REORG PLUS *for DB2*®
Reference Manual

Supporting

Version 10.1 of REORG PLUS for DB2
Version 10.1 of Database Performance *for DB2*

April 2011

www.bmc.com

Contacting BMC Software

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

United States and Canada

Address	BMC SOFTWARE INC 2101 CITYWEST BLVD HOUSTON TX 77042-2827 USA	**Telephone**	713 918 8800 or 800 841 2031	**Fax**	713 918 8000

Outside United States and Canada

Telephone	(01) 713 918 8800	**Fax**	(01) 713 918 8000

Restricted rights legend

Customer support

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

Support website

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

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

Support by telephone or e-mail

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

Before contacting BMC

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

- product information
  - — product name
  - — product version (release number)
  - — license number and password (trial or permanent)
- operating system and environment information
  - — machine type
  - — operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - — system hardware configuration
  - — serial numbers
  - — related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the issue
- commands and options that you used
- messages received (and the time and date that you received them)
  - — product error messages
  - — messages from the operating system, such as `file system full`
  - — messages from related software

License key and password information

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

- Send an e-mail message to customer_support@bmc.com.
- Use the Customer Support website at http://www.bmc.com/support.

Contents

Chapter 5 Examples of REORG PLUS jobs 373

Chapter 6 Tuning REORG PLUS jobs 521

Chapter 7 Online reorganization 569

Figures

Tables

About this book

This book contains detailed information about the REORG PLUS for DB2® product and is intended for 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
- 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 Customer Support website at http://www.bmc.com/support. Most product shipments also include the books on a documentation CD.

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

The following related publications supplement this book.

Category	Document	Description
installation and customization	*Utility Products for DB2 Installation Guide*	provides information about installing and customizing REORG PLUS and other BMC utilities for DB2
	BMC Products and Solutions for DB2 for z/OS Installation Planning Guide	contains information about BMC products and solutions for DB2, helping you understand the relationship between the products and solutions and plan their installation in your environment
	online Help panels for Installation System panels	provide information for fields that are required when installing the REORG PLUS product
core documents	*Utility Products for DB2 Messages Manual*	provides message and return code information for the BMCDSN Command Processor, CHECK PLUS *for DB2*, LOADPLUS *for DB2*, REORG PLUS, and UNLOAD PLUS *for DB2* products
documents for associated products and technology components	*APPLICATION RESTART CONTROL User Guide*	provides information about the APPLICATION RESTART CONTROL (AR/CTL) product
	DASD MANAGER PLUS for DB2 Reference Manual	provides reference information for the DASD MANAGER PLUS *for DB2* product, including the BMCTRIG and Common Statistics components
	DASD MANAGER PLUS for DB2 User Guide	provides instructions for using DASD MANAGER PLUS, including the BMCTRIG and Common Statistics components
	EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide	provides instructions for using the I/O caching and snapshot processing features of the EXTENDED BUFFER MANAGER (XBM) product and its SNAPSHOT UPGRADE FEATURE (SUF)
notices	release notes, flashes, technical bulletins	explain the latest updates to REORG PLUS

Conventions

This book uses several special conventions that are worth noting:

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

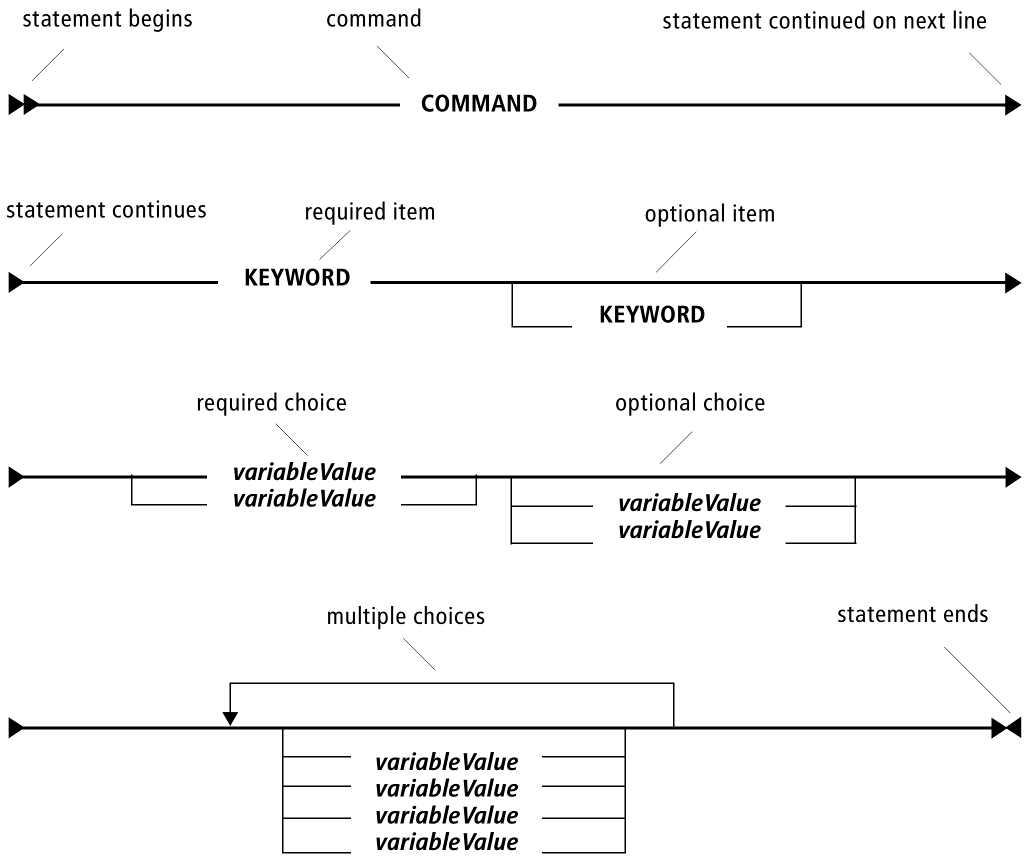
- Variable text in path names, system messages, or syntax is displayed in *italic* text:

  testsys/*instance*/*fileName*

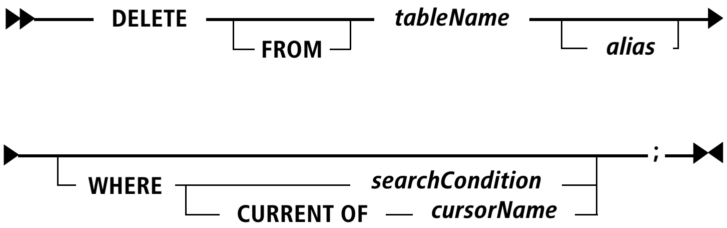
- Change bars signify changes that clarify or correct existing information, or that provide new information corresponding to product changes. This book does not use change bars to denote editorial or formatting changes, unless these updates significantly affect your use of the information.

Syntax diagrams

The following figure shows the standard format for syntax diagrams:



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



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.
- In general, IBM MVS™ commands, keywords, clauses, and data types are displayed in uppercase letters. However, if an item can be shortened, the minimum portion of the MVS command or keyword might be displayed in uppercase letters with the remainder of the word in lowercase letters (for example, CANcel).
- The following conventions apply to variables in syntax diagrams:
  - Variables typically are displayed in lowercase letters and are always italicized.
  - If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words (for example, *databaseName*).

Summary of changes

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

Version 10.1.00 April 2011

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

DB2 Version 10 support - native support

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

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

  REORG PLUS also natively reorganizes XML table spaces that contain these indexes, unless the table space was created in versioning format.
- reorganizations of an index on a table space that contains inline LOB data, unless the index contains a key derived from an expression on an inline LOB column
- reorganizations of table spaces containing an application-period temporal table, unless an index that is defined with BUSINESS_TIME WITHOUT OVERLAPS exists on the table
- reorganizations of indexes on temporal tables, unless the indexes are defined with BUSINESS_TIME WITHOUT OVERLAPS
- reorganizations of table spaces or indexes that contain pending DDL changes in those cases where the IBM DB2 REORG utility would not materialize the DDL changes

  REORG PLUS does not materialize the DDL changes.

DB2 Version 10 support - invoking DSNUTILB

REORG PLUS provides the following support by invoking DSNUTILB. For more information about requirements and restrictions when using features that invoke DSNUTILB, see “Reorganization jobs that invoke DSNUTILB” on page 70.

- the ability to reorganize LOB table spaces when reorganizing the base table space (supported by a new installation option, AUXREORG, discussed on page 628)

  You can also specify a new command option, AUX, that REORG PLUS passes to the IBM DB2 REORG utility. For more information, see “AUX” on page 233.

- reorganizations of table spaces that contain tables that are defined as ORGANIZE BY HASH (supported by a new installation option, HASHAX, discussed on page 645)

  You can also specify a new command option, AUTOESTSPACE, to indicate how to determine the hash space size. REORG PLUS passes this option to the DB2 REORG utility. For more information, see “AUTOESTSPACE” on page 233.

- indexes that contain non-key columns (supported by a new installation option, IXINCLCOL, discussed on page 651)

- reorganizations of universal table spaces that are defined as MEMBER CLUSTER (supported by a new installation option, UTSMEM, discussed on page 685)

- reorganizations of table spaces or indexes that contain pending DDL changes in cases when the IBM DB2 REORG utility would materialize the changes (supported by a new installation option, PENDDDL, discussed on page 663)

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

- the following types of temporal tables and associated objects (supported by a new installation option, TEMPRALDATA, discussed on page 679):

  — system-period temporal tables
  — history tables that are associated with system-period temporal tables
  — indexes that are defined with BUSINESS_TIME WITHOUT OVERLAPS
  — application-period temporal tables that have an index that is defined with BUSINESS_TIME WITHOUT OVERLAPS

- the following types of reorganizations that include inline LOB data:
  - — any table space reorganization (supported by a new installation option, INLOB, discussed on page 651)
  - — an index reorganization when the index is on an inline LOB column, or a table space reorganization when an index on an inline LOB column exists on the table space

    Because an index on an inline LOB column contains a key that is derived from an expression, the existing IXONEX installation option supports this feature. For more information, see page 651.
- LOB table spaces when SHRLEVEL CHANGE is in effect (supported by the LOB installation option, discussed on page 654)
- table spaces that contain XML columns that support XML versions, and their associated XML table spaces and node ID indexes (supported by the XML installation option, discussed on page 687)
- timestamp columns that are defined with a precision other than 6 (supported by a new installation option, TSPREC, discussed on page 682)
- timestamp columns that are defined as TIMESTAMP WITH TIME ZONE (supported by a new installation option, TSTZ, discussed on page 683)

Enhanced support for reorganization jobs that invoke DSNUTILB

- REORG PLUS now invokes DSNUTILB without using the DSNUTILS stored procedure. This change provides the following enhancements:
  - — You no longer need to install and activate the IBM DSNUTILS stored procedure.

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

    You no longer have to go to the message log of the started task for the DSNUTILS stored procedure to find execution messages. This change requires that you do not allocate your SYSPRINT with FREE=CLOSE.
  - — REORG PLUS no longer requires an extra DB2 thread to invoke DSNUTILB.

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

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

  — discard data sets

  REORG PLUS uses the dynamic allocation information for DDTYPE ARCHIVE to create the template for the discard data set that the DB2 REORG utility uses. For more information, see "SYSARC data set" on page 342.

  — LOAD control card data sets

  A new DDTYPE, SYSPUNCH, enables you to specify dynamic allocation information for this data set. REORG PLUS uses this information to create the template for the SYSPUNCH data set that the DB2 REORG utility uses. For more information, see "DDTYPE" on page 270 and "SYSPUNCH data set" on page 348.

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

Enhanced statistics processing

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

  The Common Statistics component is installed when you install REORG PLUS. This version of REORG PLUS requires a minimum of version 10.1.00 of the Common Statistics component.

  This feature enables the following enhancements to statistics processing in REORG PLUS:

  — REORG PLUS now collects statistics on XML objects.

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

— You can now request a statistics report without updating the statistics in the DASD MANAGER PLUS database. To support this feature, the BMCSTATS option has a new keyword, REPORT. For more information, see "REPORT" on page 237.

— You can now specify a table space sampling percentage to use for gathering statistics. A new command and installation option, TSSAMPLEPCT, supports this feature. For more information, see "TSSAMPLEPCT" on page 240 and page 682.

— You can optionally specify a separate DD statement in your JCL, ASUSRPRT DD, to contain the statistics report. For more information, see "ASUSRPRT data sets" on page 326.

This enhancement also encompasses the following changes:

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

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

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

— REORG PLUS does not collect statistics for the following objects:

- nonpartitioned indexes when running a partial reorganization
- partitions in a partition-by-growth table space that are added during the reorganization

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

— The *Utility Products for DB2 Messages Manual* now includes messages that the Common Statistics component produces.

For more information, see "Statistics options" on page 235.

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

  — BMC50510 through BMC50514

  — BMC50521 and BMC50522

  — BMC50531 and BMC50532

  — BMC50541 through BMC50547

  The following messages have been added:

  — BMC50515 through BMC50519

  — BMC50525 through BMC50529

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

Additional enhancements and changes

- REORG PLUS now provides the option to offload eligible processing to an IBM System z® Integrated Information Processor (zIIP). To enable and use zIIP processing, you must have an installed and authorized version of the EXTENDED BUFFER MANAGER (XBM) product or the SNAPSHOT UPGRADE FEATURE (SUF) technology. For this version, the minimum version of XBM or SUF is 5.6.00 with PTF BPE0313.

  NOTE

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

  The following installation and command options apply to this feature:

  — The new ZIIP installation and command options enable this functionality. For more information, see “ZIIP” on page 229 or page 688.

  — You can also use the existing XBMID installation or command option to specify an XBM subsystem to use to access this functionality. For more information, see “XBMID” on page 230 or page 686.

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

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

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

  - REORG PLUS invokes DSNUTILB to reorganize the following objects:

    - base objects when the base table space is instance number 2
    - clone objects

      A new command option, CLONE, enables you to tell REORG PLUS that you want to reorganize only the clone table in the specified table space, or the specified clone index. For more information, see “CLONE” on page 160.

    A new installation option, CLONE, enables support for these objects by invoking DSNUTILB. For more information, see page 631. For more information about DSNUTILB reorganization jobs, see “Reorganization jobs that invoke DSNUTILB” on page 70.

- You can now supply a DDLOUT DD statement in your reorganization job. The presence of this data set tells REORG PLUS to write to that data set all ALTER statements that it executes during partition rebalancing. For more information, see “DDLOUT data sets” on page 338.

- For SHRLEVEL CHANGE and SHRLEVEL REFERENCE reorganizations, you can now specify whether you want REORG PLUS to back out the reorganization when a failure occurs during processing of limit-key ALTER statements. This capability applies to ALTER statements that are processed either from your DDLIN data set or as a result of the REBALANCE option. A new installation option, ALTRFAIL, supports this change. For more information, see page 626.

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

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

  - CBUFFS on page 630
  - UBUFFS on page 683
  - WBUFFS on page 685

NOTE

If, in an earlier version, you changed the values for any of these options from the values that were shipped with REORG PLUS, review the new information for these options carefully to ensure that you obtain the intended results for this version.

- The default value for the LOB installation option has changed to YES. For more information about this option, see page 654.

- REORG PLUS now allows you to determine how extents are allocated when extending to a new data set. A new installation option, MGEXTENT, enables this support. For more information, see page 660.

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

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

  NOTE

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

- This version of REORG PLUS has the following *changes* to minimum requirements. For full requirement information, see "System setup" on page 63.

  — z/OS Version 1.10

  — DB2 Solution Common Code (SCC) version 10.1.00

  — DB2 Utilities Common Code (D2U) version 10.1.00

  — XBM or SUF version 5.6.00

  NOTE

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

- The dynamic allocation information for specific data set types that was in Chapter 2 has been moved to the relevant data set sections in Chapter 4.

Version 9.3.00 December 2009

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

- To improve REORG PLUS processing and reduce customization time, the default values for some installation options have changed. ***Because some of these changes might affect the outcome of existing jobs, BMC strongly recommends that existing customers use the Installation System option to migrate current option values when installing this new version of REORG PLUS.***

— For REORG PLUS basic installation options, changes are as follows:

Option	New default value	Old default value
COPYLVL	PART	FULL
DELFILES	YES	NO
DRAINTYP	ALL	WRITERS
DRNDELAY	1	3
DRNRETRY	255	10
DRNWAIT	NONE	UTIL
DSPLOCKS	DRNFAIL	NONE
INLINECP	YES	NO
SCPYMAX	8	*
SMAX	0	16
STOP@CMT	YES	NO
TIMEOUT	TERM	ABEND
UNLOAD	RELOAD	no default value

— Dynamic allocation is now turned on by default (ACTIVE=YES) for some DDTYPEs. However, the default value for IFALLOC remains USE, which tells REORG PLUS to allocate data sets only if none exist in the JCL. New values for the data set name pattern option (DSNPAT) ensure uniqueness.

The following table lists new values for REORG PLUS dynamic allocation installation options. The old default values for these options were as follows:

- ACTIVE=NO
- SIZEPCT=(100,100)
- DSNPAT=&&UID.&&UTILPFX.&&DDNAME

DDTYPE value	New default ACTIVE value	New default SIZEPCT value	New default DSNPAT value
UNLOAD	YES	no change	&&UID.&&JOBNAME.&&TSIX.&&DDNAME
WORK	YES	no change	&&UID.&&JOBNAME.&&TSIX.&&DDNAME
SORTWORK	no change	no change	NONE
LOCPFCPY	YES	no change	&&UID.&&DDNAME.&&TSIX..F&&PART..T&&TIME
LOCPICPY	YES	(5,100)	&&UID.&&DDNAME.&&TSIX..I&&PART..T&&TIME
LOCBFCPY	no change	no change	&&UID.&&DDNAME.&&TSIX..F&&PART..T&&TIME
LOCBICPY	no change	no change	&&UID.&&DDNAME.&&TSIX..I&&PART..T&&TIME
REMPFCPY	no change	no change	&&UID.&&DDNAME.&&TSIX..F&&PART..T&&TIME
REMPICPY	no change	no change	&&UID.&&DDNAME.&&TSIX..I&&PART..T&&TIME
REMBFCPY	no change	no change	&&UID.&&DDNAME.&&TSIX..F&&PART..T&&TIME
REMBICPY	no change	no change	&&UID.&&DDNAME.&&TSIX..I&&PART..T&&TIME

- REORG PLUS enables you to reorganize LOB table spaces. When you specify SHRLEVEL REFERENCE, REORG PLUS reorganizes the LOB table space and updates the auxiliary index. When SHRLEVEL NONE is in effect, REORG PLUS invokes DSNUTILB to reorganize the LOB table space. For more information, see "LOB data" on page 110.

- You can now use REORG INDEX to reorganize auxiliary indexes on LOB table spaces.

- REORG PLUS now optionally creates a single image copy when reorganizing a subset of physically contiguous partitions. To support this feature, REORG PLUS has added the COPYSUBSET installation option. For more information, see page 634.

NOTE

If you specify COPYSUBSET=YES, you might need to change the command syntax in existing REORG PLUS jobs to avoid syntax errors. For example, you might have a job that contains the following options on the REORG command:

```
SHRLEVEL CHANGE
PART 1:10
COPY YES COPYLVL FULL INLINE NO ICTYPE UPDATE
```

In versions earlier than 9.3, REORG PLUS changed COPYLVL FULL to COPYLVL PART in this case. REORG PLUS changed this value because ICTYPE UPDATE is not valid with INLINE NO when copying multiple partitions to a single copy data set. In version 9.3 or later, REORG PLUS still changes COPYLVL FULL to COPYLVL PART when COPYSUBSET is NO. However, if you specify COPYSUBSET=YES, REORG PLUS terminates with a syntax error.

BMC recommends that you set the INLINE command option or INLINECP installation option to YES to avoid this problem.

- This version of REORG PLUS has the following *changes* to minimum requirements. For full requirement information, see "System setup" on page 63.

  — z/OS 1.7 or later

  — DB2 Solution Common Code (SCC) version 1.6.00 with PTF BPJ0295

  — DB2 Utilities Common Code (D2U) version 9.3.00

- The new TOTALPAGEPCT installation and command options enable you to specify a percentage of total system pages (pages that are underutilized and available for use) that REORG PLUS can allocate to BMCSORT for sort processing. For more information, see page 180 and page 681.

- You can now specify an integer value for the DRAIN_WAIT command option and DRNWAIT installation option. For more information, see page 224 or page 640.

- REORG PLUS now supports the following types of indexes:

  — compressed indexes

    For restart considerations, see “Compressed indexes” on page 364.

  — indexes that contain page sizes that are greater than 4 KB

  — indexes that contain a key derived from an expression

    - REORG PLUS invokes the DSNUTILB utility control program to support these indexes. For more information about requirements and restrictions when using features that invoke DSNUTILB, see “Reorganization jobs that invoke DSNUTILB” on page 70.

    - A new installation option, IXONEX, supports this feature. For more information, see page 651.

  — indexes that contain keys with random ordering

    - REORG PLUS invokes DSNUTILB to support these indexes. For more information about requirements and restrictions when using features that invoke DSNUTILB, see “Reorganization jobs that invoke DSNUTILB” on page 70.

    - A new installation option, IXRANDOM, supports this feature. For more information, see page 652.

- REORG PLUS now allows you to provide hexadecimal values as representations of Unicode data for constants in the following specifications:

  — WHERE clauses in SELECT, UPDATE, or DELETE options
  — SET clauses in UPDATE options
  — ALTER statements in DDLIN data sets

  Consistent with the IBM DB2 REORG utility, REORG PLUS does not convert the value enclosed within apostrophes.

