

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.
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 106 on page 604.

**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 and RECOVERYICDDN command options. For performance and tuning information when using these options, see “Incremental copy data sets” on page 528 and “Improving performance” on page 605.

### 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.
The following table describes the incremental copy data sets that, when registered, correspond to the DB2 local and remote copies. The table also states when the incremental copy data sets are required.

Table 105: Corresponding incremental image copy types

<table>
<thead>
<tr>
<th>Image copy type</th>
<th>Corresponding REORG PLUS incremental copy data set (when registered)</th>
<th>When the data set is required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local primary</td>
<td>BICY or ddname1 specified in the ICDDN command or installation option</td>
<td>Always required if you specify COPY YES, unless dynamically allocating your copy data sets</td>
</tr>
<tr>
<td>Local backup</td>
<td>BICZ or ddname2 specified in the ICDDN command or installation option</td>
<td>Optional</td>
</tr>
<tr>
<td>Remote primary</td>
<td>BIRY or ddname1 specified in the RECOVERYICDDN command option or the RCVICDDN installation option</td>
<td>Required only if a BIRZ copy is made</td>
</tr>
<tr>
<td>Remote backup</td>
<td>BIRZ or ddname2 specified in the RECOVERYICDDN command option or the RCVICDDN installation option</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Allocating incremental copy data sets

Use one of the methods described in “Methods for allocating copy and work data sets” on page 335 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 106 on page 604, 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.

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 the following table. The table lists the incremental copy DD statements that REORG PLUS requires based on the table space and type of reorganization.

Table 106: Incremental copy DD statements required for SHRLEVEL CHANGE when allocating data sets in your JCL

<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 nn in the ddname. You should not specify separate data sets for nonpartitioned, multi-data-set table spaces.</td>
</tr>
</tbody>
</table>
### Table space being copied

<table>
<thead>
<tr>
<th>Type of reorganization</th>
<th>Incremental copy DD statements required</th>
</tr>
</thead>
</table>
| **Partitioned, including partition-by-growth** | ■ 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 342. |
| **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 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 342. |

---

**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 use the COPYLVL option to 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.

You can use the SIZEPCT option 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 CPYFAIL installation option determines the action that REORG PLUS takes.

**CPYFAIL=TERM**

If CPYFAIL=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.

REORG PLUS changes CPYFAIL=TERM to CPYFAIL=COPYPEND when either of the following conditions exists:

- Pending definition changes exist on the object and REORG PLUS can materialize those changes
- You are rebalancing partitions (using either the REBALANCE option or a DDLIN data set)

**CPYFAIL=COPYPEND**

If CPYFAIL=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 you are reorganizing a catalog object, REORG PLUS changes CPYFAIL=COPYPEND to CPYFAIL=TERM at the beginning of the job. If the copy registration fails, REORG PLUS backs out the renames and terminates the reorganization.
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 reorganization” on page 380.

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
- Updates 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 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.

If you cannot restart your job, you can manually complete the reorganization or back out to the starting point. For more information, see “Managing a reorganization that does not complete in the UTILTERM phase” on page 384.
Performance considerations for SHRLEVEL CHANGE reorganizations

This section describes considerations regarding performance of your online reorganizations.

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 “REORG PLUS installation options” on page 617.

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 611. 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.
Note

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.

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=\( s \)K, 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 following:

- The data spaces that you allocate with the LOGMEM command or installation option
The DASD that you allocate with the LOGSPILL command option or LOGSPIL 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 613. For guidelines for using the LOGMEM and LOGSPILL options together, see “Calculating storage requirements for log data” on page 614.

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 612.

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 the following table to help determine each number in the formula.
### Table 107: 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></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 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></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>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></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>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 *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 *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” on page 614.

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 597.

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 describes the options that you can specify during installation.

Overview of REORG PLUS installation options

Options set during installation establish default values that REORG PLUS uses during your reorganization.

The REORG PLUS product is installed by using the BMC Software Installation System. During this installation, the configuration 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:

- $730DOP2 establishes the defaults for REORG PLUS processing options.
- $532SOPT 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 configuration job is submitted, it links the ARU$OPTS installation options module in the APF-authorized library that is designated by your site.

You can configure the installation of REORG PLUS by changing the values for the REORG PLUS installation options. However, if you change any of the values in $730DOP2 or $532SOPT 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 “Building and executing REORG PLUS jobs” on page 325. For more information about customizing your installation of REORG PLUS, see the BMC Products and Solutions for DB2 Customization Guide.
To find a description of each options macro, use the following table.

Table 108: 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>$730DOP2</td>
<td>$ARUOPTS</td>
<td>Basic options</td>
<td>“Basic REORG PLUS installation options” on page 618</td>
</tr>
<tr>
<td></td>
<td>$ARUDYNA</td>
<td>Options for dynamic data set allocation</td>
<td>“Dynamic allocation installation options for REORG PLUS” on page 689</td>
</tr>
<tr>
<td>$532SOPT</td>
<td>$AUPSMAC</td>
<td>Options for BMCSORT</td>
<td>“DYNALOC installation option” on page 711</td>
</tr>
</tbody>
</table>

Basic REORG PLUS installation options

The table in this section shows the options contained in the $ARUOPTS macro in $730DOP2.

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 109: 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>ALTRFAIL</td>
<td>Not applicable</td>
<td>Obsolete</td>
<td>“ALTRFAIL (obsolete)” on page 625</td>
</tr>
<tr>
<td>ANALMAX</td>
<td>*</td>
<td>Maximum number analyze tasks to start</td>
<td>“ANALMAX=*” on page 626</td>
</tr>
<tr>
<td>ARC</td>
<td>NO</td>
<td>Whether to use the AR/CTL interface</td>
<td>“ARC=NO” on page 626</td>
</tr>
<tr>
<td>ARCHDDN</td>
<td>SYSARC</td>
<td>Default ddname for the archive data set</td>
<td>“ARCHDDN=SYSA RC” on page 626</td>
</tr>
<tr>
<td>ARCHFORMAT</td>
<td>BMC</td>
<td>Default format for the archive data set</td>
<td>“ARCHFORMAT=BMC” on page 627</td>
</tr>
<tr>
<td>AUXREORG</td>
<td>Not applicable</td>
<td>Obsolete</td>
<td>“AUXREORG (obsolete)” on page 628</td>
</tr>
<tr>
<td>Option</td>
<td>Shipped value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>------------------------</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>“AVAILPAGEPCT=100” on page 628</td>
</tr>
<tr>
<td>BILDMAX</td>
<td>300%</td>
<td>Maximum number of tasks to start for the index build process</td>
<td>“BILDMAX=300%” on page 629</td>
</tr>
<tr>
<td>BMCHIST</td>
<td>YES</td>
<td>Whether to update the BMCHIST table</td>
<td>“BMCHIST=YES” on page 629</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>“CBUFFS=30” on page 630</td>
</tr>
<tr>
<td>CONDEXEC</td>
<td>NO</td>
<td>Whether REORG PLUS should consider performing a conditional reorganization</td>
<td>“CONDEXEC=NO” on page 630</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>“COPYDDN=(BCPY,BCPZ)” on page 631</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>“COPYLVL=PART” on page 632</td>
</tr>
<tr>
<td>COPYMAX</td>
<td>*</td>
<td>Maximum number of tasks to start during the image copy process</td>
<td>“COPYMAX=*” on page 633</td>
</tr>
<tr>
<td>COPYSUBSET</td>
<td>NO</td>
<td>Whether to allow a single copy of a subset of partitions</td>
<td>“COPYSUBSET=NO” on page 633</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>“CPYRFAIL=TERM” on page 633</td>
</tr>
<tr>
<td>DATACAP</td>
<td>NO</td>
<td>Whether to turn on the Data Capture Changes flag</td>
<td>“DATACAP=NO” on page 634</td>
</tr>
<tr>
<td>DDLDDN</td>
<td>DDLIN</td>
<td>Default ddname or prefix of the DDLIN data set</td>
<td>“DDLDDN=DDLIN” on page 634</td>
</tr>
<tr>
<td>DEADLINE</td>
<td>NONE</td>
<td>Time by which the LOGFINAL phase should finish applying log records</td>
<td>“DEADLINE=NONE” on page 635</td>
</tr>
<tr>
<td>DELFILES</td>
<td>YES</td>
<td>Whether to delete data sets on completion</td>
<td>“DELFILES=YES” on page 636</td>
</tr>
<tr>
<td>DESCCEDE</td>
<td>(3,7)</td>
<td>Descriptor codes to control how WTO messages are displayed on a console</td>
<td>“DESCCEDE=(3,7)” on page 637</td>
</tr>
<tr>
<td>Option</td>
<td>Shipped value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>--------------------------------</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>“DRAINTYP=ALL” on page 637</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>“DRNDELAY=1” on page 638</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>“DRNRETRY=255” on page 638</td>
</tr>
<tr>
<td>DRNWAIT</td>
<td>NONE</td>
<td>Drain timeout value to use</td>
<td>“DRNWAIT=NONE” on page 638</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>“DSNUEXIT=(NONE,ASM)” on page 639</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>“DSNUTILB=YES” on page 640</td>
</tr>
<tr>
<td>DSPLOCKS</td>
<td>DRNFAIL</td>
<td>Whether to display claims and locks if a drain attempt times out</td>
<td>“DSPLOCKS=DRNFAIL” on page 641</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>“DSRSEXIT=(NONE,REXX)” on page 641</td>
</tr>
<tr>
<td>EXCLDUMP</td>
<td>(X37,X22,X06)</td>
<td>System codes to ignore during the dump process</td>
<td>“EXCLDUMP=(X37,X22,X06)” on page 642</td>
</tr>
<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>“FASTSWITCH=NO” on page 642</td>
</tr>
<tr>
<td>FILECHK</td>
<td>FAIL</td>
<td>Which action to take when encountering a temporary data set</td>
<td>“FILECHK=FAIL” on page 644</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>“FORCE=NONE” on page 644</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>“FORCE_AT=(START,3)” on page 645</td>
</tr>
<tr>
<td>FORCE_RPT</td>
<td>NO</td>
<td>Whether to display the canceled threads</td>
<td>“FORCE_RPT=NO” on page 646</td>
</tr>
<tr>
<td>FSFALLBACK</td>
<td>NO</td>
<td>Whether to perform rename processing when FASTSWITCH processing fails</td>
<td>“FSFALLBACK=NO” on page 646</td>
</tr>
<tr>
<td>FSTHRESHOLD</td>
<td>0</td>
<td>The minimum number of staging data sets that will trigger FASTSWITCH processing</td>
<td>“FSTHRESHOLD=0” on page 647</td>
</tr>
<tr>
<td>Option</td>
<td>Shipped value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>HASHAX</td>
<td>Not applicable</td>
<td>Obsolete</td>
<td>“HASHAX (obsolete)” on page 648</td>
</tr>
<tr>
<td>ICDDN</td>
<td>(BICY,BICZ)</td>
<td>The ddnames or ddname prefixes for the local copy data sets that receive an incremental image copy</td>
<td>“ICDDN=(BICY, BICZ)” on page 648</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>“ICTYPE=AUTO” on page 648</td>
</tr>
<tr>
<td>IDCACHE</td>
<td>10000</td>
<td>Cache size of document ID values</td>
<td>“IDCACHE=10000” on page 650</td>
</tr>
<tr>
<td>INDREFLM</td>
<td>10</td>
<td>Limit over which REORG PLUS is to reorganize the specified table space</td>
<td>“INDREFLM=10” on page 650</td>
</tr>
<tr>
<td>INLINECP</td>
<td>YES</td>
<td>Whether to create an inline image copy</td>
<td>“INLINECP=YES” on page 651</td>
</tr>
<tr>
<td>INLOB</td>
<td>Not applicable</td>
<td>Obsolete</td>
<td>“INLOB (obsolete)” on page 652</td>
</tr>
<tr>
<td>IXONEX</td>
<td>Not applicable</td>
<td>Obsolete</td>
<td>“IXONEX (obsolete)” on page 652</td>
</tr>
<tr>
<td>IXRANDOM</td>
<td>Not applicable</td>
<td>Obsolete</td>
<td>“IXRANDOM (obsolete)” on page 653</td>
</tr>
<tr>
<td>KEEPDICTIONARY</td>
<td>NO</td>
<td>Whether to keep an existing compression dictionary or build a new one</td>
<td>“KEEPDICTIONARY =NO” on page 653</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>“LEAFDSLM=200” on page 654</td>
</tr>
<tr>
<td>LOB</td>
<td>Not applicable</td>
<td>Obsolete</td>
<td>“LOB (obsolete)” on page 655</td>
</tr>
<tr>
<td>LOCKROW</td>
<td>YES</td>
<td>Serialization method for BMCSYNC and BMCUTIL</td>
<td>“LOCKROW=YES” on page 655</td>
</tr>
<tr>
<td>LOGFINAL</td>
<td>NONE</td>
<td>Time by which LOGFINAL is to start</td>
<td>“LOGFINAL=NONE” on page 655</td>
</tr>
<tr>
<td>LOGMEM</td>
<td>0</td>
<td>Amount of memory for log records</td>
<td>“LOGMEM=0” on page 656</td>
</tr>
<tr>
<td>LOGSPIL</td>
<td>(20000,10000)</td>
<td>Space allocation for spill data sets that hold the log records</td>
<td>“LOGSPIL=(20000,10000)” on page 657</td>
</tr>
<tr>
<td>LOGTHRSH</td>
<td>0</td>
<td>Number of remaining log records</td>
<td>“LOGTHRSH=0” on page 657</td>
</tr>
<tr>
<td>Option</td>
<td>Shipped value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>LONGLOG</td>
<td>CONTINUE</td>
<td>Which action to take when a long log condition occurs</td>
<td>“LONGLOG=CONTINUE” on page 657</td>
</tr>
<tr>
<td>LONGNAMETRUNC</td>
<td>MIDDLE (or M)</td>
<td>Long name truncation method for messages</td>
<td>“LONGNAMETRUNC=C=MIDDLE” on page 658</td>
</tr>
<tr>
<td>MAPTEXIT</td>
<td>(NONE,REXX)</td>
<td>Name of a user-written mapping table exit and the language in which the exit is written</td>
<td>“MAPTEXIT=(NONE,REXX)” on page 659</td>
</tr>
<tr>
<td>MAXNEWPARTS</td>
<td>2</td>
<td>Maximum number of partitions that can be added to partition-by-growth table spaces</td>
<td>“MAXNEWPARTS=2” on page 659</td>
</tr>
<tr>
<td>MAXRO</td>
<td>300</td>
<td>Maximum number of seconds that REORG PLUS spends applying log records</td>
<td>“MAXRO=300” on page 660</td>
</tr>
<tr>
<td>MAXSORTMEMORY</td>
<td>0</td>
<td>Maximum amount of memory that REORG PLUS can allocate to each sort task</td>
<td>“MAXSORTMEMORY=0” on page 661</td>
</tr>
<tr>
<td>MAXTAPE</td>
<td>3</td>
<td>Maximum number of tape devices to allocate dynamically</td>
<td>“MAXTAPE=3” on page 661</td>
</tr>
<tr>
<td>MGEVENTENT</td>
<td>CONTINUE</td>
<td>How to allocate extents when extending to a new data set</td>
<td>“MGEVENTENT=CONTINUE” on page 661</td>
</tr>
<tr>
<td>MINSORTMEMORY</td>
<td>0</td>
<td>Minimum amount of memory that REORG PLUS should allocate to each sort task</td>
<td>“MINSORTMEMORY=0” on page 662</td>
</tr>
<tr>
<td>MSGLEVEL</td>
<td>1</td>
<td>Default for the message level execution parameter</td>
<td>“MSGLEVEL=1” on page 662</td>
</tr>
<tr>
<td>OFFPOSLM</td>
<td>10</td>
<td>OFFPOS limit for table space reorganizations</td>
<td>“OFFPOSLM=10” on page 662</td>
</tr>
<tr>
<td>OPNDB2ID</td>
<td>YES</td>
<td>Whether to use the user’s RACF ID instead of the DB2 RACF ID</td>
<td>“OPNDB2ID=YES” on page 663</td>
</tr>
<tr>
<td>ORIGDISP</td>
<td>DELETE</td>
<td>Disposition of original data set during staging data set renaming process</td>
<td>“ORIGDISP=DELETE” on page 664</td>
</tr>
<tr>
<td>PENDDDL</td>
<td>Not applicable</td>
<td>Obsolete</td>
<td>“PENDDDL (obsolete)” on page 664</td>
</tr>
<tr>
<td>PLAN</td>
<td>ARU1120</td>
<td>Product plan name</td>
<td>“PLAN=ARU1120” on page 665</td>
</tr>
<tr>
<td>PREFORMAT</td>
<td>NO</td>
<td>Whether to preformat unused data set pages</td>
<td>“PREFORMAT=NO” on page 665</td>
</tr>
<tr>
<td>RCVICDDN</td>
<td>(BIRY,BIRZ)</td>
<td>The ddnames or ddname prefixes for the remote copy data sets that receive an incremental image copy</td>
<td>“RCVICDDN=(BIRY, BIRZ)” on page 665</td>
</tr>
<tr>
<td>Option</td>
<td>Shipped value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>RCVYDDN</td>
<td>(BRCY,BRCZ)</td>
<td>The ddnames or ddname prefixes for the remote copy data sets that receive a full image copy</td>
<td>“RCVYDDN=(BRCY, BRCZ)” on page 666</td>
</tr>
<tr>
<td>REDEFINE</td>
<td>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>“REDEFINE=YES” on page 666</td>
</tr>
<tr>
<td>RENMMAX</td>
<td>30</td>
<td>Maximum number of tasks to start during the staging data set rename and delete process</td>
<td>“RENMMAX=30” on page 667</td>
</tr>
<tr>
<td>RIDMDSSZ</td>
<td>2097152</td>
<td>Maximum data space size for RID maps</td>
<td>“RIDMDSSZ=2097152” on page 667</td>
</tr>
<tr>
<td>RIDMMAXD</td>
<td>1</td>
<td>Number of data spaces for RID maps</td>
<td>“RIDMMAXD=1” on page 668</td>
</tr>
<tr>
<td>RMAPMEM</td>
<td>0</td>
<td>Amount of memory for RID maps</td>
<td>“RMAPMEM=0” on page 668</td>
</tr>
<tr>
<td>RORGMAX</td>
<td>300%</td>
<td>Maximum number of tasks to start for the REORG phase</td>
<td>“RORGMAX=300%” on page 669</td>
</tr>
<tr>
<td>ROUTCDE</td>
<td>(11,1)</td>
<td>WTO console routing codes</td>
<td>“ROUTCDE=(11,1)” on page 669</td>
</tr>
<tr>
<td>SCPYMAX</td>
<td>8</td>
<td>Maximum number of tasks to start during the nonpartitioned index copy process</td>
<td>“SCPYMAX=8” on page 670</td>
</tr>
<tr>
<td>SDUMP</td>
<td>(ALLPSA,CSA,RGN,SQA,LSQA,SUM,TRT,IO)</td>
<td>System dump parameters</td>
<td>“SDUMP=(ALLPSA,CSA,RGN,SQA,LSQA,SUM,TRT,IO)” on page 670</td>
</tr>
<tr>
<td>SHORTMEMORY</td>
<td>CONTINUE</td>
<td>Which action to take when a memory shortage exists</td>
<td>“SHORTMEMORY=CONTINUE” on page 671</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>“SIXSNAP=NO” on page 672</td>
</tr>
<tr>
<td>SMAX</td>
<td>0</td>
<td>Maximum number of sort processes to invoke concurrently</td>
<td>“SMAX=0” on page 673</td>
</tr>
<tr>
<td>SMCORE</td>
<td>(0K,0K)</td>
<td>Maximum amount of sort memory</td>
<td>“SMCORE=(0K,0K)” on page 674</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>(,SYSALLDA)</td>
<td>Sort device types</td>
<td>“SORTDEVT=(,SYSALLDA)” on page 674</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>“SORTNUM=32” on page 675</td>
</tr>
<tr>
<td>Option</td>
<td>Shipped value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>SPILDSNP</td>
<td>&amp;&amp;UID</td>
<td>Pattern for spill data set name prefix</td>
<td>&quot;SPILDSNP=&amp;&amp;UID&quot; on page 676</td>
</tr>
<tr>
<td>SPILSCLS</td>
<td>NONE</td>
<td>SMS storage class for spill data sets</td>
<td>“SPILSCLS=NONE” on page 677</td>
</tr>
<tr>
<td>SPILUNIT</td>
<td>WORK</td>
<td>DASD unit for spill data sets</td>
<td>“SPILUNIT=WORK” on page 677</td>
</tr>
<tr>
<td>SQLDELAY</td>
<td>3</td>
<td>Number of seconds between retry attempts after SQLCODE -911</td>
<td>“SQLDELAY=3” on page 678</td>
</tr>
<tr>
<td>SQLRETRY</td>
<td>100</td>
<td>Number of retry attempts after SQLCODE -911</td>
<td>“SQLRETRY=100” on page 678</td>
</tr>
<tr>
<td>STAGEDSN</td>
<td>BMC</td>
<td>Naming convention to use for staging data sets</td>
<td>“STAGEDSN=BMC” on page 678</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>&quot;STOP@CMT=YES” on page 678</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>“STOPDELAY=1” on page 679</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>“STOPRETRY=300” on page 679</td>
</tr>
<tr>
<td>TAPEDISP</td>
<td>DELETE</td>
<td>Tape file disposition</td>
<td>“TAPEDISP=DELETE” on page 679</td>
</tr>
<tr>
<td>TASKMAX</td>
<td>1000%</td>
<td>Global maximum number of tasks to start</td>
<td>“TASKMAX=1000%” on page 680</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>“TERMEXIT=(NONE,REXX)” on page 680</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>TERM</td>
<td>Which action to take when a drain attempt times out</td>
<td>“TIMEOUT=TERM” on page 681</td>
</tr>
<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>“TOTALPAGEPCT=0” on page 682</td>
</tr>
<tr>
<td>TSSAMPLEPCT</td>
<td>100</td>
<td>Percentage of sampling to perform during statistics gathering</td>
<td>“TSSAMPLEPCT=100” on page 683</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>“UBUFFS=20” on page 684</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>“UNLDDN=SYSREC” on page 684</td>
</tr>
<tr>
<td>Option</td>
<td>Shipped value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UNLDMAX</td>
<td>300%</td>
<td>Maximum number of unload tasks to start</td>
<td>“UNLDMAX=300%” on page 684</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>RELOAD</td>
<td>Whether to use single- or two-phase processing</td>
<td>“UNLOAD=RELOAD” on page 685</td>
</tr>
<tr>
<td>UTILB_COLCCSID</td>
<td>Not applicable</td>
<td>Obsolete</td>
<td>“UTILB_COLCCSID (obsolete)” on page 685</td>
</tr>
<tr>
<td>UTILB_NULLIX</td>
<td>Not applicable</td>
<td>Obsolete</td>
<td>“UTILB_NULLIX (obsolete)” on page 686</td>
</tr>
<tr>
<td>UXSTATE</td>
<td>SUP</td>
<td>How to invoke DB2 user exits</td>
<td>“UXSTATE=SUP” on page 686</td>
</tr>
<tr>
<td>WBUFFS</td>
<td>(20,10)</td>
<td>Multiplier that controls the amount of buffer pool storage for each work data set</td>
<td>“WBUFFS=(20,10)” on page 686</td>
</tr>
<tr>
<td>WORKDDN</td>
<td>SYSUT1</td>
<td>Default ddname or ddname prefix for the index work data set</td>
<td>“WORKDDN=SYSUT1” on page 686</td>
</tr>
<tr>
<td>WORKUNIT</td>
<td>SYSALLDA</td>
<td>Temporary unit for work data sets</td>
<td>“WORKUNIT=SYSA LLDA” on page 687</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>“XBMID=” on page 687</td>
</tr>
<tr>
<td>XBMID example</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZIIP</td>
<td>ENABLED</td>
<td>Whether to enable zIIP processing</td>
<td>“ZIIP=ENABLED” on page 688</td>
</tr>
</tbody>
</table>

### Descriptions of basic REORG PLUS installation options

This topic describes each of the basic installation options.

For more information about setting the values of some of these options at runtime, see “Syntax of the REORG command” on page 143.

#### ALTRFAIL (obsolete)

In REORG PLUS version 11.1 and earlier, the ALTRFAIL option specified the action that REORG PLUS took when a failure occurred during processing of limit-key updates. This option applied to limit-key updates that were processed either from a DDLIN data set or as a result of the REBALANCE option, and applied only to a SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganization.
Beginning with REORG PLUS version 11.2, if a failure occurs when processing limit key updates, REORG PLUS functions as described in “General rebalancing considerations” on page 116.

**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.
- \( n \) 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 more information about using the multitasking options, see “Multitasking installation options” on page 535.

**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 sets in REORG PLUS” on page 353.

**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.

**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 sets in REORG PLUS” on page 353.

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.

- 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.

**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 244.

**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 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.

**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.
- \( n \) 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 535.

**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. See the TERMEXIT option for more information. For information
about the TERMEXIT user exit, see “Using TERMEXIT to control BMCHIST and statistics updates” on page 801.

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.

**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 527.

**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 124. For more information about conditional execution using the DASD MANAGER PLUS tables, see “Conditional reorganizations using the DASD MANAGER PLUS exceptions table” on page 127.

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 Performance for DB2 Databases or 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.

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 in REORG PLUS” on page 336.

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.
**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, ICDDN, RCVYDDN, and RCVICDDN options. The values that you specify for these options plus the highest partition number must not exceed eight characters.

  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.

  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.

**Overriding this option**

You can override the value for this option by using the COPYLVL command option.
**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
- \( n \) 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 535.

**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:
CPYFAIL=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.

REORG PLUS changes TERM to COPYPEND when any of the following conditions exists:

— Pending definition changes exist on the object and REORG PLUS can materialize those changes

— You are rebalancing partitions (using either the REBALANCE option or a DDLIN data set)

CPYFAIL=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 you are reorganizing a catalog object. This value change ensures that REORG PLUS can return the table space to its original state.

**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 “DDLIN data sets in REORG PLUS” on page 344.

*Overriding this option*

You can also specify the DDLIN data set name by using the DDLDDN command option.
**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 in REORG PLUS” on page 582.

**Overriding this option**

You can override the value for this option by using the DEADLINE command option.

**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 in REORG PLUS” on page 582.

**Overriding this option**

You can override the value for this option by using the DELAY command option.

### 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.

**DESCCDE=(3,7)**

The DESCcDE 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.

**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.

**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.

**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.

**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 the descriptions of these options.

---

**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.

**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 SPILDSNP installation option or the DSNPAT or SPILLDSNPAT 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 DSNEXIT user exit, see “Using DSNEXIT to construct data set name patterns in REORG PLUS” on page 748.

**Overriding this option**

You can override the value for this option by using the DSNEXIT command option.

**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 57.

**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 57.

*Overriding this option*

You can override the value for this option by using the DSNUTILB command option.

**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.

**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 “REORG PLUS user exits” on page 747.

Overriding this option

You can override the value for this option by using the DSRSEXIT command option.

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 and FSTHRESHOLD 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.
  
  If FASTSWITCH NO is in effect, REORG PLUS does not materialize any pending definition changes.

- **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 92.

**Overriding this option**

You can override the value for this option by using the FASTSWITCH command option.
**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 83.

- 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. 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 or DRNRETRY installation option.

**Note**

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 DRNWAIT option) is shorter than the total cancelation process time, the drain fails. The START and RETRY values allow for additional drain attempts after the start of the cancelation process; therefore, these values 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.)

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**

Basic REORG PLUS installation options

Appendix A  REORG PLUS installation options  645
You can override the value for the first parameter of this option by using the FORCE_AT suboption of the FORCE command option. You can override the value for the second parameter by using the FORCE_DELAY suboption of the FORCE command option.

**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.

**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.
  - Pending definition changes exist on the object that you are reorganizing.
  - You are rebalancing partitions.
  - REORG PLUS is invoking DSNUTILB.
  - REORG PLUS is converting objects from BRF to RRF.
  - REORG PLUS is converting RBA or LRSN format.

Overriding this option

You can override the value for this option by using the FSFALLBACK command option.

**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.

*Restrictions*

REORG PLUS ignores this option when either of the following conditions exists:

- FASTSWITCH NO is in effect.
- Pending definition changes exist on the object that you are reorganizing.