- REORG PLUS now supports reorganizing table spaces that contain columns that are defined with the DECFLOAT data type.

- REORG PLUS now allows you to reorganize, without invoking DSNUTILB, the following objects:

  — XML table spaces
  — indexes on XML table spaces
  — base table spaces that contain XML columns

In support of this feature, REORG PLUS has added the IDCACHE installation and command options. These options allow you to control the size of the cache that REORG PLUS uses to reserve a range of values for the document ID (DOCID) column of the base table space. For more information, see page 232 and page 648.

- REORG PLUS has enhanced support for reorganization jobs that invoke DSNUTILB:

  — You can now run DSNUTILB reorganization jobs when running on DB2 Version 8.

  — For copy data sets, REORG PLUS now translates the THRESHLD 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 value is exceeded. REORG PLUS uses the secondary values for UNIT, UNITCNT, VOLCNT, MAXEXTSZ, and the SMS classes.

Version 9.2.00 June 2008

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

- This version of REORG PLUS requires the following minimum versions of components:

  — BMC Software DB2 Utilities Common Code (D2U) version 1.6.00

  — BMC Software DB2 Solution Common Code (SCC) version 1.6.00

  — BMCSORT version 2.3.01

  — EXTENDED BUFFER MANAGER product (XBM) or the SNAPSHOT UPGDRADE FEATURE (SUF) of XBM version 5.5.00 when you use the Instant Snapshot technology

- REORG PLUS supports reorganizing of both types of universal table spaces (range-partitioned and partition-by-growth table spaces) and indexes defined on tables in these types of table spaces.

- REORG PLUS supports reordered row format (RRF) in two ways:

  — reorganizes data that is in RRF
  — converts data from basic row format (BRF) to RRF

  REORG PLUS does *not* support mixed format partitions in a single reorganization when you are running DB2 in either conversion mode* or enabling-new-function mode* after falling back from new function mode.

- When reorganizing a compressed table space or partition, REORG PLUS builds a new dictionary when converting data that is in basic row format (BRF) to reordered row format (RRF) *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 supports SMS class definitions in DB2 storage groups.

- REORG PLUS supports NOT LOGGED table spaces

  You can use REORG PLUS with any SHRLEVEL option except SHRLEVEL CHANGE to reorganize tables spaces that have the NOT LOGGED attribute.

- REORG PLUS supports the following data types:

  — BIGINT
  — BINARY
  — VARBINARY
  — row change timestamp

- REORG PLUS supports reorganizing XML data.

  REORG PLUS reorganizes XML data by invoking the IBM DSNUTILB utility control program. For more information, see “Reorganization jobs that invoke DSNUTILB” on page 70.

The following command and installation options have been added or enhanced to support this feature:

- The new DSNUTILB installation and command options allow you to specify whether you want REORG PLUS to invoke DSNUTILB when needed. This option must be enabled for support of certain features in REORG PLUS. For this version, REORG PLUS uses DSNUTILB to enable support for reorganizing XML data. For more information, see page 642 and page 217.

- The new installation option XML tells REORG PLUS to invoke DSNUTILB to enable reorganizing XML data.

- The new SORTDEVT installation option allows you to specify two default device types to use for sorting. The first parameter applies to sorting during a non-DSNUTILB reorganization and the second parameter applies to sorting during a DSNUTILB reorganization. For more information, see page 673.

- The new SPACE command option for dynamic allocation allows you to specify a value that overrides the value that DSNUTILB calculates for the specified data set type. For more information, see page 286.

- When invoking DSNUTILB, REORG PLUS will now pass values for the SORTKEYS command option. This option was previously provided in REORG PLUS only for consistency with the IBM DB2 REORG utility. For more information, see page 162.

- When invoking DSNUTILB, the MAPPINGTABLE option specifies the mapping table that REORG PLUS provides to the IBM DB2 REORG utility for SHRLEVEL CHANGE reorganizations. Otherwise, REORG PLUS does not use the MAPPINGTABLE option, but treats it as a comment so that the command syntax remains compatible with that of the DB2 REORG utility.

- The following new DASD MANAGER PLUS exceptions are available for use with the CONDEXEC BMC option. For more information, see "Conditional reorganization" on page 128.

  - REORMODS
  - REORMDEL
  - APPNDINS
  - AREOPEND

- In previous versions, if REORG PLUS did not perform sort tasks during unload processing, it did not build the compression dictionary. This limitation no longer applies.

- REORG PLUS no longer uses the traditional hardware snapshot function of XBM or SUF during SHRLEVEL CHANGE reorganizations. REORG PLUS uses only software snapshots because they are more efficient and require less time to register. This change does not affect use of the SIXSNAP option.

  For proper REORG PLUS performance, 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*.

- This book no longer includes descriptions of the REORGCPY and IMAGECPY installation options. In a previous version, REORG PLUS ceased requiring or supporting these options. If your installation options module contains these options, your options module will not assemble correctly. REORG PLUS always inserts a REORG LOG(NO) row into SYSIBM. SYSCOPY, preventing REORG PLUS from creating an unrecoverable situation.

- This book no longer includes a messages appendix. The information that was formerly in this appendix is now incorporated in a new manual, the *Utility Products for DB2 Messages Manual*. This new manual includes messages and code information for the REORG PLUS component and other Utility products, including the messages that are common to these products.

- In previous versions of the product, you could access message information with the BMC online message processor. With this version, you can no longer type **TSO BMCMSG *messageID*** on the TSO command line to retrieve message information. To look up a message, consult the *Utility Products for DB2 Messages Manual*.

- REORG PLUS no longer supports DB2 Version 7. Although REORG PLUS version 9.2.00 does not run in a DB2 Version 7 environment, earlier versions still support DB2 Version 7.

Chapter 1

Introduction to REORG PLUS

This chapter presents the following topics:

Overview

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

Disadvantages of disorganized data

Disorganized data decreases database efficiency in the following ways:

- In sequential processing, more I/Os are required to retrieve disorganized data than are needed to retrieve data items that are physically adjacent to one another.

- The retrieval inefficiency caused by the disorganization slows DB2 application response time and decreases user productivity.

- A disorganized database wastes DASD space.

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

The REORG PLUS solution

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

REORG PLUS benefits

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

- **reduces costs of reorganizing DB2 data** because fewer CPU cycles and EXCPs are used

- **increases availability of DB2 data** because the time needed to reorganize the data is reduced

- **improves DB2 performance** by allowing more frequent reorganizations because of reduced reorganization costs and elapsed times

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

Performing an online reorganization provides the following benefits:

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

- **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 Database Performance for DB2® solution. Database Performance helps database administrators (DBAs) determine the maintenance tasks that are required on their DB2 objects.

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

Product components

The Database Performance solution includes the following BMC products:

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

Technology components

The Database Performance solution includes the following BMC technologies:

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

Features

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

- You can use the Export utility to copy object definitions residing on a local *controlling* DASD MANAGER PLUS repository to *destination* DASD MANAGER PLUS repositories on other DB2 subsystems. For more information, see the *DASD MANAGER PLUS for DB2 User Guide*.

- You can use the value **BMC** on the CONDEXEC installation or command option to instruct REORG PLUS to use the DASD MANAGER PLUS exceptions table to determine whether an object should be reorganized. Using the exceptions table gives REORG PLUS an expanded set of conditions to determine whether the reorganization is needed. For information about the CONDEXEC options, see page 631 or page 210.

- You can use the value **BMCSTATS** on the ANALYZE command option to enhance the performance of REORG PLUS. In the ANALYZE phase, REORG PLUS uses the statistics already gathered by BMCSTATS instead of gathering the statistics itself. For information about the ANALYZE command option, see page 185.

Tasks that REORG PLUS performs

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

Table 1 REORG PLUS functional enhancements (part 1 of 3)

Functional area	Enhanced tasks
resources	■ analyzes resources for the specific reorganization ■ provides data about resource requirements ■ optionally offloads eligible processing to an IBM System z® Integrated Information Processor (zIIP) ■ optionally allocates sort work files, index work files, unload data files, archive files, and copy files dynamically
partitions	■ reorganizes any number of selected partitions of a table space and index space ■ allows select and delete operations during a partial or full reorganization ■ rebalances partitions for all access levels, including SHRLEVEL CHANGE
conditional reorganizations	■ uses the traditional values that are stored in the DB2 catalog ■ optionally uses an expanded set of exceptions and user controls from the DASD MANAGER PLUS exceptions table if you are running the Database Performance for DB2 solution

Table 1 REORG PLUS functional enhancements (part 2 of 3)

Functional area	Enhanced tasks
performance	■ provides data for fine-tuning reorganization performance ■ for an additional performance gain, offers an optional single processing phase ■ allows multitasking that is not limited by the number of CPUs
accessibility	■ optionally allows the objects that you are reorganizing to be available in read-only (RO) status during unload and reload processing ■ offers a nondestructive reorganization process, meaning that the objects that you are reorganizing remain intact throughout the reorganization ■ reclaims space that dropped tables and deleted rows occupied ■ offers an online reorganization capability through the SHRLEVEL CHANGE keywords, allowing applications to have read-write (RW) access during most of the reorganization processing ■ for a SHRLEVEL CHANGE reorganization, allows you to dynamically change the REORG command by using the XBM Utility Monitor or the MVS™ console
rebuilding objects and data	■ removes row indirection ■ optionally purges unwanted or out-of-date information and can write this data to an archive data set (for all types of reorganizations, including partial reorganizations and SHRLEVEL CHANGE) ■ optionally updates column values to a specified constant value ■ reorders out-of-cluster-order rows in single and multitable table spaces by table and clustering key or by clustering key only ■ rebuilds associated indexes ■ restores PCTFREE and FREEPAGE space in table spaces and indexes ■ adds NULL or DEFAULT values to short rows after a table is altered to add new columns ■ optionally reorganizes without reclustering rows ■ optionally deletes and redefines user-defined data sets and data sets defined in DB2 storage groups as part of the reorganization
copies	■ with a self-contained copy function, produces image copies or DSN1COPY-type copies concurrently with the reorganization ■ optionally dynamically allocates the full and incremental image copy data sets ■ optionally produces inline image copies as it reloads your tables
statistics and reports	■ produces a statistics report ■ optionally updates statistics in the DASD MANAGER PLUS statistics tables ■ optionally updates statistics in DB2 catalog tables ■ optionally maintains historical records of all REORG PLUS activities ■ resets real-time statistics and timestamp values (in both the DSNRTSDB and DB2 memory) ■ optionally provides a report listing all ALTER statements that REORG PLUS executes during partition rebalancing

Table 1 REORG PLUS functional enhancements (part 3 of 3)

Functional area	Enhanced tasks
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 a user exit point that allows 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 time

Differences between REORG PLUS and the IBM DB2 REORG utility

Table 2 summarizes the most important functional and operational differences between REORG PLUS and the IBM DB2 REORG utility. Table 3 on page 48 summarizes SHRLEVEL CHANGE differences.

NOTE

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

Table 2 Functional and operational differences between REORG PLUS and IBM DB2 REORG (part 1 of 3)

Functional or operational area	Description
authorization	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.
multiple reorganizations	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.

Table 2 Functional and operational differences between REORG PLUS and IBM DB2 REORG (part 2 of 3)

Functional or operational area	Description
UNLOAD ONLY	REORG PLUS does not have an UNLOAD ONLY option. Therefore, you cannot use REORG PLUS to generate FORMAT UNLOAD data. **Note**: The UNLOAD PLUS for DB2 product from BMC Software provides advanced unload utility functions as well as basic unload tasks.
reorganization phases	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).
indexes	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 all rows of the table space (if the table space is nonpartitioned) or all rows of the largest partition (if the table space is partitioned), not only the index keys.
catalog objects	REORG PLUS does not operate on catalog objects. If you attempt to reorganize a DB2 system table, REORG PLUS terminates.
start/stop status	For SHRLEVEL NONE (the default), REORG PLUS stops the table space and index spaces that you are reorganizing at the beginning of unload processing. The space remains stopped throughout the job. For a partial reorganization, REORG PLUS stops and starts only those partitions that you specified with the PART option of the REORG command. REORG PLUS stops and starts nonpartitioned indexes in their entirety. For SHRLEVEL REFERENCE UNLOADONLY, REORG PLUS starts the table space and index spaces that you are reorganizing in RO status at the beginning of the UNLOAD phase. REORG PLUS later stops the spaces at the beginning of the RELOAD phase, and they remain stopped until the end of the job. For a partial reorganization, REORG PLUS stops and starts only those partitions that you specified with the PART option of the REORG command. REORG PLUS stops and starts nonpartitioned indexes in their entirety. For SHRLEVEL REFERENCE, REORG PLUS starts the table space and index spaces that you are reorganizing in RO status. The spaces remain in RO status until they are stopped at the beginning of the UTILTERM phase. For a partial reorganization, REORG PLUS stops and starts only those partitions that you specified with the PART option of the REORG command. REORG PLUS stops and starts nonpartitioned indexes in their entirety.
multiple volumes	For multiple volume storage-group-defined table spaces and indexes, REORG PLUS attempts to reallocate the data set on the volume on which the data set currently resides if the volume is still defined in the storage group. After the current volume, the order of the volumes that REORG PLUS uses for allocating the DB2 VSAM data sets that you are reorganizing is unpredictable unless you use the DSRSEXIT user exit to specify a particular order.
EDITPROCs	REORG PLUS uses EDITPROCs to extract keys and update columns.

Table 2 Functional and operational differences between REORG PLUS and IBM DB2 REORG (part 3 of 3)

Functional or operational area	Description
recoverability / restartability	REORG PLUS does not have a LOG YES option. Therefore, you must create and register a full image copy to ensure recoverability of the table space after reorganizing. You can create a full image copy by using the REORG PLUS COPY YES option. You can also use the BMC COPY PLUS *for DB2*® product or the IBM DB2 COPY utility.
partition rebalancing	When you specify the REBALANCE command option, REORG PLUS computes limit key breaks based on rebalancing at the record level, which enables REORG PLUS to redistribute the number of rows across partitions. The DB2 REORG utility computes limit key breaks based on rebalancing at the page level. 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.

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

Functional or operational area	Description
performance	■ Due to efficient processing in reorganization, log apply, and termination, REORG PLUS can successfully complete a SHRLEVEL CHANGE reorganization when the DB2 objects that you are reorganizing are under a heavier application load. ■ REORG PLUS provides the ability to start the read-only phase of the reorganization at a specific time.
log apply process	■ You can specify the spill data set size. If insufficient memory exists to hold the RID translation maps or the log records, REORG PLUS allocates spill data sets on disk. ■ REORG PLUS processing does not impact the DB2 buffer pools because REORG PLUS runs outside of DB2.
RID translation map	■ The RID translation map that REORG PLUS creates is not a DB2 object and therefore does not use DB2 resources. ■ The REORG PLUS RID map processing does not require log record sequence numbers (LRSNs) or relative byte addresses (RBAs) to be kept for each RID map entry, resulting in less storage for the RID map. ■ You can control the amount of virtual storage that is allocated for the RID map by using an installation option or an option on the REORG command. ■ REORG PLUS does not require a mapping table to be predefined and does not require the mapping table name to be specified on the REORG command. ■ REORG PLUS does not require any DB2 sorting services to order the RID map records.

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

Functional or operational area	Description
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).
renaming data sets	If you use the rename process instead of the FASTSWITCH process, REORG PLUS multitasks the process of renaming the staging data sets to the original data set names. As a result, the data sets are available sooner than they would be if you used the DB2 REORG utility with FASTSWITCH NO.
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

For a more detailed explanation of the execution phases for SHRLEVEL CHANGE, see "How SHRLEVEL CHANGE works" on page 572. The figures for the various types of SHRLEVEL CHANGE reorganizations are located in Chapter 7, beginning on page 577.

REORG PLUS processing phases

Table 4 describes the REORG PLUS processing phases. Figures 1 through 5 starting on page 52 illustrate the processing phases for the following types of reorganizations:

Reorganization type	Reference
two-phase table space	Figure 1 on page 52
two-phase index	Figure 2 on page 53
single-phase table space	Figure 3 on page 54
single-phase index space	Figure 4 on page 55
invoking DSNUTILB	Figure 5 on page 56

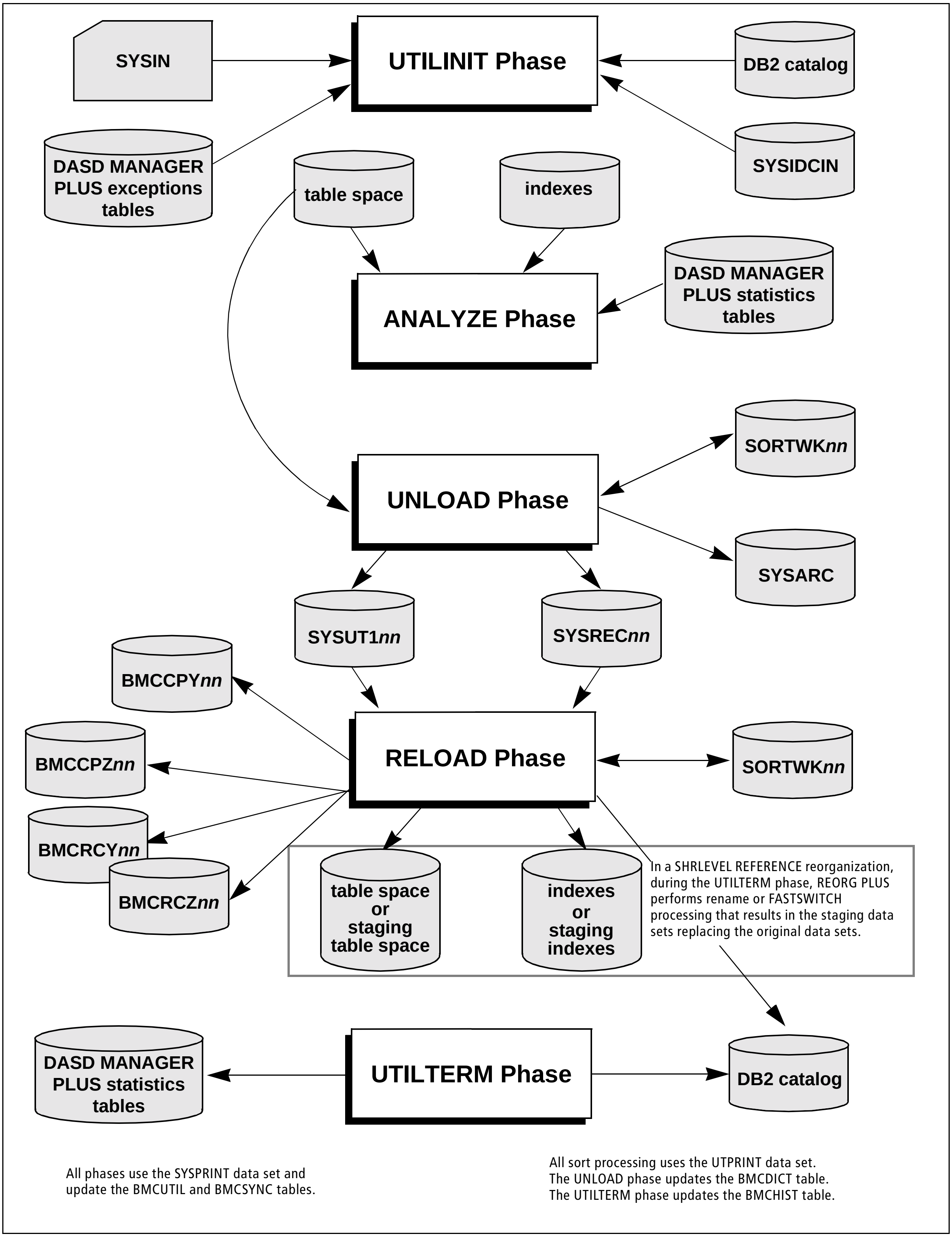
Table 4 REORG PLUS processing phases (part 1 of 2)

Phase name	Description
UTILINIT	During this phase, REORG PLUS completes the following processes: ■ initializes the job ■ performs DB2 catalog lookup ■ reads, parses, and verifies the REORG command and the IDCAMS control statements in the SYSIDCIN data set
DSNUTILB	REORG PLUS executes this phase only when the type of reorganization that you are running requires DSNUTILB. During this phase, REORG PLUS passes processing to DSNUTILB. This phase takes the place of the UNLOAD, RELOAD, and REORG phases, as well as the LOGAPPLY and LOGFINAL phases for a SHRLEVEL CHANGE reorganization. Additionally, this phase performs some of the tasks that the UTILTERM phase normally performs, such as updating statistics. For more information about the types of jobs that require this phase, see “Reorganization jobs that invoke DSNUTILB” on page 70.
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

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

Phase name	Description
RELOAD	During this phase, REORG PLUS completes the following processes: ■ redefines the VSAM data sets if the value of the REDEFINE option is YES ■ reloads the data into the table space and index spaces ■ sorts indexes as required before index build ■ collects statistics ■ produces the requested copies ■ if required, adds partitions to a partition-by-growth table space (SHRLEVEL NONE and SHRLEVEL REFERENCE only)
REORG	During this phase, REORG PLUS performs the functions of both the UNLOAD and RELOAD phase in a single processing phase. The SHRLEVEL specification affects the restartability of the job.
LOGAPPLY	During this phase, REORG PLUS applies the stored log records to the reorganized staging data sets This phase runs only if you specify SHRLEVEL CHANGE. For more information about the LOGAPPLY phase, see "How SHRLEVEL CHANGE works" on page 572.
LOGFINAL	During this phase, REORG PLUS completes the following processes: ■ prevents updates to the original data sets ■ if required, adds partitions to a partition-by-growth table space This phase runs only if you specify SHRLEVEL CHANGE. For more information about the LOGFINAL phase, see "How SHRLEVEL CHANGE works" on page 572.
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 and delete operations that are associated with the staging data sets when the following conditions exist: — you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE — you are not using the FASTSWITCH process ■ 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

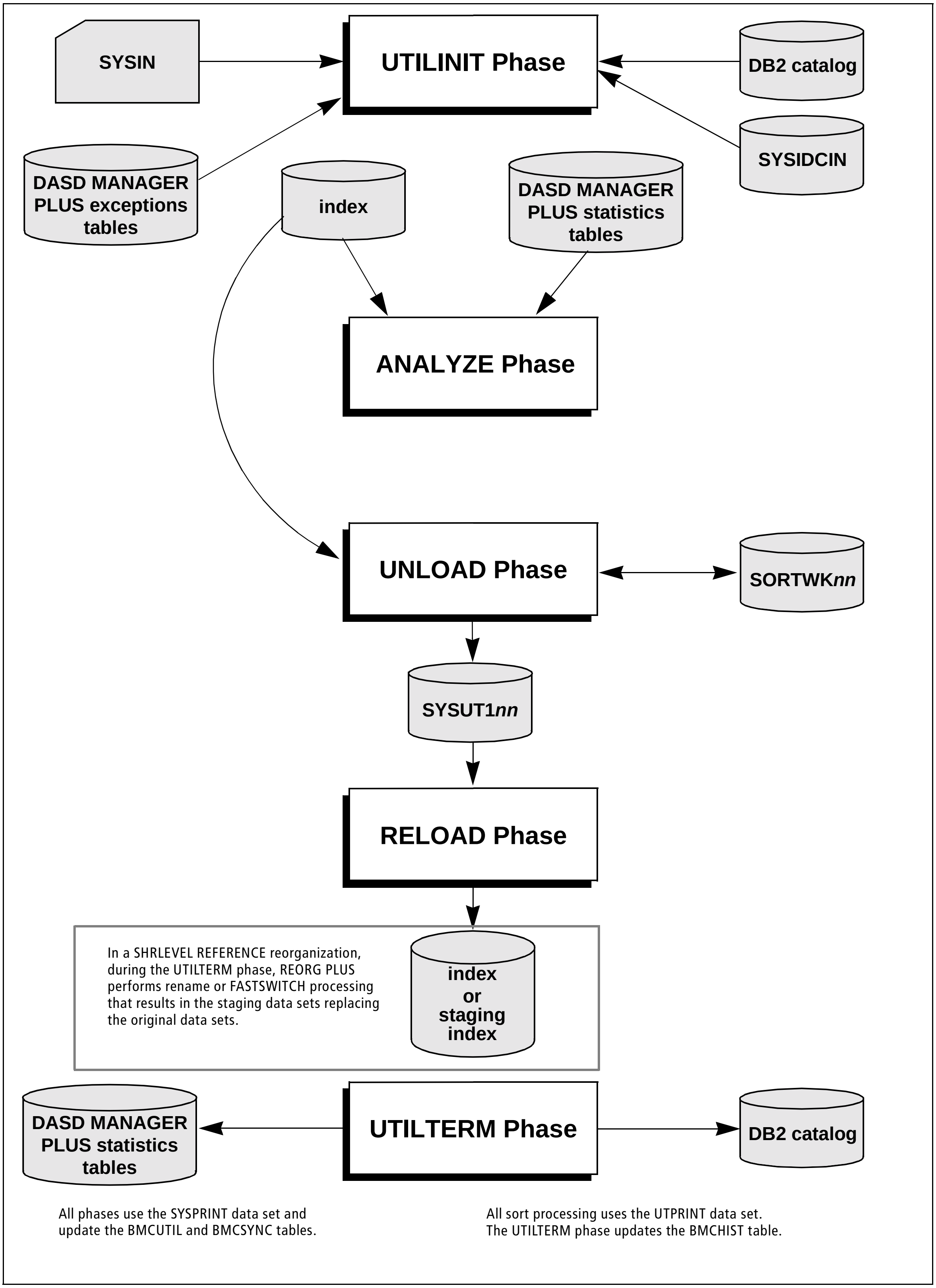


In a SHRLEVEL REFERENCE reorganization, during the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

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

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

Figure 2 Two-phase index reorganization

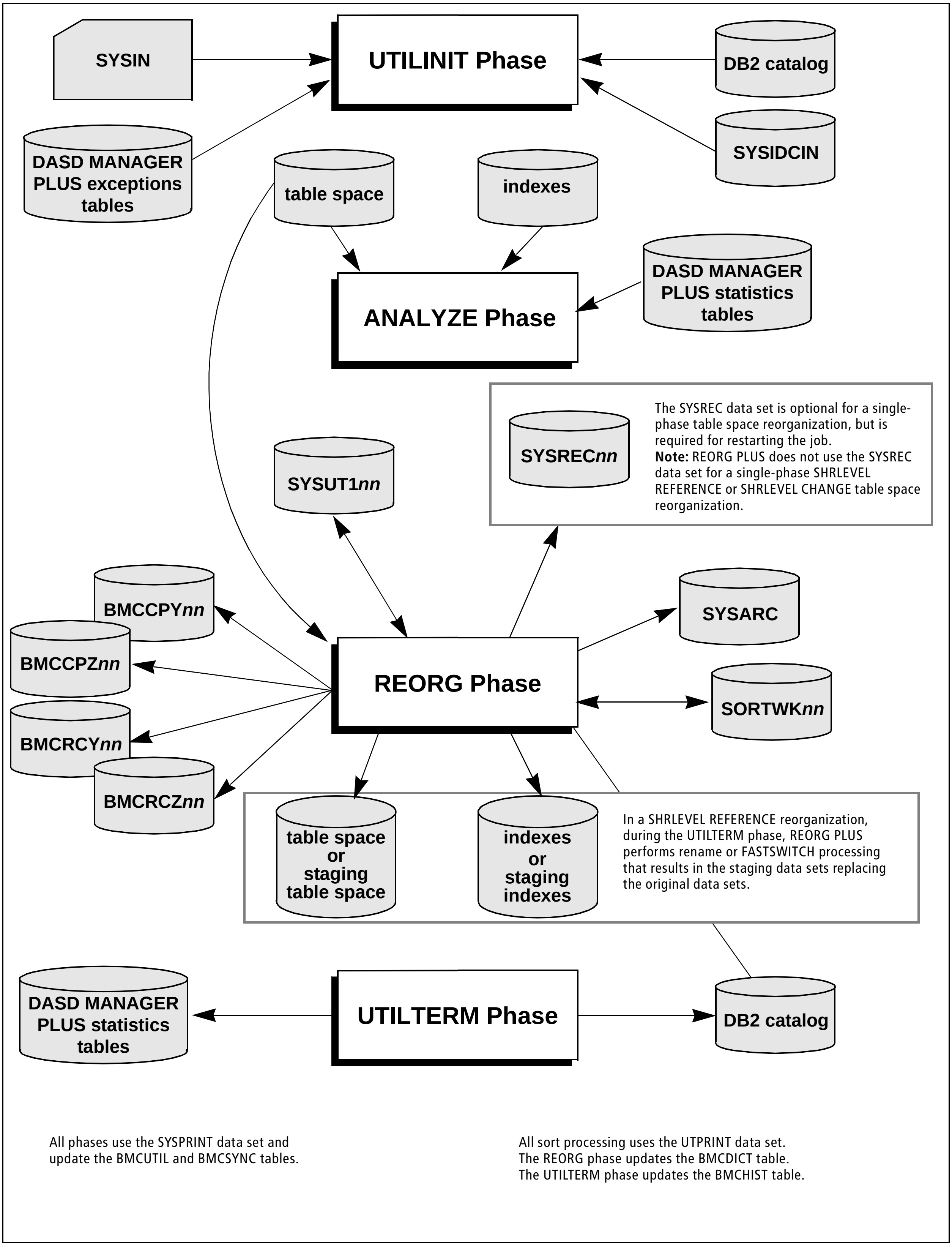


In a SHRLEVEL REFERENCE reorganization, during the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

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

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

Figure 3 Single-phase table space reorganization



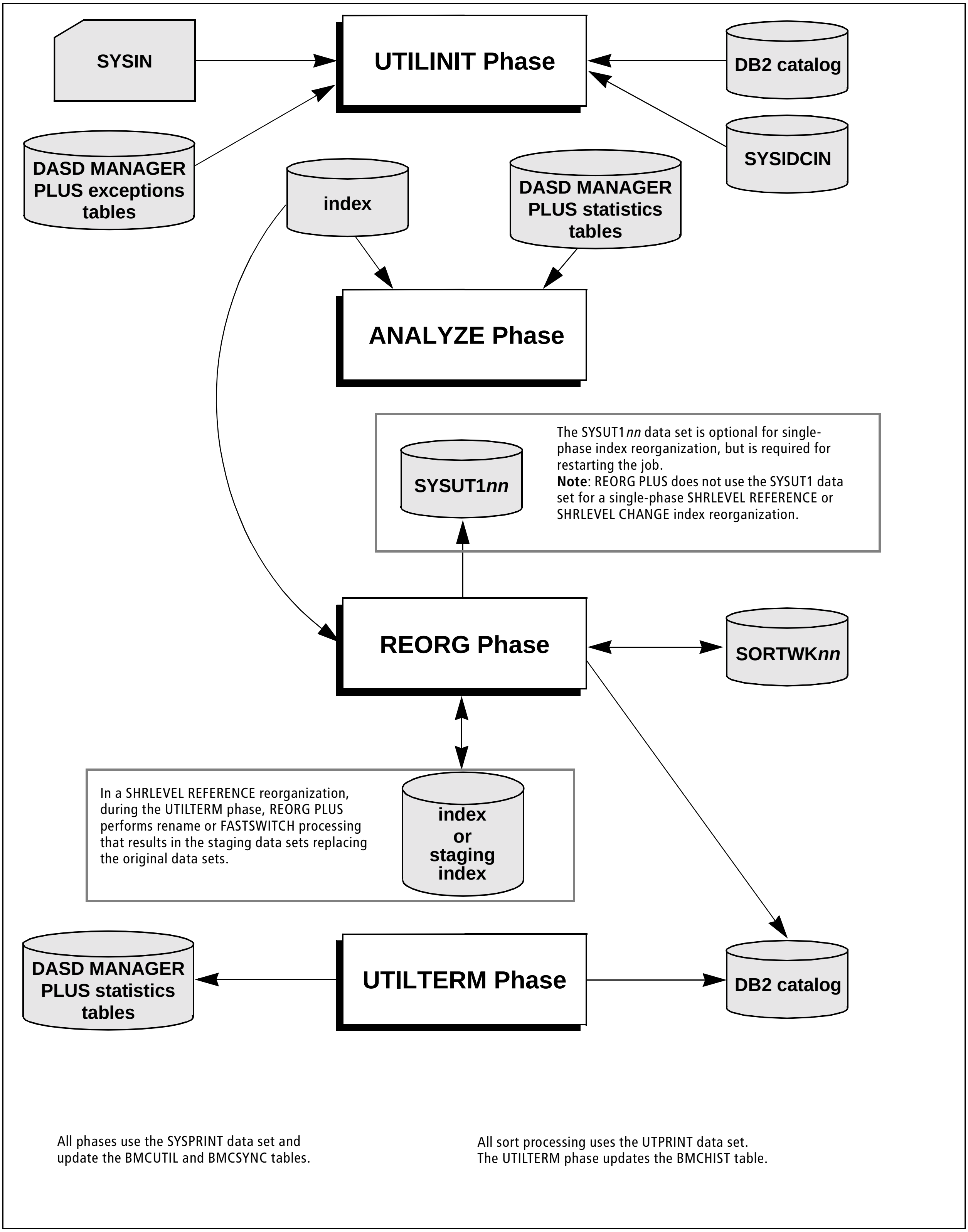
The SYSREC data set is optional for a single-phase table space reorganization, but is required for restarting the job.
Note: REORG PLUS does not use the SYSREC data set for a single-phase SHRLEVEL REFERENCE or SHRLEVEL CHANGE table space reorganization.

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

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

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

Figure 4 Single-phase index reorganization



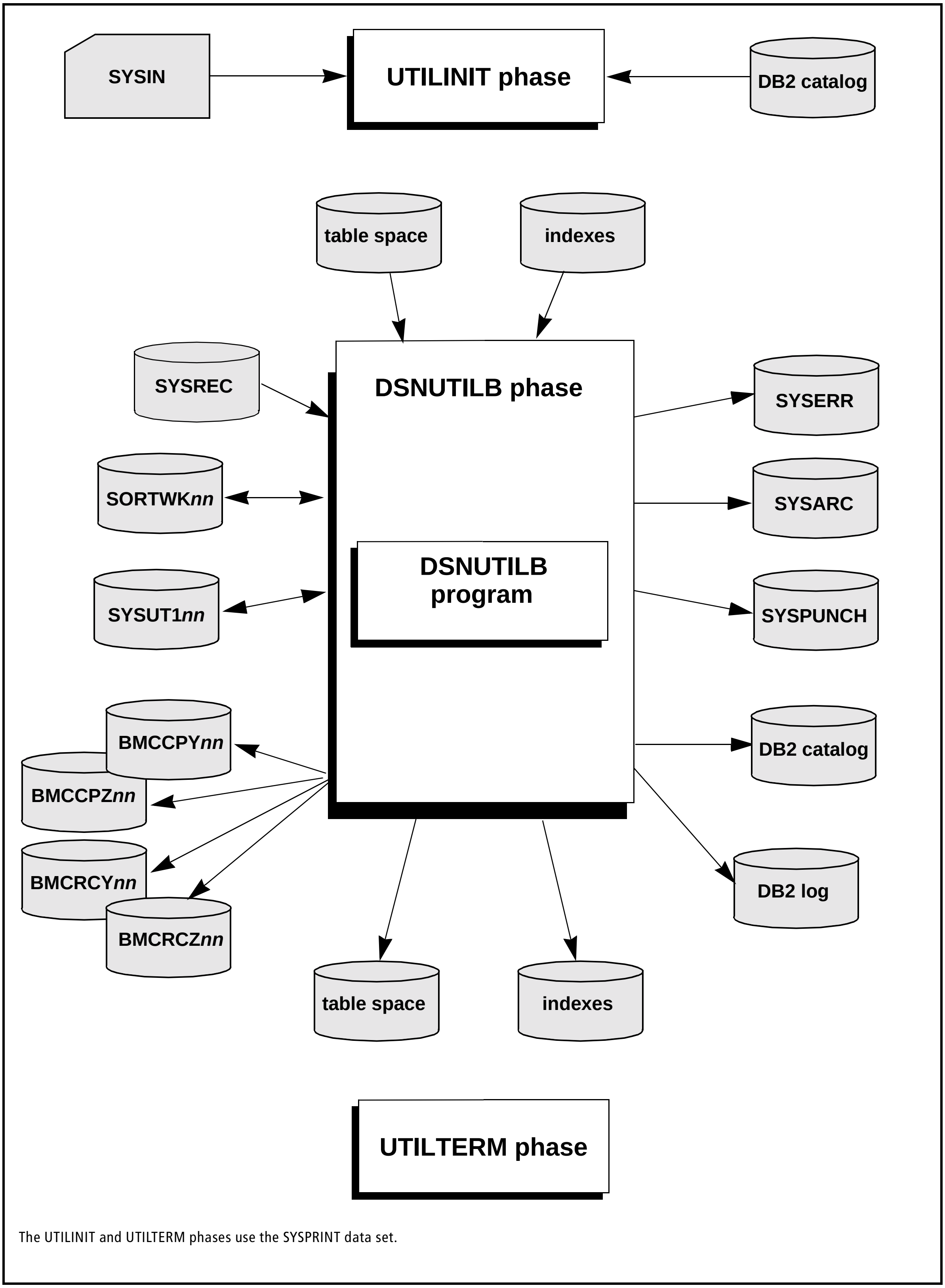
The SYSUT1*nn* data set is optional for single-phase index reorganization, but is required for restarting the job.
Note: REORG PLUS does not use the SYSUT1 data set for a single-phase SHRLEVEL REFERENCE or SHRLEVEL CHANGE index reorganization.

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

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 5 Processing phases when invoking DSNUTILB



The UTILINIT and UTILTERM phases use the SYSPRINT data set.

REORG PLUS data sets

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

Table 5 REORG PLUS data sets (part 1 of 2)

Data set or ddname	Description	See page
ASUSRPRT	ASUSRPRT is an optional data set that contains the statistics report that is generated when you specify BMCSTATS YES, BMCSTATS REPORT, or UPDATEDB2STATS YES.	326
SYSIN	SYSIN is the input data set that contains the REORG command.	347
SYSREC*nn*[a]	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.	349
SORTWK*nn*[a]	SORTWK*nn* 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.	338
SYSUT1*nn*[a]	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.	353
SYSIDCIN	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.	345
DDLIN	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.	333
DDLOUT	The DDLOUT output data set contains all SQL ALTER statements that REORG PLUS executes during partition rebalancing. REORG PLUS opens this optional data set only if executing an ALTER.	338

Table 5 REORG PLUS data sets (part 2 of 2)

Data set or ddname	Description	See page
full copy data sets[a]	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.	326
incremental copy data sets[a]	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.	244, 605
SYSARC	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.	342
SYSPUNCH	The SYSPUNCH data set is used only for DSNUTILB reorganizations and contains LOAD statements that are generated when records are discarded during the reorganization.	348
SYSPRINT	The SYSPRINT output data set contains REORG PLUS messages.	348
SYSERR[a]	This data set holds diagnostic messages that REORG PLUS might produce in error situations when running a SHRLEVEL CHANGE reorganization.	345
SYSEXEC	SYSEXEC specifies the library concatenation where REXX exits reside.	345
SYSTSPRT	REXX routes all output from the REXX 'SAY' statements to the SYSTSPRT data set.	353
UTPRINT[a]	The UTPRINT data set indicates that sort messages should be reported. However, the actual messages for each sort appear in separate SYS*nnnnn* data sets, where *nnnnn* is a system-assigned sequential number.	357
DB2 data sets	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.	not applicable

[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 described in Table 6.

Table 6 Products and components that REORG PLUS uses

Component	Description	Installation and maintenance reference
BMC Common Statistics	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.	*Utility Products for DB2 Installation Guide*
BMCSORT	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.	
DB2 Solution Common Code (SCC)	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.	
DB2 Utilities Common Code (D2U)	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.	
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 ■ zIIP processing XBM and SUF are licensed, installed, and maintained separately from REORG PLUS.	*EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE Installation Guide*

Operational considerations

This chapter presents the following topics:

System setup

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

DB2 support

This version of REORG PLUS supports DB2® Versions 8, 9, and 10.

System requirements

This version of REORG PLUS requires z/OS® Version 1.10 or later.

Software requirements

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

- REORG PLUS requires a minimum of the following versions of BMC common components:

  — version 2.3.01 of BMCSORT

  — version 10.1.00 of the DB2 Utilities Common Code (D2U)

  — version 10.1.00 of the DB2 Solution Common Code (SCC)

- To update DB2 catalog statistics or the DASD MANAGER PLUS database statistics tables, REORG PLUS requires a minimum of version 10.1.00 of the BMC Common Statistics component.

- To use the Instant Snapshot technology, REORG PLUS requires version 5.6.00 or later of either the BMC EXTENDED BUFFER MANAGER (XBM) product or its SNAPSHOT UPGRADE FEATURE (SUF) technology.

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

- To offload eligible processing to a zIIP, REORG PLUS requires version 5.6.00 of either XBM or SUF with PTF BPE0313, or a later version.

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

- To use any features that invoke DSNUTILB (see page 70), you must be licensed to use the IBM DB2 REORG utility.

Required authorization

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

Authorization verification mechanisms

Table 7 on page 65 describes how REORG PLUS verifies authorization based on the authorization verification mechanism that is available for your system.

Table 7 Authorization verification mechanisms

Available authorization verification mechanism	REORG PLUS actions
DB2 security exit	REORG PLUS uses the DSNX@XAC authorization 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. ■ Computer Associates provides the DSNX@XAC exit with CA-ACF2 Security for DB2 and CA Top Secret Security for DB2. BMC recommends this mechanism for implementing external security. The access control authorization exit must be available in the STEPLIB, JOBLIB, linklist, or in the SYS3.DSN exit.
one of the following security products from Computer Associates: ■ CA ACF2 ■ CA Top Secret	REORG PLUS uses either of these Computer Associates products with any version of DB2. REORG PLUS detects the presence of the product in the subsystem where REORG PLUS is running. To use either of these products with REORG PLUS, you must meet the following requirements: ■ The value of the ACFORTSS installation option must be YES (the default). ■ You must be using a version of your security product that enables external security calls for DB2. **Note**: If you have one of these security products installed, but the version does not support external security, you must complete one of the following tasks: ■ Change the value of the ACFORTSS installation option to NO. REORG PLUS then uses the standard DB2 method to check security. ■ Contact your security vendor for the required APAR to enable external security calls for DB2. Then, ensure that the value of the ACFORTSS installation option is YES.
none	REORG PLUS uses the standard DB2 method to check security.