Overriding this option

You can override the value for this option by using the FSTHRESHOLD command option.
**HASHAX (obsolete)**

In REORG PLUS versions earlier than 11.2, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) a table space that contained a table that was defined as ORGANIZE BY HASH.

Beginning with version 11.2, REORG PLUS no longer requires or supports the HASHAX installation 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.

**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 602.

*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.

**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).

**Overriding this option**

You can override the value for this option by using the ICTYPE command option.

**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.

**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.

\[(\text{NEARINDREF} + \text{FARINDREF}) \times 100 / \text{CARD}]=\text{integer}\]

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. 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 124.

**Tip**

You can obtain the report that recommends objects for reorganization without performing any reorganizations. To do so, specify REPORTONLY in the command.

**INLINECP=Yes**

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.

**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.

**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.

**IXONEX (obsolete)**

In REORG PLUS versions earlier than 11.2, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) indexes that contained a key derived from certain expressions or table spaces that contained this type of index.

Beginning with version 11.2, REORG PLUS no longer requires or supports the IXONEX installation 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.

**IXRANDOM (obsolete)**

In REORG PLUS versions earlier than 11.2, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) indexes that contained a key with random ordering or table spaces that contained this type of index.

Beginning with version 11.2, REORG PLUS no longer requires or supports the IXRANDOM installation 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.

**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 in REORG PLUS” on page 134.

- **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.

Overriding this option

You can override the value for this option by using the KEEPDICTIONARY command option.

**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. 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 124.

**Tip**

You can obtain the report that recommends objects for reorganization without performing any reorganizations. To do so, specify REPORTONLY in the command.
LOB (obsolete)

In REORG PLUS versions earlier than 11.2, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces that contained a LOB column, LOB table spaces, or both when a DSNUTILB reorganization was required.

Beginning with version 11.2, REORG PLUS no longer requires or supports the LOB installation 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.

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=NULL

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 in REORG PLUS” on page 582.

- You can specify a time in \texttt{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.

\textit{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.

\textit{Overriding this option}

You can override the value for this option by using the LOGFINAL command option.

\textbf{LOGMEM=0}

\textit{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 611.

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.

**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.

**LOGTHRSH=0**

*This option applies to SHRLEVEL CHANGE only.*

The LOGTHRSH 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 in REORG PLUS” on page 582.

*Overriding this option*

You can override the value for this option by using the LOGTHRESHLD command option.

**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 in REORG PLUS” on page 582.

**Overriding this option**

You can override the value for this option by using the LONGLOG command option.

### 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. The following table 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;&gt;</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.
MAPEXIT=(NONE,REXX)

The MAPEXIT 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 MAPEXIT user exit, see “Using MAPEXIT to create mapping objects” on page 808.

Restrictions

The following restrictions apply to the MAPEXIT option and the MAPEXIT user exit:

■ REORG PLUS terminates if you also specify the MAPPINGDATABASE option.

■ 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 also specify the MAPPINGTABLE command option.
  — You specify REORG INDEX.

Overriding this option

You can override the value for this option by using the MAPEXIT command option.

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.
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 in REORG PLUS” on page 336 and “SYSREC data sets in REORG PLUS” on page 362.

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.

**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 in REORG PLUS” on page 582.

**Overriding this option**

You can override the value for this option by using the MAXRO command option.
**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.

*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.

**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.

**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.

For information about how this option interacts with the SMCORE installation option, see the description of the SMCORE option.

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.

**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.

**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.

\[(\text{NEAROFFPOSF} + \text{FAROFFPOSF}) \times 100 / \text{CARDF} = \text{integer}\]

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. 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 124.

*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.

- **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 “Setting REORG PLUS authorizations” on page 53.
Using OPNDDB2ID=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.

**Tip**

**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 87.

  When invoking DSNUTILB, REORG PLUS ignores this option.

  **Overriding this option**

  You can override the value for this option by using the ORIGINADISP command option.

**PENDDDL (obsolete)**

In REORG PLUS versions earlier than 11.2, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces or indexes that contained pending DDL changes.

Beginning with version 11.2, REORG PLUS no longer requires or supports the PENDDDL installation 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.
**PLAN=ARU1120**

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.

**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 602.
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.

**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 in REORG PLUS” on page 336.

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.

**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. 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

- \( n \) 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)

When invoking DSNUTILB, REORG PLUS ignores this option.

For information about using the multitasking options, see “Multitasking installation options” on page 535.

**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 611. 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.

**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
- 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 535.

**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
- \( n \) 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 535.

**SDUMP=(ALLPSA,CSA,RGBN,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.
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).

The following table 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
- Value specified for the MINSORTMEMORY installation or command option
### Table 111: Action REORG PLUS takes when memory resources are constrained

<table>
<thead>
<tr>
<th>Memory shortage location</th>
<th>Memory shortage condition</th>
<th>SHORTMEMORY=CONTINUE</th>
<th>SHORTMEMORY=FAIL</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>
<td>REORG PLUS fails.</td>
</tr>
<tr>
<td></td>
<td>Insufficient as specified by MINSORTMEMORY</td>
<td>REORG PLUS runs one task with the available memory.</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>REORG PLUS runs one task with at least 1024 KB, whichever is greater</td>
<td>REORG PLUS fails.</td>
</tr>
<tr>
<td></td>
<td>but sufficient as specified by MINSORTMEMORY or at least 1024 KB, whichever is greater</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Insufficient as specified by MINSORTMEMORY or less than 1024 KB, whichever is greater</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Insufficient as specified by the first parameter of SMCORE</td>
<td></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.

**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 112.

For detailed information about the SIXSNAP function, see “Instant Snapshot with nonpartitioned indexes” on page 110. 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.

**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 535.

For information about the hierarchy of the SMAX, MAXSORTS, and multitasking options, see Table 101 on page 536. For more information, see “Controlling the number of sort processes” on page 533.

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.

**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 531.

**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.
**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. The following table 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 112: SORTNUM 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 350. For more information about the parameter itself, see “DYNALOC installation option” on page 711.</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 “SORTWK data sets in REORG PLUS” on page 349.

**Overriding this option**

You can override the value for this option with the SORTNUM command option.
**SPILDSNP=&&UID**

*This option applies to SHRLEVEL CHANGE only.*

The SPILDSNP 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 the following table 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 113: Symbolic variables for the SPILDSNP installation option**

<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>
<tr>
<td>&amp;USERID or &amp;UID</td>
<td>Job user ID</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td></td>
<td>You must have a security package to use the job user ID variable.</td>
<td></td>
</tr>
<tr>
<td>Symbolic variable</td>
<td>Definition</td>
<td>Length of result</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>&amp;UTIL</td>
<td>BMC utility ID</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td></td>
<td>REORG PLUS truncates longer utility IDs to eight characters.</td>
<td></td>
</tr>
<tr>
<td>&amp;UTILPFX</td>
<td>BMC utility ID prefix</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;UTILSFX</td>
<td>BMC utility ID suffix</td>
<td>8 bytes maximum</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>
</tr>
</tbody>
</table>

For more information and guidelines for specifying data set name patterns, see “SPILLDSNPAT” on page 314. User exits are discussed in “REORG PLUS user exits” on page 747.

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.

**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.

**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.

**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 87.

**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 DELETEFILES 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 535.

**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 “Using TERMEXIT to control BMCHIST and statistics updates” on page 801.

**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.
- REORG PLUS always updates DB2 statistics after materializing pending definition changes. If you have a TERMEXIT user exit that changes UPDATEDB2STATS to NO in this case, REORG PLUS changes that specification to UPDATEDB2STATS YES.

**Overriding this option**

You can override the value for this option by using the TERMEXIT command option.
**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.

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 the following table.
### Table 114: 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.

**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 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.

**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.
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 525.

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 in REORG PLUS” on page 362.

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.

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
- $n$ 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 535.

**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 “UNLOAD” on page 192.

- **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 133.

- **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.

**Overriding this option**

You can override the value for this option by using the UNLOAD command option. An additional value is available with the command option.

**UTILB_COLCCSID (obsolete)**

In REORG PLUS versions earlier than 11.2, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) a table space that contained columns defined with a CCSID specification.

Beginning with version 11.2, REORG PLUS no longer requires or supports the UTILB_COLCCSID installation 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.
**UTILB_NULLIX (obsolete)**

In REORG PLUS versions earlier than 11.2, this option told REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces that contained indexes that were defined with EXCLUDE NULL KEYS.

Beginning with version 11.2, REORG PLUS no longer requires or supports the UTILB_NULLIX installation 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. For more information, see “SYSUT1 data set” on page 526.

---

**Note**

REORG PLUS now ignores the second parameter of this option.

**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 in REORG PLUS” on page 366.

**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.

**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 for REORG PLUS” on page 140. 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.

**ZIIP=ENABLED**

The ZIIP option tells REORG PLUS whether to attempt to use IBM 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.

- 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.

**Overriding this option**

You can override the value for this option by using the ZIIP command option.
Dynamic allocation installation options for REORG PLUS

The $ARUDYNA macros produce the installation options for dynamic allocation (one macro for each data set type).

The following table 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 $730DOP2 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>“DDTYPE=UNLOAD” on page 691</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>“ACTIVE=NO or ACTIVE=YES” on page 692</td>
</tr>
<tr>
<td>ALLOC</td>
<td>ANY</td>
<td>Method to use when dynamically allocating sort work files This installation option is valid only for the SORTWORK work file type.</td>
<td>“ALLOC=ANY” on page 693</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>“AVGVOLSP=((30000,TRK),(30000,TRK))” on page 694</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>(NONE,NONE)</td>
<td>SMS data class to use</td>
<td>“DATACLAS=(NONE,NONE)” on page 695</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;DDNAME</td>
<td>Data set name pattern</td>
<td>“DSNPAT=&amp;UID.&amp;JOBNAME.&amp;TSIX.&amp;DDNAME” on page 695</td>
</tr>
<tr>
<td>Option</td>
<td>Shipped value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------</td>
<td>-------------------------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>DSNTYPE</td>
<td>(NONE,NONE)</td>
<td>Data set type</td>
<td>“DSNTYPE=(NONE,NONE)” on page 700</td>
</tr>
<tr>
<td>EXPDT</td>
<td>No value</td>
<td>Expiration date for copy and archive data sets</td>
<td>“EXPDT=“ on page 701</td>
</tr>
<tr>
<td>GDGEMPTY</td>
<td>NO</td>
<td>Whether to uncatalog all data sets when the limit is reached</td>
<td>“GDGEMPTY=NO” on page 701</td>
</tr>
<tr>
<td>GDGLIMIT</td>
<td>5</td>
<td>Number of GDG generations to keep</td>
<td>“GDGLIMIT=5” on page 702</td>
</tr>
<tr>
<td>GDGSCRATCH</td>
<td>NO</td>
<td>Whether to delete uncataloged data sets</td>
<td>“GDGSCRATCH=NO” on page 702</td>
</tr>
<tr>
<td>IFALLOC</td>
<td>USE</td>
<td>Whether to use DDs in the JCL if they are coded</td>
<td>“IFALLOC=USE” on page 702</td>
</tr>
<tr>
<td>MAXEXTSZ</td>
<td>((0,K),(0,K))</td>
<td>Maximum extent size</td>
<td>“MAXEXTSZ=((0,K),(0,K))” on page 703</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>(NONE,NONE)</td>
<td>SMS management class to use</td>
<td>“MGMTCLAS=(NONE,NONE)” on page 704</td>
</tr>
<tr>
<td>RETPD</td>
<td>No value</td>
<td>Number of days to retain copy and archive data sets</td>
<td>“RETPD=“ on page 705</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>“SIZEPCT=(100,100) or SIZEPCT=(5,100)” on page 705</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>“SMS=NO” on page 706</td>
</tr>
<tr>
<td>SMSUNIT</td>
<td>NO</td>
<td>Whether to pass the unit value to SMS</td>
<td>“SMSUNIT=NO” on page 706</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>(NONE,NONE)</td>
<td>SMS storage class to use</td>
<td>“STORCLAS=(NONE,NONE)” on page 707</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>“THRESHLD=0” on page 707</td>
</tr>
</tbody>
</table>
Option | Shipped value | Brief description | Reference
--- | --- | --- | ---
UNIT | (SYSALLDA,SYSALLDA) | Unit names used for dynamic allocation | “UNIT=(SYSALLDA,SYSALLDA)” on page 709
UNITCNT | (0,0) | Number of devices to dynamically allocate | “UNITCNT=(0,0)” on page 709
VOLCNT | (25,25) | Largest number of volumes to process | “VOLCNT=(25,25)” on page 710

**Descriptions of dynamic allocation installation options for REORG PLUS**

This section describes each of the dynamic allocation installation options.

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 for REORG PLUS” on page 277.

**DDTYPE=UNLOAD**

The DDTYPE option specifies the data set type for which you are establishing dynamic allocation options. The following table lists the valid values for the DDTYPE option. Each instance of the $ARUDYNA macro must contain a different value for this option.

**Table 116: Valid DDTYPE values**

<table>
<thead>
<tr>
<th>DDTYPE value</th>
<th>Corresponding default ddname</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLOAD</td>
<td>SYSREC</td>
</tr>
<tr>
<td>WORK</td>
<td>SYSUT1</td>
</tr>
<tr>
<td>SORTWORK</td>
<td>SORTWK</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>SYSARC</td>
</tr>
<tr>
<td>SYSPUNCH</td>
<td>SYSPUNCH</td>
</tr>
<tr>
<td>LOCPFCPY</td>
<td>BCPY</td>
</tr>
<tr>
<td>LOCPICPY</td>
<td>BICY</td>
</tr>
<tr>
<td>LOCBFCPY</td>
<td>BCPZ</td>
</tr>
<tr>
<td>LOCBICPY</td>
<td>BICZ</td>
</tr>
</tbody>
</table>
Dynamic allocation installation options for REORG PLUS

<table>
<thead>
<tr>
<th>DDTYPE value</th>
<th>Corresponding default ddname</th>
</tr>
</thead>
<tbody>
<tr>
<td>REMPFCOPY</td>
<td>BRCY</td>
</tr>
<tr>
<td>REMPICPY</td>
<td>BIRY</td>
</tr>
<tr>
<td>REMBFCOPY</td>
<td>BRCZ</td>
</tr>
<tr>
<td>REMBICPY</td>
<td>BIRZ</td>
</tr>
<tr>
<td>LOCPFCPY</td>
<td>for future use</td>
</tr>
<tr>
<td>LOCPICPY</td>
<td>for future use</td>
</tr>
<tr>
<td>REMPLCPY</td>
<td>for future use</td>
</tr>
<tr>
<td>REMBLCPY</td>
<td>for future use</td>
</tr>
</tbody>
</table>

**Overriding this option**

You can override each instance of the DDTYPE installation option by using the DDTYPE command option.

**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 117: ACTIVE option default values by DDTYPE**

<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>REMPFCPY</td>
<td>NO</td>
</tr>
<tr>
<td>REMPICPY</td>
<td>NO</td>
</tr>
<tr>
<td>REMBFCOPY</td>
<td>NO</td>
</tr>
</tbody>
</table>
Additional considerations

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.

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.

**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.

REORG PLUS uses the second parameter (`integer2`) when the value for the THRESHLD installation option or command 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.

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.

**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 the description of the THRESHLD option.

Overriding this option

You can override the value for this option by using the DATACLAS command option.

**DSNPAT=&amp;UID.&amp;JOBNAME.&amp;TSIX.&amp;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 118: 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;&amp;JOBNAME.&amp;&amp;TSIX.&amp;&amp;DDNAME</td>
</tr>
<tr>
<td>WORK</td>
<td>&amp;&amp;UID.&amp;&amp;JOBNAME.&amp;&amp;TSIX.&amp;&amp;DDNAME</td>
</tr>
<tr>
<td>SORTWORK</td>
<td>NONE</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>&amp;&amp;UID.&amp;&amp;UTILPFX.&amp;&amp;DDNAME</td>
</tr>
<tr>
<td>SYSPUNCH</td>
<td>&amp;&amp;UID.&amp;&amp;UTILPFX.&amp;&amp;DDNAME</td>
</tr>
<tr>
<td>LOCPFCPY</td>
<td>&amp;&amp;UID.&amp;&amp;DDNAME.&amp;&amp;TSIX..F&amp;&amp;PART..T&amp;&amp;TIME</td>
</tr>
<tr>
<td>LOCPICPY</td>
<td>&amp;&amp;UID.&amp;&amp;DDNAME.&amp;&amp;TSIX..I&amp;&amp;PART..T&amp;&amp;TIME</td>
</tr>
<tr>
<td>LOCBFCPY</td>
<td>&amp;&amp;UID.&amp;&amp;DDNAME.&amp;&amp;TSIX..F&amp;&amp;PART..T&amp;&amp;TIME</td>
</tr>
<tr>
<td>LOCBICPY</td>
<td>&amp;&amp;UID.&amp;&amp;DDNAME.&amp;&amp;TSIX..I&amp;&amp;PART..T&amp;&amp;TIME</td>
</tr>
<tr>
<td>REMPFCPY</td>
<td>&amp;&amp;UID.&amp;&amp;DDNAME.&amp;&amp;TSIX..F&amp;&amp;PART..T&amp;&amp;TIME</td>
</tr>
<tr>
<td>REMPICPY</td>
<td>&amp;&amp;UID.&amp;&amp;DDNAME.&amp;&amp;TSIX..I&amp;&amp;PART..T&amp;&amp;TIME</td>
</tr>
<tr>
<td>REMBFCPY</td>
<td>&amp;&amp;UID.&amp;&amp;DDNAME.&amp;&amp;TSIX..F&amp;&amp;PART..T&amp;&amp;TIME</td>
</tr>
<tr>
<td>REMBICPY</td>
<td>&amp;&amp;UID.&amp;&amp;DDNAME.&amp;&amp;TSIX..I&amp;&amp;PART..T&amp;&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 &VCAT, &DATEJ, and &TIME4 to generate unique names across multiple reorganizations.

- You can use text or any of the symbolic variables in the table in this section 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 297.

- When specifying a pattern in your installation options, you must precede each REORG variable with an additional ampersand (&) 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 the table in this section) 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, 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 119: Symbolic variables for the DSNPAT installation option

<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>The 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 In a nondata-sharing environment, GRPNM contains the DB2 SSID.</td>
<td>4 bytes</td>
<td>Value passed</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 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.</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>
<tr>
<td>Symbolic variable</td>
<td>Definition</td>
<td>Length of result</td>
<td>DSNUTILB reorganization</td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>-----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>&amp;PART5</td>
<td>Partition being used 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. 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>5 bytes</td>
<td>Variable passed</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 REORG PLUS ignores PROC names.</td>
<td>8 bytes maximum</td>
<td>Variable passed</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 You must have a security package to use this variable.</td>
<td>8 bytes maximum</td>
<td>Variable passed</td>
</tr>
<tr>
<td>&amp;UTIL a</td>
<td>BMC utility ID REORG PLUS truncates longer utility IDs to eight characters.</td>
<td>8 bytes maximum</td>
<td>Variable translated to the IBM &amp;UTILID variable</td>
</tr>
<tr>
<td>&amp;UTILPFX a</td>
<td>BMC utility ID prefix</td>
<td>8 bytes maximum</td>
<td>Value passed</td>
</tr>
<tr>
<td>&amp;UTILSFX a</td>
<td>BMC utility ID suffix</td>
<td>8 bytes maximum</td>
<td>Value passed</td>
</tr>
</tbody>
</table>
Utility IDs that include special characters might cause REORG PLUS to generate invalid data set names. For more information, see “DSNPAT” on page 297.

**GDG names**

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 in REORG PLUS” on page 79.

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)'
```

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 “DSNPAT” on page 297.

**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.

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.

- **EXTRREQ** 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.

**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.

**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.

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.

**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

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. 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 the description of the THRESHLD option.

**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.

**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 the description of the THRESHLD option.

**Overriding this option**

You can override the value for this option by using the MGMTCLAS command option.
**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 or EXPDT command option. 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.
**SMS=NO**

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.

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.

**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.
**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 the description of the THRESHLD option.

### Overriding this option

You can override the value for this option by using the STORCLAS command option.

**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)
  - VOLUMCNT (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.

**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 “THRESHLD” on page 295.

**Overriding this option**

You can override the value for this option by using the THRESHLD command option.
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.

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 the description of the THRESHLD option.

Overriding this option

You can override the value for this option by using the UNIT command option.

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 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 the description of the THRESHLD option.
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.

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.

If you specify a second value (integer2), REORG PLUS uses that value when the value for the THRESHLD installation or command option 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 VOLCNT parameter in that template. For information about when REORG PLUS builds a secondary template, see the description of the THRESHLD option.

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.

---

**DYNALOC installation option**

The DYNALOC installation option of BMCSORT 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 $532SOPT follows, showing DYNALOC and the values that are shipped with BMCSORT.

```
$AUPSMAC DYNALOC=(SYSDA,3,ON,ON,6000000,3000000,3390,SC=,RETRY=(0,0)) X
```

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 120 on page 712 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 one 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 120: DYNALOC parameters

<table>
<thead>
<tr>
<th>Parameter name or position</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position 1</td>
<td>This parameter specifies the generic unit name from which the BMC product should dynamically allocate SORTWK data sets. This parameter applies only when the Data Facility Storage Management System (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>
</tr>
<tr>
<td></td>
<td>Initial value: SYSDA</td>
</tr>
<tr>
<td></td>
<td>Valid values: Use a unit name up to 8 characters.</td>
</tr>
<tr>
<td>Position 2</td>
<td>Do not change this value. The BMC product does not use this parameter, but the parameter is required for proper assembly of the installation options macro.</td>
</tr>
<tr>
<td></td>
<td>Initial value: 3</td>
</tr>
<tr>
<td></td>
<td>Valid values: Do not change this value.</td>
</tr>
<tr>
<td>Position 3</td>
<td>This parameter tells the BMC product whether to dynamically allocate SORTWK files. BMC recommends that you not change this value.</td>
</tr>
<tr>
<td></td>
<td>Initial value: ON</td>
</tr>
<tr>
<td></td>
<td>Valid values: ON dynamically allocates SORTWK. OFF does not dynamically allocate SORTWK.</td>
</tr>
<tr>
<td>Position 4</td>
<td>Do not change this value. The BMC product does not use this parameter, but the parameter is required for proper assembly of the installation options macro.</td>
</tr>
<tr>
<td></td>
<td>Initial value: ON</td>
</tr>
<tr>
<td></td>
<td>Valid values: Do not change this value.</td>
</tr>
<tr>
<td>Position 5</td>
<td>Do not change this value. The BMC product does not use this parameter, but the parameter is required for proper assembly of the installation options macro.</td>
</tr>
<tr>
<td></td>
<td>Initial value: 6000000</td>
</tr>
<tr>
<td></td>
<td>Valid values: Do not change this value.</td>
</tr>
<tr>
<td>Position 6</td>
<td>Do not change this value. The BMC product does not use this parameter, but the parameter is required for proper assembly of the installation options macro.</td>
</tr>
<tr>
<td></td>
<td>Initial value: 3000000</td>
</tr>
<tr>
<td></td>
<td>Valid values: Do not change this value.</td>
</tr>
<tr>
<td>Position 7</td>
<td>This parameter specifies the DASD type with the smallest track capacity that a dynamically allocated SORTWK data set might encounter at your site.</td>
</tr>
<tr>
<td></td>
<td>Initial value: 3390</td>
</tr>
<tr>
<td></td>
<td>Valid values: 3380, track capacity of 47968 3390, track capacity of 56664 9345, track capacity of 46456</td>
</tr>
<tr>
<td>Parameter name or position</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SC</td>
<td>This parameter specifies the name of the DFSMS storage class from which to</td>
</tr>
<tr>
<td></td>
<td>dynamically allocate SORTWK. If DFSMS is active and you do not specify a</td>
</tr>
<tr>
<td></td>
<td>value for this parameter, the BMC product uses the value from the first</td>
</tr>
<tr>
<td></td>
<td>DYNALOC parameter.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If your installation has an automatic class selection (ACS)</td>
</tr>
<tr>
<td></td>
<td>routine, it can override this specification.</td>
</tr>
<tr>
<td>RETRY</td>
<td>This parameter specifies how you want the BMC product to handle retry</td>
</tr>
<tr>
<td></td>
<td>attempts for SORTWK dynamic allocation:</td>
</tr>
<tr>
<td></td>
<td>■ The first subparameter indicates the number of times that you want the</td>
</tr>
<tr>
<td></td>
<td>BMC product to retry the request.</td>
</tr>
<tr>
<td></td>
<td>■ The second subparameter indicates the number of minutes to wait</td>
</tr>
<tr>
<td></td>
<td>between each retry.</td>
</tr>
<tr>
<td></td>
<td>Using this parameter allows you to avoid a capacity-exceeded condition</td>
</tr>
<tr>
<td></td>
<td>when disk space is not immediately available for a SORTWK dynamic allocation</td>
</tr>
<tr>
<td></td>
<td>request.</td>
</tr>
<tr>
<td></td>
<td>BMC recommends that you do not change this value because it can affect</td>
</tr>
<tr>
<td></td>
<td>the elapsed time of your jobs. However, if you currently use SyncSort and</td>
</tr>
<tr>
<td></td>
<td>rely on the retry function, BMC recommends that you use the same values as</td>
</tr>
<tr>
<td></td>
<td>your SyncSort RETRY installation parameter.</td>
</tr>
</tbody>
</table>

Appendix A  REORG PLUS installation options 713
Common utility tables

This chapter describes the contents of the common utility tables, considerations for these tables, and how to maintain them if necessary.

Overview of common utility tables

The BMC common utility tables contain information about the BMC utilities that you generate and submit through a BMC utility product.

Table 121 on page 715 lists the tables that each utility uses and each table’s default name and synonym.

Table 121: Common utility tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Default name</th>
<th>Synonym</th>
<th>Utilities that use this table</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCDICT</td>
<td>CMN_BMCDICT</td>
<td>BMC_BMCDICT</td>
<td>■ LOADPLUS&lt;br&gt; ■ REORG PLUS</td>
</tr>
<tr>
<td>BMCHIST</td>
<td>CMN_BMCHIST</td>
<td>BMC_BMCHIST</td>
<td>■ CHECK PLUS&lt;br&gt; ■ COPY PLUS&lt;br&gt; ■ LOADPLUS&lt;br&gt; ■ RECOVER PLUS&lt;br&gt; ■ REORG PLUS&lt;br&gt; ■ UNLOAD PLUS</td>
</tr>
</tbody>
</table>
### Table of Common Utility Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Default name</th>
<th>Synonym</th>
<th>Utilities that use this table</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCLGRNX</td>
<td>CMN_BMCLGRNX</td>
<td>BMC_BMCLGRNX</td>
<td>- COPY PLUS&lt;br&gt;- Log Master&lt;br&gt;- RECOVER PLUS&lt;br&gt;- RECOVERY MANAGER</td>
</tr>
<tr>
<td>BMCSYNC</td>
<td>CMN_BMCSYNC</td>
<td>BMC_BMCSYNC</td>
<td>- CHECK PLUS&lt;br&gt;- COPY PLUS&lt;br&gt;- DASD MANAGER PLUS (BMCSTATS)&lt;br&gt;- LOADPLUS&lt;br&gt;- RECOVER PLUS&lt;br&gt;- RECOVERY MANAGER&lt;br&gt;- REORG PLUS&lt;br&gt;- UNLOAD PLUS</td>
</tr>
<tr>
<td>BMCTRANS</td>
<td>CMN_BMCTRANS</td>
<td>BMC_BMCTRANS</td>
<td>- Log Master&lt;br&gt;- RECOVERY MANAGER</td>
</tr>
<tr>
<td>Table</td>
<td>Default name</td>
<td>Synonym</td>
<td>Utilities that use this table</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
<td>---------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>CMN_BMCUTIL</td>
<td>BMC_BMCUTIL</td>
<td>• CHECK PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• COPY PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DASD MANAGER PLUS (BMCSTATS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• RECOVER PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• RECOVERY MANAGER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• UNLOAD PLUS</td>
</tr>
<tr>
<td>BMCXCOPY</td>
<td>CMN_BMCXCOPY</td>
<td>BMC_BMCXCOPY</td>
<td>• COPY PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Log Master</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• RECOVER PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• RECOVERY MANAGER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• UNLOAD PLUS</td>
</tr>
</tbody>
</table>

**Warnings and considerations for common utility tables**

This topic describes important information that you need to know when using the common utility tables.
The following warnings apply to the common utility tables:

- 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 any BMC table except 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.

Note the following considerations:

- 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

- Supported versions of the following BMC products support the LOCKROW installation option:
  
  - CHECK PLUS
  - LOADPLUS
  - REORG PLUS
UNLOAD PLUS

If the value of the option is YES (which is the default value), the products use MVS enqueues instead of SQL LOCK TABLE statements to serialize updates to the BMCSYNC and BMCUTIL tables.

The following BMC products always use MVS enqueues for serialization when updating the BMCSYNC and BMCUTIL tables:

— COPY PLUS
— DASD MANAGER PLUS
— RECOVER PLUS

Managing common utility tables

This topic 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 query the tables

1  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:

```
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:

```
SELECT DBNAME, SPNAME, CHAR(ELAPSED, ISO), CHAR(TIME, ISO)
FROM creatorName.CMN_BMCHIST
WHERE UTILID = 'utilityID';
```

To display BMC utility status

1. 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:

```
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

1. To terminate a BMC utility, perform one of the following actions:

   ■ To terminate a BMC utility that is executing, use the following SQL statements:

   ```
   DELETE FROM creatorName.CMN_BMCUTIL
   WHERE UTILID = 'utilityID';
   DELETE FROM creatorName.CMN_BMCSYNC
   WHERE UTILID = 'utilityID';
   DELETE FROM creatorName.CMN_BMCDICT
   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 the TERM restart parameter.