DB2 authorization

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

- sufficient DB2 authority to execute the REORG PLUS plan and all packages that the REORG PLUS plan uses

- authorization equivalent to the authorization that the comparable IBM DB2 REORG utility requires

- ALTER INDEX and ALTER TABLE privileges for the database containing the named table space or index (if not implicit in the authority that you have)

NOTE

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

Additional authorizations for SHRLEVEL CHANGE

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

- TRACE privilege
- MONITOR2 privilege
- DISPLAY privilege (if not already granted to PUBLIC)

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

Additional authorizations for XML reorganizations

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

- SYSIBM.SYSSEQUENCES
- SYSIBM.SYSSEQUENCESDEP

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

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

Additional authorizations for using DSRSEXIT

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

- For the ALTER TABLESPACE statement, you need *one* of the following authorizations:

  — ownership of the table space
  — DBADM authority for the database that contains the table
  — SYSADM or SYSCTRL authority

- For the ALTER INDEX or ALTER TABLE statement, you need *one* of the following authorizations:

  - ownership of the index
  - ownership of the table on which the index is defined
  - DBADM authority for the database that contains the table
  - SYSADM or SYSCTRL authority

Additional authorizations for using XBM or SUF

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

Data set authorization

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

- If you use RACF, specify OPNDB2ID=YES in your installation options.
- Establish authorization as described in “Establishing authorization when OPNDB2ID=NO” on page 68.

Using RACF and OPNDB2ID=YES

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

NOTE

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

Establishing authorization when OPNDB2ID=NO

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

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

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

- For STAGEDSN=BMC:

  — *VCAT*.BMCDBC.*database.object*.I0001
  — *VCAT*.BMCDBD.*database.object*.I0001
  — *VCAT*.OLDDBC.*database.object*.I0001
  — *VCAT*.OLDDBD.*database.object*.I0001
  — *VCAT*.BMCDBD.*database.object*.J0001
  — *VCAT*.BMCDBC.*database.object*.J0001
  — *VCAT*.OLDDBD.*database.object*.J0001
  — *VCAT*.OLDDBC.*database.object*.J0001

- For STAGEDSN=DSN (the default when you use the FASTSWITCH process):

  — *VCAT*.DSNDBC.*database.object*.I0001
  — *VCAT*.DSNDBD.*database.object*.I0001
  — *VCAT*.DSNDBC.*database.object*.J0001
  — *VCAT*.DSNDBD.*database.object*.J0001
  — *VCAT*.DSNDBC.*database.object*.S0001
  — *VCAT*.DSNDBD.*database.object*.S0001

For more information, see “Staging data sets and the rename process” on page 100 and “Staging data sets and the FASTSWITCH process” on page 102.

Using a security package other than RACF

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

1. Associate users with a security group.
2. Grant EXECUTE privileges on the REORG PLUS product program (ARUUMAIN) to the security group.
3. Grant the minimum data set authority levels to ARUUMAIN, described in "Establishing authorization when OPNDB2ID=NO" on page 68.

MEMLIMIT system parameter

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

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

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

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

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

Number of DB2 threads that REORG PLUS uses

The maximum number of batch DB2 threads that REORG PLUS uses concurrently is six per job. REORG PLUS uses up to six 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.

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

- LOB objects:

  — reorganizing LOB table spaces when SHRLEVEL NONE or SHRLEVEL CHANGE is in effect

  — reorganizing LOB table spaces and the base table space at the same time

  — reorganizing table spaces that contain inline LOB data

  — reorganizing indexes that contain a key derived from an expression on an inline LOB column, or reorganizing table spaces that contain those indexes

  For more information about how REORG PLUS reorganizes LOB objects, see "LOB data" on page 110.

- clone objects:

  — reorganizing clone objects

  — reorganizing base objects that participate (or have participated) in a clone relationship when the instance number of the base table space is 2

  When base objects participate in a clone relationship and the instance number of the base table space is 1, REORG PLUS reorganizes those objects natively.

- temporal table objects:

  — reorganizing table spaces that contain system-period temporal tables

  — reorganizing table spaces that contain the history tables that are associated with system-period temporal tables

  — reorganizing indexes that are defined with BUSINESS_TIME WITHOUT OVERLAPS, and reorganizing table spaces that contain those indexes

  REORG PLUS reorganizes natively those table spaces that contain a business-period temporal table that do not have an index that is defined with BUSINESS_TIME WITHOUT OVERLAPS.

- reorganization of additional types of table spaces:
  - — table spaces that contain XML columns that support XML versions, and their associated XML 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.
  - — universal table spaces that are defined as MEMBER CLUSTER

    REORG PLUS natively reorganizes indexes on universal table spaces that are defined as MEMBER CLUSTER.
  - — table spaces that contain pending DDL changes in those cases when the IBM DB2 REORG utility would materialize those changes

    If the DB2 REORG utility would *not* materialize the pending changes, REORG PLUS reorganizes the table space natively (but does not materialize the changes).
  - — table spaces that contain timestamp columns that are defined with a precision other than 6
  - — table spaces that contain timestamp columns that are defined as TIMESTAMP WITH TIME ZONE
  - — table spaces that contain any of the following types of indexes:
    - unique indexes that contain non-key columns
    - indexes that contain a key derived from an expression
    - indexes that contain keys with random ordering
- reorganization of additional types of indexes:
  - — node ID indexes associated with table spaces that contain XML columns that support XML versions
  - — indexes that contain non-key columns
  - — indexes that contain pending DDL changes in those cases when the IBM DB2 REORG utility would materialize those changes

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

- — indexes that contain a key derived from an expression
- — indexes that contain keys with random ordering

NOTE

REORG PLUS invokes DSNUTILB to enable new features quickly. REORG PLUS generally will provide 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.

Enabling REORG PLUS to invoke DSNUTILB

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

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

2. For each feature that requires invoking DSNUTILB, ensure that the installation option that enables the feature is set to YES.

   Table 8 on page 73 lists the option associated with each feature.

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

Option	Required setting	More information
DSNUTILB	YES	page 217 or page 642
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 needed	page 272 or page 691
MAPPINGTABLE	(*for a SHRLEVEL CHANGE reorganization*) a valid mapping table	page 314
SORTNUM	2 or greater	page 174 or page 673

4. 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.

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

For existing jobs, you might need to make changes to your JCL. For example, Table 10 on page 76 describes which options will cause your job to fail when invoking DSNUTILB.

Installation options that enable DSNUTILB

In addition to specifying DSNUTILB=YES, you must specify YES (or DEFAULT, in the case of AUXREORG) for the installation options shown in Table 8 to enable the features for which REORG PLUS invokes DSNUTILB.

Table 8 Installation options that enable DSNUTILB features (part 1 of 2)

Feature	Installation option	See page
ability to reorganize LOB table spaces when reorganizing the base table space **Note**: For this option, you must specify DEFAULT to enable this feature. YES is not a valid value.	AUXREORG	628
base objects that participate in a clone relationship when the base table space instance number is 2	CLONE	631
clone objects	CLONE	631
indexes that contain non-key columns	IXINCLCOL	651
indexes that contain a key derived from an expression, including indexes on inline LOB columns **Note**: An index on an inline LOB column contains a key that is derived from an expression.	IXONEX	651
indexes that contain keys with random ordering	IXRANDOM	652
LOB data when SHRLEVEL NONE or SHRLEVEL CHANGE is in effect	LOB	654
table spaces that contain inline LOB columns **Note**: If the table space also contains an index on the inline LOB column, you must also specify YES for the IXONEX installation option.	INLOB	651
table spaces and indexes that contain pending DDL changes	PENDDDL	663
table spaces that contain XML columns that support XML versions, and associated XML table spaces	XML	687
tables defined as ORGANIZE BY HASH	HASHAX	645
certain temporal tables and indexes on those tables (as described in the list of features for which REORG PLUS invokes DSNUTILB)	TEMPRALDATA	679
timestamp columns defined as TIMESTAMP WITH TIME ZONE	TSTZ	683

Table 8 Installation options that enable DSNUTILB features (part 2 of 2)

Feature	Installation option	See page
timestamp columns defined with a precision other than 6	TSPREC	682
universal table spaces defined with MEMBER CLUSTER	UTSMEM	685

General restrictions

Note the following general restrictions when running a DSNUTILB reorganization:

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

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

- Do not specify FREE=CLOSE for your SYSPRINT data set.

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

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

Data set allocation

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

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

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

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

With the exceptions described in Table 9, 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 9 Dynamic allocation options considerations for a DSNUTILB reorganization

Command option	Considerations
ACTIVE	requires a value of YES for all work files, and for at least the primary copy data set if you specify COPY YES If ACTIVE is NO for these data sets, REORG PLUS issues message BMC50178E and terminates. Additionally, if the DB2 REORG utility requires the discard and LOAD control card data sets, you must specify YES for the ARCHIVE and SYSPUNCH DDTYPEs. If these data sets are required but are not dynamically allocated, the reorganization terminates.
ALLOC	ignores this option DSNUTILB dynamically allocates SORTWK DD names.
AVGVOLSP	ignores this option
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 43 on page 290. Also, be aware that user-defined variables are not valid for a DSNUTILB reorganization.
IFALLOC	ignores this option If you specify DD statements in your JCL for any of the data sets required for this 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

Other REORG PLUS options

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

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

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

Table 10 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" on page 74.

Table 10 Options that are incompatible or translated for a DSNUTILB reorganization (part 1 of 3)

Command option	REORG PLUS response if you include the option
ANALYZE ONLY	issues message BMC50178E and terminates
ANALYZE (all others)	ignores this option
AMENDED	ignores this option
ARCROWS	ignores this option
ASSOCIATE	ignores this option
AVAILPAGEPCT	ignores this option
AVGVOLSP	ignores this option
BMCHIST	ignores this option
BMCSTATS	ignores this option
COPY YES	invokes the copy function
COPYLVL PART	if COPY YES, converts to COPYLVL FULL
DDLDDN	ignores this option
DELETE	issues message BMC50178E and terminates
DSNUEXIT	ignores this option
DSPLOCKS	ignores this option
DSRSEXIT	ignores this option
exitPoint	ignores this option
ICDDN	ignores this option
ICTYPE INCREMENTAL	if COPY YES, converts to ICTYPE UPDATE
IDCACHE	ignores this option
IDCDDN	ignores this option

Table 10 Options that are incompatible or translated for a DSNUTILB reorganization (part 2 of 3)

Command option	REORG PLUS response if you include the option
INLINE NO	ignores this option
KEEPDICTIONARY	NO: ignores this option YES: ■ if reorganizing a LOB table space using SHRLEVEL CHANGE, ignores this option ■ otherwise, passes as KEEPDICTIONARY
LOGMEM	ignores this option
LOGSPILL	ignores this option
LOGTHRESHLD	ignores this option
LONGNAMETRUNC	ignores this option
MAXNEWPARTS	ignores this option
MAXSORTMEMORY	ignores this option
MAXSORTS	ignores this option
MAXTAPE	ignores this option
MINSORTMEMORY	ignores this option
NLPCTFREE	ignores this option
ON FAILURE	ignores this option
ON MESSAGE	ignores this option
ORDER YES ASSOCIATE BYTABLE	ignores this option
ORIGINALDISP	ignores this option
PREFORMAT	NO: ignores this option YES: passes as PREFORMAT
RECOVERYICDDN	ignores this option
REDEFINE YES	passes as REUSE
REGISTER	ignores this option
RIDMAPMEM	ignores this option
SELECT	issues message BMC50178E and terminates
SHORTMEMORY	ignores this option
SHRLEVEL REFERENCE UNLOADONLY	issues message BMC50178E and terminates
SIXSNAP	ignores this option
SORTDEVT	if reorganizing a LOB table space, ignores this option
SPILLDSNPAT	ignores this option
SPILLSTORCLAS	ignores this option
SPILLUNIT	ignores this option
STORCLAS	ignores this option

Table 10 Options that are incompatible or translated for a DSNUTILB reorganization (part 3 of 3)

Command option	REORG PLUS response if you include the option
SYNC	ignores this option
TIMEOUT TERM,*rc*	ignores this option
TERMEXIT	ignores this option
TOTALPAGEPCT	ignores this option
TSSAMPLEPCT	ignores this option
UPDATE	issues message BMC50178E and terminates
UPDATEDB2STATS YES	passes as STATISTICS TABLE (ALL) INDEX (ALL) REPORT YES UPDATE ALL
WTOMSG	ignores this option
XBMID	ignores this option
ZIIP	ignores this option

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.

Concurrent execution of BMC utilities

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 724.

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

The BMCSYNC table allows multiple BMC utilities (or multiple instances of a utility) to operate concurrently on different partitions of a DB2 space if no nonpartitioning indexes are involved. In addition, some BMC utilities can operate concurrently on the same object or partition. For information about which products can operate concurrently, see Table 11 on page 79.

The “Access level” column in Table 11 refers to the value of the **SHRLEVEL** column in the BMCSYNC table. The level can be one of the following values:

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

Table 11 Executing BMC utilities concurrently (part 1 of 2)

Product	Access level	Additional information
CHECK PLUS	S	none
COPY PLUS	S or blank	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).
DASD MANAGER PLUS (BMCSTATS)	S	none
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

Table 11 Executing BMC utilities concurrently (part 2 of 2)

Product	Access level	Additional information
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 execute an IBM utility that attempts to manipulate data within the same objects that a BMC utility is processing.

The setting of the LOCKROW installation option determines whether REORG PLUS uses MVS™ enqueues or SQL LOCK TABLE statements to serialize the BMCSYNC and BMCUTIL tables. For information about the LOCKROW option, see Appendix A, “REORG PLUS installation options.”

Object status for SHRLEVEL NONE

For SHRLEVEL NONE reorganizations (the default), this section describes the initial status requirements, restrictive statuses, 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

If you are using a DDLIN data set or the REBALANCE command option, the database must be in RW status. Otherwise, for SHRLEVEL NONE, REORG PLUS requires the following initial statuses:

- The database associated with the objects that are participating in the reorganization must have one of the following initial statuses:
  — read / write (RW)
  — read-only (RO)

- Any table space or index space that participates in the reorganization must have one of the following initial statuses:

  — read / write (RW)
  — read-only (RO)
  — utility-only (UT)

Restrictive statuses for SHRLEVEL NONE

Table 12 lists the possible restrictive statuses that prevent a successful SHRLEVEL NONE reorganization, depending on the type of reorganization being performed. An X in the table column indicates that the reorganization fails if the object is in the listed status for that type of reorganization.

Table 12 Restrictive statuses that are not permitted for SHRLEVEL NONE

Status that is not permitted	Reorganizations that fail (X)		
	Full table space	Partial table space	Index
AREST (advisory restart pending)	X	X	X
DBETE (DBET error)	X	X	X
GRECP (group RECOVER pending)	X[a]	X	X
LPL (logical page list)	X	X	X
PSRBD (page set REBUILD pending)		X	X
RBDP (REBUILD pending)		X	X
RBDP* (logical part REBUILD pending)		X	X
RECP (RECOVER pending)	X[a]	X	X
REFP (refresh pending)	X	X	X
RESTP (restart pending)	X	X	X
UTRO (utility restrictive state, read-only access allowed)	X	X	X
UTRW (utility restrictive state, read / write access allowed)	X	X	X
UTUT (utility restrictive state, utility exclusive control)	X	X	X
WEPR (write page error range)	X	X	X

[a] The table space cannot be in this status, but the associated indexes can be.

Status changes for SHRLEVEL NONE

Table 13 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 13 Status changes during a SHRLEVEL NONE reorganization

Phase	Status
UTILINIT and ANALYZE	The status of the spaces remains RW, RO, or UT (the initial status).
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. The 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 that 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.

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, restrictive statuses, 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

If you are using a DDLIN data set or the REBALANCE command option with any type of SHRLEVEL, the database must be in RW status. Otherwise, REORG PLUS requires the following initial statuses, depending on the type of SHRLEVEL:

- For SHRLEVEL REFERENCE UNLOADONLY and SHRLEVEL REFERENCE:
  - The database associated with the objects that are participating in the reorganization must have one of the following initial statuses:
    - read / write (RW)
    - read-only (RO)
  - Any table space or index space that participates in the reorganization must have one of the following initial statuses:
    - read / write (RW)
    - read-only (RO)
    - utility-only (UT)
- For SHRLEVEL CHANGE:
  - The database associated with the objects that are participating in the reorganization must be in RW status.
  - Any table space or index space that participates in the reorganization must have one of the following initial statuses:
    - read / write (RW)
    - utility-only (UT)

Restrictive statuses for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE

Table 14 shows the restrictive statuses that REORG PLUS does not permit for the object that is being reorganized, depending on the type of SHRLEVEL that you specify on the command. If the object is in a restrictive status that REORG PLUS does not permit, REORG PLUS terminates the job. An X in a column in Table 14 indicates the following information:

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

Table 14 Restrictive statuses that are not permitted for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE

Status that is not permitted	Reorganizations that fail (X)		
	SHRLEVEL REFERENCE UNLOADONLY	SHRLEVEL REFERENCE	SHRLEVEL CHANGE
AREST (advisory restart pending)	X	X	X
GRECP (group RECOVER pending)		X	X
LPL (logical page list)	X	X	X
PSRBD (page set REBUILD pending)	X	X[a]	X
RBDP (REBUILD pending)	X	X[a]	X
RBDP* (logical part REBUILD pending)	X	X	X
RECP (RECOVER pending)	X	X	X
REFP (refresh pending)	X	X	X
RESTP (restart pending)	X	X	X
UTRO (utility restrictive state, read-only access allowed)	X	X	X
UTRW (utility restrictive state, read / write access allowed)	X	X	X
UTUT (utility restrictive state, utility exclusive control)	X	X	X
WEPR (write page error range)	X	X	X

[a] 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 through Table 17 on page 86 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.

Additional considerations

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

Phase	Status
UTILINIT and ANALYZE	The status of the spaces remains RW, RO, or UT (the initial status).
beginning of UNLOAD	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.
beginning of RELOAD	■ REORG PLUS stops the table space and associated indexes (for a table space reorganization) or the index space (for an index reorganization). ■ Processing continues in the RELOAD and UTILTERM phases as with SHRLEVEL NONE.

Table 16 Status changes during a SHRLEVEL REFERENCE reorganization (part 1 of 2)

Phase	Status
UTILINIT and ANALYZE	The status of the spaces remains RW, RO, or UT (the initial status).
beginning of UNLOAD (two-phase reorganization) *or* beginning of REORG (single-phase reorganization)	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.

Table 16 Status changes during a SHRLEVEL REFERENCE reorganization (part 2 of 2)

Phase	Status
beginning of UTILTERM	■ REORG PLUS stops and sets the RECOVER pending status of the table space and all associated index spaces (for a table space reorganization) or the index space (for an index reorganization). ■ After both of the following steps complete, REORG PLUS starts the table space and index spaces in the status that they were in when the reorganization began, and resets the RECOVER pending status: — the renaming 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.

Table 17 Status changes during a SHRLEVEL CHANGE reorganization

Phase	Status
UTILINIT and ANALYZE	The status of the spaces remains RW, RO, or UT (the initial status).
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.[a]
beginning of 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 renaming or FASTSWITCH operations that are associated with the staging data sets and registering the REORG or COPY row in the SYSIBM.SYSCOPY catalog table, REORG PLUS allows all access to the objects to resume. The table space and index spaces remain in the status that they had when the reorganization began. REORG PLUS then resets the RECOVER pending status.[b]

[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 you are not using the FASTSWITCH process, 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.

DEFINE NO objects

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

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

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

Multi-data-set DB2 objects

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

- For multi-data-set table spaces and index spaces that are VCAT-defined, you must perform one of the following tasks to prevent REORG PLUS from terminating:

  — Predefine all required data sets.

  — Specify REDEFINE YES and include a SYSIDCIN data set that defines the required data sets.

- For storage-group-defined multi-data-set objects, REORG PLUS defines additional data sets if needed for expansion, even if the value of the REDEFINE command or installation option is NO.

- 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 117.

- You can use the DSRSEXIT user exit to redefine a multi-data-set object. For more information, see “Sample DSRSEXIT REXX user exit” on page 790.

Dynamic data set allocation

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

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

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 (BMCCPY, BMCCPZ, BMCRCY, and BMCRCZ)
- incremental copy data sets (BMCICY, BMCICZ, BMCIRY, and BMCIRZ)

For information about allocating a particular data set type, see “REORG PLUS DD statements” on page 323.

Enabling dynamic allocation

To enable and use dynamic data set allocation quickly and simply, specify ACTIVE YES on your REORG PLUS command or in your installation options for each DDTYPE to dynamically allocate. For more options that you can use with dynamic allocation, see “Dynamic allocation options” on page 269.

Running in a worklist environment

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

Generating data set names

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

Names created with DSNPAT

The DSNPAT installation or command option allows you to specify text and variable data for building data set names. If you cannot construct a data set name that meets your organization's standards by using the text and the supplied variables, REORG PLUS provides an exit point that allows you to create your own variables for use with DSNPAT. Sample exits written in assembler, COBOL, C, and LE C are described in Appendix C, "REORG PLUS user exits," and are provided in the *HLQ*.*LLQ*SAMP library. (*HLQ* is the high-level qualifier specified during installation and *LLQ* is the low-level qualifier or prefix set during installation.) For more information about the DSNPAT option, see page 289.

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

GDG names

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

GDG name format

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

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

GDG base

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

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

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

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

- The GDGLIMIT installation or command option allows you to specify the number of generations to keep.

- If the GDGLIMIT value is exceeded, the GDGEMPTY option tells the system to uncatalog either all preexisting generations of this data set or only the oldest generation.