BMCDICT table

The BMCDICT table stores the compression dictionary during load or reorganization processing.

Table 122 on page 721 describes the contents of the BMCDICT table.
Table 122: Contents of the BMCDICT table

<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></td>
<td></td>
<td>For a nonpartitioned table space, the value is 0.</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>

BMCDICT table considerations

This topic describes important information that you need to know about the BMCDICT table:

- 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 or 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.

On rare occasions, you might need to take action to control expansion of the BMCDICT table.

To control expansion of the BMCDICT table

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:

```
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**

The BMCHIST table 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.

For COPY PLUS, if the value is SUMMARY, the utility inserts only summary information about the COPY PLUS execution into the BMCHIST table. This option provides less information than the YES option.

Table 123 on page 722 describes the contents of the BMCHIST table.

**Table 123: Contents of the 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>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 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 (COPY PLUS) or the PART option</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note the following conditions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If the list of partitions exceeds 1011 bytes, the utility truncates the value that is stored in this column.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- For UNLOAD PLUS, if you specified LOGICAL PART, these partitions are the physical partitions that correspond to the logical partitions that you specified.</td>
</tr>
<tr>
<td>OBJNAME</td>
<td>VARCHAR(27)</td>
<td>Fully qualified object name</td>
</tr>
<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>
</tbody>
</table>
BMCHIST table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>

**BMCHIST table considerations for COPY PLUS**

COPY PLUS uses the BMCHIST table to record completed COPY and COPY IMAGECOPY command executions.

HISTRETN is available as a COPY PLUS installation option or as an option on the OPTIONS command. HISTRETN tells COPY PLUS the number of days to keep entries in the BMCHIST table.

**WARNING**

If you want to use BMCHIST, allocate adequate space for the table. COPY PLUS makes an entry in the table for every copied space. If you are copying a large number of partitions, you might need to increase the size of the BMCHIST table space from the standard size that was allocated during installation.

**BMCHIST table considerations for RECOVER PLUS**

For each execution of AFRMAIN, RECOVER PLUS writes a single row to the BMCHIST table.

DBNAME, SPNAME, and OBJNAME columns will always be blank.

RECOVER PLUS accumulates elapsed time for each of the following phases using the RECOVER PLUS phase shown:

- PHASE_1: LOGSORT
- PHASE_2: MERGE (includes RESTORE phase)
- PHASE_3: SNAP
PHASE_4: REBUILD (includes UNLOAD phase)

PHASE_5: DB2UTIL (the time spent in DSNUTILB)

The elapsed time for each of the phases is a sum for all objects. The utility elapsed time, ELAPSED, is the duration from the start of the utility to until it finishes. Because RECOVER PLUS multitasks, the sum of the phases might be greater than the total elapsed time of the utility. The elapsed time columns have a limit of 24 hours.

**Maintaining the BMCHIST table**

When a utility completes successfully, it inserts a row into the BMCHIST table. You can control expansion of this table by deleting old rows. If you use REORG PLUS, you can also control inserts into the BMCHIST table.

**To delete old rows from the BMCHIST table**

1. To delete selected rows from the BMCHIST table based on the date that the utility completed, use the following sample SQL statement:

   ```sql
   DELETE
   FROM creatorName.CMN_BMCHIST
   WHERE DATE < 'yyyy-mm-dd';
   ```

**To control inserts into the BMCHIST table (REORG PLUS only)**

1. Use the TERMEXIT option to specify a user exit that controls inserts into the BMCHIST table.

**BMCLGRNX table**

The BMCLGRNX table contains log ranges that show when a table space was open for updates.

Table 124 on page 725 describes the contents of the BMCLGRNX table.

**Table 124: Contents of the BMCLGRNX table**

<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 (mmddyy)</td>
</tr>
</tbody>
</table>
BMCSYNC table

The BMCSYNC table contains information about the status of the objects that the currently executing utilities are accessing.

Table 125 on page 726 describes the contents of the BMCSYNC table. 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 125: Contents of the BMCSYNC table

<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 (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>
</tbody>
</table>

Note

RECOVERY MANAGER uses the BMCLGRNXX table only for DB2 Versions 9 and 10. RECOVERY MANAGER uses the SYSIBM.SYSLGRNXX table for DB2 versions greater than Version 10.
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| NAME1      | CHAR(8)    | Database name or creator name  
(DASD MANAGER PLUS) This value is the database name.  
(CHECK PLUS, LOADPLUS, REORG PLUS, and UNLOAD 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      | CHAR(18)   | Space, table, or index name  
(DASD MANAGER PLUS) The BMCSTATS utility always inserts the space name (limited to a maximum of 8 characters).  
(CHECK PLUS, LOADPLUS, REORG PLUS, and UNLOAD PLUS) If the value for NAME1 would exceed 8 bytes or the value for NAME2 would exceed 18 bytes, NAME2 contains the table OBID or index ISOBID of the object in hexadecimal format. |
| KIND       | CHAR(2)    | Type of object:  
■ IP (index partition)  
■ IX (index)  
■ TB (table)  
■ TP (table space partition)  
■ TS (table space)  
■ DD, DW, D1, D2 (dynamic work file allocation)  
■ CI (copy information)  
■ RD (restart data set block) |
| PARTITION  | SMALLINT   | Physical partition number:  
■ Null or 0 for a single data set nonpartitioned space  
■ Data set number for a multi-data-set, nonpartitioned space  
■ Partition number for a partitioned space  
(CHECK PLUS, COPY PLUS, DASD MANAGER PLUS, LOADPLUS, REORG PLUS, and UNLOAD PLUS) The value is null or 0 for any nonpartitioned space. |
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCID</td>
<td>SMALLINT</td>
<td>Internal identifier of the object. DASD MANAGER PLUS does not use this column.</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>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 “Shared access levels of BMC utilities” on page 67.</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>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</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>(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(10)</td>
<td>(RECOVER PLUS) Log point at which this space was externalized</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVER PLUS serialization logic uses this column. The other utilities do not use this column.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> RECOVER PLUS no longer uses EXTRBA.</td>
</tr>
<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) Instance number of the current base 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>

**BMCSYNC table considerations**

This topic contains important information that you need to know about the BMCSYNC table:

- By default, DASD MANAGER PLUS uses the BMCSYNC table to synchronize access to DB2 spaces. However, if you want to turn this feature off, you may do so by specifying No for the BMCSYNC installation option. If you specify No for this option, DASD MANAGER PLUS does not use the BMCSYNC table and the product bypasses BMCUTIL table access, UTILID enqueue logic, and object name enqueue logic used for BMC utility concurrency control. Turning this feature off can lead to VSAM data set access failures in BMCSTATS or other utilities due to utility conflicts that are no longer detected.
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 a partition-by-growth table space.
  Estimate this allocation based on the following factors:
  — Number of utilities that you are running concurrently
  — Value of MAXPARTITIONS
  — 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

— 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
Maintaining the BMCSYNC table

When a utility abends, rows might remain in the BMCSYNC table. On rare occasions, you might need to take action to control expansion of the BMCSYNC table.

To control expansion of the BMCSYNC table

1. Use one of the following methods to delete rows in the BMCSYNC table:

   - Use the TERM restart parameter on the EXEC statement to delete rows from both the BMCUTIL and BMCSYNC tables. Do not delete any rows for instances of utilities that are awaiting restart.

   - Delete invalid rows from 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:

   ```sql
   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.

Cleaning up RECOVER UNLOADKEYS entries

Successful completion of a RECOVER UNLOADKEYS job leaves rows in BMCSYNC with blank utility IDs for table space partitions and indexes related to the unloaded keys. The table space rows prevent other BMC utilities from obtaining exclusive control of the table space.

To clean up RECOVER UNLOADKEYS entries

1. Use one of the following methods to remove the invalid BMCSYNC rows:

   - Run a RECOVER BUILDINDEX job.

   - Run a job that uses the following statement for the table space and each index:

     ```sql
     DELETE FROM creatorName.CMN_BMCSYNC
     WHERE UTILID=''
     AND NAME1='databaseName'
     AND NAME2='spaceName'
     AND UTILNAME='RECOVER';
     ```
Shared access levels of BMC utilities

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.

**Note**

All BMC 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 736.

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 the following table. For additional serialization and concurrency issues for each utility, see that utility’s reference manual.

The "Access level" column in the following table refers to the value of the SHRLEVEL column name in the BMCSYNC table (“BMCSYNC table” on page 726). 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 126: Shared access levels of BMC utilities**

<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>Product</td>
<td>Access level</td>
<td>Additional information</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| DASD MANAGER PLUS (BMCSTATS)    | S            | ■ 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. |
| LOADPLUS                        | X            | 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. |
| 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.

---

**BMCTRANS table**

The BMCTRANS table contains information that RECOVERY MANAGER and Log Master use for transaction recovery.

Table 127 on page 734 describes the contents of the BMCTRANS table. The table contains one row for each execution of Log Master (that is, one row for each log scan performed).

**Table 127: Contents of the BMCTRANS table**

<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>
</tbody>
</table>
| STATE       | SMALLINT NOT NULL | Level of recovery analysis performed:
<p>| | 0 (only UNDO analysis has been performed) |
| | 1 through 9999 (UNDO and PIT analysis have been performed) |
| | Greater than 10000 (UNDO, PIT, and REDO analysis have been performed) |
| PITTIME     | TIMESTAMP NOT NULL WITH DEFAULT | Timestamp for the PIT RBA |</p>
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
<tr>
<td>Column Name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REDOJOBSTATUS</td>
<td>SMALLINT</td>
<td>Code indicating the status of a REDO 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>
<tr>
<td>ENDTIME</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Transaction end time</td>
</tr>
<tr>
<td>ACTION</td>
<td>SMALLINT</td>
<td>Code indicating what recovery, if any, has been performed on the transaction</td>
</tr>
</tbody>
</table>
Table 128: Contents of the BMCUTIL table

<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>
</tbody>
</table>
| STATUS      | CHAR(1)   | Execution status of the utility:  
|             |           | - A (active, not executing command)  
|             |           | - I (initializing)  
|             |           | - P (pausing or pause-stopped)  
|             |           | - S (stopped)  
|             |           | - T (terminating)  
|             |           | - X (executing command)  
|             |           | *(DASD MANAGER PLUS)* The value for this column is always X. |
| UTILNAME    | CHAR(8)   | Name of the executing utility:  
|             |           | - CHECK  
|             |           | - COPY  
|             |           | - STATS  
|             |           | - LOAD  
|             |           | - RECOVER  
|             |           | - REORG  
|             |           | - UNLOAD |
| PHASE       | CHAR(8)   | Current phase of the utility  
<p>|             |           | COPY PLUS does not use this column. |
| USERID      | CHAR(8)   | User ID executing the utility |
| SSID        | CHAR(4)   | DB2 subsystem where the utility is running |</p>
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| RESTART        | CHAR(1)        | Restart option:  
|                |                | ■ N (not restart)  
|                |                | ■ P (RESTART(PHASE))  
|                |                | ■ Y (RESTART)  
|                |                | DASD MANAGER PLUS does not use this column.                                                                                                                                                           |
| NOTEID         | CHAR(8)        | TSO user ID to be notified  
|                |                | DASD MANAGER PLUS does not use this column.                                                                                                                                                           |
| DBNAME         | CHAR(8)        | (RECOVER PLUS and REORG PLUS) Name of the database containing the table or index space for which the last checkpoint was taken  
|                |                | This value can be blank.  
|                |                | The other utilities do not use this column.                                                                                                                                                           |
| SPNAME         | CHAR(8)        | (RECOVER PLUS and REORG PLUS) Name of the table or index space for which the last checkpoint was taken  
|                |                | This value can be blank.  
|                |                | The other utilities do not use this column.                                                                                                                                                           |
| SPSTATUS       | CHAR(5)        | (REORG PLUS) Space status before the utility stopped  
|                |                | The other utilities do not use this column.                                                                                                                                                           |
| COMMANDNO      | SMALLINT       | Not used (always 0)                                                                                                                                                                                   |
| COMMAND        | VARCHAR(256)   | First 256 characters of the utility command text  
|                |                | RECOVER PLUS, DASD MANAGER PLUS, and COPY PLUS do not use this column.                                                                                                                                  |
| STATE          | LONG VARCHAR   | Utility state and sync information  
|                |                | DASD MANAGER PLUS does not use this column.                                                                                                                                                           |
| START_TIMESTAMP| TIMESTAMP      | Starting timestamp of the utility                                                                                                                                                                     |
Maintaining the BMCUTIL table

When a utility abends, rows might remain in the BMCUTIL table.

On rare occasions, you might need to take action to control expansion of the BMCUTIL table.

To control expansion of the BMCUTIL table

1. Use one of the following methods to delete rows from the BMCUTIL table:
   - Use the TERM restart parameter on the EXEC statement to delete rows from both 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.

BMCXCOPY table

The BMC utilities use the BMCXCOPY table to track registered copies.

Table 129 on page 740 describes the contents of the BMCXCOPY table, which contains information about 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 that are not registered as Flash Copies in SYSCOPY with the BMC EXTENDED BUFFER MANAGER (XBM) product or BMC 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 129: Contents of the BMCXCOPY 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</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>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------</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, 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>
<tr>
<td></td>
<td></td>
<td>- m (indicates that the table space was exported by the COPY PLUS EXPORT command or migrated by the RECOVER PLUS IMPORT command)</td>
</tr>
<tr>
<td>ICDATE</td>
<td>CHAR(6)</td>
<td>Date of the entry (ymmd)</td>
</tr>
<tr>
<td>START_RBA</td>
<td>VARCHAR(10)</td>
<td>The relative byte location of a point in the DB2 recovery log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The indicated point as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- For ICTYPE F, the starting point for all updates since the image copy was taken</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- For COPY_TYPE O, the minimum of the consistent point and the oldest inflight URID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <em>(RECOVERY MANAGER)</em> For ICTYPE C, the consistent log point for the copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— RBA for non-data-sharing systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— LRSN for data sharing systems</td>
</tr>
<tr>
<td>FILESEQNO</td>
<td>INTEGER</td>
<td>Tape file sequence number of the copy</td>
</tr>
<tr>
<td>DEVTYPE</td>
<td>CHAR(8)</td>
<td>Type of device on which the copy resides</td>
</tr>
</tbody>
</table>

BMCXCOPY table
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBMREQD</td>
<td>CHAR(1)</td>
<td>Whether the row came from the basic machine-readable material (MRM) tape:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (YES)</td>
</tr>
<tr>
<td>DSNAMEN</td>
<td>CHAR(44)</td>
<td>Name of the data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If STYPE V, DSNAMEN is the name of the VSAM data component.</td>
</tr>
<tr>
<td>ICTYPEF</td>
<td>CHAR(6)</td>
<td>Time at which this row was inserted (hhmmss)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The insertion takes place after the completion of the operation that the row represents.</td>
</tr>
<tr>
<td>ICTYPEF</td>
<td>CHAR(1)</td>
<td>SHRLEVEL parameter on COPY if ICTYPE F:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ C (change)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ R (reference)</td>
</tr>
<tr>
<td>DSVOLSER</td>
<td>VARCHAR(1784)</td>
<td>Volume serial numbers of the data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Commas separate items in a list of 6-byte numbers. This column is blank if the data set is cataloged.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>Date and time when the row was inserted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This column contains 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.</td>
</tr>
<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>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</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>COPYPAGESF</td>
<td>FLOAT(53)</td>
<td>Number of pages written to the copy data set</td>
</tr>
<tr>
<td>NPAGE$F$</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>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</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. 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>
<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></td>
<td></td>
<td>RBA/LRSN format for the space or partition:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ B (basic 6-byte format)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ E (extended 10-byte 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>■ X (export copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ I (import copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (default value)</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>NOTE_VALUE</td>
<td>CHAR(4)</td>
<td>Encoded value that quickly locates data for a specific space in a cabinet copy. 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. 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>
<tr>
<td>EXPSSID</td>
<td>VARCHAR(8)</td>
<td>Source location SSID of the migration file (valid with COPY_TYPE = I)</td>
</tr>
<tr>
<td>EXPSLRSN</td>
<td>VARCHAR(10)</td>
<td>Indicates the SYNC AUTO point on the source (valid with COPY_TYPE = I and COPY_TYPE = X)</td>
</tr>
<tr>
<td>EXPTLRSN</td>
<td>VARCHAR(10)</td>
<td>Indicates the SYNC AUTO point on the target (valid with COPY_TYPE = I)</td>
</tr>
</tbody>
</table>
Maintaining the BMCXCOPY table

Periodically, you should review BMCXCOPY and delete old rows to control its expansion.

To control expansion of the BMCXCOPY table

1. To delete all rows from the BMCXCOPY table that are older than 30 days, run an SQL DELETE statement, using the following statement as an example:

   ```sql
   DELETE
   FROM creatorName.CMN_BMCXCOPY
   WHERE DAYS(CURRENT TIMESTAMP) - DAYS(TIMESTAMP) > 30;
   ```
REORG PLUS user exits

This chapter describes user-written exit routines that you can use to customize your reorganization processing.

Overview of REORG PLUS user exits

The table in this topic lists the following information about user-written exit routines that you can use with REORG PLUS:

- 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 130: User exits that REORG PLUS provides

<table>
<thead>
<tr>
<th>Exit routine purpose</th>
<th>Command option</th>
<th>Language</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating data set name patterns</td>
<td>DSNUEXIT</td>
<td>■ Assembler</td>
<td>“Using DSNUEXIT to construct data set name patterns in REORG PLUS” on page 748</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ COBOL II</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ LE COBOL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ C</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ LE C</td>
<td></td>
</tr>
</tbody>
</table>
### Exit routine purpose

<table>
<thead>
<tr>
<th>Exit routine purpose</th>
<th>Command option</th>
<th>Language</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing the redefinition of DB2 VSAM data sets</td>
<td>DSRSEXIT</td>
<td>REXX</td>
<td>&quot;Using a DSRSEXIT, TERMEXIT, or MAPTEXIT user exit&quot; on page 784, &quot;Using DSRSEXIT to manage VSAM data set redefinition&quot; on page 787</td>
</tr>
<tr>
<td>Controlling updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS updates</td>
<td>TERMEXIT</td>
<td>REXX</td>
<td>&quot;Using a DSRSEXIT, TERMEXIT, or MAPTEXIT user exit&quot; on page 784, &quot;Using TERMEXIT to control BMCHIST and statistics updates&quot; on page 801</td>
</tr>
<tr>
<td>Creating mapping objects for use with DSNUTILB</td>
<td>MAPTEXIT</td>
<td>REXX</td>
<td>&quot;Using a DSRSEXIT, TERMEXIT, or MAPTEXIT user exit&quot; on page 784, &quot;Using MAPTEXIT to create mapping objects&quot; on page 808</td>
</tr>
</tbody>
</table>

### Accessing the REORG PLUS sample user exits

Source code for the sample user exits is distributed in the REORG PLUS HLQ. LLQSAMP library (where HLQ is the high-level qualifier specified during installation and LLQ is the low-level qualifier or prefix set during installation).

The following table lists the library members that contain the sample user exits.

**Table 131: Library member names of user exits**

<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>DSRSEXIT</td>
<td>REXX</td>
<td>DSRSREXX</td>
</tr>
<tr>
<td>MAPTEXIT</td>
<td>REXX</td>
<td>MAPTREXX</td>
</tr>
<tr>
<td>TERMEXIT</td>
<td>REXX</td>
<td>TERMREXX</td>
</tr>
</tbody>
</table>

### Using DSNUEXIT to construct data set name patterns in REORG PLUS

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 SPI LDSNP installation or SPI LLD S N P A T command option.

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 for REORG PLUS

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.

- 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.

- 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 for REORG PLUS**

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.

The following table lists where you can find the return code.

**Table 132: DSNUEXIT return code location**

<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 or LE_COBOL</td>
<td>Contents of the RETURN-CODE variable</td>
</tr>
<tr>
<td>C or LE_C</td>
<td>Returned by the return function</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 for REORG PLUS**

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 SPILDSNP installation option or SPILDSNPAT 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 for REORG PLUS**

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 (shown in the following figure) 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 65: DSNUEXIT assembler exit parameter block**

```
* 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 , PAMRS PASSED TO EXIT
* INPUT AREA
* XPJOBN DS CL8 JOBNAME
XPSTEP DS CL8 STEPNAME
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
XPDATE DS CL6 UTILITY EXECUTION DATE MMDDYY
XPTIME DS CL6 UTILITY EXECUTION TIME HHMMSS
XPUTILID DS CL16 UTILITY ID
XPDATE8 DS CL8 UTILITY EXECUTION DATE MMDDYYYY
XPGPNM DS CL4 DATA SHARING GROUP NAME
XPVCAT DS CL8 VCAT NAME (FROM 1ST PART IF PARTITNED)
XPDATEJ DS CL7 UTILITY EXECUTION DATE CCYYDDD
   DS CL13 RESERVED FOR REORG PLUS
* USER WORK AREA
* XPUSRWD1 DS F USER WORD 1
XPUSRWD2 DS F USER WORD 2
XPUSRWD3 DS F USER WORD 3
XPUSRWD4 DS F USER WORD 4
XPUSRWD5 DS F USER WORD 5
XPUSRWD6 DS F USER WORD 6
XPUSRWD7 DS F USER WORD 7
XPUSRWD8 DS F USER WORD 8
* OUTPUT AREA
* XPUVAREA DS 100CL17 USER VARIABLE AREA
XPUVENT# EQU 100 NUMBER OF ENTRIES IN OUTPUT AREA
XP$ EQU *-ARUDSNXP
```

**DSECT fields**

The following table describes the major DSECT fields for the DSNUEXIT assembler user exit parameter block and their uses.
### Table 133: Major DSECT fields for the DSNUEXIT assembler user exit parameter block

<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>XPRTYPE</td>
<td>Type of reorganization being performed, table space (TS) or index (IX)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>XPUSER</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>XPDATE</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>XPDATE8</td>
<td>Date of the execution of the utility, in the format MMDDYYYY</td>
<td>8 bytes</td>
</tr>
<tr>
<td>XPGRPNM</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>XPVCAT</td>
<td>VCATNAME specified in the DB2 catalog for the table space being reorganized,</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>or for the first partition if the table space is partitioned</td>
<td></td>
</tr>
<tr>
<td>XPDATEJ</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>Work space</td>
<td>Up to 4 bytes each</td>
</tr>
<tr>
<td>XPUVAREA</td>
<td>Area containing user-defined variable information For information about</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>establishing user-defined variables, see “DSNUEXIT user-defined variables</td>
<td></td>
</tr>
<tr>
<td></td>
<td>for REORG PLUS” on page 750.</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>

**Variable mapping block DSECT**

The variable mapping block DSECT, as shown in the following figure, 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.

**Figure 66: DSNUEXIT assembler variable mapping block**

* *---------------------------------------------------------------------*
* 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 *
* 3. MAY NOT CONTAIN EMBEDDED BLANKS                            *
* 4. NON-BLANK CHARACTERS MUST BE 'NATIONAL' CHARACTERS WHICH ARE *
*     A THROUGH Z, 0 THROUGH 9, #, @, $, -, AND .                *
* *                                                             *
* *---------------------------------------------------------------------*

**DSECT fields**

The following table describes the major DSECT fields for the DSNUEXIT assembler variable mapping block and their uses.

**Table 134: 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

The following figure shows an example of this user exit written in assembler.