- The GDGSCRATCH installation option tells the system whether to delete the entry that was just uncataloged from the volume's table of contents (VTOC). If the entry is deleted, the space on the volume becomes available to other users.

For more information, see the installation option descriptions in Appendix A, starting on page 699, and see "GDGLIMIT" on page 297.

Specifying ddname prefixes

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

- acceptable set:

```
BMCRD
BMCRDWK
```

- not acceptable set:

```
BMCRD
BMCRD11
```

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

Deleting dynamically allocated data sets

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

You can also specify this preference with the DELFILES installation option. For more information about these options, see page 200 or page 637.

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.

Giving larger data sets different properties than smaller data sets

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

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

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

For more information about the THRESHLD option and the options that it affects, see "THRESHLD" on page 286.

Using SMS ACS routines

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

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

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

Reaching the MAXTAPE limit

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

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

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

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

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

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

Changing options on restart

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

The following restrictions apply to changes that you make to dynamic allocation options before restarting a job:

- You cannot change the value for the ACTIVE option on any restart.

- Changing any option on restart such that it results in different ddnames or a different number of DDs than the original option can produce an error. If you need to change the number of SYSREC and SYSUT1 work files, resubmit the job with a parameter of NEW.

- To change the value of other dynamic data set allocation options, specify RESTART(PHASE).

Work file validity and integrity checks

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

- check for data set attributes

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

- check for data integrity

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

Check for data set attributes

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

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

SYSREC, SYSUT1, and SYSARC data sets

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

FILECHK=FAIL

If your installation options specify FILECHK=FAIL and if 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 the REORG PLUS installation options. 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 following copy data sets are temporary data sets, REORG PLUS continues processing and issues a warning message, regardless of the value of FILECHK:

- BMCCPY*nn*
- BMCCPZ*nn*
- BMCRCY*nn*
- BMCRCZ*nn*

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 following full or incremental copy files are temporary data sets, regardless of the value of FILECHK:

- BMCCPY*nn*
- BMCCPZ*nn*
- BMCRCY*nn*
- BMCRCZ*nn*
- BMCICY*nn*
- BMCICZ*nn*
- BMCIRY*nn*
- BMCIRZ*nn*

Check for data integrity

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

SHRLEVEL considerations

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

- SHRLEVEL NONE (the default), 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

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 REFERENCE UNLOADONLY

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

The objects that you are reorganizing have status requirements that are different from those for SHRLEVEL NONE. For more information, see “Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE” on page 82.

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

- single-phase reorganization
- DSNUTILB reorganization

SHRLEVEL REFERENCE

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

The objects that you are reorganizing have status requirements that are different from those for SHRLEVEL NONE. For more information, see “Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE” on page 82.

SHRLEVEL REFERENCE uses staging data sets as described in “Staging data sets” on page 98.

SHRLEVEL CHANGE

This option allows the table space and associated indexes (for a table space reorganization) or the index space (for an index reorganization) to remain in RW status during all of the reorganization, except for a brief time during the UTILTERM phase when REORG PLUS replaces the original data sets with the staging data sets. For more information, see “Staging data sets” on page 98.

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 82.

For information regarding a SHRLEVEL CHANGE reorganization, see Chapter 7, "Online reorganization."

Staging data sets

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

- REORG PLUS uses the rename process by default. REORG PLUS "replaces" the original data sets with the staging data sets by renaming the VSAM data sets (discussed in "Staging data sets and the rename process" on page 100).

- In the FASTSWITCH process, REORG PLUS bypasses the VSAM rename process and changes the DB2 catalog entry to point to the staging data sets (discussed in "Staging data sets and the FASTSWITCH process" on page 102).

  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.

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.

Use Table 18 on page 99 to determine what action to take, regarding staging data sets, to enable REORG PLUS to perform your job correctly.

Table 18 Staging data set actions

REDEFINE command or installation option	Object you are reorganizing	Action
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.

For a partial reorganization, 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. For information about restart processing, see "Specifying the RESTART and RESTART(PHASE) options" on page 362.

Staging data sets and the rename process

If you use the rename process, you can choose which naming convention you want REORG PLUS to use to allocate the staging data sets. If the value of the STAGEDSN installation option is BMC, REORG PLUS names the staging data sets by replacing the **DSN**DBC node of the cluster and the **DSN**DBD node of the data component with **BMC**DBC and **BMC**DBD. Table 19 illustrates this naming convention.

Table 19 Naming convention for the rename process with STAGEDSN=BMC

Existing data set name	Staging data set name
VCAT.**DSN**DBC.*database*.*tableSpace*.I0001.A*nnn*	*VCAT*.**BMC**DBC.*database*.*tableSpace*.I0001.A*nnn*
VCAT.**DSN**DBD.*database*.*tableSpace*.I0001.A*nnn*	*VCAT*.**BMC**DBD.*database*.*tableSpace*.I0001.A*nnn*
VCAT.**DSN**DBC.*database*.*tableSpace*.J0001.A*nnn*	*VCAT*.**BMC**DBC.*database*.*tableSpace*.J0001.A*nnn*
VCAT.**DSN**DBD.*database*.*tableSpace*.J0001.A*nnn*	*VCAT*.**BMC**DBD.*database*.*tableSpace*.J0001.A*nnn*

If your SMS, RACF, or other data-set-allocation rules require you to use a second-node qualifier of DSNDBC (or DSNDBD) at times, you can set STAGEDSN=DSN. This setting instructs REORG PLUS to use the fifth-node qualifier to create your unique staging data set names. Table 20 illustrates this naming convention.

Table 20 Naming convention for the rename process with STAGEDSN=DSN

Existing data set name	Staging data set name
VCAT.**DSN**DBC.*database*.*tableSpace*.**I**0001.A*nnn*	*VCAT*.**DSN**DBC.*database*.*tableSpace*.**J**0001.A*nnn*
VCAT.**DSN**DBD.*database*.*tableSpace*.**I**0001.A*nnn*	*VCAT*.**DSN**DBD.*database*.*tableSpace*.**J**0001.A*nnn*
VCAT.**DSN**DBC.*database*.*tableSpace*.**J**0001.A*nnn*	*VCAT*.**DSN**DBC.*database*.*tableSpace*.**I**0001.A*nnn*
VCAT.**DSN**DBD.*database*.*tableSpace*.**J**0001.A*nnn*	*VCAT*.**DSN**DBD.*database*.*tableSpace*.**I**0001.A*nnn*

Data set rename process

After writing the reorganized objects to the staging data sets, REORG PLUS renames the data sets by using the following process. Figure 6 on page 102 illustrates this process.

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

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

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

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

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

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

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

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

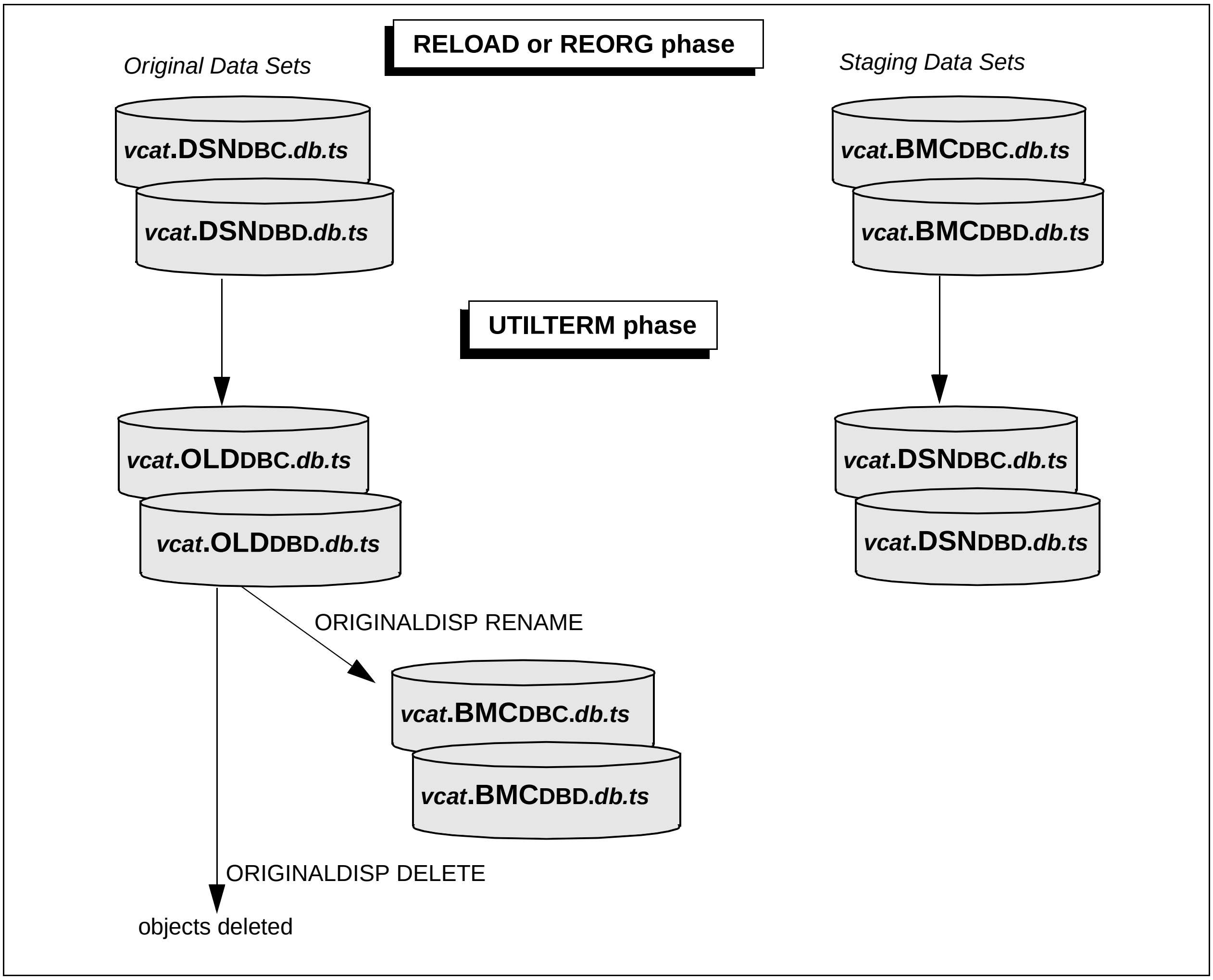
   - If the value for the ORIGDISP or ORIGINALDISP option is RENAME, REORG PLUS renames the original data sets to the staging data set names, depending on the value of STAGEDSN, as follows:

     — If STAGEDSN=BMC, REORG PLUS replaces **OLD** with **BMC** in the names.

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

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

Figure 6 Data set rename process when STAGEDSN=BMC



Tasks to perform if your reorganization job terminates

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

Staging data sets and the FASTSWITCH process

If you are using FASTSWITCH processing and you want to preallocate or define the staging data sets, you must use the original DB2 VSAM data set names as a starting point, but replace the **I** qualifier with **J**, or the **J** qualifier with **I**. Table 21 on page 103 illustrates this naming convention (which is equivalent to specifying STAGEDSN=DSN).

Table 21 Naming convention for the FASTSWITCH process

Existing data set name	Staging data set name
VCAT.DSNDBC.*database.object*.I0001.A*nnn*	*VCAT*.DSNDBC.*database.object*.J0001.A*nnn*
VCAT.DSNDBD.*database.object*.I0001.A*nnn*	*VCAT*.DSNDBD.*database.object*.J0001.A*nnn*
VCAT.DSNDBC.*database.object*.J0001.A*nnn*	*VCAT*.DSNDBC.*database.object*.I0001.A*nnn*
VCAT.DSNDBD.*database.object*.J0001.A*nnn*	*VCAT*.DSNDBD.*database.object*.I0001.A*nnn*

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 104 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.

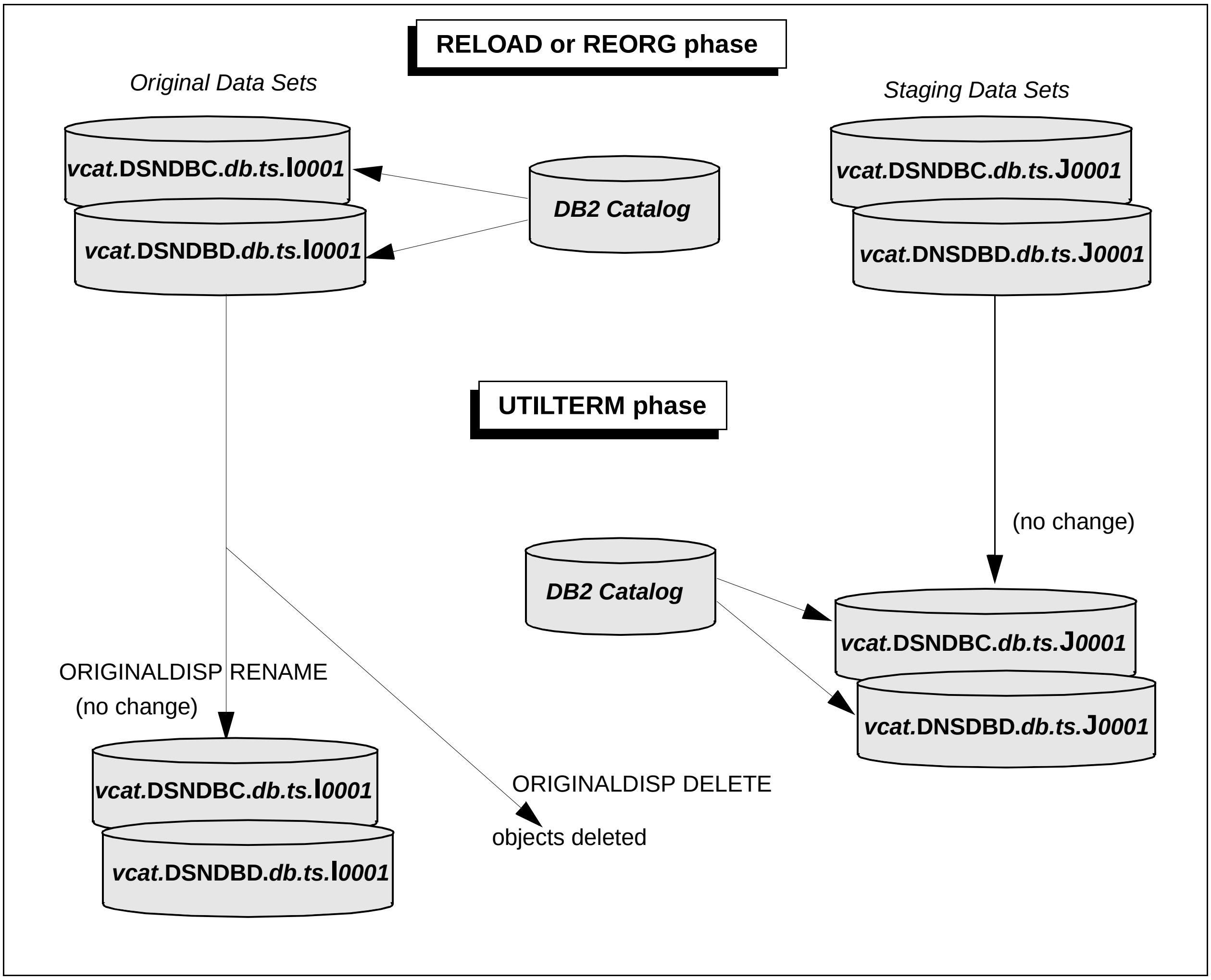
3. The next step depends on the value for the ORIGDISP or ORIGINALDISP option:

   - 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 “Not completing in the UTILTERM phase” on page 370.

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 command and installation option values

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

- If you need to make copies, consider one of the following options to avoid encountering data set allocation restrictions of the operating system or REORG PLUS memory restrictions:

  — Limit the number of copies per partition.

  — If your site's recovery strategy allows for full copies, use one of the following options:

    - If you are reorganizing all partitions, create a single copy by specifying COPYLVL FULL.

    - If you are reorganizing a subset of partitions, create a single copy by specifying a single physically contiguous range of partitions, COPYSUBSET=YES in the installation options module, and COPYLVL FULL on your REORG command.

  For more information, see the command option "COPYLVL" on page 246 and the COPYSUBSET installation option on page 634.

- If you dynamically allocate partition-level copies, REORG PLUS appends the partition number to the default ddname prefix. If you dynamically allocate partition-level copy data sets for more than 99 partitions, ensure that you change the value of the COPYDDN installation or command option to specify a ddname prefix; the prefix plus the highest partition number must not exceed eight characters. If you are also dynamically allocating remote copies, change the value of the RECOVERYDDN command option (or RCVYDDN installation option) to specify a ddname prefix for your remote copy data sets; the prefix plus the highest partition number must not exceed eight characters. For more information, see the command options "COPYDDN" on page 248 and "RECOVERYDDN" on page 250 or the COPYDDN and RCVYDDN installation options on page 632 and page 665.

- 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. For more information, see the command option "REDEFINE" on page 195 or the REDEFINE installation option on page 665.

- Ensure that the value of your LOCKROW installation option is YES. For more information about the LOCKROW installation option, see page 655.

- Consider using VOLCNT (AUTO,AUTO) rather than the default of 25 to avoid data set allocation limitations when 25 is excessive. For more information about the VOLCNT installation option, see page 707.

Constrained resources

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

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.

XML data

When you are running on DB2 Version 9 or later, REORG PLUS provides the capability to reorganize XML data.

Considerations when reorganizing the base table space

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

XML columns that support XML versions

If the XML column supports XML versions, REORG PLUS invokes DSNUTILB to reorganize the table space. For more information, see “Reorganization jobs that invoke DSNUTILB” on page 70.

NOTE

REORG PLUS natively reorganizes indexes on these table spaces when you specify REORG INDEX.

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. For more information about these options, see page 232 or page 648.

Authorization

For authorization requirements that apply when reorganizing tables that contain an XML column, see “Additional authorizations for XML reorganizations” on page 66.

Considerations when reorganizing the XML table space

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

NOTE

You can also reorganize the indexes on an XML table space with the REORG INDEX command option. For information about authorization requirements when reorganizing a user-defined XML index, see “Additional authorizations for XML reorganizations” on page 66.

XML columns that support XML versions

If the associated XML column supports XML versions, REORG PLUS invokes DSNUTILB to reorganize the XML table space and to reorganize the node ID index of the XML table space. For more information, see “Reorganization jobs that invoke DSNUTILB” on page 70.

NOTE

REORG PLUS natively reorganizes document ID and user-defined indexes associated with these table spaces when you specify REORG INDEX.

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 exist:

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

NOTE

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

Partition-by-growth table spaces

If your XML table space is a partition-by-growth table space, see "Partition-by-growth table spaces" on page 118 for additional considerations.

LOB data

This section provides instructions and considerations for reorganizing LOB data.

Reorganizing the base table space and its indexes

How REORG PLUS handles reorganizing the base table space (and its indexes) for LOB data depends on the type of reorganization and whether the table space includes inline LOB data. Table 22 on page 111 describes these conditions and results.

Table 22 How REORG PLUS reorganizes base table spaces and their indexes for LOB data

Type of object	Inline LOB data?	How REORG PLUS handles the request
index on an inline LOB column, or a table space that contains this index	yes	REORG PLUS invokes DSNUTILB to reorganize the index or table space.
any other base table space	no	REORG PLUS reorganizes the table space natively.
	yes	REORG PLUS invokes DSNUTILB to reorganize the table space.
any other index on the base table space	either	REORG PLUS reorganizes the index natively.

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 70.

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

2 Ensure that the installation and command options are set appropriately for a DSNUTILB reorganization job, as described in "Reorganization jobs that invoke DSNUTILB" on page 70.

3 For a REORG TABLESPACE, ensure that the value of the INLOB installation option is YES.

For more information about this option, see page 651.

4 In either of the following cases, also ensure that the value of the IXONEX installation option is YES:

- You specify REORG INDEX.
- You specify REORG TABLESPACE and an index on that table space is defined on an inline LOB column.

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

Create your job as usual, making any necessary adjustments based on the information in "Considerations when reorganizing the base table space" on page 112.

Considerations when reorganizing the base table space

The following considerations apply when you are reorganizing a table space that contains a LOB column:

- 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 a LOB column

  — a WHERE clause that references a LOB column
- You cannot alter a partition limit key on a table space that contains a LOB column.
- If you are reorganizing a partition-by-growth table space that contains a LOB column, see "Partition-by-growth table spaces" on page 118 for additional considerations.

Reorganizing LOB table spaces

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

- For SHRLEVEL REFERENCE, REORG PLUS natively reorganizes the LOB table space and updates the auxiliary index.
- For SHRLEVEL NONE and SHRLEVEL CHANGE, REORG PLUS invokes DSNUTILB to reorganize the LOB table space.

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 SHRLEVEL NONE or SHRLEVEL CHANGE is in effect

1. Ensure that you meet the requirements for a DSNUTILB job, as described in "Reorganization jobs that invoke DSNUTILB" on page 70.

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

2. Ensure that the installation and command options are set appropriately for a DSNUTILB reorganization job, as described in “Reorganization jobs that invoke DSNUTILB” on page 70.

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

   For more information about this option, see page 654.

To reorganize LOB table spaces when SHRLEVEL REFERENCE is in effect

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

Considerations when reorganizing LOB table spaces (SHRLEVEL REFERENCE)

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

Object status requirements

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

Data set requirements

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

Incompatible REORG PLUS command options

Some REORG PLUS command options are not available when you are reorganizing a LOB table space. Table 23 on page 114 describes the options that are not available and how REORG PLUS responds if your job contains these options.