Figure 67: DSNUEXIT sample assembler user exit

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$OPTS DSNUEXIT PARAMETER OR IN THE DSNUEXIT PARAMETER OF
THE REORG COMMAND. THE SPECIFICATION IN THE REORG COMMAND WILL
OVERRISE THE SPECIFICATION IN THE ARU$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

*---------------------------------------------------------------------*

*                           N O T E S                              *
*---------------------------------------------------------------------*

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-
ICATIONS 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 -                                                                *
* 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                        E
R7       EQU   7                         R
R8       EQU   8
R9       EQU   9                           E
R10      EQU   10                           Q
R11      EQU   11                            U
R12      EQU   12                             A
R13      EQU   13                              T
R14      EQU   14                               E
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                STEPNAME
XPDBNAME DS    CL8                DATABASE NAME
XPSNAME  DS    CL8                SPACE NAME
XPRTYPE  DS    CL2                REORG TYPE (TS OR IX)
XPUSER   DS    CL8                USER ID
XPSSID   DS    CL4                DB2 SUBSYSTEM ID
XPDATE   DS    CL6                UTILITY EXECUTION DATE MMDDYY
Using DSNUEXIT to construct data set name patterns in REORG PLUS

XPTIME   DS    CL6                UTILITY EXECUTION TIME HHMMS
XPUTILID DS    CL16               UTILITY ID
XPDATE8  DS    CL8                UTILITY EXECUTION DATE MMDDYYYY
XPGRPNM DS    CL4                DATA SHARING GROUP NAME
XPVCAT   DS    CL8                VCAT NAME (FROM 1ST PART IF PARTITIONED)
XPDATEEJ DS    CL7                UTILITY EXECUTION DATE CCYDDD
       DS    CL13               RESERVED FOR REORG PLUS

* USER WORK AREA

  * XPUSRWD1 DS    F                  USER WORD 1
  XPUSRWD2 DS    F                  USER WORD 2
  XPUSRWD3 DS    F                  USER WORD 3
  XPUSRWD4 DS    F                  USER WORD 4
  XPUSRWD5 DS    F                  USER WORD 5
  XPUSRWD6 DS    F                  USER WORD 6
  XPUSRWD7 DS    F                  USER WORD 7
  XPUSRWD8 DS    F                  USER WORD 8

  * OUTPUT AREA

  * XPUVAREA DS 100CL17            USER VARIABLE AREA
  XPUVENT# EQU 100                NUMBER OF ENTRIES IN OUTPUT AREA
  XPS$   EQU *-ARUDSNXP

*---------------------------------------------------------------------*
*                                                                     *
* 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 *
*   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 ,                  MAP A VARIABLE ENTRY
XPUVNAME DS    CL9                VARIABLE NAME
XPUVDATA DS    CL8                VARIABLE DATA
XPUVENT$ EQU *-XPUVNAME         LENGTH OF EACH ENTRY
EJECT

*---------------------------------------------------------------------*
*                                                                     *
* LOCAL PROGRAM WORK AREA DSECT POINTED TO BY R11                    *
*                                                                     *
*---------------------------------------------------------------------*
WORKAREA DSECT
SAVE DS I8F LOCAL SAVE AREA
DWORK DS D DOUBLEWORD WORK AREA
CWORK DS CL8 CHARACTER WORK AREA
WRKAREA$ EQU */-WORKAREA
EJECT

* *---------------------------------------------------------------------*  
* PROGRAM START                                                   *  
* *---------------------------------------------------------------------*  

ARUEDSNA CSECT

* >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
* UNCOMMENT THE NEXT TWO INSTRUCTIONS TO DISABLE THIS EXIT.          *
* >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
*                                                                 *
* LA R15,0 RC=0, GOOD RETURN                                         *
* BSM 0,R14 JUST RETURN TO CALLER                                    *
* >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
* UNCOMMENT THE PREVIOUS TWO INSTRUCTIONS TO DISABLE THIS EXIT.       *
* >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
*                                                                 *
* STM R14,R12,12(R13) SAVE CALLERS REGS                              *
* LR R12,R15 R12 IS MY BASEREG                                       *
* USING ARUEDSNA,R12 ESTABLISH ADDRESSABILITY                        *
*                                                                 *
* 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=BELLOW GET WORK AREA                   *
* LTR R15,R15 OK?                                                    *
* BNZ DSNXRC08 ..NO, ERROR                                           *
* LR R11,R1 R11 = A(USER WORK AREA)                                  *
* ST R11,XPUSRD1 SAVE A(USER WORK AREA)                              *
* USING WORKAREA,R11 ADDRESS WORKAREA DSECT                          *
* LA R15,SAVE MY SAVE AREA@                                          *
* ST R15,8(.R13) SAVE IN CALLERS SAVE AREA                           *
* 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                                     *
* *---------------------------------------------------------------------*  

Using DSNUEXIT to construct data set name patterns in REORG PLUS
* _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
MVC  XPUVNAME,=CL9'_JDATE'   SET VAR NAME
LA    R2,MONTHTAB            @(MONTH TABLE - NO LEAP YEAR)
PACK  DWORK(8),XPDATE8+4(4)  GET YEARS
CVB   R4,DWORK               INTO R4
LR    R7,R4                  SAVE A COPY IN R7
SRDL  R4,32                  PREPARE FOR DIVIDE
D     R4,=F'4'               DIVIDE BY 4
LTR   R4,R4                  LEAP YEAR?
BNZ   DSNX0110                NO ->
LR    R4,R7                  GET YEAR AGAIN
SRDL  R4,32                  PREPARE FOR DIVIDE
D     R4,=F'100'             DIVIDE BY 100
LTR   R4,R4                  LEAP YEAR?
BNZ   DSNX0100                YES ->
LR    R4,R7                  GET YEAR AGAIN
SRDL  R4,32                  PREPARE FOR DIVIDE
D     R4,=F'400'             DIVIDE BY 400
LTR   R4,R4                  LEAP YEAR?
BNZ   DSNX0110                NO ->

DSNX0100 DS    0H

LA    R2,LEAPTAB             @(MONTH TABLE - LEAP YEAR)

DSNX0110 DS    0H

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    R6,0(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),PL4PAT  GET EDIT PATTERN
ED    CWORK(L'PL4PAT),DWORK+4 EDIT YYDDD
MVI  XPUVDATA,C'D'           SET VAR DATA
MVC  XPUVDATA+1(5),CWORK+3   SET VAR DATA
LA    R3,XPUVENT$(,R3)       @(NEXT ENTRY)
MVC  XPUVNAME(XPUVENT$),BLANKS CLEAR ENTRY
MVC  XPUVNAME,=CL9'_JCDATE'  SET VAR NAME

Appendix C  REORG PLUS user exits 759
CVD R5, DWORK MAKE IT PACKED
MVC CWORK(L'PL4PAT), PL4PAT GET EDIT PATTERN
ED CWORK(L'PL4PAT), DWORK+4 EDIT YYYYDDD
MVI XPUVDATA, C'D' SET VAR DATA
MVC XPUVDATA+1(7), CWORK+1 SET VAR DATA

*---------------------------------------------------------------------*
*                                                                     *
*  BUILD _UTILPFX AND _UTILSFX USER VARIABLES                         *
*                                                                     *
*---------------------------------------------------------------------*
*
LA R3, XPUVENT$(, R3) @(NEXT ENTRY)
MVC XPUVNAME(XPUVENT$), BLANKS CLEAR ENTRY

MVC XPUVNAME, =CL9' _UTILPFX' SET VAR NAME
TRT XPUTILID, TRTAB LOOK FOR DELIMITER
BZ DSNX0200 NONE ->

LA R2, XPUTILID @(UTIL ID)
LR R4, R1 SAVE @(DELIMITER)
SR R1, R2 LENGTH OF FIRST PART
BZ DSNX0200 NONE ->

C R1, =F'8' MORE THAN 8 BYTES LONG
BNH DSNX0130 NO ->
LA R1, 8 MAKE IT 8 BYTES

DSNX0130 DS 0H
*
BCTR R1.0 SUBTRACT 1 FOR EXECUTED MVC
LA R5, XPUTILID @(UTILID)
EX R1, DSNXMCU MOVE UTILID PREFIX

LA R3, XPUVENT$(, R3) @(NEXT ENTRY)
MVC XPUVNAME(XPUVENT$), BLANKS CLEAR ENTRY

MVC XPUVNAME, =CL9' _UTILSFX' SET VAR NAME
LA R5, 1(, R4) @1ST CHAR PAST DELIMITER
LA R1, XPUTILID+L'XPUTILID @(END OF UTILID)
SR R1, R5 LENGTH OF SECOND PART OF UTILID
BZ DSNXRCCO0 NONE ->

C R1, =F'8' MORE THAN 8 BYTES LONG
BNH DSNX0140 NO ->
LA R1, 8 MAKE IT 8 BYTES

DSNX0140 DS 0H
*
BCTR R1.0 SUBTRACT 1 FOR EXECUTED MVC
EX R1, DSNXMCU MOVE UTILID PREFIX

B DSNXRCCO0 RETURN RC=0

DSNX0200 DS 0H
*
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+8, XPUTILID+8 GET NEXT 8 BYTES

B DSNXRCCO0 RETURN RC=0
DSNXMVCU MVC XPUDATA(0),0(R5) *** EXECUTED INSTRUCTION ***

DROP R3

EJECT

*---------------------------------------------------------------------*
*                                                                     *
* SET RETURN CODE AND EXIT                                           *
*                                                                     *
*---------------------------------------------------------------------*

DSNXRC00 DS 0H *

LA R3,0       SAVE RETURN CODE
B DSNX9000    GOOD RETURN

DSNXRC08 DS 0H *

LA R3,8       SAVE RETURN CODE
TERMINATE REORG+ EXECUTION

DSNX9000 DS 0H *

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 0H *

LR R15,R3     RESTORE RETURN CODE
L R14,12(,R13)
LM R0,R12,20(R13)
BSM 0,R14 RETURN

EJECT

*---------------------------------------------------------------------*
*                                                                     *
*                         C O N S T A N T S                               *
*                                                                     *
*---------------------------------------------------------------------*

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRTAB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X'00000000000000000000000000000000' 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X'00000000000000000000000000000000' 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X'00000000000000000000000000000000' 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>X'00000000000000000000000000000000' 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>X'FF000000000000000000000000000000' 4 (SP) . +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| DC    | X'00000000000000000000000000000000' 5 :
| DC    | X'FFFF0000000000000000000000000000' 6 - / (X'6A') . _
| 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 \ |
| DC    | X'00000000000000000000000000000000' F |

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>MONTHTAB DC H'0'                JANUARY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>H'31'                FEBRUARY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>H'59'                MARCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DSNUEXIT COBOL II and LE COBOL user exit for REORG PLUS

This topic describes 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 (as shown in the following figure) 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 68: DSNUEXIT COBOL II and LE COBOL exit parameter record*

```cobol
*-----------------------------------------------------------------  
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).  
      10  EXIT-TSNAME     PIC X(8).  
      10  EXIT-REORG-TYPE PIC X(2).  
      10  EXIT-FILLER1    PIC X(2).  
```
Parameter record fields

The following table describes the major fields for the DSNUEXIT COBOL II and LE COBOL user exit parameter record and their uses.

### Table 135: Major parameter record fields for the DSNUEXIT COBOL II and LE COBOL user exit

<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>
<tr>
<td>Field</td>
<td>Description</td>
<td>Length</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<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...WORK-AREA-8</td>
<td>8 parameters for work space</td>
<td>Up to 4 bytes each</td>
</tr>
</tbody>
</table>

**COBOL II and LE COBOL variable mapping record**

The COBOL II and LE COBOL variable mapping record (as shown in the following figure) 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.

**Figure 69: DSNUEXIT COBOL II and LE COBOL variable mapping record**

```cobol
05 USER-DEFINED-VARIABLE-TABLE OCCURS 100 TIMES.
10 VARIABLE-NAME            PIC X(9).
10 VARIABLE-VALUE           PIC X(8).
10 FILLER REDEFINES VARIABLE-VALUE.
15 VARIABLE-PREFIX          PIC X(1).
15 VARIABLE-JUL-DATE        PIC X(7).  
```

**Variable mapping record fields**

The following table describes the major variable mapping record fields for the DSNUEXIT COBOL II and LE COBOL user exit and their uses.
Sample COBOL II and LE COBOL user exit

The following figure shows an example of this user exit written in COBOL.

Figure 70: DSNUEXIT sample COBOL II and LE COBOL user exit

```
* THIS IS A SAMPLE REORG PLUS USER EXIT. THIS EXIT WOULD BE USED
* TO DEFINE USER VARIABLES FOR BUILDING DATA SET NAMES
* FOR DYNAMIC WORKFILE ALLOCATION WITH THE DSNPAT KEYWORD AND FOR
* SHRLEVEL CHANGE WITH THE SPILLDNAT 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 ARU$OPTS DURING INSTALLATION OR IN THE
* DSNUEXIT PARAMETER OF THE REORG COMMAND. THE REORG COMMAND
* OPTION WILL OVERRIDE THE PARM IN THE ARU$OPTS MACRO.

* ENVIRONMENT DIVISION.
* INPUT-OUTPUT SECTION.
* FILE-Control.
* DATA DIVISION.
* FILE SECTION.
* EJECT
* WORKING-STORAGE SECTION.

01 FILLER PIC X(16) VALUE 'WORKING STORAGE'.
```
**MISCELLANEOUS LITERALS, TABLE SUBSCRIPTS, NUMERIC VALUES**

**FOR JULIAN-DATE OR JULIAN/CENTURY-DATE CONVERSION, ETC.**

```
01 MISCELLANEOUS.
  05 SAVE-DATA PIC X(8) VALUE SPACES.
  05 DAYS-SUBX PIC S9(3) COMP VALUE ZERO.
  05 SUBX PIC S9(3) COMP VALUE ZERO.
  05 ONE PIC S9(1) COMP-3 VALUE +1.
  05 TWO PIC S9(1) COMP-3 VALUE +2.
  05 FOUR PIC S9(1) COMP-3 VALUE +4.
  05 MAX-LENGTH PIC S9(3) COMP-3 VALUE +16.
  05 NINETEEN PIC S9(3) COMP-3 VALUE +19.
  05 TWENTY PIC S9(3) COMP-3 VALUE +20.
  05 ONE-HUNDRED PIC S9(3) COMP-3 VALUE +100.
  05 FOUR-HUNDRED PIC S9(3) COMP-3 VALUE +400.
  05 JULIAN-DATE-DESC PIC X(9) VALUE '_JDATE'.
  05 JULIAN-CDATE-DESC PIC X(9) VALUE '_JCDATE'.
  05 UTILITY-PREFIX PIC X(9) VALUE '_UTILPFX'.
  05 UTILITY-SUFFIX PIC X(9) VALUE '_UTILSFX'.
  05 UTILID-PREFIX PIC X(8) VALUE SPACES.
  05 UTILID-SUFFIX PIC X(8) VALUE SPACES.
  05 UTILID-POINTER PIC S9(3) VALUE ZERO.
  05 UTILID-COUNTER PIC S9(3) VALUE ZERO.
  05 UTILID-TALLY PIC S9(3) VALUE ZERO.

01 DATE-WORK-AREA.
  05 CONVERTED-DATE.
    10 DATE-PREFIX PIC X(1) VALUE 'D'.
    10 JULIAN-CDATE PIC 9(7) VALUE ZERO.
    10 FILLER REDEFINES JULIAN-CDATE.
    15 JULIAN-CC PIC 9(2).
    15 JULIAN-DATE PIC 9(5).
    15 FILLER REDEFINES JULIAN-DATE.
    20 JULIAN-YY PIC 9(2).
    20 JULIAN-DAYS PIC 9(3).
  05 WORK-YEAR PIC S9(5) COMP-3 VALUE ZERO.
  05 YEAR-ANSWER PIC S9(3) COMP-3 VALUE ZERO.
  05 YEAR-REMAINDER PIC S9(9) COMP-3 VALUE ZERO.
  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.
```

Using DSNUEXIT to construct data set name patterns in REORG PLUS
05 LEAP-MARCH  PIC 9(3)  COMP-3  VALUE 60.
05 LEAP-APRIL   PIC 9(3)  COMP-3  VALUE 91.
05 LEAP-MAY     PIC 9(3)  COMP-3  VALUE 121.
05 LEAP-JUNE    PIC 9(3)  COMP-3  VALUE 152.
05 LEAP-JULY    PIC 9(3)  COMP-3  VALUE 182.
05 LEAP-AUGUST  PIC 9(3)  COMP-3  VALUE 213.
05 LEAP-SEPTEMBER PIC 9(3)  COMP-3  VALUE 244.
05 LEAP-OCTOBER PIC 9(3)  COMP-3  VALUE 274.
05 LEAP-NOVEMBER PIC 9(3)  COMP-3  VALUE 305.
05 LEAP-DECEMBER PIC 9(3)  COMP-3  VALUE 335.
01 FILLER REDEFINES LEAP-MONTHS.
05 LEAP-MONTH-DAYS PIC 9(3)  COMP-3  OCCURS 12 TIMES.

*-----------------------------------------------------------------
LINKAGE SECTION.
*-----------------------------------------------------------------

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).
  10 EXIT-TSNAME   PIC X(8).
  10 EXIT-REORG-TYPE PIC X(2).
  10 EXIT-FILLER1  PIC X(2).
  10 EXIT-USERID   PIC X(8).
  10 EXIT-DB2-SSID PIC X(4).
  10 EXIT-DAY.
    15 EXIT-MM PIC 9(2).
    15 EXIT-DD PIC 9(2).
    15 EXIT-YY PIC 9(2).
  10 EXIT-TIME   PIC X(6).
  10 EXIT-UTILID-PARM PIC X(16).
  10 FILLER REDEFINES EXIT-UTILID-PARM.
  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(2).
  10 EXIT-GRPNM   PIC X(4).
  10 EXIT-VCAT    PIC X(8).
 10 EXIT-DAYJ.
    15 EXIT-DAYJ-YEAR PIC 9(4).
  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).
05 USER-DEFINED-VARIABLE-TABLE OCCURS 100 TIMES.
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.

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 ','
   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
      WITH POINTER UTILID-POINTER
   END-IF.
PERFORM 1500-UTILID-PARMS.

1400-EDIT-UTILID-EXIT.
EXIT.

1500-UTILID-PARMS.
MOVE ONE TO SUBX.
MOVE UTILITY-PREFIX TO VARIABLE-NAME(SUBX).
MOVE UTILID-PREFIX TO VARIABLE-VALUE(SUBX).

ADD ONE TO SUBX.
MOVE UTILITY-SUFFIX TO VARIABLE-NAME(SUBX).
MOVE UTILID-SUFFIX TO VARIABLE-VALUE(SUBX).

ADD ONE TO SUBX.
MOVE JULIAN-DATE-DESC TO VARIABLE-NAME(SUBX).
MOVE DATE-PREFIX TO VARIABLE-PREFIX(SUBX).
MOVE JULIAN-DATE TO VARIABLE-JUL-DAT(SUBX).

ADD ONE TO SUBX.
MOVE JULIAN-CDATE-DESC TO VARIABLE-NAME(SUBX).
MOVE CONVERTED-DATE TO VARIABLE-VALUE(SUBX).

1500-UTILID-PARMS-EXIT.
EXIT.
DSNUEXIT C user exit for REORG PLUS

This topic describes 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 C.

C exit parameter structure

The C exit parameter structure (as shown in the following figure) 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 71: DSNUEXIT C exit parameter structure

```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 xuser[8];  /* user id */
  char xpssid[4];  /* db2 subsystem id */
  char xpdate[6];  /* utility execution date mmddyy */
  char xptime[6];  /* utility execution time hhmms */
  char xputilid[16];  /* utility id */
  char xpdate8[8];  /* utility execution date mmddyyyy */
  char xpgrpnm[4];  /* data sharing group name */
  char xpvcat[8];  /* vcat name (from 1st part if partitnd*/
  char xpdatej[7];  /* utility execution date ccyyddd */
  char xpresrv2[13]; /* reserved for REORG PLUS */
  int xpusrw1;  /* user word 1*/
  int xpusrw2;  /* user word 2*/
  int xpusrw3;  /* user word 3*/
  int xpusrw4;  /* user word 4*/
  int xpusrw5;  /* user word 5*/
  int xpusrw6;  /* user word 6*/
  int xpusrw7;  /* user word 7*/
  int xpusrw8;  /* user word 8*/
...
```

Structure fields

The following table describes the major structure fields for the DSNUEXIT C user exit and their uses.

Table 137: Major structure fields for the DSNUEXIT C exit parameter

<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>
</tbody>
</table>
### Field Description Length

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>xpdname</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>xptuser</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>xptdate</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>xptdate8</td>
<td>Date of the execution of the utility, in the format MMDDYYY</td>
<td>8 bytes</td>
</tr>
<tr>
<td>xpgpnm</td>
<td>DB2 data sharing group name</td>
<td>Not applicable</td>
</tr>
<tr>
<td>xpcat</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>xptdatej</td>
<td>Julian date of the execution of the utility, in the format CCYYDDD</td>
<td>7 bytes</td>
</tr>
<tr>
<td>xpswrd1...xpswrd8</td>
<td>8 parameters of work space</td>
<td>Up to 4 bytes each</td>
</tr>
</tbody>
</table>

### C exit variable mapping structure

The C exit variable mapping structure, as shown in the following figure, 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 72: DSNUEXIT C exit variable mapping structure**

```c
struct XPUVAREA {
    char xpuvname(|9|);
    char xpuvdata(|8|);
    } xpuvars(|XPUVENT|);  
```

**Structure fields**

The following table describes the major variable mapping structure fields for the DSNUEXIT C user exit and their uses.
Table 138: Major variable mapping structure fields for the DSNUEXIT COBOL II and LE COBOL 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. For information about establishing user-defined variables, see “DSNUEXIT user-defined variables for REORG PLUS” on page 750.</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 C user exit**

The following figure shows an example of this user exit written in C.

**Figure 73: DSNUEXIT sample C user exit**

```c
/* 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 SPILDSNPAT 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 */
/* MAY HAVE IN THIS AREA. */
/* */
/* PHONE: 1-800-537-1813 */
/* */
/* N OTES */
/* */
/* 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 ARU$SNXP STRUCT. */
/* THIS CALL IS FROM ASSEMBLER MODULE DYNWDSNX */
/* */
/* UPON EXIT RETRUN CODE FROM THIS EXIT */
```

772  REORG PLUS for DB2 Reference Manual
/*                      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 dynwdsnux(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 */
    };
}
Using DSNUEXIT to construct data set name patterns in 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 xptime[6]; /* utility execution time hhmmss */
    char xputilid[16]; /* utility id */
    char xptime8[8]; /* utility execution time mmddyy */
    char xpgpnm[4]; /* data sharing group name */
    char xpvcat[8]; /* vcat name (from 1st part if partitnd*/
    char xptimej[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 aruedsc (struct arudsnpx *xparm) {
    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;
    tp = xparm->xpuvars;
    x = l = wday = wyear = iyear = imonth = totdays = 0;
    xdisable = i = pfxlen = sfxstrt = sfxlen = 0;
    /* to disable this exit set xdisable = 1 */
}
```
Using DSNUEXIT to construct data set name patterns in REORG PLUS

/*--------------------------------------------------------------------*/
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
*/
/*
*/
/*--------------------------------------------------------------------*/
/*--------------------------------------------------------------------*/
/*
*/
/* BUILD _JDATE AND _JCDATE USER VARIABLES
*/
/*
*/
/*--------------------------------------------------------------------*/
memcpy(month,xparm->xpdate8,2);
memcpy(day,xparm->xpdate8+2,2);
memcpy(yr,xparm->xpdate8+4,4);
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);
/*--------------------------------------------------------------------*/

Appendix C REORG PLUS user exits 775


Using DSNUEXIT to construct data set name patterns in REORG PLUS

/* sprintf(tp->xpuvdata, "%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 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 '/':
        case 'X'6A':
        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;
}
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->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 */
/* */
DSNUEXIT LE C user exit for REORG PLUS

This topic describes 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 (as shown in the following figure) 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 74: DSNUEXIT LE C exit parameter structure

```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       xpdate[6];       /* utility execution date mmddyy   */
    char       xptime[6];       /* utility execution time hhmmss   */
    char       xputilid[16];    /* utility id                      */
    char       xpdate8[8];      /* utility execution date mmddyyyy */
    char       xpgprm[4];       /* data sharing group name         */
    char       xpvcat[8];  /* vcat name (from 1st part if partitnd) */
    char       xpdarvst[7];    /* utility execution date ccyyddd  */
    char       xpresrvv[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*/
    ...
```

Structure fields
The following table describes the major structure fields for the DSNUEXIT LE C user exit and their uses.

Table 139: Major structure fields for the DSNUEXIT LE C user exit parameter

<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>xprtype</td>
<td>Type of reorganization being performed, table space (TS) or index (IX)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>xpuser</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>xupdate</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>xupdate8</td>
<td>Date of the execution of the utility, in the format MMDDYYYY</td>
<td>8 bytes</td>
</tr>
<tr>
<td>xpgprpnm</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>xpvcat</td>
<td>VCATNAME specified in the DB2 catalog for the table space being reorganized,</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>or for the first partition if the table space is partitioned</td>
<td></td>
</tr>
<tr>
<td>xupdatej</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>

**LE C exit variable mapping structure**

The LE C exit variable mapping structure (as shown in the following figure) 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 75: DSNEXIT LE C exit variable mapping structure**

```c
struct XPUVAREA {
    char xpuvname[9];
    char xpuvdata[8];
} xpuvars[XPUVENT];
```

**Structure fields**

The following table describes the major variable mapping structure fields for the DSNEXIT LE C user exit and their uses.

**Table 140: Major variable mapping structure fields for the DSNEXIT 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 “DSNEXIT user-defined variables for REORG PLUS” on page 750.</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**

The following figure shows an example of this user exit written in LE C.

**Figure 76: DSNEXIT sample LE C user exit**

```c
/* DISCLAIME 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 SPILDSNPAT 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 */
```
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 ASSEMBLER 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 MODIFICATIONS 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 .
Using DSNUEXIT to construct data set name patterns in REORG PLUS

#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 xspname[8]; /* space name */
  char xprtype[2]; /* REORG type (ts or ix) */
  char xpresrv1[2]; /* reserved for REORG PLUS */
  char xuserid[8]; /* user id */
  char xpssid[4]; /* db2 subsystem id */
  char xptime[6]; /* utility execution time hhmms */
  char xpuvname[9]; /* utility execution date mmddyy */
  char xpuvdata[8]; /* utility execution time hhmms */
  char xpvcat[8]; /* vcat name (from 1st part if partition) */
  char xptimej[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 */
}

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 wyyear;
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 = 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
 * */
/*
 * BUILD _JDATE AND _JCDATE USER VARIABLES
 * */
memory(month,xparm->xpdate8,2);
memory(day,xparm->xpdate8+2,2);
memory(yr,xparm->xpdate8+4,4);
sscanf(yr,"%d",&wyyear);
sscanf(day,"%d",&wday);
sscanf(month,"%d",&imonth);
imonth = imonth - 1;
if (wyyear % 4 == 0 && wyyear % 100 != 0 || wyyear % 400 == 0)
todays = wday + leaptab[imonth];
else
    totdays = wday + monthtab[imonth];
sprintf(yr,"%d",wyyear);
sprintf(day,"%03.03d",totdays);
/*
 * clear the user variable area to blanks
 * */
memset(tp->xpuvname,' ', sizeof(struct XPUVAREA));
*/
/*
Using DSNUEXIT to construct data set name patterns in REORG PLUS
*/
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

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
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 '/':
        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;
}
Using a DSRSEXIT, TERMEXIT, or MAPTEXIT user exit

This section describes the REXX-based user exits that you can use with REORG PLUS.

You can use the following user exits written in REXX:

- **DSRSEXIT** to manage the redefinition of DB2 VSAM data sets (see “Using DSRSEXIT to manage VSAM data set redefinition” on page 787)
- **TERMEXIT** to control updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS (see “Using TERMEXIT to control BMCHIST and statistics updates” on page 801)
- **MAPTEXIT** to create mapping objects for use with DSNUTILB (see “Using MAPTEXEXIT to create mapping objects” on page 808)

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 141 on page 785 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.

The following table lists the variables that REORG PLUS passes to all REXX user exits. You cannot change the value of any of these variables.

<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>BMCStringUtil</td>
<td>REORG</td>
</tr>
<tr>
<td>BMC_REORG_TYPE</td>
<td>Reorganization type, either TABLESPACE or INDEX</td>
</tr>
<tr>
<td>Variable name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BMC.Utility_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 REORG PLUS truncates longer utility IDs to eight characters.</td>
</tr>
<tr>
<td>BMC_UtilPFX</td>
<td>BMC utility ID prefix</td>
</tr>
<tr>
<td>BMC_UtilSuffix</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_Db2_Ssid</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 SSSID.</td>
</tr>
<tr>
<td>BMC_Date</td>
<td>Current date (in the form MMDDYY)</td>
</tr>
<tr>
<td>BMC_DateJ</td>
<td>Current Julian date (in the form YYYYDDD)</td>
</tr>
<tr>
<td>BMC_Date8</td>
<td>Current date (in the form MMDDYYYY)</td>
</tr>
<tr>
<td>BMC_Time</td>
<td>Current time (in the form HHMMSS)</td>
</tr>
<tr>
<td>BMC_Time4</td>
<td>Current time (in the form HHMM)</td>
</tr>
</tbody>
</table>
Using DSRSEXIT to manage VSAM data set redefinition

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 787)
- Update the DB2 catalog with the changed values for use with subsequent allocations unless you indicate otherwise (see “Resizing DB2 objects” on page 787)
- Selectively tell REORG PLUS to specify REDEFINE NO for an object (see “Setting REDEFINE NO” on page 788)
- 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 789)
- 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 789)
- Add the DATACLAS, MGMTCLAS, and STORCLAS for storage groups (see “Using the DSRSEXIT user exit to add SMS classes” on page 789)

**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 enables 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 completing 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 795.

**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 796.

**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 796.

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 user exits, the following requirements and restrictions apply specifically 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 sets in REORG PLUS” on page 357.
- 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 “Setting REORG PLUS authorizations” on page 53.
- 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 198.

- 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 user exit

This topic describes how to run the DSRSEXIT.

**Before you begin**

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.

   - On the REORG command, specify the DSRSEXIT option with the name of your REXX user exit.

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 “Setting REORG PLUS authorizations” on page 53 and the OPNDB2ID=YES installation option.
DSRSEXIT variables

In addition to the variables common to all REXX exits, REORG PLUS passes other variables to the DSRSEXIT user exit.

Table 142 on page 791 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. This table also lists the following information about these variables:

- Variable name
- Brief description of the variable
- 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 795.

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 142: Variables that REORG PLUS passes to DSRSEXIT

<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>Variables that apply to all objects</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| BMC_HIGH_USED_RBA         | High-used relative byte address (HURBA)  
This 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. | No      |
| BMC_HIGH_ALLOC_RBA        | High-allocated relative byte address  
This 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. | No      |
<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_HIGH_USED_RBN</td>
<td>High-used relative block number (number of 4 KB blocks used in the data set) This 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) This 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>
<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>Variable name</td>
<td>Description and DB2 or ICF column name, or valid values</td>
<td>Update?</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>BMC_DB2_MGMTCLAS</td>
<td>Specify the name of the SMS MGMTCLAS to add&lt;br&gt;Note: You cannot use the DSRSEXIT to change an existing specified SMS MGMTCLAS.</td>
<td>No</td>
</tr>
<tr>
<td>BMC_DB2_STORCLAS</td>
<td>Specify the name of the SMS STORCLAS to add&lt;br&gt;Note: You cannot use the DSRSEXIT to change an existing specified SMS STORCLAS.</td>
<td>No</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)&lt;br&gt;The default is YES.</td>
<td>Yes</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&lt;br&gt;The default is YES.</td>
<td>Yes</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)&lt;br&gt;The default is NO.</td>
<td>Yes</td>
</tr>
<tr>
<td>BMC_STOGROUP_VOLUME.0</td>
<td>Stem variable that contains the number of volumes in the storage group list.&lt;br&gt;BMC_STOGROUP_VOLUME.0= n, where n is the number of volumes.&lt;br&gt;BMC_STOGROUP_VOLUME.1 through BMC_STOGROUP_VOLUME. n contain the actual names of volumes in the storage group</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Variables that apply only to a table space**

<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_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>Variable name</td>
<td>Description and DB2 or ICF column name, or valid values</td>
<td>Update?</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------</td>
<td>---------</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>
<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 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 795)
- Bypass redefining the data set (see “Modifying the variable for selective REDEFINE” on page 796)
- Order the storage group volumes (see “Modifying the variables for the storage group volumes” on page 796)

### 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 the following 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 the following 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, set the following variables as follows:

- 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.

- 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.
Example

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

The figure in this topic illustrates a sample REXX exit that shows one way of setting the variables in the DSRSEXIT user exit.

Figure 77: Sample DSRSEXIT REXX user exit

```rexx
/* REXX */
/* **************************************************************** */
/*                                                                  */
/* SAMPLE REXX EXIT FOR THE DATASET REDEFINE 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'.  THEN  */
/*    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 *RUIN* VALUES IN YOUR REXX VARIABLES TO BE LOST.    */
/* IT *IS* OK TO USE 'CALL' TO REFERENCE LABELED SUBROUTINES         */
/* LOCATED WITHIN THE SAME REXX PROGRAM.                            */
```

Using DSRSEXIT to manage VSAM data set redefinition

798  REORG PLUS for DB2 Reference Manual
Using DSRSEXIT to manage VSAM data set redefinition

Appendix C  REORG PLUS user exits 799
/* **************************************************************** */
/*                                                                  */
/* 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          */
/* SYSIBM.SYSINDEXPART.                                              */
/*                                                                  */
/* 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
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 “Syntax of the REORG command” on page 143. For information about the corresponding installation options, see “REORG PLUS installation options” on page 617.

Within this exit, you can also use SQL to maintain your BMCHIST tables. The example in the following figure deletes all rows over 90 days old.

**Figure 78: Example maintaining BMCHIST tables using SQL**

```
DELETE FROM STRIP(BMC_TBCREATOR_BMCHIST)||.||BMC_TBNAME_BMCHIST
WHERE  DBNAME = BMC_DBNAME
AND SPNAME = BMC_TSIX
AND UTILID = BMC_UTILID
AND DATE < ( CURRENT DATE - 90 DAYS );
COMMIT;
```

### Additional TERMEXIT restrictions

In addition to the common restrictions that apply to DSRSEXIT, TERMEXIT, and MAPTEXIT user exits, the following restrictions apply specifically 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.
- REORG PLUS always updates DB2 statistics after materializing pending definition changes. If you have a TERMEXIT user exit that changes UPDATEDB2STATS to NO in this case, REORG PLUS changes that specification to UPDATEDB2STATS YES.

For information about the common restrictions for the REXX exits, see “DSRSEXIT, TERMEXIT, and MAPTEXIT common restrictions” on page 785.

### Running the TERMEXIT exit

This topic describes how to run a TERMEXIT user exit.

**Before you begin**

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.
- On the REORG command, specify the TERMEXIT option with the name of your
  REXX user exit.

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
   “Setting REORG PLUS authorizations” on page 53 and the OPNDB2ID=YES
   installation option.

TERMEXIT variables

In addition to the variables common to all REXX exits, REORG PLUS passes
additional variables to the TERMEXIT user exit.

The following table describes the variables specific to this exit and indicates whether
you can update the variable (yes or no). (For a list of the common variables for all
REXX exits, see Table 141 on page 785.)

Table 143: 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>
</tbody>
</table>
### Variable name | Description | Update?
--- | --- | ---
BMC_OPT_UPDATEDB2STATS | 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. | Yes
BMC_OPT_BMCSTATS | 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. | Yes
BMC_CREATOR_BMCHIST | The value of this variable is populated with your BMCHIST table CREATOR. | No
BMC_TBNAME_BMCHIST | The value of this variable is populated with your BMCHIST table NAME. | No
BMC_REORG_PART_NUMBERS.n | The value of this stem variable is populated with the partition numbers of any partition involved in the reorganization. BMC_REORG_PART_NUMBERS.0=n, where n 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. | No

---

### Sample TERMEXIT REXX user exit

The figure in this topic illustrates a sample REXX exit that shows one way of setting the variables in the TERMEXIT user exit.

**Figure 79: Sample TERMEXIT REXX user exit**

```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 REORGLASTTIME 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.

****************************************************************** */
```

---

804  REORG PLUS for DB2 Reference Manual
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 '* BMC_BMCHIST TABLE NAME = ' ,
    STRIP(BMC_TBCREATOR_BMCHIST)||'.'||BMC_TBNAME_BMCHIST
/* ******************************************************************
DISPLAY ALL PART NUMBERS INVOLVED IN THIS REORGANIZATION
****************************************************************** */
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 ' ' /* ****************************************************************
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
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'
Using TERMEXIT to control BMCHIST and statistics updates
Appendix C  REORG PLUS user exits 805
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 ** 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 */
SQLSTMTS = 'SELECT REORGLASTTIME FROM SYSIBM.TABLESPACESTATS '  
'WHERE DBNAME = ? AND NAME = ? AND PARTITION = ? ' 

SQLSTMIX = '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 STMIX'  
ADDRESS DSNREXX  
'EXECSQL PREPARE S3 FROM :SQLSTMIX'  
END ELSE DO                                        /* ELSE REORG INDEX   */ 
CURR_FUNC = 'PREPARE SQL STMTS'  
ADDRESS DSNREXX  
'EXECSQL PREPARE S3 FROM :SQLSTMTS'  
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. 
************************************************************************/ 
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 */
Using MAPTEXIT to create mapping objects

The MAPTEXIT exit 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 “Setting REORG PLUS authorizations” on page 53.

- REORG PLUS terminates if you specify both the MAPTEXIT option and the MAPPINGDATABASE option.

- REORG PLUS ignores any MAPTEXIT user exit that you specify when any of the following conditions exists:
  — 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 “REORG PLUS installation options” on page 617.

Supplying MAPTEXIT in a SYSEXEC DD statement

This topic describes how to use a SYSEXEC DD statement to supply a MAPTEXIT user exit.

Before you begin

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 “REORG PLUS installation options” on page 617.

   - 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 231.

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:

   //SYSTSPRT 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 “Setting REORG PLUS authorizations” on page 53 and the OPNDB2ID=YES installation option.

**MAPTEXIT variables**

In addition to the variables common to all REXX exits, REORG PLUS passes additional variables to the MAPTEXIT user exit.

The following table describes the variables specific to this exit and indicates whether you can update the variable (yes or no).

**Table 144: Variables that REORG PLUS passes to MAPTEXIT**

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Update?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_MAPT_NOCEXEC</td>
<td>Whether the exit was called from the STEPLIB (Y) or a SYSEXEC DD (N)</td>
<td>No</td>
</tr>
<tr>
<td>BMC_MAPT_TYPE</td>
<td>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).</td>
<td>No</td>
</tr>
<tr>
<td>BMC_MAPT_AUTH</td>
<td>Authorization ID that the exit uses in the SET CURRENT SQLID SQL statement</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>The initial value of this variable is set to the authorization ID that is in effect for the REORG PLUS job.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If you change the value for this variable, the authorization ID that you supply must have the authorizations as described in <strong>CR</strong>&quot;Additional authorizations for using MAPTEXIT&quot; on page 68.</td>
<td></td>
</tr>
<tr>
<td>BMC_MAPT_CREATOR a</td>
<td>Creator portion of the fully qualified mapping table name and mapping table index name</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>The initial value of this variable is set to the user ID of the job.</td>
<td></td>
</tr>
<tr>
<td>BMC_MAPT_SUFX</td>
<td>Suffix concatenated to the BMC_MAPT_TABLE and BMC_MAPT_INDEX values to ensure that fully qualified object names are unique to a job.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>The initial value of this variable is set to the job name.</td>
<td></td>
</tr>
</tbody>
</table>
Variable name | Description | Update?
---|---|---
BMC_MAPT_TABLE | 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 | Address of the creator length, used only when BMC_MAPT_NOCEXEC=Y | No
BMC_MAPT_CRNM_ADDR | Address of the creator name, used only when BMC_MAPT_NOCEXEC=Y | No
BMC_MAPT_TBLN_ADDR | Address of the table name length, used only when BMC_MAPT_NOCEXEC=Y | No
BMC_MAPT_TBNM_ADDR | Address of the table name, used only when BMC_MAPT_NOCEXEC=Y | No

a The exit passes this value back to REORG PLUS to pass to DSNUTILB with the MAPPINGTABLE keyowrd.

Sample MAPTEXIT REXX exit

The figure in this topic shows the sample MAPTEXIT REXX exit provided with REORG PLUS.

Figure 80: Sample MAPTEXIT REXX user exit

```rexx
/* 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 ARUMAIN CALL TO DSNUTILB OR...
   DROPS MAPPING TABLE FOR IBM SHRLEVEL CHANGE REORG AFTER ARUMAIN 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,
  BMC_UTILITY_SHRLEVEL,
  BMC_JOBNAME,
  BMC_STEPNAME,
  BMC_DBNAME,
  BMC_TSIx,
  BMC_USERID,
  BMC_VCATNAME,
  BMC_DB2_SSID,
  BMC_DB2_GROUPNAME,
  BMC_DATE,
  BMC_TIME,
  BMC_DATE8,
  BMC_TIME4,
  BMC_DATEJ,
  BMC_UTILPFX,
  BMC_UTILSFX,
  BMC_REORG_TYPE,
  BMC_MAPT_TYPE,
  BMC_MAPT_AUTH,
  BMC_MAPT_CRLN_ADDR,
  BMC_MAPT_CRNM_ADDR,
  BMC_MAPT_TBLN_ADDR,
  BMC_MAPT_TBNM_ADDR
END
SAY '**** START *********** ' BMC_EXIT_POINT ' ************'
SAY ' ' IF BMC_MAPT_NOCEXEC = 'N' THEN DO
  SAY '*   EXECUTING REXX VIA SYSEXEC DD'
END ELSE DO
  SAY '*   EXECUTING COMPILED REXX VIA STEPLIB DD'
END

/* ****************************************************************
IF DESIRED, MODIFY VALUES FOR BMC_MAPT_AUTH, BMC_MAPT_CREATOR,
BMC_MAPT_SUFX, BMC_MAPT_TABLE, OR BMC_MAPT_INDEX AFTER THIS
COMMENT. OTHERWISE, DEFAULT VALUES WILL BE USED.
DO NOT MODIFY THE VALUE OF ANY OTHER VARIABLE.
*******************************************************************/
BMC_MAPT_CREATOR = BMC_USERID
BMC_MAPT_SUFX = BMC_JOBNAME
Using MAPEXIT to create mapping objects

BMC_MAPT_TABLE = 'MAP_TBL_'||STRIP(BMC_MAPT_SUFX)
BMC_MAPT_INDEX = 'XMAP_TBL_'||STRIP(BMC_MAPT_SUFX)

/* ****************************************************************
MODIFICATIONS AFTER THIS COMMENT ARE NOT RECOMMENDED.
*******************************************************************/

SAY '*   BMC_DB2_SSID               = ' BMC_DB2_SSID
SAY '*   BMC_DB2_RELEASE            = ' BMC_DB2_RELEASE
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
*******************************************************************/