Table 23 Command options that are incompatible when reorganizing LOB table spaces

Command option	REORG PLUS response
ANALYZE BMCSTATS	ignores this option
ANALYZE ANALYZE SAMPLE ANALYZE SCAN	does not sample or scan, but adds high-used relative byte addresses (HURBAs) from all data sets to estimate copy data set size
BMCSTATS YES	changes this option to BMCSTATS NO
CONDEXEC BMC CONDEXEC YES	issues message BMC50056E and terminates
DELETE	issues message and terminates
INDREFLIMIT	issues message BMC50056E and terminates if specifying this option would cause REORG PLUS to check criteria for conditional reorganization
OFFPOSLIMIT	issues message BMC50056E and terminates if specifying this option would cause REORG PLUS to check criteria for conditional reorganization
ORDER YES	changes this option to ORDER NO
SELECT	issues message and terminates
SHRLEVEL CHANGE	issues message BMC50115E and terminates
UNLOAD CONTINUE	changes this option to UNLOAD RELOAD
UNLOAD PAUSE	issues message BMC50115E and terminates
UPDATE	issues message and terminates
UPDATEDB2STATS YES	changes this option to UPDATEDB2STATS NO

Floating-point data

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

Partial reorganization

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

General considerations

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

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

- Because REORG PLUS does not rebuild indexes during a partial reorganization, it cannot update a row change timestamp column when a row is modified if the column that is implicitly updated is part of a nonpartitioned index key.

- REORG PLUS does not collect statistics for nonpartitioned indexes when running a partial reorganization.

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

SHRLEVEL NONE considerations

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

- REORG PLUS stops and starts only those partitions specified with the PART option of the REORG command. REORG PLUS stops or starts nonpartitioned indexes in their entirety.

- If a failure occurs after REORG PLUS starts updating one of the following indexes, you can restart the reorganization, but the index will be left in RBDP status after the restarted job completes:

  — a compressed, non-unique, nonpartitioned index

  — a document ID index for which REORG PLUS has generated document ID values

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

SHRLEVEL REFERENCE or SHRLEVEL CHANGE considerations

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

- If nonpartitioned indexes are defined, REORG PLUS copies the original nonpartitioned indexes to the staging data sets during the UNLOAD or REORG phase. For more information, see “Instant Snapshot with nonpartitioned indexes.”

- 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 the supported intelligent storage devices and you must specify YES or AUTO for the SIXSNAP command or installation option. For a list of supported devices, see the *EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide*.

Table 24 shows how the value of the SIXSNAP command or installation option determines the action that REORG PLUS takes if a copy failure occurs.

Table 24 SIXSNAP and Instant Snapshot

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

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 118.

Table 25 shows how the REDEFINE command or installation option can also impact the type of copies that REORG PLUS makes.

Table 25 REDEFINE and SIXSNAP

REDEFINE value	SIXSNAP value	REORG PLUS action
NO	AUTO	changes the value of SIXSNAP to NO and makes a software-based copy
	YES	terminates the job
YES	AUTO or YES	performs the SIXSNAP function

For more information about invoking Instant Snapshot, see "SIXSNAP" on page 228. For information about the XBM and SUF requirements, see "XBM and SUF considerations" on page 140.

Considerations for SIXSNAP use

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

Timeout considerations

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

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

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

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

Hardware vendor considerations

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

- **SMS controlled**

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

- **EMC**

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

- **STK**

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

Partition-by-growth table spaces

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

NOTE

If you are reorganizing partition-by-growth table spaces that are defined with MEMBER CLUSTER, REORG PLUS invokes DSNUTILB. The information in this section does not apply in this case.

The strategy that REORG PLUS uses to place rows within the table space depends on whether the table contains a LOB column, as follows:

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

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

When processing partition-by-growth table spaces, REORG PLUS starts at most one unload, reload, and reorg task per range of contiguous partitions. The task that handles the last partition handles the partitions that might be added during the reorganization process.

Partition-by-growth table spaces have special considerations for compression dictionaries, restarting, copy data sets, and SYSREC data sets. For more information, see the following sections:

- “Building a dictionary” on page 136
- “Partition-by-growth table spaces” on page 363
- “Copy data sets” on page 326
- “SYSREC data sets” on page 349

Table space extension by REORG PLUS

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

- The table has no LOB columns.

- You include the last partition in the reorganization, either by running a full table space reorganization, or by specifying it using the PART option (page 163).

- The value of the DB2 MAXPARTITIONS parameter is greater than the number of partitions in the table space.

- The value of the MAXNEWPARTS option (page 164) 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 331.
- REORG PLUS does not collect statistics for partitions that are added during the reorganization.

Table space extension by DB2

DB2 might add partitions during a SHRLEVEL CHANGE reorganization of a partition-by-growth table space or an index defined on its table. REORG PLUS can support added partitions up to the limit specified by the MAXNEWPARTS option (page 164). If the number of partitions that DB2 adds exceeds MAXNEWPARTS, REORG PLUS terminates and issues message BMC53025E.

Note the following considerations when DB2 adds partitions to a partition-by-growth table space during a REORG PLUS reorganization:

- 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 331.

- REORG PLUS does not collect statistics for partitions that are added during the reorganization.

Partition rebalancing

REORG PLUS supports partition rebalancing in the following ways:

- REORG PLUS reorganizes partitions that have had the values of their partition keys previously altered and are now in REORG pending (REORP) status.
- REORG PLUS reorganizes logically contiguous partitions for any type of SHRLEVEL table space reorganization where you alter the partition's limit key during the reorganization by using the DDLIN data set (defined by the DDLIN DD statement). For more information, see "DDLIN data set" on page 124.

- 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 167.

To rebalance partitions

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

2. Include a SYSARC data set, if needed.

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

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

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

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

5. Specify COPY YES on the REORG command.

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

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

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

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

General restrictions and considerations

The following general restrictions and considerations apply to partition rebalancing:

- You cannot perform partition rebalancing on a table space if the table space or an index on the table space has pending DDL changes. REORG PLUS issues message BMC51334E and terminates.
- When you include both a DDLIN data set and the REBALANCE command option in a reorganization job, DDLIN data set ALTER statements override the REBALANCE option for the partitions specified in the DDLIN ALTER statements.
- REORG PLUS processes all related partitions in the same unload task. Consequently, the task distribution might not be as even as in a normal reorganization.
- You can rebalance up to 255 ranges of logically contiguous partitions in a single execution of REORG PLUS.
- With SHRLEVEL CHANGE, you can use the DDLOUT DD statement to obtain only a report (without completing the reorganization) of the ALTER statements that REORG PLUS would use to rebalance partitions. For more information, see "DDLOUT data sets" on page 338.
- For SHRLEVEL CHANGE and SHRLEVEL REFERENCE, you can use the ALTRFAIL installation option (page 626) to tell REORG PLUS how to handle failures that occur when processing limit-key ALTER statements. This capability applies to ALTER statements that are processed either from your DDLIN data set or as a result of the REBALANCE option.

Limit key considerations

REORG PLUS supports the following data types in limit keys when you rebalance partitions:

- CHAR
- DATE
- DECIMAL
- INTEGER
- SMALLINT
- BIGINT
- TIME
- TIMESTAMP
- VARCHAR

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

- The following types of constants as limit key values:

  — floating point
  — LOBs
  — 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.

- Partition limit key alters on tables that contain LOB columns (because DB2 does not support them).

Table space considerations

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
  — XML table spaces

- 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, using an internal format. You can use the BMC LOADPLUS for DB2 product 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.

- For table spaces that contain an XML column, REORG PLUS terminates if either of the following conditions exists:
  - — You specify the REBALANCE command option.
  - — A DDLIN data set in your JCL contains 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 that is defined with the LARGE or DSSIZE attribute

- 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

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

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

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

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

DDLIN data set

The DDLIN data set should contain only an SQL ALTER INDEX statement or, for table-controlled partitioning, an ALTER TABLE statement for each limit key that you want to change. REORG PLUS ignores ALTER statements that do not apply to the current reorganization. Thus, you can use a single DDLIN data set for multiple jobs. For more information about this data set, see “DDLIN data set” on page 333.

REORG PLUS does not explicitly execute the SQL statements. Instead, REORG PLUS extracts the partition number and the associated limit key values from the statements. REORG PLUS reorganizes the data in the partitions based on the new key values and alters the limit keys during the UTILTERM phase, thus rebalancing the data in the partitions.

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

When it processes 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 335 and “Sample ALTER TABLE statements and resulting messages” on page 336.

Additional considerations

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

- 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 SYSREC*nn* DD statements for these types of partitions in the JCL. Continue to allow REORG PLUS to dynamically allocate all other SYSREC*nn* data sets.

- You can use a previously created DDLOUT data set as DDLIN input. For more information about this data set, see “DDLOUT data sets” on page 338.

REBALANCE option

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

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.

Partition number	
Logical	Physical
1	1
2	2
3	3
4	4

You then alter the table space to rotate the first partition to last, resulting in the following mapping of logical and physical partitions:

Partition number	
Logical	Physical
1	2
2	3
3	4
4	1

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

Partition number	
Logical	Physical
1	2
2	3
3	4
4	1
5	5

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

Partition number	
Logical	**Physical**
1	3
2	4
3	1
4	5
5	2

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.

```
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 REORG PLUS performs the reorganization only if it is needed. The CONDEXEC command and installation options tell REORG PLUS whether to check the criteria for a conditional reorganization. The CONDEXEC options also tell REORG PLUS whether to use information from the DB2 catalog (CONDEXEC YES) or the DASD MANAGER PLUS exceptions table (CONDEXEC BMC). Using the exceptions table allows you to set more conditions that might cause the reorganization to be performed. However, the BMC value is available only if you are using REORG PLUS as part of the Database Performance for DB2 solution. For more information about the BMC value, see “Using the DASD MANAGER PLUS exceptions table” on page 130.

NOTE

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

Using the DB2 catalog

After REORG PLUS determines that it will check to see if a conditional reorganization should be performed by using the DB2 catalog, it uses the limit command and installation options along with the CONDEXEC option to determine what action to take. The limit command options are OFFPOSLIMIT, INDREFLIMIT, and LEAFDISTLIMIT, and the limit installation options are OFFPOSLM, INDREFLM, and LEAFDSLM.

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. If you want REORG PLUS to issue the report with the recommendations but without performing any reorganizations, specify REPORTONLY in the command.

Table 26 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

CONDEXEC value	REORG PLUS action	Reference
command option is NO	ignores all limit command and installation option values and performs the reorganization	NA
command option is YES	uses the limit command or installation option values to determine whether to perform the reorganization	Table 27
installation option is NO	ignores the limit installation options and performs the reorganization unless a limit command option is specified	
installation option is YES	uses the limit installation options to determine whether to perform the reorganization, unless those options are overridden by command options	
command or installation option is BMC	ignores all limit command and installation options and uses the DASD MANAGER PLUS exceptions table to determine whether to perform the reorganization BMC is valid only if you are using REORG PLUS as part of the Database Performance for DB2 solution.	"Using the DASD MANAGER PLUS exceptions table" on page 130

Limit options

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

Table 27 on page 130 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

Limit command option	Limit installation option value	REORG PLUS action
specified with no value	*integer* or NONE	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.
specified with a value	*integer* or NONE	uses the command option value and does not consider any limit option that is not on the command
(nothing specified)	*integer*	honors the limit installation option and uses its value
(nothing specified)	NONE	honors the values on the other limit installation options but does not consider the option with NONE specified

The command syntax for CONDEXEC and the limit options begins on page 210. For a description of the installation options, see Appendix A, “REORG PLUS installation options.”

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 examines the DASD MANAGER PLUS exceptions table to determine whether the reorganization is needed. If REORG PLUS finds an exception for the object to be reorganized and the exception is active, REORG PLUS reorganizes the object. It then updates the exceptions table to indicate that the exception is not active. If REORG PLUS does not find an exception for the object, it does not perform the reorganization.

Using the DASD MANAGER PLUS exceptions table allows REORG PLUS to use an expanded set of conditions to trigger a conditional reorganization. Also, 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. If you want REORG PLUS to issue the report with the exceptions but not to perform any reorganizations, also specify REPORTONLY in the command. For a description of the CONDEXEC command option, see page 210. For a description of REPORTONLY, see page 216.

NOTE

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

Table 28 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 28 Exceptions that initiate a reorganization (part 1 of 3)

Reorganization type	Value in the exceptions table	DASD MANAGER PLUS field name	Brief description of the exception
either table space or index	CARD	Card	percent increase in the cardinality of a table or index
	DSEXTENT	Dsextents	number of extents at the data set level
	EXTENTS	Extents	table spaces or indexes that have reached a certain number of extents (evaluated at the partition level)
	REORMDEL	Mass del reorg	mass deletion (evaluated at the partition level)
	REORMODS	Mods since reorg	initiates a reorganization after a large number of rows have been modified (evaluated at the partition level)
	REORSPAC	Reorgspace	percentage of space used that is over or under the minimum space required for the object
	SPACE	Space	percentage of change in tracks that an object uses
	BMCRUSER	BMCRUSER	user-defined exception in DASD MANAGER PLUS **Note**: For information about setting up a user-defined exception and naming it BMCRUSER, see the *DASD MANAGER PLUS for DB2 User Guide*.

Table 28 Exceptions that initiate a reorganization (part 2 of 3)

Reorganization type	Value in the exceptions table	DASD MANAGER PLUS field name	Brief description of the exception
table space	FARIND	Farind	percentage of rows in a table space that are more than 16 pages from their original page
	FAROFF	Faroff	percentage of rows that are more than 16 pages from the optimal position (an index-based trigger) This exception is evaluated only for clustering indexes. A high Faroff percentage indicates that clustering might be degrading.
	PACTHI	PctActivHi	table spaces in which the percentage of active pages is greater than or equal to the specified value
	PACTLO	PctActivLo	table spaces in which the percentage of active pages is less than or equal to the specified value
	PCTCLUS	PctClus	cluster ratio of the object (an index-based trigger)
	PCTDROP	Pct Dropped Rows	percentage of space that is occupied by dropped rows
	REORPEND	Reorg Pend	indication that the partition or object is in REORG pending status
	TOTALIND	Totalind	percentage of rows in a table that are not in their optimal positions
	TOTALOFF	Totaloff	percentage of table space rows that are not in optimal position by the index key (an index-based trigger)

Table 28 Exceptions that initiate a reorganization (part 3 of 3)

Reorganization type	Value in the exceptions table	DASD MANAGER PLUS field name	Brief description of the exception
index	APPNDINS	Append inserts	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
	AREOPEND[a]	Advisory pending	whether the index is in advisory REORG pending status
	LEAFDIST	Leafdist	the distance in page IDs between successive leaf pages during a sequential access of the index
	LEAFFOFF	LeafFarOff	percentage of leaf pages that are physically located far from the previous leaf page
	LEAFTOFF	LeafTotOff	percentage of leaf pages that are not in optimal position
	LEVELINC	LevelInc	increase in the number of index levels
	LEVELMIN	LevelMin	number of index levels that are greater than the minimum number required
	LEVELS	Levels	number of index levels
	NUNIFORM	NonUniform	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.
	PSEUDODL	Pseudo Deleted Key	percentage of keys that have been pseudo-deleted within an index partition (evaluated at the partition level)
	ROWS/KEY	Rows/Key	average number of rows per key value

DB2 real-time statistics

REORG PLUS resets the real-time statistics and updates timestamp values (in both the DSNRTSDB and DB2 memory) during the UTILTERM phase. Table 29 on page 134 lists the columns in tables SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS that REORG PLUS resets for each partition and the updated value.

NOTE

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

Table 29 Real-time statistics updates

Table	Column	Updated value
SYSIBM.SYSTABLESPACESTATS[a]	UPDATESTATSTIME	timestamp of the update
	TOTALROWS	actual value
	NACTIVE	actual value
	SPACE	actual value
	EXTENTS	actual value
	COPYLASTTIME	timestamp of the update when REORG PLUS took the copy
	COPYUPDATEDPAGES	zero when a copy is taken
	COPYCHANGES	zero when a copy is taken
	COPYUPDATELRSN	null when a copy is taken
	COPYUPDATETIME	null when a copy is taken
	REORGLASTTIME	timestamp of the reset
	REORGINSERTS	zero
	REORGDELETES	zero
	REORGUPDATES	zero
	REORGDISORGLOB	zero
	REORGUNCLUSTINS	zero
	REORGMASSDELETE	zero
	REORGNEARINDREF	zero
	REORGFARINDREF	zero
SYSIBM.SYSINDEXSPACESTATS[a, b]	UPDATESTATSTIME	timestamp of the reset
	TOTALENTRIES	actual value
	NLEVELS	actual value
	NACTIVE	actual value
	SPACE	actual value
	EXTENTS	actual value
	REORGLASTTIME	timestamp of the reset
	REORGINSERTS	zero
	REORGDELETES	zero
	REORGAPPENDINSERT	zero
	REORGPSEUDODELETES	zero
	REORGMASSDELETE	zero
	REORGLEAFNEAR	zero
	REORGLEAFFAR	zero
	REORGNUMLEVELS	zero

[a] In DB2 Version 8, these table names are SYSIBM.TABLESPACESTATS and SYSIBM.INDEXSPACESTATS.

[b] For nonpartitioned indexes in partial reorganizations, REORG PLUS updates only the SPACE and EXTENTS columns of this 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 using the single REORG phase with SHRLEVEL NONE (the default), you can omit the SYSREC data set for a table space reorganization or the SYSUT1 data set for an index reorganization. Omitting the data set provides a performance boost. However, if you omit the data set, your job might not be restartable. For more information about the requirements for the SYSREC data set, see "SYSREC data sets" on page 349.

- When using the single REORG phase with SHRLEVEL REFERENCE, REORG PLUS does not use the SYSREC data set for a table space reorganization or the SYSUT1 data set for an index reorganization (even if specified). However, the job is restartable due to the nondestructive nature of this type of reorganization. For information about restarting the job during a single-phase reorganization, see "Restarting REORG PLUS" on page 361.

- When using the single REORG phase with SHRLEVEL CHANGE, REORG PLUS does not use the SYSREC data set for a table space reorganization or the SYSUT1 data set for an index reorganization (even if specified). 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.

Table space compression

This section describes how REORG PLUS handles table space compression.

The KEEPDICTIONARY installation option (page 652) 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 exist:

- The value of KEEPDICTIONARY is NO.

- The value of KEEPDICTIONARY is YES, and one of the following conditions is in effect:

  — A dictionary does not exist.

  — REORG PLUS is converting your data from BRF to RRF and the value of the DB2 subsystem parameter HONOR_KEEPDICTIONARY is NO.

- You are rebalancing partitions of a table space with compression. (For more information about partition rebalancing, see page 120.)

After REORG PLUS builds the dictionary, REORG PLUS uses it to compress the data rows. REORG PLUS stores the dictionary in the BMCDICT table until the reorganization completes.

NOTE

REORG PLUS stores a dictionary in BMCDICT for each partition that you are reorganizing. If you are reorganizing a table space with a large number of partitions, you might need to increase the size of the BMCDICT table space from the standard size that was allocated during installation. For more information about the BMCDICT table, see page 715.

Note the following considerations about building a dictionary:

- REORG PLUS reserves enough pages for the largest dictionary size.
- If REORG PLUS does not unload enough rows to build a complete dictionary, REORG PLUS

  — does not compress any rows

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

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

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

Keeping a dictionary

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

Index compression

The following considerations apply to index compression:

- When the following conditions exist, you must specify REDEFINE YES:
  - Before the reorganization, an ALTER occurred that changed the compression attribute of a participating index.
  - The DSVCI system parameter is set to YES, indicating that the control interval (CI) size is variable.
- Compressed indexes have additional restart considerations. For more information, see "Compressed indexes" on page 364.

Reordered row format

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

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

Considerations

The following considerations apply to RRF support in REORG PLUS:

- If any table in the table space has an EDITPROC or VALIDPROC, REORG PLUS does not perform the conversion for the table space.
- REORG PLUS does not support mixed-format partitions in a single reorganization when either of the following conditions exists:
  - You are running DB2 Version 9 in either conversion mode* or enabling-new-function mode* (after falling back from new-function mode).
  - You are running DB2 Version 9 in new-function mode or a later version of DB2, but you have disabled RRF.

  Under these conditions, 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 the following conditions exist:

  — REORG PLUS is running on a subsystem that has been migrated directly from DB2 Version 8 to DB2 Version 10.

  — The subsystem has not yet been migrated to DB2 Version 10 new-function mode.

Recoverability of the reorganized table space

If you specify COPY NO (or do not specify a COPY option), you must take some action to ensure that DB2 can recover the table space, if necessary, after the reorganization. For example, you can

- Create an image copy by running COPY PLUS or the IBM DB2 COPY utility.
- Execute DSN1COPY or any other suitable substitute that your system provides.

If you use the table space as a read-only table space, specify COPY NO or specify COPY YES REGISTER NONE and DD DUMMY for the copy data sets.

Referential integrity

REORG PLUS performs no checking for referential constraint violations, nor does it set the CHKP status for reorganizations that result in violations. To maintain referential integrity, consider the following information:

- If you use the UPDATE option and update a column that is part of a foreign key, ensure that the new column value does not violate any referential constraints.

- If you use the SELECT or DELETE option to delete rows from a primary table, you must ensure that the dependent rows are deleted as well.

DB2 user exits

REORG PLUS invokes EDITPROCs during processing. If the value of the UXSTATE installation option is SUP, REORG PLUS invokes the exit in supervisor state (and PSW key=7). If you can ensure that all EDITPROCs called by REORG PLUS are able to run in problem state, you can specify UXSTATE=PROB in the REORG PLUS installation options module.

XBM and SUF considerations

To enhance performance, REORG PLUS uses several features of XBM or SUF during portions of the reorganization process. If you have installed the required version of XBM or SUF, REORG PLUS can use the following XBM or SUF functions:

- software snapshot functions for a SHRLEVEL CHANGE reorganization

  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.

- 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 the Database Performance for DB2 solution, your license authorizes you to use SUF, not the full version of XBM.

For more information, see the following documents:

- For information about installing SUF and XBM, see the *EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE Installation Guide* or the *Database Performance for DB2 Installation Guide*.

- For information about the various types of snapshot technology, the devices that support Instant Snapshot, zIIP processing options, and corresponding restrictions, see the *EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide*.

Using XBM or SUF with SHRLEVEL CHANGE

By using the software snapshot functions of XBM or SUF when performing a SHRLEVEL CHANGE reorganization, REORG PLUS can reorganize and apply the log records to a consistent image of the data while the data is available for RW operations. Using XBM or SUF also allows you to monitor and alter the execution of the reorganization by using the MVS console or the XBM Utility Monitor.

To use XBM or SUF with a SHRLEVEL CHANGE reorganization

Ensure that you have a currently supported version of XBM or SUF, and then complete the following steps:

1 Create and activate a management set that contains the appropriate snapshot object definition.

2 Create and activate a configuration that contains the appropriate cache attributes.

3 Specify SHRLEVEL CHANGE on the REORG command statement (see page 162).

4 Ensure that a value is specified for the XBMID installation or command option (see page 230 or page 686).

5 Specify any additional SHRLEVEL CHANGE options as needed (as discussed on page 300).

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).

To use Instant Snapshot technology to copy nonpartitioned indexes

1. Ensure that you have a currently supported version of XBM or SUF as described in “Software requirements” on page 63.

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

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

   For information about the SIXSNAP command option, see “SIXSNAP” on page 228.

Using XBM or SUF to enable zIIP processing

You can use a component of XBM or SUF to enable zIIP processing during your reorganization.

To enable zIIP processing

1. Ensure that you have a version of XBM or SUF that supports zIIP processing, as described in “Software requirements” on page 63.

2. Ensure that an XBM subsystem is started.

3. (*optional*) Use the XBMID installation or command option to specify the XBM subsystem that you want REORG PLUS to access.

   For information about the XBMID command option, see “XBMID” on page 230.

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

   For information about the ZIIP command option, see “ZIIP” on page 229.

DB2 features that REORG PLUS does not support

This version of REORG PLUS does not support, or provides limited support, for the following features of DB2.

- REORG PLUS does not support the following features:

  — 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.

Chapter 3

Syntax of the REORG command

This chapter presents the following topics:

Command syntax rules for REORG PLUS

The following general rules apply to the REORG command syntax:

- In a REORG command, REORG PLUS considers any line beginning with an asterisk (*) in column 1 to be a comment and ignores it.

- When encountering two consecutive hyphens in a line (except within a delimited token), REORG PLUS considers everything on that line after the hyphens to be a comment and ignores the remainder of the line.

- If you specify the same command option more than once, REORG PLUS uses only the last option that you specify. For example, if you specify the following options, REORG PLUS accepts UNLOAD CONTINUE as the processing option:

```
UNLOAD RELOAD UNLOAD CONTINUE
```

- When you use a signed token, do not place a space between the sign and the value.

- You can split a token (such as a keyword, identifier, or constant) across a line. However, REORG PLUS ignores anything in columns 73 through 80.

  A blank, or a delimiter if the token is delimited, indicates the end of the token. If an undelimited token ends in column 72, column 1 on the next line must be blank.

NOTE

If you use applications that automate JCL submission and resolve symbolic variables within your JCL, the resulting control cards might not appear as they do in the JCL that you created. These applications might produce an invalid command statement.

- In the syntax diagrams in the following pages, underlined options indicate default options. For more information about how to read syntax diagrams, see “Syntax diagrams” on page 23.

- When specifying object names, you can specify objects that contain Unicode data. However, you cannot specify delimited object names that do not have a character representation in EBCDIC.

Alphabetical listing of REORG PLUS command options

As a quick reference, Table 30 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 30 REORG PLUS command options (part 1 of 4)

Command option	See page	Valid for REORG INDEX?
ACTIVE	272	yes
ALLOC	275	yes
AMENDED	199	no
ANALYZE	185	yes
ARCHDDN	172	no
ARCROWS	209	no
ASSOCIATE	177	no
AUTOESTSPACE	233	no
AUX	233	no
AVAILPAGEPCT	179	yes
AVGVOLSP	284	yes
BMCHIST	193	yes

Table 30 REORG PLUS command options (part 2 of 4)

Command option	See page	Valid for REORG INDEX?
BMCSTATS	236	yes
CLONE	160	yes
CLUSTERRATIO (no longer supported)	238	yes
CONDEXEC	210	yes
COPY	241	no
COPYDDN	248	no
COPYLVL	246	no
DATACLAS	299	yes
DDLDDN	172	no
DDTYPE	270	yes
DEADLINE	301	yes
DELAY	304	yes
DELETE	260	no
DELETEFILES	200	yes
DRAIN	314	yes
DRAIN_WAIT	224	yes
DSNPAT	289	yes
DSNUEXIT	221	yes
DSNUTILB	217	yes
DSPLOCKS	226	yes
DSRSEXIT	222	yes
exitPoint	220	yes
EXPDT	296	no
FASTSWITCH	226	yes
GDGLIMIT	297	no
ICDDN	252	no
ICTYPE	244	no
IDCACHE	232	no
IDCDDN	199	yes
IFALLOC	273	yes
INDREFLIMIT	213	no
INLINE	242	no
KEEPDICTIONARY	191	no
LEAFDISTLIMIT	214	yes
LOG	194	no
LOGFINAL	312	yes
LOGMEM	311	yes

Table 30 REORG PLUS command options (part 3 of 4)

Command option	See page	Valid for REORG INDEX?
LOGSPILL	312	yes
LOGTHRESHLD	300	yes
LONGLOG	303	yes
LONGNAMETRUNC	166	yes
MAPPINGTABLE	314	yes
MAXEXTSZ	278	yes
MAXNEWPARTS	164	no
MAXRO	300	yes
MAXSORTMEMORY	183	yes
MAXSORTS	175	yes
MAXTAPE	269	yes
MGMTCLAS	299	yes
MINSORTMEMORY	183	yes
NLPCTFREE	195	yes
NOSYSREC	190	no
OFFPOSLIMIT	211	no
ON FAILURE	205	yes
ON MESSAGE	203	no
ORDER	176	yes
ORIGINALDISP	202	yes
PART	163	yes
PREFORMAT	208	yes
REBALANCE	167	no
RECOVERYDDN	250	no
RECOVERYICDDN	254	no
REDEFINE	195	yes
REGISTER	243	no
REORG INDEX	160	yes
REORG TABLESPACE	159	no
REPORTONLY	216	yes
RETPD	296	no
RETRY	225	yes
RETRY_DELAY	225	yes
REUSE	198	yes
RIDMAPMEM	310	yes
SELECT	258	no
SHORTMEMORY	181	yes
SHRLEVEL	161	yes

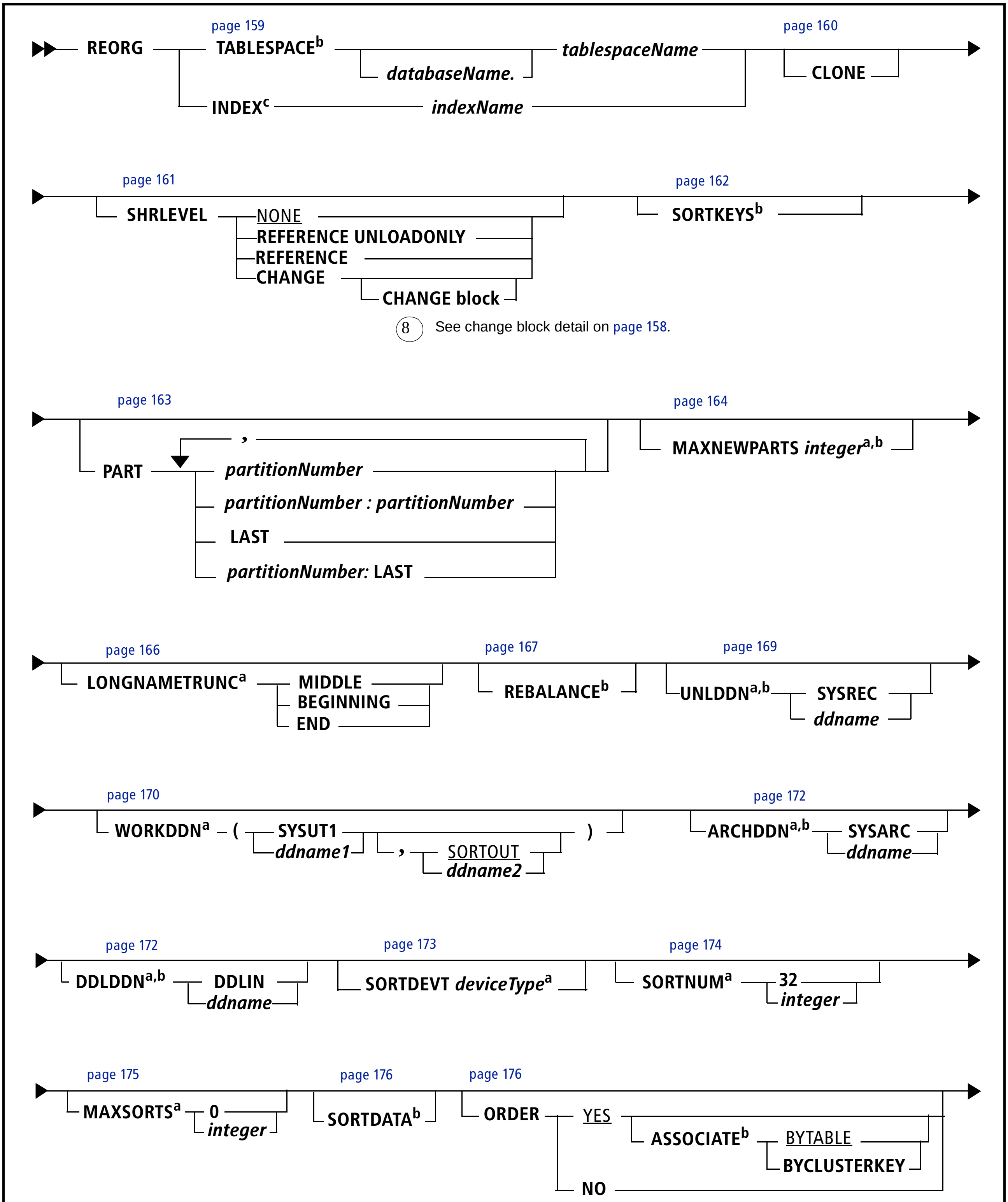
Table 30 REORG PLUS command options (part 4 of 4)