CREATE_TABLE:
CURR_FUNC = 'CREATE_TABLE'
IF BMC_DB2_RELEASE < 1100 THEN DO
  SQLCRTTB = 'CREATE TABLE '.
    STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE,
    ' (TYPE CHAR( 01 ) NOT NULL, '.
    ' SOURCE_RID CHAR( 05 ) NOT NULL, ';
  "EXECSQL " SQLCRTTB
ADDRESS DSNREXX SQLCRTTB
END
Using MAPTEXIT to create mapping objects

```
  TARGET_XRID  CHAR( 09 ) NOT NULL,
  LRSN        CHAR( 06 ) NOT NULL)
END
ELSE DO
  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( 10 ) NOT NULL) ' END

SAY '*** CREATING MAPPING TABLE ***'
SAY ' '
SQLCALL = "EXEC SQL " 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
      SAY ' ',
      END
    ELSE DO
      SIGNAL RESTART_PROCESS
      END
  END
  OTHERWISE SIGNAL BAD_SQLCODE
END
CURR_FUNC = 'COMMIT TABLE CREATE'
ADDRESS DSNREXX 'EXEC SQL 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 ' ',
      END
    ELSE DO
      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 'EXECSQL 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 = "EXECSQL " 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 'EXECSQL COMMIT'
IF SQLCODE = 0 THEN DO
  SIGNAL CREATE_INDEX
END
ELSE DO
  SAY '*   COMMIT FAILED'
  SAY ' '.
  SIGNAL BAD_SQLCODE
END

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, ' WILL BE REUSED.'
    SAY ' '.
  END
END

Using MAPTEXIT to create mapping objects
Using MAPEXIT to create mapping objects
'IMPLICITLY DROPPED.'

SAY ' ' END

ELSE DO
  SAY '*   TABLE ' STRIP(BMC_MAPT_CREATOR)||'.'||BMC_MAPT_TABLE,
  ' DOES NOT EXIST.'
  SAY ' ' END

END OTHERWISE SIGNAL BAD_SQLCODE END

CURR_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
SAY '*** ERROR DURING DB2 ""CURR_FUNC"" SQLCODE ' SQLCODE
SAY
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'   = ' 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'   = ' 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
Using MAPEXIT to create mapping objects
Index

Symbols

&DATE variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&DATEJ variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&DB variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&DDNAME variable 298, 695

&GRPNM variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&JDATE variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&JOBNAME variable 298, 676
  with DSNPAT option 298
  with SPIILDSNPAT option 314

&PART variable 298, 695

&PART5 variable 298, 695

&RTYPE variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&SSID variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&STEPNAME variable

&TIME variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&TIME4 variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&TSIX variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&UID variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&USERID variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&UTIL variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&UTILID variable 695

&UTILPFX variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&UTILSFX variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

&VCAT variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPILLDSNPAT option 314

$532SOPT job 617
ABEND keyword of TIMEOUT option 226
abends
and data capture flag 607
excluding from a system dump 642
JES3 369
recovering from, using the ON FAILURE option 210
restarting from 374
restarting with SHRLEVEL CHANGE 607
taking a system dump 670
using ANALYZE statistics to avoid 520
above-the-bar storage 56
above-the-line storage 182, 532
unused pages 628
access, controlling
in REORG PLUS 53
access, shared 726
accessibility of objects
SHRLEVEL option 85, 162
tuning considerations 549
ACF2. See CA ACF2
ACFORTSS option 26
activating dynamic allocation
ACTIVE option 280, 692
example 462
ACTIVE option
command 280
installation 692
restart considerations 83
advisory pending trigger 129
advisory REORG-pending (AREOR) status 98
advisory restart pending (AREST) status 69, 73
ALL keyword
of DRAIN option 322
of DRAINTYP option 637
of FORCE option 238, 644
of REGISTER option 255
ALLOC option 283, 693
allocating data sets 335
avoiding I/O queueing 539
copy data sets 338
dynamically. See dynamic allocation of data sets
estimating size with ANALYZE 370, 520, 551
incremental copy data sets 602
inline copy data sets 253, 651
multiple 534
pattern for spill data set names 676
reallocating during a job 190
SORTWK 349
staging data set requirements 87
VIO, restrictions 352
See also dynamic allocation of data sets
allocation method, specifying for dynamic allocation 283
allocation size
adjusting 293, 705
limiting 290, 703
ALTER authority 53
ALTER SQL statement
in DDLOUT data set 349
with DDLIN data set 344
with primary, secondary quantities 787
altered index compression definition 136
altering
data set allocations 787
limit keys 344
altering a SHRLEVEL CHANGE job dynamically 589
ALTRFAIL option 625
AMEND function of DATA PACKER (for DB2) 202
AMENDED option 202
ampersand
in DSNPAT option 695
in SPIFDSNPAT option 676
ANALMAX option
considerations 536
description 626
ANALYZE option
description 187
dynamic allocation considerations 187, 372
example 399, 443, 494
meanings of subparameters 187
retrieving DASD MANAGER PLUS statistics 192
ANALYZE phase
allocation-related messages 523
bypassing gathering statistics 520
data set allocation estimates 370
determining method to use 520
multitasking considerations 536
pausing and continuing after analysis 187
primary functions 40
restarting 382
specifying ON FAILURE 211
tuning considerations 523, 551

ANY keyword of ALLOC option 283, 693
APF-authorization requirements 617
append inserts trigger 129
application defaults module 327
applying log records, online REORG 516
APPNDINS exception 129
AR/CTL (APPLICATION RESTART Control)
    implementing the interface 599
    purpose 599
    requirements 626
ARC option 626
ARCHDDN option
    command 174
    example 451
    installation 626
ARCHFORMAT option 175, 627
archive data sets 175, 353, 627
See also SYSARC data sets
ARCHIVE keyword of DDTYPE option 278, 691
archive-enabled table spaces 344
archive-enabled tables 264
ARCROWS option 216
AREOPEND exception 129
AREOR (advisory REORG-pending) status 98
AREST (advisory restart pending) status 69, 73
ARUSOPTS load module 617
ARUUMAIN module 326, 374
ASM keyword of DSNUEXIT option 226
assembler, sample DSNUEXIT exit 751
assigning copy data sets for partial reorganization 633
ASSOCIATE option of ORDER YES option 180
ASUSRPRPT data set 245, 335
attributes of data sets, checks for 83
authorization
    MONITOR2 53
authorizations
ALTER authority 53
APF 617
CONTROL authority 53
DB2 and data set
    REORG PLUS 53
DBADM 53
DBCTRL 53
DISPLAY 53
mechanisms, description 52
needed to execute 52
needed to execute REORG PLUS 53
RACF 53
required for canceling threads 53
required for XML reorganizations 53
TRACE 53
verifying 52
AUTO keyword
    of ICTYPE option 255
    of SIXSNAP option 238
    of VOLCNT option 287, 710
AUTOESTSPACE option 243
automatic backout of reorganization 386
AUX option 244
AUXREORG option 628
availability of objects
    SHRLEVEL option 85, 162
    tuning considerations 549
available pages 182, 628
AVAILPAGEPCT option 182, 628
average row length, determining 187
AVGVOLSP option 289, 694

B
backing out reorganization 386, 625
backing up BMC tables 717
base objects that participate in a clone relationship 161
BASIC keyword
    of DSNTYPE option 292
    of RBALRSN_CONVERSION option 214
BASIC keyword of DSNTYPE option 700
basic row format 138
batch applications, using with AR/CTL 599
BEGINNING keyword of LONGNAMETRUNC option 168, 658
below-the-line storage 532, 533

Index 823
of an online reorganization 30, 35
of REORG PLUS 30
BILDMAX option 537, 629
binary strings 270, 275
BMC Common Statistics component, integrating with other BMC products 51
BMC High Speed Utilities (for DB2) solution 30
BMC keyword
of ARCHFORMAT option 175, 627
of CONDEXEC option 217
of NLPCTFREE option 197
of STAGEDSN option 678
BMC Performance for DB2 Databases solution 30
BMC utilities
displaying status 719
running concurrently 67, 732
terminating 719
BMCDICT table
considerations 721
contents 720
maintaining 721
with large number of partitions 95
BMCFORCE data set 336
BMCHIST installation option, BMCHIST table 722
BMCHIST option 196, 629
BMCHIST table
backing up 717
contents 722
controlling update of 196
COPY PLUS considerations 724
maintenance 725
querying 719
RECOVER PLUS considerations 724
BMCLGRNX table 725
BMCFSWD library 331
BMCRUSER exception 129
BMCSORT
controlling 530, 533
data set that contains messages 49, 369
description 49
dynamic allocation 176, 177
MAXSORTS option 178
multitasking 537, 670, 680
SMAX option 673
SMCORE option 674
SORTDEVT option 176, 674
SORTNUM option 177, 675
version requirement 51
BMCSORT installation option 711
BMCTSTRS keyword of ANALYZE option 192
BMCTSTRS option
description 246
example 399
BMCSYNC table
backing up 717
cleaning up RECOVER PLUS UNLOADKEYS 731
considerations 729
contents 726
frequency of rows written to, controlling 196
LOB data considerations 729
maintaining 731
running utilities concurrently 67, 732
serializing 655
terminating a job 380
when updated 41
with large number of partitions 95
XML data considerations 729
BMCTRANS table 734
BMCTRIG utility, using 127
BMCUUTIL table
backing up 717
contents 736
maintaining 739
ON FAILURE TERMINATE, effect of 211
serializing 655
utility IDs 328
when updated 41
BMCCXCOPY table
backing up 717
contents 739
maintaining 746
querying 719
buffers
assigned to DB2 by REORG PLUS 528
controlling usage 525
virtual storage 540
VSAM and the 16-megabyte line 525
BUILD phase, multitasking 537
building REORG PLUS jobs 325
BYCLUSTERKEY keyword of ASSOCIATE option 180
BYTABLE keyword of ASSOCIATE option 180
C
C keyword of DSNUEXIT option 226
C language sample DSNUEXIT user exit 770
CA ACF2 security product 52
CA Top Secret security product 52

CA cache for populating document ID column 102, 242, 650
canceling or terminating a job 380
canceling threads
  authorizations required 53
  FORCE option 235, 644
  FORCE_AT option 645
  FORCE_RPT option 646
  reporting 237, 336, 646
  specifying when to cancel 236, 645

CARD exception 129
cardinality
  changes, when to run 248
  how REORG PLUS determines 551
  telling REORG PLUS how to determine 187

CATALOG MANAGER
  dynamic data set allocation considerations 79, 280
  multiple reorganizations 365
catalog, DB2
  reorganizing objects in 100
  switching I and J names 92
  updating data set allocation 787
  updating table statistics 248
  using for conditional reorganization 124

CBUFFS option 630

CCSID, specified at the column level 59, 685

CHANGE MANAGER
  dynamic data set allocation considerations 79, 280
  multiple reorganizations 365

CHANGE option of SHLEVEL option 164
changes to the product 16

character constants, string rules 275
character data, translation of 265

CHECK pending (CHKP) status
  limitations on setting 139
  with DELETE 268
  with SELECT 266
  with UPDATE 270

checks performed on work files 83

cloning data sets 515
clone objects 161

CONEX option 161

COBOL II option
  sample DSNEXIT user exit 762
  specifying user exit language 226

COBOL2 keyword of DSNEXIT option 226
codes, return

for DSNEXIT user exit 750
for REXX user exits 784
specifying with ON FAILURE 210, 212
specifying with ON MESSAGE 209
specifying with TIMEOUT 224, 681
column-level security 142
columns
  defined with CCSID specification 59, 685
  inline LOB 58, 652
  limit key number 171
  non-key in indexes 22
  TIMESTAMP WITH TIME ZONE 266
c comma as separator 275
command constant, data translation 265
command options 160
See also keywords
command syntax
  alphabetical listing of command options 144
  diagrams 148
  option descriptions 160
  rules 143
common components 49
Common Statistics component 49, 245
Common Statistics Component 51
common utility tables 95
comparison operators 274
components that REORG PLUS uses 49
See also the individual components
compressed indexes 136, 377
  indexes 25
  versioned 25
compression
  BMCDICT table 720
  dictionary 134, 194, 653
  how REORG PLUS uses and performs 134
  indexes 136, 377
  KEEPDICTIONARY option 194, 653
  phases that perform 40
  restart considerations 377
  restrictions with partition rebalancing 135
  SORTW data set 373
concatenation rules
  for DSN pattern 302
  for SPIILDSN pattern 317
concurrency issues 66, 595
concurrent sort processes 531
CONEX option
command 217
considerations 124
installation 630
interaction between command and installation options 124
condition, specifying for WHERE clause 273
conditional reorganization
considerations 124
limit command options 219, 221
limit installation options 654, 662
REPORTONLY option 222
specifying 630
using the DASD MANAGER PLUS exceptions table 127
using the DB2 catalog 124
console, MVS, using with SHRLEVEL CHANGE 593
constants
description and use 270, 275
floating point 275
rules for 275
specifying for a comparison 275
translation of 265
with UPDATE 270
with WHERE 275
CONTINUE keyword
of LONGLOG option 311, 657
of MGEXTENT option 661
of SHORTMEMORY option 184, 671
of UNLOAD option 193, 685
CONTINUE UTILITY keyword of ON MESSAGE option 209
CONTROL authority 53
control card data sets, LOAD 60
control interval (CI) size, variable 136
controlling access
in REORG PLUS 53
controlling BMCHIST updates 801
conventions, documentation 14
conversion of table space type 98
converting RBA or LRSN format 214
copies, image
considerations for large number of partitions 547
difference between inline and standard 343
FlashCopy 142
incremental 255
inline 253
large number of partitions 547
memory considerations 547
partition-by-growth table spaces 342
partition-level 341, 633
registering 255
REORG PLUS 529, 540
specifying inline option 253
specifying option to enable 251
subset of partitions in single copy 633
See also inline image copies
copies, incremental 255, 540
copies, snapshot 516
copy data set
soverriding default ddnames, remote 260
copy data sets
allocating 337
buffer usage, controlling 527, 630
considerations when not creating 139
DD DUMMY 83
default DD names example 392
DISP=MOD considerations 83
dynamic allocation 278
for partition-by-growth table spaces 342
GDG names 79
incremental 255
inline 253
multitasking 536
soverriding default ddnames, local 259
performance, improving 343
registering 255, 338
registration failure during SHRLEVEL CHANGE 606
specification and usage 336
subset of partitions in single data set 633
temporary 83
See also copy data sets, dynamic allocation of
See also incremental copy data sets
copy data sets, dynamic allocation of 278
considerations 338
DDTYPE option 278
expiration date 304
GDG names 79
limiting GDG versions 305
naming 79, 80
retention period 305
See also dynamic allocation of data sets
COPY option
  COPY NO considerations 139
description 251
examples 392, 416
copy options 251
COPY pending status
  example 439
  setting with COPY NO 252
copy registration
  BMCXCOPY table 739
  REGISTER option 255
copy, partition level 341
COPYDDN option
  command 259
  examples 399, 451
  installation 631
COPYLVL option
  command 257
  copies, image
    specifying full or partial 257
  copy data sets, dynamic allocation of
    assigning partitions 257
  examples 399, 416
  FULL keyword of COPYLVL option 257
  image copies
    specifying full or partial 257
  image copy data sets, dynamic allocation of
    assigning partitions 257
  installation 632
  keywords
    FULL keyword of COPYLVL option 257
    PART keyword of COPYLVL option 257
  PART keyword of COPYLVL option 257
COPYMAX option 633
COPYSUBSET option 633
CPUs, number of 547
CPYRFAIL option
  actions during failure 606
  description 633
  with DDLIN data set 119
creating additional data sets as needed (REDEFINE option) 515
creating data set name 79
CURRENT DATE keyword
  of predicate 277
  of UPDATE SET option 270
current date variable
  with DSNPAT option 298, 695
  with SPI LDSNP option 676
  with SPI LLDSNPAT option 314
current Julian date variable
  with DSNPAT option 298, 695
  with SPI LDSNP option 676
  with SPI LLDSNPAT option 314
CURRENT TIME keyword of UPDATE SET option 270
current time variable
  with DSNPAT option 298, 695
  with SPI LDSNP option 676
  with SPI LLDSNPAT option 314
current time variable, short form
  with DSNPAT option 298, 695
  with SPI LDSNP option 676
  with SPI LLDSNPAT option 314
CURRENT TIMESTAMP keyword
  of predicate 277
  of UPDATE SET option 270

D
D2U (DB2 Utilities Common Code) 49, 51
DASD MANAGER PLUS
  BM CSTATS keyword of ANALYZE option 192
  BM CSTATS option 246
  BM CT RIG utility 127
  conditional reorganizations 127
  dynamic data set allocation considerations 79, 280
  exceptions table 127
  multiple reorganizations 365
  statistics retrieved by REORG PLUS 520
  updating statistics 246
DASD vendor considerations for Instant Snapshot 112
data availability
  during reorganization 85
  maximizing 549
  performance information 541
data capture flag
abnormal termination 607
REORG PLUS resetting 570
REORG PLUS setting 568
setting with DATACAP option 634
data classes, SMS 306, 695
data integrity, checks for 85
DATA PACKER (for DB2), AMEND function 202
data set allocation. See allocating data sets
data set name pattern
  concatenation 302
  specifying 297, 695
  user exit for 749
data set name pattern user exit
  assembler example 751
  C example 770
  COBOL II example 762
  description 749
  LE_C example 777
  LE_COBOL example 762
  specifying 639
  user-defined variables 750
data set redefine user exit
  considerations for nonpartitioned objects 788
  description 787
  example 798, 804
  invoking the user exit 790
  ordering STOGROUP volumes 789
  resizing data sets 787
  selectively redefining 788
  specifying 641
  variables passed 791
data set rules 541
data set type, specifying for dynamic allocation 292, 700
data sets, DB2
  changing size of 787
  controlling buffer usage 528
  extending 200, 661
  multi-data-set spaces 97
  preformatting unused portion 213
  SHRLEVEL CHANGE 87
  SHRLEVEL REFERENCE 87
  used by REORG PLUS 528
data sets, REORG PLUS
  allocating 335
  ASUSRPRT 245, 335
  attributes, checks for 83
  BMCFORCE 336
  calculating sizes 370
  control card, LOAD 60, 278, 361
  corresponding image copy types 602
  creating additional 515
  deleting 81, 202
  deleting and renaming 204
  description 46
  discard 60, 353
  disposition 83
  DSN pattern 516
  dynamic allocation 280
  extended sequential
    allocating 292, 700
    SORTWK restriction 352
    SYSREC restriction 365
  FASTSWITCH process 92
  large format 292, 700
  LOAD control card 60, 278, 361
  log spill records 657, 676
  pattern for names 516
  prefixes, specifying 80
  rename and delete process for staging data sets 90, 92
  resizing 794
  specifying DD statements 332, 370
  staging 87
  striped 352
  SYSPUNCH 60, 278, 361
  See also allocating data sets
  See also dynamic allocation of data sets
data sharing
  group name variable for DSNPAT option 298
  group name variable for SPILLDSNPAT option 314
  restart considerations 378
  using XBM 241
  when DB2 versions are not the same 327
  when using table space compression 327
data space
  allocating for log records 319
  allocating for RID maps 318
  estimating size for log records 611
  restricting size for RID maps 610
data-sorting indexes, definition 180
data, deleting during reorganization 268
database name variable
with DSNPAT option 298, 695
with SPILDSNP option 676
with SPILLDSNPAT option 314
Database Performance (for DB2) solution 30, 192, 217
database status 69
database, mapping 323
database, specifying table space to be reorganized 160
DATALCAP option 634
DATAclas option 306, 695
DATAclas value in DSNZPARMs 199
date rules for constants 275
DATE SQL function 137
date variable, current
  with DSNPAT option 298, 695
  with SPILDSNP option 676
  with SPILLDSNPAT option 314
date variable, Julian
  with DSNPAT option 298, 695
  with SPILDSNP option 676
  with SPILLDSNPAT option 314
DAY SQL function 137
DB2
  authority to execute REORG PLUS 53
  changing the size of VSAM data sets 787
  DSNUTILB program 57
  REORG utility 51
  REORG utility compared to REORG PLUS 35
  security exit 52
  subsystem identifier (SSID) parameter of EXEC statement 327
  system setup 51
  TEMPLATE control statements 62
  user exits 139, 534
  UTILITY_OBJECT_CONVERSION parameter 214
  versions supported 51
  versions, when different on data sharing group 327
DB2 catalog
  reorganizing objects in 100
  switching I and J names 92
  updating data set allocation 787
  updating table statistics 248
  using for conditional reorganizations 124
DB2 data sets. See data sets, DB2
DB2 directory 100
DB2 keyword of ARCHFORMAT option 175, 627
DB2 Solution Common Code (SCC). See SCC
DB2 Utilities Common Code. See D2U
DBCNTL installation data set 389
DBC, support for 265
DBET error status 69
DBETE status 69
DD DUMMY 83, 245, 368
DD statements
  $SORTPARM 370
  ASUSRPRT 335
  BMCFORCE 336
  copy data sets 602
  DDLIN 344
  DDLOUT 349
  description 332
  DUMMY 83, 368
  for REXX exits 790, 802, 810
  SORTWK 349
  SYASC 353
  SYSErr 356
  SYSEXEC 357
  SYSIDCIN 357
  SYSIN 360
  SYSPRIN2 361
  SYSPRINT 360
  SYSEREC 362
  SYSTERM 366
  SYSTSPRT 366
  SYSTUT1 366
  using with dynamic allocation 282, 702
  UTPRINT 369
DDL
  pending changes 59, 98, 664
  to rebalance partitions 120
DDLDDN option 176, 634
DDLIN data set
  considerations 116
  contents 120
  example 477
  failure during UTILTERM 625
  other requirements 120
  sample statements and messages 347
  specification and usage 344
  supported data types in limit keys 117
  unsupported data types in limit keys 117
DDLOUT data set 349
ddname variable 298, 695
ddnames, overriding
archive data set 174, 626
DDLIN data set 176, 634
IDCAMS input data set 201
incremental local copy data sets 262, 648
incremental remote copy data sets 263, 665
index work data set 173, 686
local copy data sets 259, 631
output data set 171, 684
remote copy data sets 260, 666
SYSARC 174, 626
SYSREC 171, 684
SYSUT 173, 686

DDTYPE option
command 278
examples 416, 462
installation 691

deactivating dynamic allocation 280, 692

DEADLINE option
command 309
example 477
installation 635

decimal floating point 270, 275, 345
decimal point as comma 275
decimal, rules for constants 275
defaults, installation options 618
DEFER keyword of MAXRO option 308, 660
DEFINE NO attribute, restrictions 97
DELAY option
command 312
example 486
installation 635
delete and rename process for staging data sets
backing out after failure 386
backing out after limit key update failure 625
canceling during 380, 608
description 90
failure during 384
falling back from FASTSWITCH process 207, 646
object status following 74
threshold for 208, 515, 647
DELETE keyword
of ORIGDISP option 664
of ORIGINALDISP option 204
of TAPEDISP option 679
DELETE option
description 268
referential integrity issues 139
SHRLEVEL CHANGE considerations 596
DELETEFILES option
changing the default value 636
data sets deleted 202
description 202
restart considerations 379
when processing performed 202, 636
with dynamic allocation 81
with ON FAILURE TERMINATE 202, 212
with TIMEOUT TERM 202
deleting data during reorganization 268
deleting dynamically allocated work files 81, 202, 636
DELFILES option 636
delimited tokens in command syntax 143
DESCCDE option 637
device type, specifying 176
diagrams, syntax 148
dictionaries, compression
BMCDICT table 720
for partition-by-growth table spaces 136
KEEPDICTIONARY option 194, 653
restrictions with partition rebalancing 135
using existing 136
using with table space compression 134
directory, DB2 100
DISABLED keyword of ZIIP option 240, 688
disabling zIIP processing 240, 688
discard data sets. See SYSARC data sets
discarding rows
to a discard data set, with DSNUTILB 60, 356
to an archive data set 353
with DELETE 268
with SELECT 266
DISP (data set disposition) 83
DISPLAY option
description 589
log apply control 583
sample output, terse 589
sample output, verbose 589
DISPLAY privileges 53
displaying drain locks 234, 641
displaying status of BMC utilities 719
document ID column
cache for 102, 242, 650
restrictions on 270
document ID indexes 376
documentation information 13
double-byte character set, support for 265
drain
action on timeout 224, 681
canceling threads for 235, 644
DELETEFILES option, processing 202
displaying locks 234, 641
read claimers, canceling 236, 238, 644
specifying the number of retry attempts 233,
638
specifying the time between retry attempts 234,
638
specifying the timeout value 232, 638
specifying type for SHRLEVEL CHANGE 322,
637
write claimers, canceling 238, 644
DRAIN keyword of LONGLOG option 657
DRAIN option 322
DRAIN_WAIT option 232
DRAINTPY option 637
DRNDELAY option
description 638
performance tuning 542
DRNFAIL keyword of DSPLOCKS option 234, 641
DRNRTRY option
description 638
performance tuning 542
DRNWAITE option
description 638
performance tuning 542
DROP COLUMN, pending 98
DSECT, assembler parameter block 751
DSEXTENT exception 129
DSN pattern 297, 695
DSN=NULLFILE 83, 368
DSN1COPY, making as part of reorganization 252
DSNDB01 database 100
DSNDB06 database, reorganizing objects in 100
DSNHDECP
settings for changing decimal points to comma 275
SSID from 327
DSNPAT option
command 297
examples 456, 462
installation 695
DSNTYPE option 292, 700
DSNUEXIT option 639
DSNUEXIT user exit
assembler example 751
C example 770
COBOL II example 762
description 749
LE_C example 777
LE_COBOL example 762
specifying 639
user-defined variables 750
DSNUILTB
specifying 60
when invoked 57
DSNUILTB option 223, 640
DSNUILTB reorganization
considerations 57
data set name pattern, symbolic variables 297,
695
diagram of phases 39
dynamic data set allocation 62, 77
example 462
how to run 60
multi-table table spaces 61
object status 69, 72
processing phases 39
recovery 380
requirements for 60
restrictions 61
software requirements 60
specifying 223, 640
statistics 131, 248
symbolic variables for data set name pattern
297, 695
threads used 57
user-defined variables 298
DSNZPARMs
SMS DATACLAS value 199
utility timeout value 232, 638
DSPLOCKS option
command 234
installation 641
performance tuning 542
DSRSEXIT option 228, 641
DSRSEXIT user exit
considerations for nonpartitioned objects 788
description 787
eexample 798, 804
invoking 790
ordering STOGROUP volumes 789
resizing data sets 787
selectively redefining data sets 788
specifying 641
variables passed 791
DSSIZE, estimating copy data set size 338, 602
DSSIZE, pending 98
dump, system
excluding abend codes 642
generating 670
duration, labeled, specifying as part of a condition 277
DYNALLOC option, RETRY parameter 711
DYNALOC installation option 177, 675, 711
dynamic allocation of data sets 79, 82, 280, 689
activating and deactivating 78, 280, 692
allocation method 283, 693
ANALYZE option 187, 372
ARCHDDN option 174
archive (SYSARC) 174, 176, 353
BMCSORT 177
CATALOG MANAGER considerations 79, 280
CHANGE MANAGER considerations 79, 280
command options 277
copy, full image 278
COPYDDN considerations 259
DASD MANAGER PLUS considerations 79, 280
data set type, specifying 292, 700
DD statements with 282, 702
DD type specification 278, 691
DDLDDN considerations 176, 634
deleting 81, 202, 636
description 77
discard data sets (SYSARC) for DSNUTILB jobs 353
DSNUTILB reorganization 62
enabling 78
eexample 399
expiration date 304
extended sequential 292, 700
GDG names 79
ICDDN considerations 262
incremental image copy 47, 278
index work files (SYSUT) 173, 278, 366
installation options 689, 691
large format 292, 700
limiting size 290, 703
LOAD control card data sets (SYSPUNCH) for DSNUTILB jobs 361
local full copy data sets 259
local incremental copy data sets 262
merging with DD statements 282, 702
method 283, 693
name pattern user exit 747
naming
using DSNPATT 79, 297, 695
using DSNUEXIT 750
options 277
pattern for data set names 297, 695, 750
percentages of allocation size 293, 705
performance 529
prefix (data set) considerations 80, 171, 173
RECOVERYDDN considerations 260
RECOVERYICDDN considerations 263
Index

remote full copy data sets 260
remote incremental copy data sets 263
restarting a job 83, 374, 379
retention period 305
SMS classes 306, 695
sort work files (SORTWK) 278, 349
SPACE option 294
tape devices, number of 278
terminating job 380
threshold for secondary unit/class 707
threshold for secondary values 295, 707
unit count 286
unit names 285, 709
UNLDDN considerations 171
unload work files (SYSREC) 278, 362
user exits 226, 639, 641
using 78
volume count 287
WORKDDN considerations 173
worklist environment 280
dynamic allocation of data set copy, full image 47
dynamic allocation of tape units 82
dynamic allocation options 277
dynamically altering a SHRLEVEL CHANGE
reorganization 589
DYNAMNBR parameter 95, 326

example REORG PLUS jobs 389
exceptions, generating a reorganization 127
EXCLDUMP option 642
EXEC statement
description 326
utility parameters 327
EXECUTE privileges
REORG PLUS 53
executing REORG PLUS jobs 374
execution phases of REORG PLUS
compared to DB2 35
primary functions 39
SHRLEVEL CHANGE 567, 572
tuning considerations 550
execution phases of REORG PLUS DSNUTILB
processing phase
DSNUTILB 39
exits
security 52
exits, DB2 139
exits, user

E

EA-enabled data sets 98
EDITPROCs
and UXSTATE value 139
invoking with an AMENDED table 202
libraries that contain 331
row format conversion 138
rows compressed by 134
electronic documentation 13
ENABLE ARCHIVE, tables defined with 264
ENABLED keyword of ZIIP option 240, 688
enabling dynamic allocation 78
enabling zIIP processing 140, 240, 688
END keyword of LONGNAMETRUNC option 168, 658
ENDING AT clause of SQL ALTER statement 26
enqueues, specifying 655
ERP environments, REORG PLUS
recommendations 541
estimating data set sizes, with ANALYZE 520
evaluation of predicate 273
accessing the source 748
authorizations for 53, 789
controlling BMCHIST update 801
controlling statistics updates 801
creating mapping objects 808
data set name pattern
  assembler example 751
  C example 770
  COBOL II example 762
description 749
  invoking 639
  LE COBOL example 762
  LE_C example 777
data set redefine
  description 787
  example 798
  invoking 641
dropping mapping objects 808
DSNUEXIT
  assembler example 751
  C example 770
  COBOL II example 762
description 749
  invoking 639
  LE_C example 777
  LE_COBOL example 762
data set redefine
  description 787
  example 798
  invoking 641
dropping mapping objects 808
DSRSEXIT
  description 787
  example 798
  invoking 641
list of exits provided 747
mapping table
  description 808
  example 812
  invoking 231, 659
MAPTEXIT
  description 808
  example 812
  invoking 231, 659
REXX
  requirements 809
  restrictions 784
  setting up 790, 802, 810
sample library 748
TERMEXIT
  description 801
  example 804
  invoking 680
variables
for DSN patterns 298
for SPILLDSN patterns 314
EXPDT option 304, 701
expiration date for certain data sets 304, 701
expression, indexes created on 59, 137, 652
EXTENDED BUFFER MANAGER. See XBM
EXTENDED BUFFER MANAGER (XBM)
  REORG PLUS requirements 53
EXTENDED keyword of
  RBALRSN_CONVERSION option 214
extended extended sequential data set
  sallocating 292
extended sequential data sets
  allocating 700
  SORTWK restriction 352
  SYSREC restriction 365
extending data sets 200, 661
EXTENTS exception 129
extents, secondary 661
EXTPREFIX keyword of DSNTYPE option 292, 700
EXTREQ keyword of DSNTYPE option 292, 700
F
FAIL keyword
  of FILECHK option 644
  of SHORTMEMORY option 184, 671
failure
  cannot restart in UTILTERM 384
  during FASTSWITCH processing 94, 207, 646
  during index reorganization 382
  during limit key updates 386
  during rename processing 386
  handling for Instant Snapshot 111
  inadequate space 379
  restarting from 374
  specifying how to handle 210
  while setting restrictive statuses 386
falling back from FASTSWITCH to rename process 207, 646
FARIND exception 129
FAROFF exception 129
FASTSWITCH option 205, 642
FASTSWITCH process

834  REORG PLUS for DB2 Reference Manual
Index

backing out after failure in UTILTERM 386, 625
command option 205
completing manually after failure in UTILTERM 384
failure during 94
falling back to rename process 207, 646
installation option 642
naming conventions 92
object status following 92
restarting in UTILTERM 380
threshold for 208, 515, 647
with ORIGINALDISP 204, 664
features of REORG PLUS 32
FIELDPROCs
in WHERE clause 142, 273
libraries that contain 331
statistics gathered 246
FILECHK option 83, 644
five-byte partition variable 298, 695
fixes applied, generating a report for 330
FlashCopy image copies 142
floating point
ALTER to before the reorganization 108
columns unsupported in WHERE clauses 142
decimal 270, 275, 345
rules for constants 275
FORCE option 235, 644
FORCE_AT option
command, of FORCE option 236
installation 645
FORCE_DELAY option 237
FORCE_RPT option
command, of FORCE option 237
installation 646
FORCEROLLUP Common Statistics component option 248
FORMAT BMC option of LOADPLUS 175, 627
format of archive data sets 175, 627
FORMAT UNLOAD data 35, 175, 627
FREE keyword of IFALLOC option 282, 702
free space in nonleaf index pages, controlling 197
FSFALLBACK option 207, 646
FSTHRESHOLD option 208, 647
FULL keyword of COPYLVL option 632
functions, SQL, indexes created on expressions that contain 137

G
GDG. See generation data group
GDGEMPTY option 701
GDGLIMIT option 305, 702
GDGSCRATCH option 702
generation data group (GDG) 79
limiting number of generations 305, 702
removing from the VTOC 702
uncataloging 701
granting privileges to PUBLIC 53
graphic strings rules for constants 275
GRECP (group RECOVER pending) status 69, 73
group name variable
with DSNPAT option 298, 695
with SPIILDSNP option 676
with SPIILDSNPAT option 314
group RECOVER pending status (GRECP) 69, 73

H
hardware compression
BMCDICT table 720
See also compression
hash-organized tables 58, 243, 648
HASHAX option 648
hexadecimal strings rules for constants 275
hierarchy of log apply control options 585
high-used RBA
ANALYZE HURBA option 522
average row length, determining 522
cardinality estimate 191, 522
recovering objects 32
REDEFINE NO option 200
resetting 200
HISTORY Common Statistics component option 248
HISTORY installation option
BMCHIST table 722
HISTRETN installation option
BMCHIST table 724
HURBA keyword of ANALYZE option 191

I
I/O
performance improvement 367, 526, 538
performance messages 542
processing, to maximize 539
queueing, avoiding 539
I/O performance for partitioned objects 525
I/O, maximum virtual storage 540
ICDDN option 262, 648
ICF (Integrated Catalog Facility) 53
ICTYPE option
command 255
examples 486
installation 648
performance 540
ID
subsystem variable
with DSNPAT option 298, 695
with SPILDSNP option 676
with SPILLDSNPAT option 314
user variable
with DSNPAT option 298, 695
with SPILDSNP option 676
with SPILLDSNPAT option 314
utility (parameter of EXEC statement) 328
utility prefix variable
with DSNPAT option 298, 695
with SPILDSNP option 676
with SPILLDSNPAT option 314
utility suffix variable
with DSNPAT option 298, 695
with SPILDSNP option 676
with SPILLDSNPAT option 314
utility variable
with DSNPAT option 298, 695
with SPILDSNP option 676
with SPILLDSNPAT option 314
XBM subsystem 241, 687
IDCACHE option 242, 650
IDCAMS, specifying commands for data set redefinition 357, 392, 580
IDCDDN option 201
identity column restrictions 270
IFALLOC option
command 282
example 462
installation 702
image copies
considerations for large number of partitions 547
difference between inline and standard 343
FlashCopy 142
incremental 255
inline 253
large number of partitions 547
memory considerations 547
partition-level 341, 633
registering 255
REORG PLUS 540
specifying inline option 253
specifying option to enable 251
subset of partitions in single copy 633
See also inline image copies
image copy data sets
allocating 337
buffer usage, controlling 527, 630
considerations when not creating 139
DD DUMMY 83
default DD names example 392
DISP=MOD considerations 83
for partition-by-growth table spaces 342
GDG names 79
incremental 255
inline 253
multitasking 536
overriding default ddnames, local 259
overriding default ddnames, remote 260
performance, improving 343
registering 255, 338
registration failure during SHRLEVEL CHANGE 606
specification and usage 336
subset of partitions in single data set 633
temporary 83
See also inline image copies
See also incremental copy data sets
image copy data sets, dynamic allocation of considerations 338
DDTYPE option 278, 691
expiration date 304
GDG names 79
limiting GDG versions 305
naming 79, 80
retention period 305
implicit time zone 266
IMPLICIT_TIMEZONE DSNHDECP value 266
IMPLICIT_TZ option 266
inadequate space failure 379
incremental copy data sets
  buffer usage, controlling 528
  corresponding image copy types 602
  ddnames required 602
  improving performance 602
  overriding default ddnames 262, 263
  performance, improving 602
  registering 602
  specifying and using 602
  specifying with ICTYPE 255
incremental image copies 255, 540
INCREMENTAL keyword of ICTYPE option 255, 648
INDEX option 161
index reorganization
  ANALYZE option 522
  example 431
  failure during 382
  falling back from FASTSWITCH to rename process 207
INDEX option 161
performing conditionally 124
processing phases
  RELOAD 555
  REORG phase 559
  single-phase reorganization 39
  single-phase reorganization with SHRLEVEL CHANGE 572
  two-phase reorganization 39
  two-phase reorganization with SHRLEVEL CHANGE 572
SHRLEVEL CHANGE option 87
SHRLEVEL REFERENCE option 87
SHRLEVEL REFERENCE UNLOADONLY option 86
  single-phase 133
SORTWK data sets 349, 530
  specifying 161
status requirements 69, 72
SYSUT1 data sets 366, 526
UNLOAD phase 552
 See also indexes
index space name variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPIILDSNPAT option 314
index work data sets (SYSUT1)
allocating 335
controlling buffer usage 526
DD DUMMY 83, 368
deleting automatically after job 202
description 48
DSN=NULLFILE 83, 368
dynamic allocation
  DDTYPE option 278
  description 77
  syntax 277
  XML table space reorganization 190
multiple 538
overriding default ddbname 686
performance considerations 538
  tuning considerations 538
XML table space reorganization 103
indexes
catalog, reorganizing 100
compressed 136, 377
created on expression 59, 137, 652
data-sorting definition 180
document ID 376
LOB 106
  multiple 541
  node ID 190
non-key columns in 22
nonpartitioned 109
null keys 59, 686
on a clone table 161
on hash-organized tables 648
on inline LOB columns 58, 652
random keys 59, 462, 653
secondary, REDEFINE 198
SHRLEVEL CHANGE access to objects 74
  using Instant Snapshot 110
versioned 100
  with pending DDL changes 59
XML 53, 103
indexes on hash-organized tables 58
INDREFLIM option 220
INDREFLM option 650
initial status of objects 69
inline image copies 253
  performance considerations 540
  restart considerations 379
  specifying 253, 651
  with ICTYPE UPDATE 255, 648
  inline LOB columns 58, 137, 652
  INLINE option of COPY YES option

Index 837
description 253  
performance information 540  
INLINECP option  
description 651  
performance information 540  
INLOB option 652  
input data set 48, 360  
installation options  
  $ARUDYNA macro 689  
  $ARUOPTS macro 618  
  basic REORG PLUS 618  
  BMCHIST 722  
  dynamic data set allocation 689  
  HISTORY 722  
  HISTRETN 724  
  list of 618, 689  
  multiple modules 331, 617  
Installation System 617  
Instant Snapshot  
  DASD vendor considerations 112  
  handling failures 111  
  specifying 238, 672  
  with multi-data-set objects 97  
  with nonpartitioned indexes 110  
integer rules for constants 275  
integrity checks performed on work files 83  
interface  
  with AR/CTL 599  
  with XBM or SUF 589  
invoking DSNUTILB  
  considerations 57  
  DSNUTILB option 223, 640  
  example 462  
  See also DSNUTILB reorganization  
invoking REORG PLUS 374  
IRLMWAIT, used with drain timeout 232, 638  
IXONEX option 652  
IXRANDOM option 653  

J  
JCL  
  example REORG PLUS jobs 389  
  in DBCNTL installation data set 389  
JES3 limitation 369  
JOB name variable  
  with DSNPAT option 298, 695  
  with SPIILDSNP option 676  
  with SPIILLDSNPAT option 314  
JOBS statement 325  
jobs, REORG PLUS  
  building 325  
  example 389  
  executing 374  
  for installation options 617  
  restarting and recovering 374  
  restarting during UTILTERM 380, 608  
  terminating or canceling 380, 608  
K  
KEEPDICTIONARY option  
  command 194  
  how REORG PLUS uses 134  
  installation 653  
  performance considerations 515  
key compression 193  
KEYCARD Common Statistics component option 248  
keys  
  derived from an expression 59, 137, 652  
  limit 117, 120  
  null 59, 686  
  random 59, 462, 653  
keywords
ABEND keyword of TIMEOUT option 226
ACTIVE option 280, 692
ALL keyword
  of DRAIN option 322
  of DRAINTYP option 637
  of FORCE option 238, 644
  of REGISTER option 255
ALLOC option 283, 693
ALTRFAIL option 625
AMENDED option 202
ANALMAX option 626
ANALYZE option 187
ANY keyword of ALLOC option 283, 693
ARC option 626
ARCHDDN option 174, 626
ARCHFORMAT option 175, 627
ARCHIVE keyword of DDTYPE option 278, 691
ARCROWS option 216
ASM keyword of DSNUEXIT option 226
ASSOCIATE option of ORDER YES option 180
AUTO keyword
  of ICTYPE option 255
  of SIXSNAP option 238
  of VOLCNT option 287
AUTO keyword of VOLCNT option 710
AUTOESTSPACE option 243
AUX option 244
AUXREORG option 628
AVAILPAGEPCT option 182, 628
AVGVOLSP option 289, 694
BASIC keyword
  of DSNTYPE option 292
  of RBALRSN_CONVERSION option 214
BASIC keyword of DSNTYPE option 700
BEGINNING keyword of
  LONGNAMETRUNC option 168, 658
BILDMAX option 629
BMC keyword
  of ARCHFORMAT option 175, 627
  of CONDEXEC option 217
  of NLPCTFREE option 197
  of STAGEDSN option 678
BMCHIST option 196, 629
BMCMSTATS keyword of ANALYZE option 192
BMCMSTATS option 246
BYCLUSTERKEY keyword of ASSOCIATE option 180
BYTEABLE keyword of ASSOCIATE option 180
C keyword of DSNUEXIT option 226
CBUFFS option 630
CHANGE option of SHRLEVEL option 164
CLONE option 161
COBOL2 keyword of DSNUEXIT option 226
CONDEXEC option 217, 630
CONTINUE keyword
  of LONGLONG option 311, 657
  of MGEXTENT option 661
  of SHORTMEMORY option 184, 671
  of UNLOAD option 193, 685
CONTINUE UTILITY option of ON
  MESSAGE option 209
COPY option 251
COPYDDN option 259, 631
COPYLVL option 257, 632
COPYMAX option 633
COPYSUBSET option 633
CPYFAIL option 633
CURRENT DATE keyword
  of predicate 277
  of UPDATE SET option 270
CURRENT TIME keyword of UPDATE SET option 270
CURRENT TIMESTAMP keyword
  of predicate 277
  of UPDATE SET option 270
DATACAP option 634
DATACLAS option 306, 695
DB2 keyword of ARCHFORMAT option 175, 627
DDLDDN option 176, 634
DDTYP option 278, 691
DEADLINE option 309, 635
DETER keyword of MAXRO option 308, 660
DELAY option 312, 635
DELETE keyword
  of ORIGDISP option 664
  of ORIGINALDISP option 204
  of TAPEDISP option 679
DELETE option 268
DELETFILES option 202
DELFIL option 636
DESCCDE option 637
description of
  command options 160
  installation options 625
DISABLED keyword of ZIIP option 240, 688
DISPLAY option 589
DRAIN keyword of LONGLONG option 657
DRAIN option 322
DRAIN_WAIT option 232
DRAINTYP option 637
DRNDELAY option 638
DRNFAIL keyword of DSPLOCKS option 234, 641
DRNRETRY option 638
DRNWAIT option 234, 641
DSNPAT option 297, 695
DSNUEXIT option 226, 639
DSNUTILB option 223, 640
DSPLOCKS option 234, 641
DSRSEXIT option 228, 641
ENABLED keyword of ZIIP option 240, 688
END keyword of LONGNAMETRUNC option 168, 658
EXCLDUMP option 642
EXPDT option 304, 701
EXTENDED keyword of RBALRSN_CONVERSION option 214
EXTPREF keyword of DSNTYPE option 292, 700
EXTREQ keyword of DSNTYPE option 292, 700
FAIL keyword
of FILECHK option 644
of SHORTMEMORY option 184, 671
FASTSWITCH option 205, 642
FILECHK option 644
FORCE option 235, 644
FORCE_AT option
command, of FORCE option 236
installation 645
FORCE_DELAY option of FORCE option 237
FORCE_RPT option
command, of FORCE option 237
installation 646
FREE keyword of IFALLOC option 282, 702
FSFALLBACK option 207, 646
FSTHRESHOLD option 208, 647
FULL keyword of COPYLVL option 632
GDGEMPTY option 701
GDGLIMIT option 305, 702
GDGSCRATCH option 702
HASHAX option 648
HURBA keyword of ANALYZE option 191
ICDDN option 262, 648
ICTYPE option 255, 648
IDCACHE option 242, 650
IDCDDN option 201
IFALLOC option 282, 702
IMPLICIT_TZ option 266
INCREMENTAL keyword of ICTYPE option 255, 648
INDEX option 161
INDREFLIMIT option 220
INDREFLM option 650
INLINE option of COPY YES option 253
INLINECP option 651
INLOB option 652
IXONEX option 652
IXRANDOM option 653
KEEPDICTIONARY option 194, 653
keywords
SUFSTART keyword of WTOMSG option 242
LARGE keyword of DSNTYPE option 292, 700
LAST keyword of PART option 167
LASTRETRY keyword of FORCE_AT option 236, 645
LE_C keyword of DSNUEXIT option 226
LE_COBOL keyword of DSNUEXIT option 226
LEADISTLIMIT option 221
LEAFDSLM option 654
LOB option 655
LOCBFCPY keyword of DDTYPE option 278, 691
LOCICPY keyword of DDTYPE option 278, 691
LOCKROW option 655
LOCPFCPY keyword of DDTYPE option 278, 691
LOCPICPY keyword of DDTYPE option 278, 691
LOG NO option 197
LOGFINAL option 320, 655
LOGMEM option 319, 656
LOGSPIL option 657
LOGSPILL option 320
LOGTHRESHLD option 308
LOGTHRSH option 657
LONGLG option 311, 657
LONGNAME option 168, 658
MAPPINGDATABASE option 323
MAPPINGTABLE option 323
MAPEXIT option 231, 659
MAXEXTSZ option 290, 703
MAXNEWPARTS option 167, 659
MAXRO option 308, 660
MAXSORTMEMORY option 186, 661
MAXSORTS option 178
MAXTAPE option 278, 661
MGEXTENT option 661
MGMTCLAS option 306, 704
MIDDLE keyword of LONGNAMETRUNC option 168, 658
MINIMAL keyword of ALLOC option 283, 693
MINSORTMEMORY option 186, 662
MGMTCLAS option 306, 704
MIDDLE keyword of LONGNAMETRUNC option 168, 658
MINIMAL keyword of ALLOC option 283, 693
MINSORTMEMORY option 186, 662
MSGLEVEL option 662
NLPCTFREE option 197
NONE keyword
  of DATACLAS option 695
  of DEADLINE option 309, 635
  of DRAIN_WAIT option 232
  of DRNWAIT option 638
  of DSNTYPE option 292, 700
  of DSNUEXIT option 226, 639
  of DSPLOCKS option 234, 641
  of DSRSEXIT option 228, 641
  of FORCE option 644
  of INDREFLM option 650
  of LEAFDSL option 654
  of LOGFINAL option 320, 655
  of MAPTEXIT option 231, 659
  of MGMTCLAS option 704
  of OFFPOSMLM option 662
  of RAIBALRSN_CONVERSION option 214
  of REGISTER option 255
  of SHRLEVEL option 162
  of SPILLSTORCLAS option 313
  of SPILLUNIT option 313
  of SPILOTSLS option 677
  of SPILUNIT option 677
  of STORCLAS option 707
  of TERMEXIT option 229, 680
NOSYSREC option 192
NULL keyword
  of predicate 277
  of UPDATE SET option 270
OFFPOSLIMIT option 219
OFFPOSMLM option 662
ON FAILURE option 210
ON MESSAGE option 209
ON n COLUMNS keyword of REBALANCE option 171
ONLY keyword of ANALYZE option 190
OPNDB2ID option 663
OPTIMIZED keyword of ALLOC option 283, 693
ORDER option 179
ORIGDISP option 664
ORIGINALDISP option 204
PART keyword of COPYLVL option 632
PART option 165
PAUSE keyword
  of ANALYZE option 190
  of UNLOAD option 193
PENDDDL option 664
PLAN option 665
PREFORMAT option 213, 665
RBALRSN_CONVERSION option 214
RCVICDDN option 665
RCVRPEND keyword of ALTRFAIL option 625
RCYDDN option 666
READERS keyword of FORCE option 236, 644
REBALANCE option 169
RECOVERYDDN option 260
RECOVERYICDDN option 263
REDEFINE option 198, 666
REFERENCE keyword of SHRLEVEL option 163
REFERENCE UNLOADONLY keyword of SHRLEVEL option 163
REGISTER keyword of COPY YES option 255
RELOAD keyword of UNLOAD option 193, 685
REMBFCPY keyword of DDTYPE option 278, 691
REMBICPY keyword of DDTYPE option 278, 691
REMPFCPY keyword of DDTYPE option 278, 691
REMPICPY keyword of DDTYPE option 278, 691
RENAME keyword
  of ORIGDISP option 664
  of ORIGINALDISP option 204
RENMMAX option 667
REORG option 160
REPORT keyword of BMCSTATS option 246
REPORTONLY keyword of FORCE option 238
REPORTONLY option 222
RESET keyword of MGEXTENT option 661
RETCODE keyword
  of ON FAILURE TERMINATE
    UTILITY option 212
  of ON MESSAGE CONTINUE
    UTILITY option 209
RETDPD option 305, 705
RETRY keyword
  of FORCE_AT option 236, 645
RETRY keyword of DSPLOCKS option 234, 641
RETRY option 233
RETRY_DELAY option 234
REUSE option 201
REXX keyword
  of DSRSEXIT option 228, 641
  of MAPTEXIT option 231, 659
  of TERMEXIT option 229, 680
RIDMAPMEM option 318
RIDMDSSZ option 667
RIDMAXD option 668
RMAPMEM option 669
ROUTCDE option 669
SAMPLE keyword of ANALYZE option 190
SCAN keyword of ANALYZE option 190
SCPYMAX option 670
SDUMP option 670
SELECT option 266
SHORTMEMORY option 184, 671
SHRLEVEL option 162
SIXSNAP option 238, 672
SIZEPCT option 293, 705
SMAX option 673
SMCORE option 674
SMS option 284, 706
SMSUNIT option 285, 706
SORTDATA option 179
SORTDEVT option 176, 674
SORTKEYS option 165
SORTNUM option 177, 675
SORTWORK keyword of DDTYPE option 278, 691
SPACE option 294
SPIILDSNP option 676
SPIILDSNPAT option 314
SPIILLSTORCLAS option 313
SPIILLUNIT option 313
SPIILSCLS option 677
SPIILLUNIT option 677
SQL keyword
  of DRAIN_WAIT option 232
  of DRNWAIT option 638
SQLDELAY option 678
SQLRETRY option 678
STAGEDSN option 678
STANDARD keyword of NLPCTFREE option 197
START keyword of FORCE_AT option 236, 645
STOP UTILITY keyword
  of ON FAILURE option 213
  of ON MESSAGE option 209
STOP@CMT option 678
STOPDELAY option 679
STOPRETRY option 679
STORCLAS option 306, 707
SUPSTART keyword of WTOMSG option 242
SYNC option 196
SYSPUNCH keyword of DDTYPE option 278, 691
TABLESPACE option 160
TAPEDISP option 679
TASKMAX option 680
TERM keyword
  of ALTRFAIL option 625
  of CPYRFAIL option 633
  of LONGLOG option 311, 657
  of TIMEOUT option 225, 681
TERMEXIT option 229, 680
TERMINATE UTILITY keyword of ON
  FAILURE option 211
TERSE keyword of DISPLAY option 589
THRESHLD option 295, 707
TIMEOUT option 224, 681
TOTALPAGEPCT option 183, 682
TSSAMPLEPCT option 250, 683
UBUFFS option 684
UNIT option 285, 709
UNITCNT option 286, 709
UNLDDN option 171, 684
UNLDMAX option 684
UNLOAD keyword of DDTYPE option 278, 691
UNLOAD option 192, 685
UNLOADONLY keyword of SHRLEVEL
  REFERENCE option 163
UPDATE keyword of ICTYPE option 255, 648
UPDATE option 270
UPDATEDB2STATS option 248
USE keyword of IFALLOC option 282, 702
UTIL keyword
  of DRAIN_ALLOC option 232
  of DRNWAIT option 638
UTILB_COLCCSID option 685
UTILB_NULLIX option 686
UXSTATE option 686
VERBOSE keyword of DISPLAY option 589
VOLCNT option 287, 710
WARN keyword of FILECHK option 644
labeled duration, specifying as part of a condition 277
large format sequential data sets 292, 700
LARGE keyword of DSNTYPE option 292, 700
large number of partitions, considerations for 95
LAST keyword of PART option 167
LASTRETRY keyword of FORCE_AT option 236, 645
LE C language sample DSNUEXIT user exit 777
LE COBOL language sample DSNUEXIT user exit 762
LE_C keyword of DSNUEXIT option 226
LE_COBOL keyword of DSNUEXIT option 226
LEAFDIST exception 129
LEAFDISTLIMIT option 221
LEAFDSLM option 654
LEAFFOFF exception 129
LEAFTOFF exception 129
LEVELINC exception 129
LEVELMIN exception 129
LEVELS exception 129
limit keys
backing out changes 625
changing 98, 116, 120
number of columns for 171
report of 349
supported data types 117
limit options
command 219, 221
determining which values to use 124
installation 654, 662
LIMIT value of TEMPLATE control statement 295, 707
LOAD control cards. See SYSPUNCH data sets
LOADPLUS utility, loading from SYSARC data set 175, 354
LOB data
BMCSYNC table considerations 729
considerations 107
DATE SQL function 137
DAY SQL function 137
how to reorganize 106, 655
inline 58, 137, 652
length parameter, in UPPER or LOWER SQL functions 137
locale, in UPPER or LOWER SQL functions 137
LOWER SQL function 137
MONTH SQL function 137
nested functions in an expression 137
RBA or LRSN format conversion 214
restart considerations 376
UPPER SQL function 137
YEAR SQL function 137
LOB indexes 106
LOB option 655
LOB table spaces
considerations 107
how to reorganize 106, 655
in catalog, reorganizing 100
RBA or LRSN format conversion 214
reorganizing when reorganizing the base table space 58, 244, 628
restarting 376
LOCBFCPY keyword of DSNTYPE option 278, 691
LOCBICPY keyword of DSNTYPE option 278, 691
LOCKROW option 655
locks, displaying 234, 641
LOCPFCPY keyword of DSNTYPE option 278, 691
LOCPICPY keyword of DSNTYPE option 278, 691
log apply control options
displaying values 589
hierarchy diagram 585
summary 585
using the MVS console to change 593
using Utility Monitor to view/change 589
log control task 568
LOG NO option 197
log range table 725
log records
application 569
defining 319, 320
performance considerations 516, 611
specifying logfinal start time 320
spill data set for 657
LOGAPPLY phase
description 569
primary function 40
restarting 382
specifying ON FAILURE 212
LOGFINAL option 320, 655
LOGFINAL phase
criteria for entering 583
defining time to end 309
defining time to start 320
restarting 382
specifying drain type 322, 637
logical page list status 69, 73
logical part REBUILD pending status 69, 73
LOGMEM option
command 319
installation 656
performance considerations 611
LOGSPIL option 657
LOGSPILL option 320
LOGTHRESHLD option 308
LOGTHRSCH option 657
long object name
ssyntax rules 143
long object names
truncating in messages 168, 658
longlog condition
benefits due to REORG PLUS technology 579
how determined 579
LONGLOG option
command 311
example 486
installation 657
LONGNAMETRUNC option 168, 658
LOWER SQL function 137
LPL status 69, 73
LRSN, converting format 214

M

macros for installation options 617, 689, 691
MAINT parameter of EXEC statement 330
maintaining common utility tables 717
management classes, SMS 306, 704
mapping table
specifying 60, 323
user exit for 808
mapping table user exit
description 808
example 812
restrictions 231, 659
specifying 231, 659
variables for 811
MAPPINGDATABASE option 323
MAPPINGTABLE option 323
MAPTEXIT option 231, 659
MAPTEXIT user exit
description 808
example 812
restrictions 231, 659
specifying 231, 659
variables for 811
variables passed 811
Mass del reorg exception 129
MAXCC, specifying in SYSIDCIN 358
MAXEXTSZ option 290, 703
maximum amount of sort memory 661
MAXNEWPARTS option
command 167
example 469
installation 659
MAXPARTITIONS, pending 98
MAXPRIME keyword of TEMPLATE control statement 290, 703
MAXRO option
command 308
example 477
installation 660
MAXSORTMEMORY option 186, 661
MAXSORTS option 178
MAXTAPE option
command 278
installation 661
reaching the limit 82
MAXVALUE keyword of SQL ALTER statement 26
MEMBER CLUSTER, universal table spaces defined as 23
MEMLIMIT system parameter 56, 540
memory
above the bar 56
above the line 182, 532, 628
below the line 532, 533
message level (MSGLEVEL) parameter
installation option 662
multitasking enabling 534
I/O bound phases 536
installation options 670, 680
multiple index data sets 538
phases that invoke sort 537
specifying 535
multitasking considerations 537
multivolume restrictions 352
MVS console, using with SHRLEVEL CHANGE 593
MVS enqueues, specifying 655

N

names of common utility tables, determining 719
names of objects, truncating in messages 168, 658
naming data sets
dynamically allocated 79
GDG names 79
original 90, 92
patterns for dynamic allocation 79
prefixes for ddnames 80
staging 90, 92
user exit 749

NEW parameter of EXEC statement 328
NEW/RESTART parameter of EXEC statement 328
NEW/RESTART(PHASE) parameter of EXEC statement 328
NLPCTFREE option 197
NOCEXEC option of the REXX compiler 809
node ID indexes 190
NOLIMIT value of MEMLIMIT system parameter 56, 326, 540, 609
non-key columns, indexes containing 22
NONE keyword
keywords

NONE keyword

of FORCE option 236
of DATACLAS option 695
of DEADLINE option 309, 635
of DRAIN_WAIT option 232
of DRNWAIT option 228, 641
of DSNUEXIT option 228, 641
of FORCE option 236, 644
of INDREFLM option 650
of LEAFDSDL option 654
of LOGFINAL option 320, 655
of MAPTEXIT option 231, 659
of MGMTCLAS option 704
of OFFPOSLIMIT option 219
of OFFPOSLM option 662
of RBALRSN_CONVERSION option 214
of REGISTER option 255
of SPILLSTORCLAS option 313
of SPILLUNIT option 313
of SPILSCLS option 677
of SPILUNIT option 677
of STORCLAS option 707
of TERMEXIT option 228, 680
NONE keyword of SHRLEVEL option 162
nonleaf index page, controlling the amount of free space 197
nonpartitioned indexes, using Instant Snapshot with indexes

nonpartitioned 110

XBM

Instant Snapshot function with
nonpartitioned indexes 110
nonpartitioned objects, considerations when redefining 788
NOSYSREC option 192
null keys 59, 686
NULL keyword

of predicate 277
of UPDATE SET option 270
NULLFILE DSN 83, 368
number of CPUs 547
numeric variables 298, 317
NUNIFORM exception 129

object availability, SHRLEVEL option 85, 162, 549
object names, long

syntax rules 143
truncating in messages 168, 658
object status 69, 72
OFFPOSLIMIT option 219
OFFPOSLM option 662
ON FAILURE option
description 210
example 456
with DELETEFILES option 212
ON MESSAGE option 209
ON n COLUMNS keyword of REBALANCE option 171
online reorganization. See SHRLEVEL CHANGE
operating system requirements 51
operators, comparison 274
operators, with predicate 274
OPNDB2ID

installation option 663
OPNDB2ID option
effects on RACF authorizations 53
optimal dynamic allocation method 283
OPTIMIZED keyword of ALLOC option 283, 693
options

command 160
copy 251
dynamic allocation 277
installation 617
processing of REORG PLUS 160
selective unload and update 264
SHRLEVEL CHANGE 307
statistics 245
options module. See installation options
ORDER option
description 179
example 443
memory usage 547
performance considerations 530, 547
ordering rows. See ORDER option
ordering STOGROUP volumes 789
ORIGDISP option 664
original data set
deleting and renaming 204
FASTSWITCH processing 205, 642
ORIGINALDISP option 204
overriding default ddnames
  BCPY 259
  BCPZ 259
  BICY 262
  BICZ 262
  BIRY 263
  BIRZ 263
  BRCY 260
  BRCZ 260
  SYSARC 174
  SYSIDCIN 201
  SYSREC 171
  SYSUT1 173

partition-by-growth table spaces
  allocated as DSSIZE 338, 602
  building a dictionary 136
  copies for 342
  defined as MEMBER CLUSTER 23
  example 469
  extension by DB2 115
  extension by REORG PLUS 114
  inline image copies 253
  invalid with REBALANCE 169
  LAST command option 167
  MAXNEWPARTS option 167, 659
  partition-level SYSRECs 363
  partitions added by REORG PLUS 114
  partitions copied 252
  rebalancing considerations 119
  restart considerations 376
  when partitions are added 555, 559, 570

partition-level copies 633
  naming data sets 341

partitioned objects, reorganizing
  examples 399, 416, 435
  I/O performance 525
  large number of partitions 95
  PART option 165
  table space memory considerations 547
  See also partition-by-growth table spaces

partitions, large number of 95

partition, data set name
  concatenation 302
  description 297
  installation option 695
  user exit for 639, 749
  variables for 750

partition, spill data set name
  concatenation 317
  description 314
  user exit for 639, 749
  variables for 750

PAUSE keyword
  of ANALYZE option 190
  of UNLOAD option 193

pausing after analysis 190

pausing after unloading data 193

Pct Dropped Rows exception 129

PctActivHi exception 129

PctActivLo exception 129

PCTCLUS exception 129

PCTDROP exception 129

PENDDDL option 664

partition rebalancing
  considerations 116
  DDLIN data set 120
  requirements 116
  using REBALANCE option 169

partial reorganization
  assigning copy data sets 633
  compressed indexes 377
  considerations 109
  document ID indexes 376
  falling back from FASTSWITCH to rename process 207
  REDEFINE 198
  secondary indexes 198
  SHRLLEVEL CHANGE 110, 568
  SHRLLEVEL NONE 69, 110
  SHRLLEVEL REFERENCE 74, 110
  SHRLLEVEL REFERENCE UNLOADONLY 74
  specifying copy data sets 633

partition number variable 298, 695

partition rebalancing
  considerations 116
  DDLIN data set 120
  requirements 116
  using REBALANCE option 169

PACTHI exception 129

PACTLO exception 129

page set REBUILD pending status 69, 73
parallel sort processes 531
parameters, EXEC statement 327
PART keyword of COPYLVL option 632

PART option
  considerations 109
  description 165
  example 416, 435

patterns, data set name
  concatenation 302
  description 297
  installation option 695
  user exit for 639, 749
  variables for 750

patterns, spill data set name
  concatenation 317
  description 314
  user exit for 639, 749
  variables for 750

partition, data set name
  concatenation 302
  description 297
  installation option 695
  user exit for 639, 749
  variables for 750
pending DDL changes 59, 98, 378, 664
pending statuses
  for SHRLEVEL NONE 69
  for SHRLEVEL REFERENCE, UNLOADONLY, and CHANGE 73
percent of table space pages to sample for statistics 250, 683
percentages of data set allocation size 293, 705
performance considerations
  dynamic allocation 529
  ERP environments 541
  full copy data sets 343
  general 515
  I/O 536, 539
  incremental copy data sets 602
  index work files 538
  inline image copies 540
  KEEPDICTIONARY option 515
  log records 611
  multiple data sets 534
  multitasking 534
  ORDER option 547
  processing phases 550
  sort 537
  spill data sets 613
  SYSTEM data sets 538
Performance for DB2 Databases solution 192, 217
persistent read only (PRO) status 69, 73
phases of REORG PLUS
  compared to DB2 REORG 35
  primary functions 39
  SHRLEVEL CHANGE 567, 572
  tuning considerations 550
PLAN option 665
plan, specifying product plan 665
  REORG PLUS 53
predicate
  evaluation 273
  option (with WHERE) 274
prefix variable, utility ID
  with DSNPAT option 298, 695
  with SPILSDSNP option 676
  with SPILLSDSNP option 314
prefix, specifying ddname
COPYDDN 259
  general considerations 80
  ICDDN 262
  RECOVERYDDN 260
  RECOVERYICDDN 263
  UNLDDN 171
  WORKDDN 173
PREFORMAT option
  command 213
  installation 665
PRO (persistent read only) status 69, 73
processing options of REORG PLUS 160
processing phases of REORG PLUS
  compared to DB2 REORG 35
  primary functions 39
  SHRLEVEL CHANGE 567
  tuning considerations 550
processor requirements 51
product changes 16
product plan, specifying 665
PSEUDODL exception 129
PSRBD status 69, 73
PUBLIC, granting to 53
publications, related 13
queueing, I/O, avoiding 539
RACF
  security exit 52
RACF (IBM Resource Access Control Facility)
  authority 53
random keys in indexes 59, 462, 653
range-partitioned table spaces
  defined as MEMBER CLUSTER 23
  rebalancing 161, 244
RBA, converting format 214
RBA, high-used
  ANALYZE HURBA option 522
  average row length, determining 522
  cardinality estimate 191, 522
  recovering objects 32
  REDEFINE NO option 200
  resetting 200
RBALRSN_CONVERSION option 214
RBDP (REBUILD pending) status 69, 73, 625
<table>
<thead>
<tr>
<th>RBDP* status 69, 73</th>
<th>REDEFINE option 198</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCVICDDN option 665</td>
<td>example 416</td>
</tr>
<tr>
<td>RCVRPEND keyword of ALTRFAIL option 625</td>
<td>installation 666</td>
</tr>
<tr>
<td>RCVYDDN option 666</td>
<td>performance considerations 515, 517</td>
</tr>
<tr>
<td>read claimers, canceling 236, 238, 644</td>
<td>requirements 88</td>
</tr>
<tr>
<td>read-only status</td>
<td>with data set redefine exit 789</td>
</tr>
<tr>
<td>and SHRLEVEL CHANGE 74</td>
<td>redefining data sets</td>
</tr>
<tr>
<td>read-or-replication-only (RREPL) status 59, 60, 69, 73, 264</td>
<td>ordering STOGROUP volumes 789</td>
</tr>
<tr>
<td>read/write status and SHRLEVEL CHANGE 74, 568</td>
<td>pausing a job after analysis 190</td>
</tr>
<tr>
<td>READERS keyword of FORCE option 236, 644</td>
<td>pausing a job after unloading data 193</td>
</tr>
<tr>
<td>real-time statistics 131</td>
<td>providing IDCAMS commands in SYSIDCIN 357</td>
</tr>
<tr>
<td>reallocation of data sets, by user 379</td>
<td>REDEFINE option, specifying 198, 666</td>
</tr>
<tr>
<td>REBALANCE option</td>
<td>requirements 88</td>
</tr>
<tr>
<td>command 169</td>
<td>resizing 787</td>
</tr>
<tr>
<td>initial status requirements 69, 73</td>
<td>selectively redefining 788</td>
</tr>
<tr>
<td>rebalancing partitions</td>
<td>STOGROUP-defined data sets 198</td>
</tr>
<tr>
<td>DDLIN data set 120, 344</td>
<td>SYSDICIN 357</td>
</tr>
<tr>
<td>initial status requirements 69, 73</td>
<td>VCAT-defined data sets 198, 357</td>
</tr>
<tr>
<td>REBALANCE option 121, 169</td>
<td>with user exit 789</td>
</tr>
<tr>
<td>requirements and considerations 115</td>
<td>REFERENCE keyword of SHRLEVEL option 163</td>
</tr>
<tr>
<td>REBUILD pending (RBDP) status 69, 73, 625</td>
<td>REFERENCE UNLOADONLY keyword of SHRLEVEL option 163</td>
</tr>
<tr>
<td>recopied data sets</td>
<td>referential integrity considerations</td>
</tr>
<tr>
<td>considerations 377</td>
<td>deleting data during reorganization 268</td>
</tr>
<tr>
<td>restarting 377</td>
<td>general issues 139</td>
</tr>
<tr>
<td>RECOVER pending (RECP) status</td>
<td>selecting data for unload/reload 266</td>
</tr>
<tr>
<td>after limit key update failure 625</td>
<td>updating value in a foreign key column 270</td>
</tr>
<tr>
<td>restrictions and usage with failures 382</td>
<td>REFP status 69, 73</td>
</tr>
<tr>
<td>with SHRLEVEL NONE 69</td>
<td>refresh pending status 69, 73</td>
</tr>
<tr>
<td>with SHRLEVEL REFERENCE, UNLOADONLY, and CHANGE 73</td>
<td>REGION parameter 326, 542</td>
</tr>
<tr>
<td>recovery</td>
<td>buffers 540</td>
</tr>
<tr>
<td>after terminating a DB2 object 380</td>
<td>REGISTER keyword of COPY YES option</td>
</tr>
<tr>
<td>considerations when not making copies 139</td>
<td>corresponding image copy types (incremental) 602</td>
</tr>
<tr>
<td>during SHRLEVEL CHANGE 607</td>
<td>description 255</td>
</tr>
<tr>
<td>during UTILTERM phase 384</td>
<td>example 451</td>
</tr>
<tr>
<td>from an abnormal termination 210</td>
<td>registering copy data sets 255, 338</td>
</tr>
<tr>
<td>of reorganized table space 139</td>
<td>related publications 13</td>
</tr>
<tr>
<td>RECOVERYDDN option</td>
<td>RELOAD keyword of UNLOAD option 193, 685</td>
</tr>
<tr>
<td>command 260</td>
<td>RELOAD phase</td>
</tr>
<tr>
<td>example 451</td>
<td>check for available resources 555</td>
</tr>
<tr>
<td>RECOVERYICDDN option 263</td>
<td>functions with SHRLEVEL CHANGE 569</td>
</tr>
<tr>
<td>RECP (RECOVER pending) status 69, 73, 625</td>
<td>primary functions 40</td>
</tr>
<tr>
<td>REDEFINE option</td>
<td>specifying ON FAILURE 212</td>
</tr>
<tr>
<td></td>
<td>tuning considerations 555</td>
</tr>
<tr>
<td></td>
<td>REMBFCPY keyword of DDTYPE option 278, 691</td>
</tr>
<tr>
<td></td>
<td>REMBICPY keyword of DDTYPE option 278, 691</td>
</tr>
<tr>
<td></td>
<td>REMPFCPY keyword of DDTYPE option 278, 691</td>
</tr>
<tr>
<td></td>
<td>REMPICPY keyword of DDTYPE option 278, 691</td>
</tr>
</tbody>
</table>
rename and delete process for staging data sets
backing out after failure in UTILTERM 386, 625
canceling during 380, 608
completing manually after failure in
UTILTERM 384
description 90, 92
failure during 384
falling back from FASTSWITCH process 207, 646
multitasking 536
object status following 74
threshold for 208, 515, 647
RENAME keyword
of ORIGDISP option 664
of ORIGINALDISP option 204
rename process for original data sets
description 90
multitasking 536
RENNMMAX option 667
reordered row format 138
REORG option 160
REORG pending (REORP) status 119, 244
REORG phase
general considerations 133
multitasking 537
primary functions 40
restarting 382
tuning considerations 559
REORG type variable
with DSNPAT option 298, 695
with SPIILDSPN option 676
with SPIILDSPNAT option 314
REORG utility of IBM
invoking DSNUTILB to use 57, 223, 640
requirement for 51
STATISTICS option 248
REORG_MAPPING_DATABASE DB2 subsystem
parameter 323
REORMDEL exception 129
REORMMODS exception 129
REORP (REORG pending) status 119, 244
REORPEND exception 129
REORSPAC exception 129
report
of applied fixes 330
of canceled threads 237, 336, 646
of SQL ALTER statements 349
statistics 245, 246
REPORT keyword of BMCSTATS option 246
REPORTONLY keyword of FORCE option 238
REPORTONLY option 222
requirements, system and software 51
RESET keyword of MGEXTENT option 661
resetting a column value to a constant 270
resetting DASD MANAGER PLUS exceptions 127
resizing data sets 787, 794
resources, check for in RELOAD phase 555
RESTART parameter of EXEC statement 328
restart pending status 69, 73
RESTART(PHASE) parameter of EXEC statement
328
restarting
ACTIVE option considerations 83
after inadequate space failure 379
compressed indexes considerations 377
considerations 375
data sharing considerations 378
DELETEFILES option considerations 379
document ID indexes considerations 376
during UTILTERM phase 380, 608
dynamic allocation considerations 83, 374, 379
from failure 374
inline image copies considerations 379
LOB table spaces considerations 376
paused job, example 439
pending DDL changes considerations 378
prior to UTILTERM for SHRLEVEL CHANGE 607
RESTART parameters 328
restart parameters of EXEC statement 328
restrictions 375
statistics considerations 378, 608
work files considerations 379
XML table spaces considerations 376
RESTP status 69, 73
restrictive statuses
for SHRLEVEL NONE 69
for SHRLEVEL REFERENCE, UNLOADONLY, and CHANGE 73
RETCODE keyword
of ON FAILURE TERMINATE UTILITY option 212
of ON MESSAGE CONTINUE UTILITY option 209
retention period for certain data sets 305, 705
RETID option 305, 705
RETPD option 305
RETRY keyword
of FORCE_AT option 236, 645
RETRY keyword of DSPLOCKS option 234, 641
RETRY option 233
RETRY parameter of DYNALOC option 711
RETRY_DELAY option 234
retrying drain 233, 638
retrying dynamic allocation of SORTWK 711
return codes
  for DSNUEXIT user exit 750
  retrying SQL -911 678
  REXX exits 784
  specifying with ON FAILURE 210, 212
  specifying with ON MESSAGE 209
  specifying with TIMEOUT 224, 681
REUSE option 201
REXX exits
  accessing the source 748
  common variables 784
  data set redefine
    description 787
    example 798, 804
    installation option 641
DSRSEXIT
  description 787
  example 798, 804
  installation option 641
mapping table creation 808
MAPTEXIT
  description 808
  example 812
  invoking 231, 659
restrictions 784
return codes 784
sample library 748
setting up 790, 802, 810
TERMEXIT 801
REXX keyword
  of DSRSEXIT option 228, 641
  of MAPTEXIT option 231, 659
  of TERMEXIT option 229, 680
RID translation map
  benefits due to REORG PLUS technology 580
  defining memory for 318
  description 568
  restricting memory for 610
RIDMAPMEM option
  command 318
  restricting size 667, 668
RIDMDSSZ option 667
RIDMMAXD option 668
RMAPMEM option 668
RORGMAX option 537, 669
ROUTCDE option 669
row change time stamp column 109, 270
row format 138
row length, determining average 187
row-level security 142
ROWS/KEY exception 129
RREPL (read-or-replication-only) status 59, 60, 69, 73, 264
rules for constants 275
running BMC utilities concurrently 67, 732
RUNSTATS cardinality changes 248

S
SAMPLE keyword of ANALYZE option 190
samples. See examples
sampling
  cardinality 190, 520
  for statistics 250, 683
SAS/C runtime library 331
SCAN keyword of ANALYZE option 190
scanning, controlling 190
SCC (DB2 Solution Common Code)
  description 49
  load library 331
SCC (DB2 Solutions Common Code)
  version requirement 51
SCPYMAX option 670
SDUMP option 670
secondary extents 661
secondary indexes
  multitasking 536
  REDEFINE 198
secondary templates, switching to 295, 707
security
  exit 52
  REORG PLUS 53
  requirements 52
  row- and column-level 142
  verification 52
SEGSIZE, pending 98
SELECT option
  example 451
  referential integrity issues 139
  SHRLEVEL CHANGE considerations 596
  specification information 266
SELECT privileges 53
selecting data, specifying conditions 266
selective unload and update options 264
selectively redefining VSAM data sets 788
Send New command of XBM 589
sequential data sets, extended
  allocating 292, 700
  SORTWK restriction 352
  SYSREC restriction 365
serialization issues 66, 595
shadow data sets. See staging data sets
SHORTMEMORY option 184, 671
SHRLEVEL 726
SHRLEVEL CHANGE
  altering dynamically 589
  batch applications using AR/CTL 599, 626
  benefits 30
  change block options 307
  command option 164
  concurrency issues and restrictions 595
  conditional reorganization 124
  considerations when using SELECT or DELETE 596
  controlling the log apply process 582, 585
  copy data sets 601
  criteria for ending 583
  criteria for entering LOGFINAL 583
  data capture flag 568, 570
  DB2 resources used 579
  description 87
  diagram of log apply control options 585
  diagrams of phases 572
  displaying job information 589
  examples 477, 486, 494
  execution phases, description 567, 572
  failure during 625
  flow diagram 566
  incompatible options 596
  incremental copy data sets 602
  information displayed by XBM 589
  initial object status 73
  options that control 583
  overview 566
  recovery 607
  restart 607
  restrictive statuses 73
  specifying drain type 322, 637
  staging data sets 87
  using XBM to control execution 589, 593
SHRLEVEL NONE
  command option 162
  initial object status 69
  restrictive statuses 69
  status changes during reorganization 69
SHRLEVEL option
  considerations for use 85, 549
  description 162
SHRLEVEL REFERENCE
  command option 163
  description 87
  examples 399, 416
  failure during 625
  initial object status 73
  restrictive statuses 73
  staging data sets 87
  status changes during reorganization 74
  terminating job 380
SHRLEVEL REFERENCE UNLOADONLY
  command option 163
  initial object status 73
  restrictive statuses 73
  status changes during reorganization 74
single-phase reorganization
  diagrams 39
  issues 133
  performance considerations 550, 559
SHRLEVEL CHANGE diagrams 572
  specifying 193, 685
  SYSREC considerations 363, 525
  SYSUT1 considerations 526
SIXSNAP option
  command 238
  installation 672
SIXSNAP processing
  DASD vendor considerations 112
  handling failures 111
  with multi-data-set objects 97
SIZEPCT option 293, 705
  sliding-scale calculation for secondary extents 661
SMAX option
  controlling BMCSORT 533
  description 673
  performance considerations 547
  used with MAXSORTS 178
SMCORE option
  controlling BMCSORT 531
  description 674
SMS (IBM Storage Management Subsystem)
  classes 306, 695
  dynamic data set allocation 284, 706
  SMS extended sequential data sets
allocating 292, 700
SORTWK restriction 352
SYSREC restriction 365
SMS option 284, 706
SMSUNIT option 285, 706
snapshot copy 516
SNAPSHOT UPGRADE FEATURE (SUF)
description 140
for nonpartitioning indexes 141
integrating with other BMC products 51
requirements 111
using to enable zIIP processing 140, 240, 688
using with SHRLEVEL CHANGE option 568
software compression. See compression software requirements 51
Solution Common Code. See SCC
sort processing
concurrent processes 531
controlling 530
multitasking 537
sort processing options
AVAILPAGEPCT 182, 628
MAXSORTMEMORY 186, 661
MAXSORTS 178
MINSORTMEMORY 186, 662
SHORTMEMORY 184, 671
SMAX 673
SMCORE 674
SORTDATA 179
SORTDEVT 674
SORTKEYS 165
SORTNUM 177, 547, 675
sort work data set. See SORTWK data set
SORTDATA option
description 179
example 456
SORTDEVT option 176, 674
sorting data, memory usage 547
SORTKEYS option 165
SORTNUM option
command 177
example 462
installation 675
turning on BMCSORT 547
SORTWK data sets
allocating 349
compression and ANALYZE option 373
deleting 202
description 47
dynamic allocation
description 77
SORTDEVT option 176
syntax 277
with SORTNUM 177
performance information 530
size estimate provided by ANALYZE 370
specification and usage 349
SORTWK data sets dynamic allocation
with SORTNUM 675
SORTWORK keyword of DDTYPE option 278, 691
space
allocations 339
failure due to inadequate 379
key compression of nonleaf pages in unload phase 193
specifying for allocation 294
SPACE exception 129
SPACE option 294
specifying installation options module 331
SPILDSNP option 516, 676
spill data sets
calculating size 613
description of options 596
naming 314
pattern
concatenation 317
spill data sets pattern
description 314
variables for 750
SPILDSNPAT option 314, 518
SPILLSTORCLS option 313, 518
SPILLUNIT option 313, 518
SPILLSCLS option 516, 677
SPILLUNIT option 516, 677
SQL -911 return code, retrying 678
SQL functions, indexes created on expressions that contain 137
SQL keyword
of DRAIN_WAIT option 232
of DRNWAIT option 638
SQL statements
altering data set sizes 787
altering limit keys 115, 344
altering objects in DDLIN data set 344
deleting rows from the BMCDICT table 721
deleting rows from the BMCHIST table 725
deleting rows from the BMCSYNC table 731, 739
deleting rows from the BMCSYNC table for RECOVER UNLOADKEYS 731
deleting rows from the BMCHIST table 721
querying BMCHIST table 719
querying BMCSYNC table 719
reporting ALTER statements in DDLOUT data set 349
terminating BMC utilities 719
SQLDELAY option 678
SQLRETRY option 678
SSID (DB2 subsystem identifier) parameter of EXEC statement 327
STAGEDSN option
    and rename process 90
    authority needed 53
    data set rules 541
    description 678
    performance consideration 515
staging data sets
authorization 549
canceling or terminating job during UTILTERM 380, 608
defining with IDCAMS commands in SYSIDCIN 357
description 87
example 399, 416
failure during switching of data sets 384
falling back from FASTSWITCH to rename process 207, 646
FASTSWITCH process 92
naming conventions 90, 92
preallocation requirements 88, 200
rename and delete process 90, 92
SHRLEVEL CHANGE used in RELOAD phase 569
    threshold for determining switch process 208, 515, 647
STANDARD keyword of NLPCTFREE option 197
START keyword of FORCE_AT option 236, 645
bypassing 520
DASD MANAGER PLUS 192, 246, 520
DB2 catalog updates 248
    options 245
output data set, optional 335
    real-time 131
report output 245, 246
restart considerations 378, 608
    sampling 250, 683
    with DSNUTILB reorganization 131, 248
STATISTICS option of IBM DB2 REORG utility 248
status requirements of objects for reorganization 69, 72
status, BMC utilities 719
STEP name variable
    with DSNPAT option 298, 695
    with SPIILDSNP option 676
    with SPIILLDSNPAT option 314
STEPLIB DD statement 331, 809
STOGROUP-defined data sets
    ordering through user exit 789
    redefining data sets during reorganization 198
    requirements 88
    using DSNZPARM values 199
STOP status
    read-only status
        and SHRLEVEL REFERENCE 35
        and SHRLEVEL REFERENCE UNLOADONLY 35
    EDITPROCs
        DB2 REORG comparison 35
        LOG YES option 35
        REORP (REORG pending) status 35
        STOP status 35
STOP UTILITY keyword
    of ON FAILURE option 213
    of ON MESSAGE option 209
STOP@CMT option 678
STOPDELAY option 679
STOPRETRY option 679
storage above the bar 56
storage classes, SMS 306, 707
Storage Management Subsystem. See SMS
storage, virtual 56
STORCLAS option 306, 707
striped data sets 352
subset of partitions in single image copy 633
SUBSTR SQL function 137
subsystem ID parameter of EXEC statement 327
subsystem ID variable
  with DSNPAT option 298, 695
  with SPILDSNP option 676
  with SPILLDSNPAT option 314
subsystem ID, XBM 241, 687
SUF. See SNAPSHOT UPGRADE FEATURE
SUF (SNAPSHOT UPGRADE FEATURE)
  requirement for REORG PLUS 51
  See also individual entries for XBM and SUF
suffix variable, utility ID
  with DSNPAT option 298, 695
  with SPILDSNP option 676
  with SPILLDSNPAT option 314
summary of changes 16
supported data types in limit keys 117
supported DB2 versions 51
suspend and resume interface, with AR/CTL 599
switch process for staging data sets
  description 92
  falling back after failure 207
  FASTSWITCH option 205, 642
  FSFALLBACK option 207, 646
  FSTHRESHOLD option 208, 647
  threshold for 208, 515, 647
switching, template 295, 707
symbolic variables
  DSN pattern 298, 695
  SPIILDSNP pattern 676
  SPIILLDSN pattern 314
SYNC option 196
syntax
  alphabetical listing of command options 144
  command option descriptions 160
  diagrams 148
  rules 143
SYSALLDA unit name 709
SYSARC data sets
changing the default ddname 626
description 353
determining size with ARCROWS 216
dynamic allocation
  DDTYPE option 278
  description 77
  limiting GDG versions 305
  specifying expiration date 304
  specifying retention period 305
  using GDG names 79
example 451
format of, specifying 175, 627
overriding default ddname 174
specification and usage 353
with partition rebalancing 116
SYSCOPY table
  duplicate entries 255
  registering copies in 100, 255
  registration information 338, 602
SYSCOPY table space, reorganizing 100
SYSERR data sets, specification and usage 356
SYSEXEC data sets, specification and usage 357
SYSIBM.SYSCOLUMNNS table 248
SYSIBM.SYSCOPY table
  duplicate entries 255
  registering copies in 100, 255
  registration information 338, 602
SYSIBM.SYSFIELDS table 246
SYSIBM.SYSINDEXSPACESTATS real-time
  statistics table 131
SYSIBM.SYSEQUENCES table 53, 242, 650
SYSIBM.SYSEQUENCESDEP table 53
SYSIBM.SYSTABLESPACESTATS real-time
  statistics table 131
SYSIBM.SYXMLRELS table 53
SYSIDCIN data sets
  description 48
  example 392
  overriding default ddname 201
  specification and usage 357
  when required 88
SYSIDCIN DD statement 357
SYSIN data sets
  description 48
  specification and usage 360
SYSPRIN2 data sets
  changing message levels 331
  description 48
  specification and usage 361
worklist environment considerations 361
SYSPRINT data sets
  changing message levels 331
  description 48
  example REORG PLUS job output 389
  specification and usage 360
SYSPUNCH data sets
  allocating 278
  description 48
  specification and usage 361
SYSPUNCH keyword of DDTYPE option 278, 691
SYSREC data sets
  controlling buffer usage 525
  DD DUMMY 83
  deleting automatically after job 202
  description 48
  DSN=NULLFILE 83
  dynamic allocation
    DDTYPE option 278
    description 77
    syntax 277
  multiple 48, 525, 538
  overriding default ddname 171
  partition rebalancing considerations 365
  partition-level for partition-by-growth table spaces 363
  performance considerations 538
  setting default 684
  single-phase issues 133
  size estimate provided by ANALYZE 370
  SMS extended sequential data sets 365
  specification and usage 366
  worklist environment considerations 365
system authorization
  verification 52
system dump
  excluding abend codes 642
  generating 670
system period temporal tables 21, 264
system requirements 51
system setup 51
SYSTERM data sets, specification and usage 366
SYSTSIPT table space, reorganizing 100
SYSTSPRT data sets, specification and usage 366
SYSUT1 data sets
  allocating 335
  controlling buffer usage 526
  DD DUMMY 83, 368
  deleting automatically after job 202
  description 48
  DSN=NULLFILE 83, 368
  dynamic allocation
    DDTYPE option 278
    description 77
    syntax 277
  XML table space reorganization 190
  multiple 538
  overriding default ddname 173, 686
  performance considerations 538
  size estimate provided by ANALYZE 370
  SMS extended sequential data sets 368
  specification and usage 366
  tuning considerations 538
  worklist environment considerations 368
  XML table space reorganization 103

T
  table names, determining 719
  table space compression
    description 134
    KEEPDICTIONARY option 194, 653
    restrictions with partition rebalancing 135
  table space reorganization
    performing conditionally 124
    status requirements 69, 72
  table spaces
archive-enabled 344
catalog, reorganizing 100
defined as MEMBER CLUSTER 23
LOB 104, 214, 376
multi-table 61
name variable
with DSNPAT option 298, 695
with SPIILDSNP option 676
with SPIILDDSNPAT option 314
partition-by-growth 119
partitioned, large number of partitions 547
range-partitioned 23, 161
recoverability 139
SHRLEVEL CHANGE access to objects 74
specifying for reorganization 160
type conversion 98
universal 23
XML 103
table-controlled partitioning 354, 494
tables
archive enabled 264
clone 161
mapping 60, 231, 323
organized by hash 58, 243, 648
temporal 21, 264
tables, BMC
backing up 717
BMCHIST 722
BMCLGRNX 725
BMCSYNC 41, 726
BMCTRANS 734
BMCUUTIL 41, 736
BMXCOPY 739
common utility 95
considerations 717
determining names 719
querying 719
warnings 717
tables, BMCDICT 720
TABLESPACE option 160
tape
changing work file disposition 679
devices, number of 82, 278, 661
with MAXTAPE 82
TAPEDISP option 679
TASKMAX option 680
tasks
performed by REORG PLUS 32
specifying maximum for sort 178
TEMPLATE control statements 62, 295, 707
template switching 295, 707
temporal tables 21, 264
temporary data sets
definition 83
how REORG PLUS handles 83
TERM keyword
of ALTRFAIL option 625
of CPYRFAIL option 633
of LONGLOG option 311, 657
of TIMEOUT option 225, 681
TERM parameter 380
TERM parameter of EXEC statement 328
TERMEXIT option 229, 680
TERMEXIT user exit
description 801
invoking 802
specifying 680
using 801
variables passed 803
TERMINATE UTILITY keyword of ON FAILURE option
DELETETFILES option processing 202
description 211
terminating
DB2 object, recovery after 380
or canceling a job 380
SHRLEVEL CHANGE reorganization 583
terminating BMC utilities 719
TERSE keyword of DISPLAY option 589
threads, canceling
authorizations required 53
FORCE option 235, 644
FORCE_AT option 645
FORCE_RPT option 646
reporting 237, 336, 646
specifying when to cancel 236, 645
threads, number used 57
THRESHLD option 295, 707
threshold for secondary values 81
threshold for determining switch process
performance recommendation 515
specifying 208, 647
threshold for secondary values 295, 707
time
specifying with DEADLINE 309
specifying with LOGFINAL 320
time rules for constants 275
time variable, current
with DSNPAT option 298, 695
with SPILDSNP option 676
with SPILLDSNPAT option 314
time variable, short form
with DSNPAT option 298, 695
with SPILDSNP option 676
with SPILLDSNPAT option 314
time zone, timestamp
data type 117
implicit time zone 266
with CURRENT TIMESTAMP keyword of predicate 277
with DEADLINE 309
with LOGFINAL 320
with UPDATE SET CURRENT TIMESTAMP 270
timeout
displaying locks 234, 641
specifying the number of retry attempts 233, 638
specifying the time between retry attempts 234, 638
specifying the value for 232, 638
TIMEOUT option
cmd 224
installation 681
with DELETEFILES processing 202
timeout, transactions retry after 542
timestamp
specifying with DEADLINE 309
specifying with LOGFINAL 320
with time zone
data type 117
implicit time zone 266
with CURRENT TIMESTAMP keyword of predicate 277
with DEADLINE 309
with LOGFINAL 320
with UPDATE SET CURRENT TIMESTAMP 270
TIMESTAMP WITH TIME ZONE, defined on timestamp columns 117
tokens in syntax, rules for 143
Top Secret. See CA Top Secret
TOTALIND exception 129
TOTALOFF exception 129
TOTALPAGEPCT option 183, 682
TRACE authority 53
translation of command constants 265
triggers
advisory pending 129
append inserts 129
generating a reorganization 127
truncating long names in messages 168, 658
TSSAMPLEPCT option 250, 683
tuning REORG PLUS
dynamic allocation of data sets 529
I/O processing 539
improving memory use 547
installation options 515
memory use 547
multitasking 534
phases 550
two-phase reorganization
diagrams 39
example 451
SHRLEVEL CHANGE diagrams 572
specifying 192, 685

U

UBUFFS option 684
UNCLUST exception 129
Unclust inserts exception 129
underutilized pages 183, 682
Unicode data 142, 143
unit names, for dynamic allocation 285, 709
UNIT option
cmd 285
installation 709
with SMSUNIT option 82, 285
UNITCNT option 286, 709
description 684
unload data sets. See SYSREC data sets
UNLOAD keyword of DDTYPE option 278, 691
UNLOAD option
cmd 192
CONTINUE example 451
installation 685
PAUSE example 435
with REDEFINE 193
UNLOAD phase
functions with SHRLEVEL CHANGE 568
multitasking 537
ORDER option 547
partial reorganization 198
pausing and continuing after unloading data 193
primary functions 40
restarting 382
specifying ON FAILURE 212
tuning considerations 552
unload work files. See SYSREC data sets
UNLOADKEYS entries, cleaning up 731
UNLOADONLY keyword of SHRLEVEL
REFERENCE option 163
unsupported data types in limit keys 117
unsupported features of DB2 142
unused pages 182
UPDATE keyword of ICTYPE option 255, 648
UPDATE option
description 270
example 399
referential integrity issues 139
UPDATEDB2 Common Statistics component
option 248
UPDATEDB2STATS option
description 248
example 392, 462
with DSNUTILB reorganization 248
UPPER SQL function 137
USE keyword of IFALLOC option 282, 702
user exits
accessing the source 748
authorizations for 53, 789
controlling BMCHIST update 804
controlling statistics updates 804
creating mapping objects 808
data set name pattern
assembler example 751
C example 770
COBOL II example 762
description 749
invoking 639
LE COBOL example 762
LE_C example 777
variables 298, 314
data set redefine
description 787
example 798
invoking 641
DB2 139, 534
dropping mapping objects 808
DSNUEXIT
assembler example 751
C example 770
COBOL II example 762
description 749
invoking 639
LE COBOL example 762
LE_C example 777
DSRSEXIT
description 787
example 798
invoking 641
list of exits provided 747
mapping table
description 808
example 812
invoking 231, 659
MAPTEXIT
description 808
example 812
invoking 231, 659
REXX
description 808
example 812
invoking 231, 659
TERMEXIT
description 808
example 812
invoking 231, 659
sample library 748
TERMEXIT
controlling BMCHIST update 801
controlling statistics updates 801
description 801
example 804
user ID variable
  with DSNPAT option 298, 695
  with SPILDSNP option 676
  with SPILLDSNPAT option 314
user-defined variables 298, 750
USERID (user identifier) parameter of EXEC statement 330
UTIL keyword
  of DRAIN_WAIT option 232
  of DRNWAIT option 638
UTILB_COLCCSID option 685
UTILB_NULLIX option 686
UTILID (utility identifier) parameter of EXEC statement 328
UTILINIT phase
  primary functions 41, 196, 629
  restarting 382
  specifying ON FAILURE 211
utility ID prefix variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPIILDSNPAT option 314
utility ID suffix variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPIILDSNPAT option 314
utility ID variable
  with DSNPAT option 298, 695
  with SPIILDSNP option 676
  with SPIILDSNPAT option 314
utility identifier (UTILID) parameter of EXEC statement 328
Utility Monitor
  altering a SHRLEVEL CHANGE job 589
  monitoring a job 589
utility parameters, EXEC statement 327
utility restrictive state, read-only access allowed (UTRO) 69, 73
utility restrictive state, read/write access allowed (UTRW) 69, 73
utility restrictive state, utility exclusive control (UTUT) 69, 73
utility status, restrictions and usage with SHRLEVEL CHANGE 74
utility timeout value in DSNZPARMs 232, 638

UTILITY_OBJECT_CONVERSION DB2 subsystem parameter 214
UTILTERM phase
  backing out after failure 386, 625
  completing manually after failure 384
  failure involving DDLIN processing 625
  primary functions 41, 196, 629
  recovery during 384
  restart during 382, 608
  specifying ON FAILURE 212
  unable to restart 384
UTILTERM phase for SHRLEVEL CHANGE 571
UTPRINT data set
  description 49
  specification and usage 369
UTRO (utility restrictive state, read-only access allowed) 69, 73
UTRW (utility restrictive state, read/write access allowed) 69, 73
UTUT (utility restrictive state, utility exclusive control) 69, 73
UXSTATE option 534, 686

V

VALIDPROCs
  libraries that contain 331
  row format conversion 138
variable control interval (CI) size 136
variables
  for DSN pattern
    concatenation 302
    description 695
    numeric 298
    user-defined 750
  for SPIILDSNP pattern
    concatenation 317
    description 314, 676
    numeric 317
    user-defined 750
  passed to DSRSEXIT user exit 791
  passed to MAPTEXIT user exit 811
  passed to TERMEXIT user exit 803
  REXX user exits, common for 784
variables for DSN pattern
  description 298
VCAT-defined data sets
  redefining during reorganization 198, 357
  requirements 88
VCAT-defined table space, example 392
VCATNAME variable
  with DSNPAT option 298, 695
  with SPILDSNP option 676
  with SPILDSNPAT option 314
VERBOSE keyword of DISPLAY option 589
  verifying
  authorization 52
  versioned indexes
    compressed 25
    in SYSTSIPT table space 100
versions of components, required 51
versions of DB2
  supported 51
  when different on data sharing group 327
versions, XML 21, 214
VIO data sets, restrictions 352
virtual storage
  above the bar 56
  above the line 540
  and MEMLIMIT 56
  below the line 540
  providing maximum 540
VOLCNT option 287, 710
volume count 287
VSAM buffers and the 16-megabyte line 525
VSAM data sets
  FASTSWITCH process 92
  ordering STOGROUP volumes 789
  redefining during reorganization 199
  redefining through a user exit 787
  rename and delete process 90
  resizing 787
  selectively redefining 788

W
WARN keyword of FILECHK option 644
WBUFFS option 686
WEPR status 69, 73
WHERE option
  DELETE usage 268
  description 273
  example 451
  SELECT usage 266
  UPDATE usage 270
WITH TIME ZONE keyword of CURRENT TIMESTAMP keyword of predicate 277
of UPDATE SET CURRENT TIMESTAMP option 270
work files
  deleting 81, 202
  dynamic allocation 176, 634
  multiple 534
  restart considerations 379
  sort (SORTWK) 349
  See also dynamic allocation of data sets
WORK keyword of DDTYPE option 278, 691
WORKDDN option 173, 686
worklist environment
  dynamic data set allocation 79, 280
  multiple reorganization considerations 365
  restriction when deleting files 636
  SMS extended sequential data sets 365
  viewing utility output 361
WORKUNIT option 687
write claimers, canceling 238, 644
write page error status 69, 73
WRITERS keyword
  of DRAIN option 322
  of DRAINTYP option 637
WTO messages, specifying routing code 669
WTOMSG option 242

X
XBM
data sharing 241
description for SHRLEVEL CHANGE 141
for nonpartitioning indexes 141
information displayed for SHRLEVEL CHANGE 589
initialization 568
requirements 111
Send New command 589
using the MVS console 593
using the Utility Monitor 589
using to control a SHRLEVEL CHANGE reorganization 589, 593
using to enable zIIP processing 140, 240, 688
XBMGROUP name 241
XBMID option
  command 241
  example 477
  installation 687
XML data
   BMCSYNC table considerations 729
   considerations when reorganizing 102
   document ID column 102
   restrictions on partition rebalancing 102
XML data, authorizations 53
XML indexes 103
XML indexes, authorizations for reorganizing 53
XML table spaces
   ANALYZE option 103, 187
   considerations when reorganizing 103
   restrictions on ANALYZE SCAN 190
   SYSUT1 data sets for reorganization 190
XML versions 21, 214

Y

YEAR SQL function 137

Z

z Integrated Information Processor (zIIP). See zIIP
ZIIP option 240, 688
zIIP processing
   enabling 140
   option to enable or disable 240, 688
REORG PLUS requirements 51
requirements 51