Command option	See page	Valid for REORG INDEX?
SIXSNAP	228	no
SIZEPCT	285	yes
SMS	276	yes
SMSUNIT	277	yes
SORTDATA	176	no
SORTDEVT	173	yes
SORTKEYS	162	no
SORTNUM	174	yes
SPACE	286	yes
SPILLDSNPAT	305	yes
SPILLSTORCLAS	305	yes
SPILLUNIT	304	yes
STORCLAS	299	yes
SYNC	193	yes
TERMEXIT	223	yes
THRESHLD	286	yes
TIMEOUT	218	yes
TOTALPAGEPCT	180	yes
TSSAMPLEPCT	240	no
UNIT	280	yes
UNITCNT	280	yes
UNLDDN	169	no
UNLOAD	189	yes
UPDATE	262	no
UPDATEDB2STATS	238	yes
VOLCNT	282	yes
WHERE condition	264	no
WORKDDN	170	yes
WTOMSG	231	yes
XBMID	230	yes
ZIIP	229	yes

Command syntax diagrams for REORG PLUS

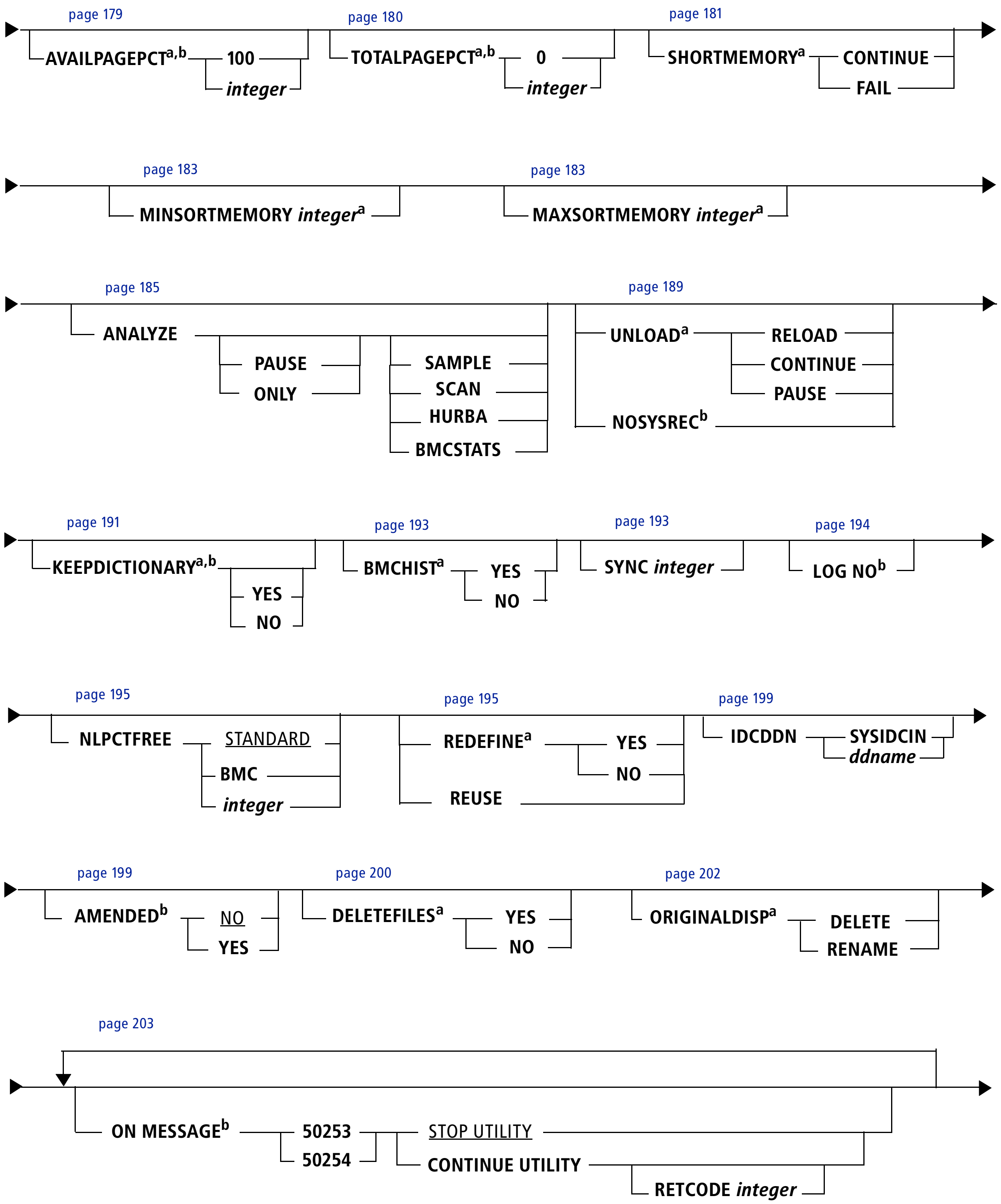
Figure 8 on page 150 shows the possible options allowed on the REORG command. For information about how to read the syntax diagrams, see “Syntax diagrams” on page 23.

Figure 8 REORG PLUS command syntax diagram (part 1 of 5)



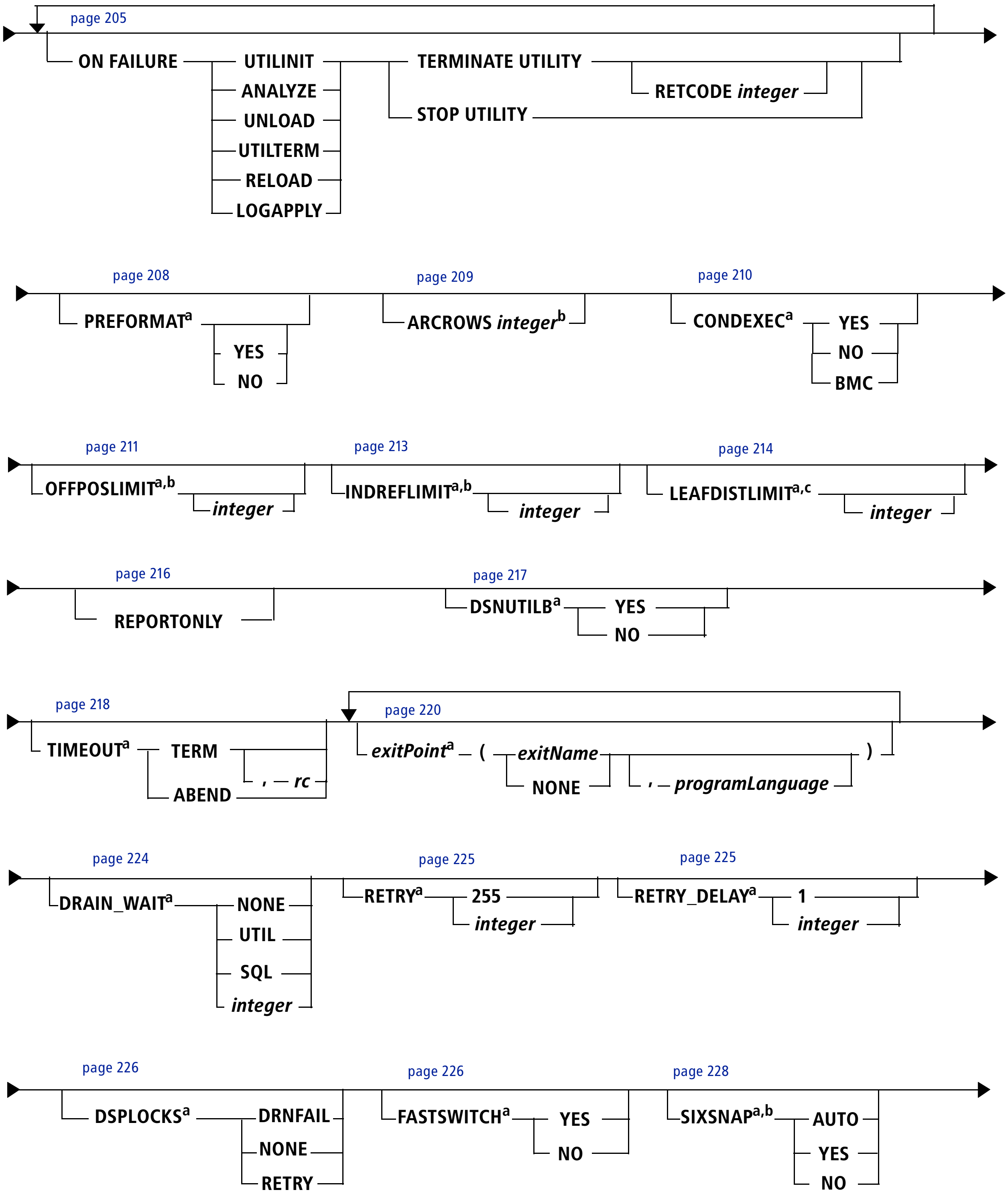
[a] Option has a corresponding installation option.
[b] This option is applicable for a table space reorganization only.
[c] This option is applicable for an index reorganization only.

Figure 8 REORG PLUS command syntax diagram (part 2 of 5)



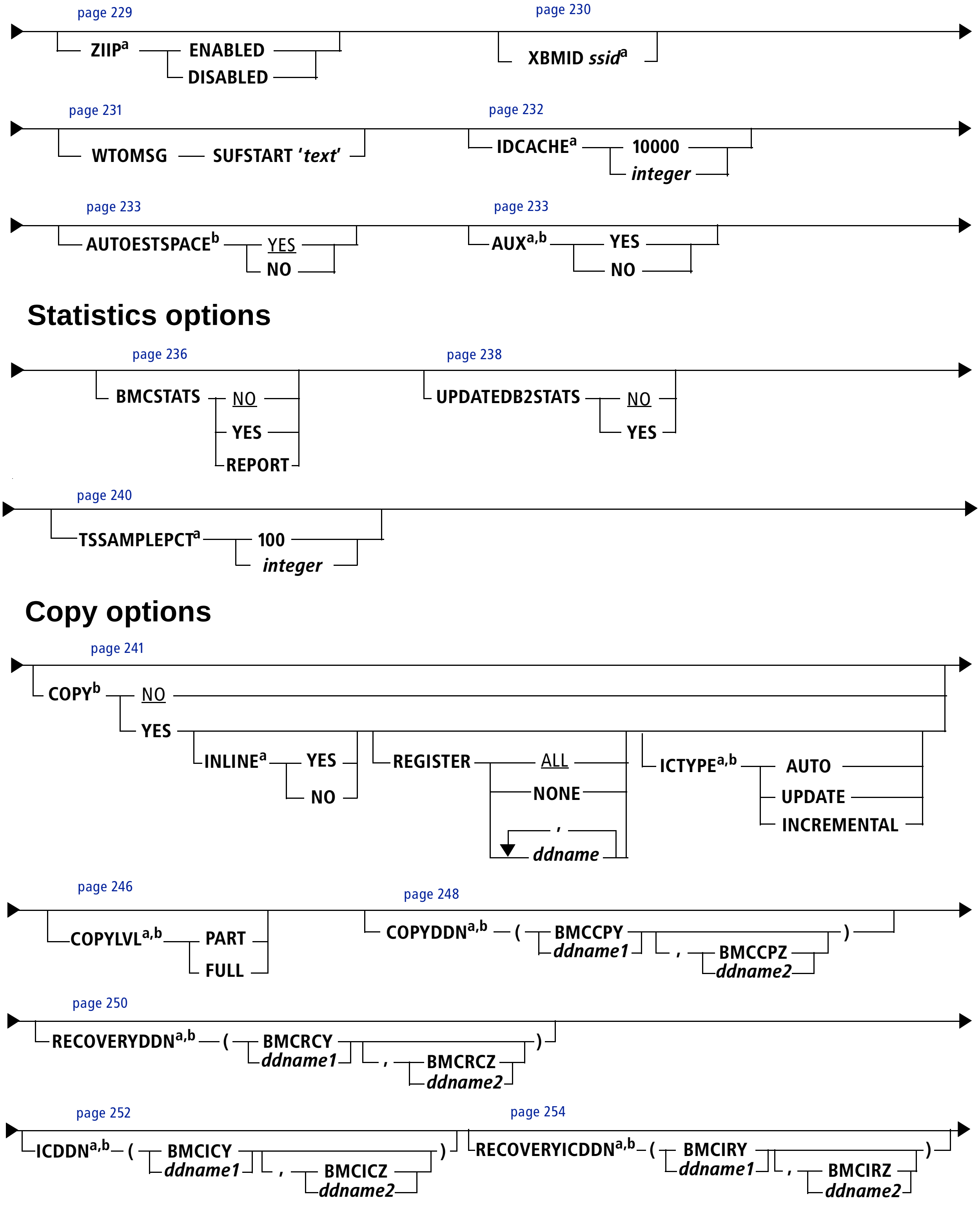
[a] Option has a corresponding installation option.
[b] This option is applicable for a table space reorganization only.

Figure 8 REORG PLUS command syntax diagram (part 3 of 5)



[a] Option has a corresponding installation option.
[b] This option is applicable for a table space reorganization only.
[c] This option is applicable for an index space reorganization only.

Figure 8 REORG PLUS command syntax diagram (part 4 of 5)



[a] Option has a corresponding installation option.

[b] This option is applicable for a table space reorganization only.

Figure 8 REORG PLUS command syntax diagram (part 5 of 5)

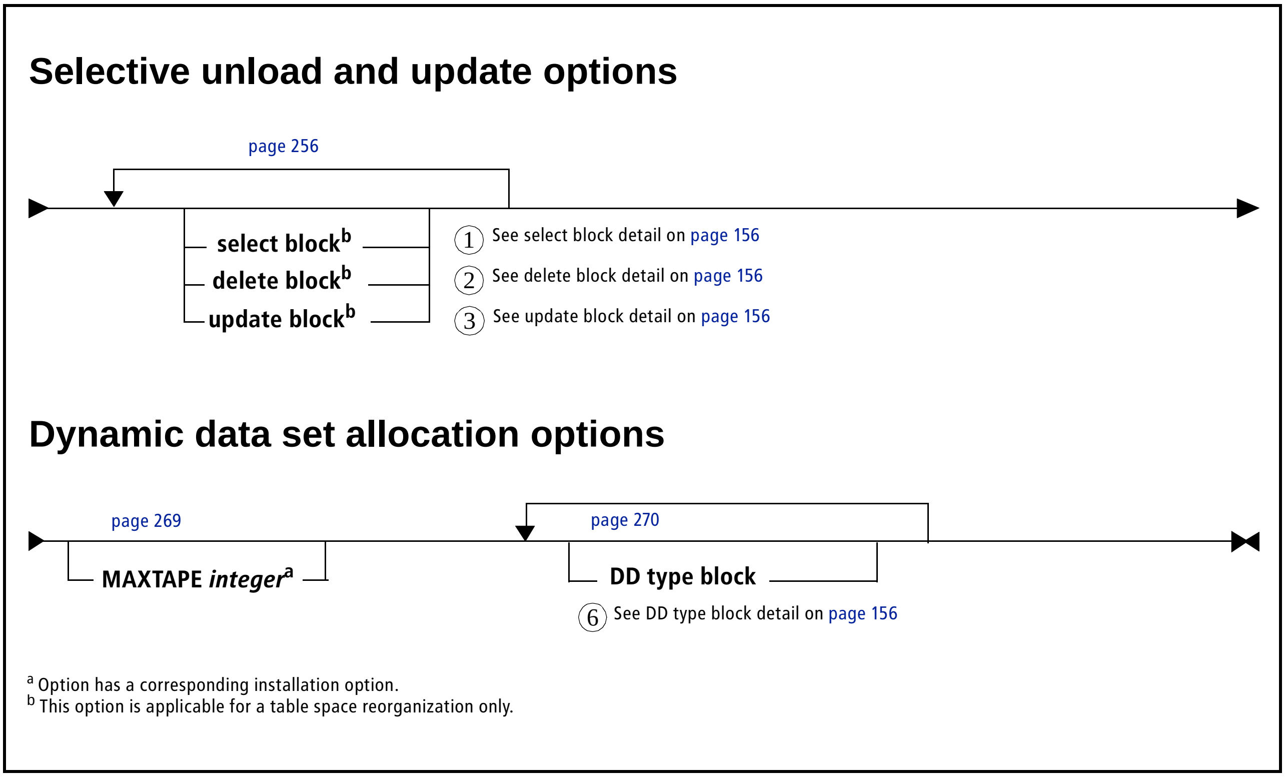

Selective unload and update options
page 256
select block[b]
delete block[b]
update block[b]
1 See select block detail on page 156
2 See delete block detail on page 156
3 See update block detail on page 156
Dynamic data set allocation options
page 269
MAXTAPE integer[a]
page 270
DD type block
6 See DD type block detail on page 156
[a] Option has a corresponding installation option.
[b] This option is applicable for a table space reorganization only.

Figure 9 shows syntax diagram details. The number on each detail diagram corresponds to a preceding section of the REORG PLUS command syntax or to another diagram in this section.

Figure 9 Detail syntax diagrams (part 1 of 5)

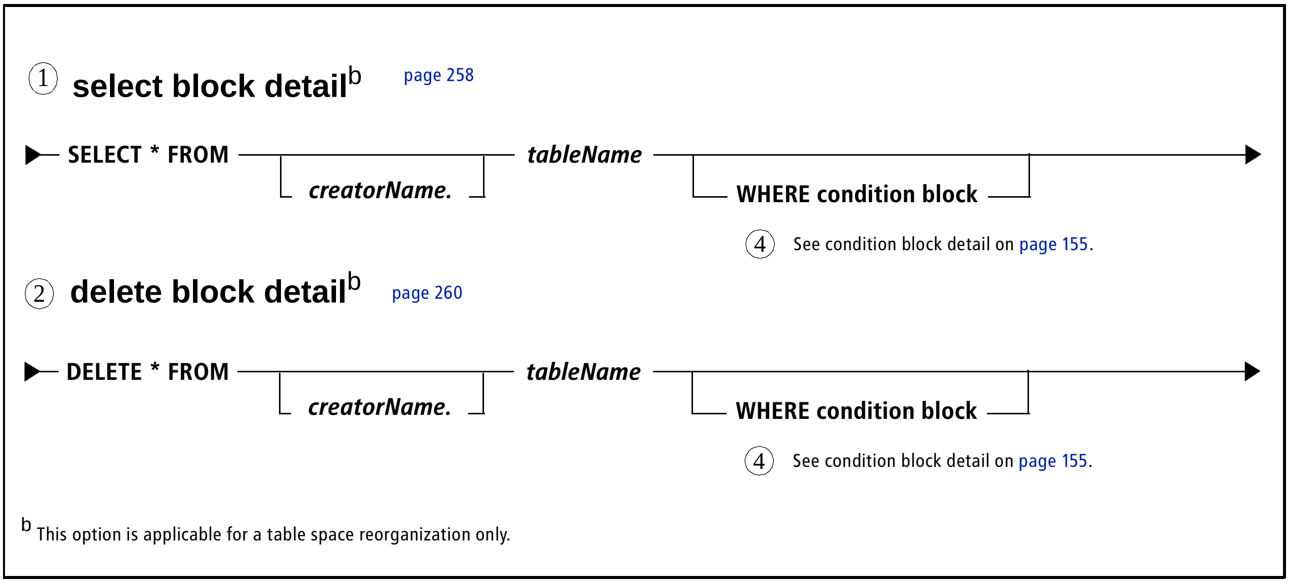
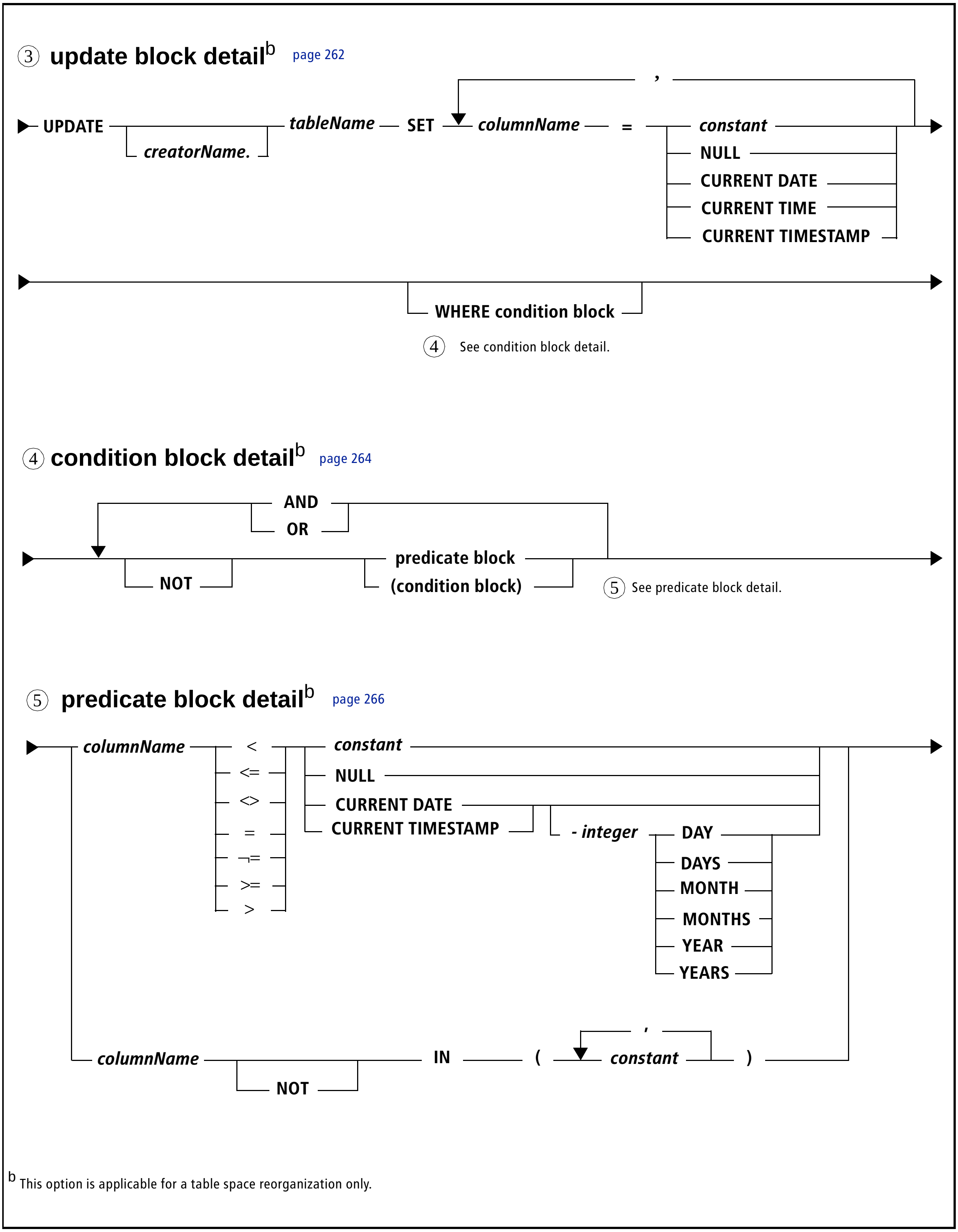


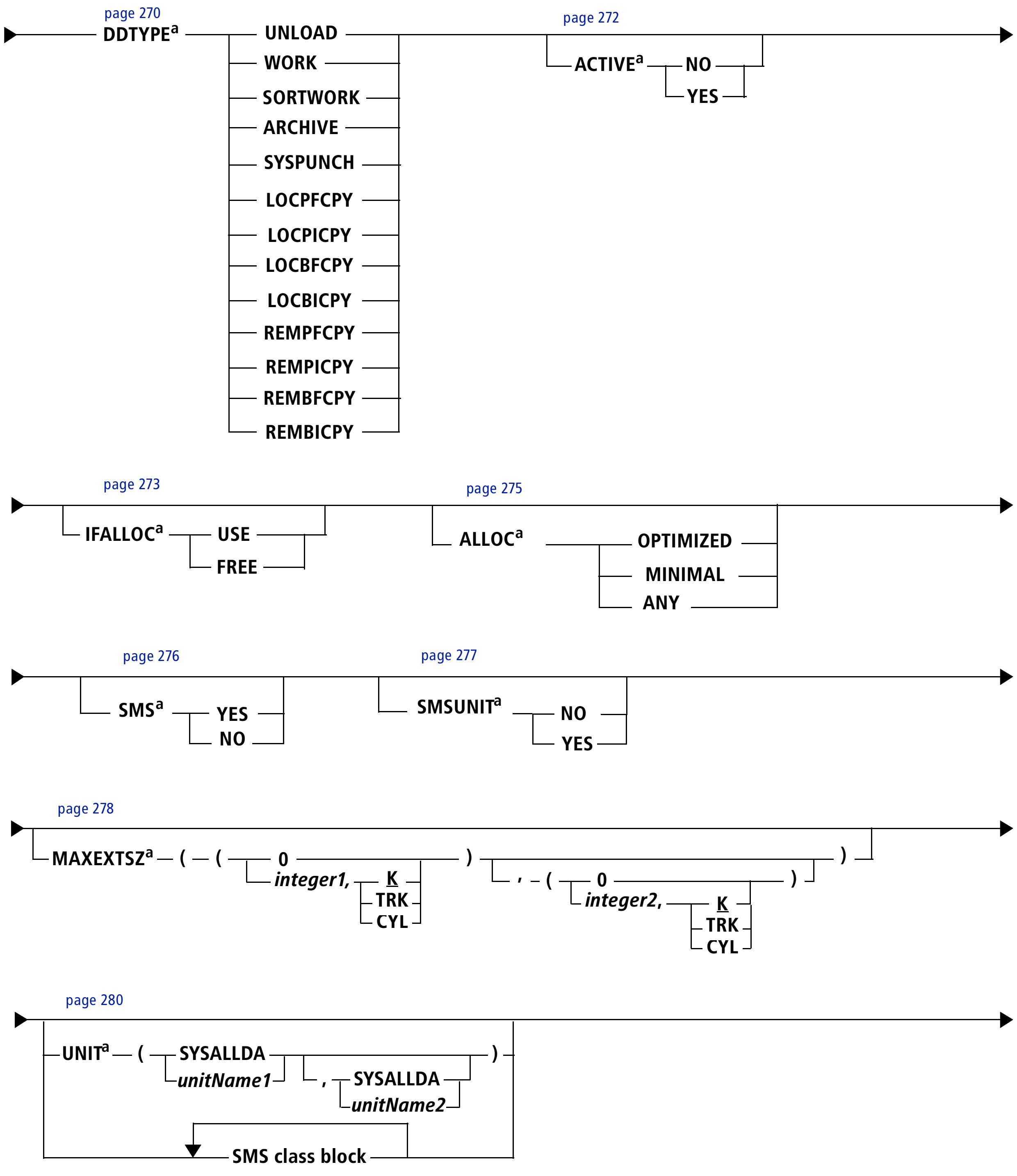
Figure 9 Detail syntax diagrams (part 2 of 5)



[b] This option is applicable for a table space reorganization only.

Figure 9 Detail syntax diagrams (part 3 of 5)

⑥ DD type block detail page 270

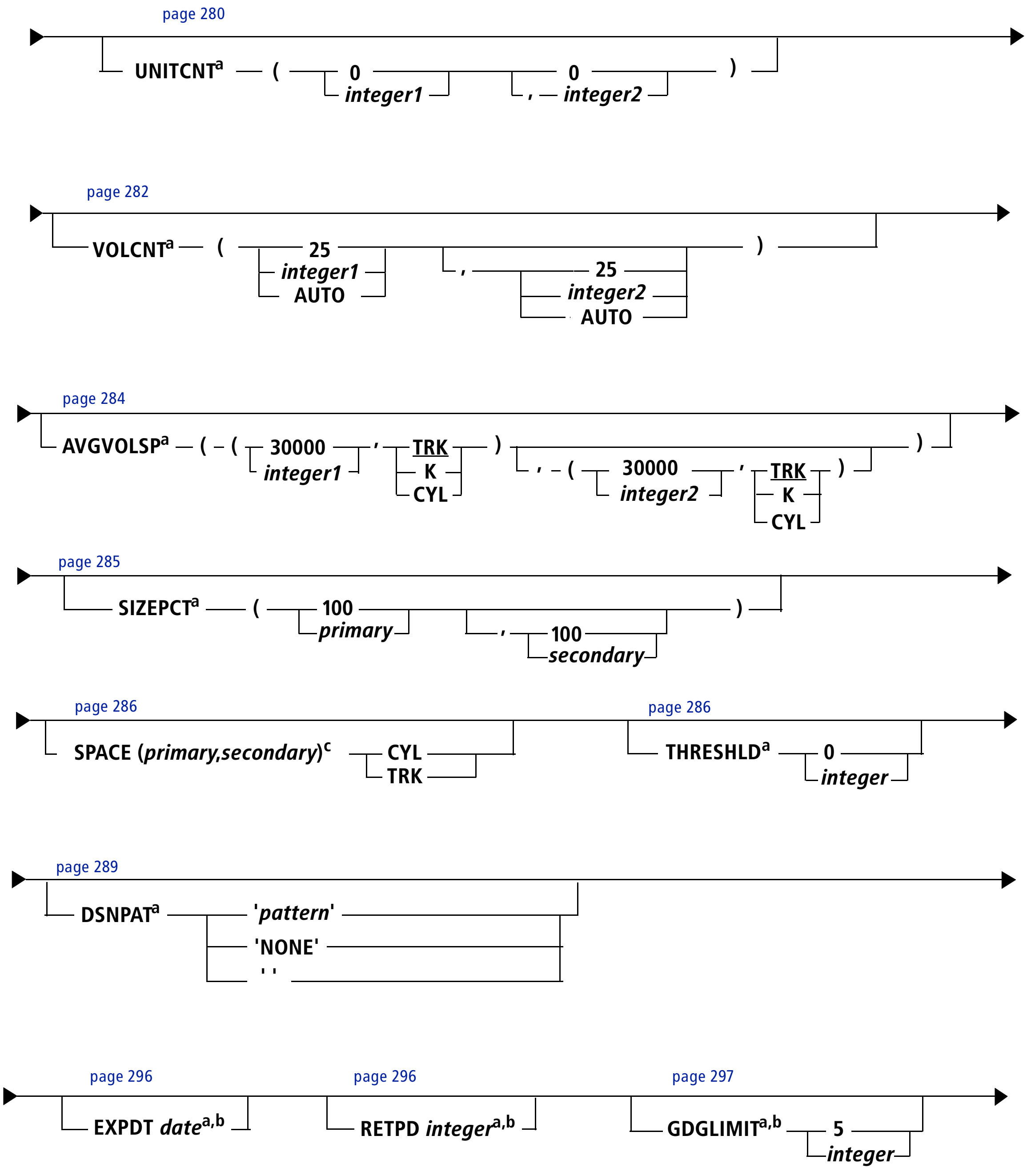


⑦ See SMS class block detail on page 158.

[a] Option has a corresponding installation option.

Figure 9 Detail syntax diagrams (part 4 of 5)

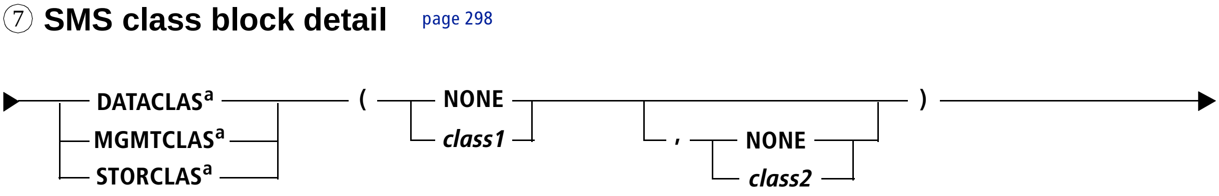
⑥ DD type block detail *continued*



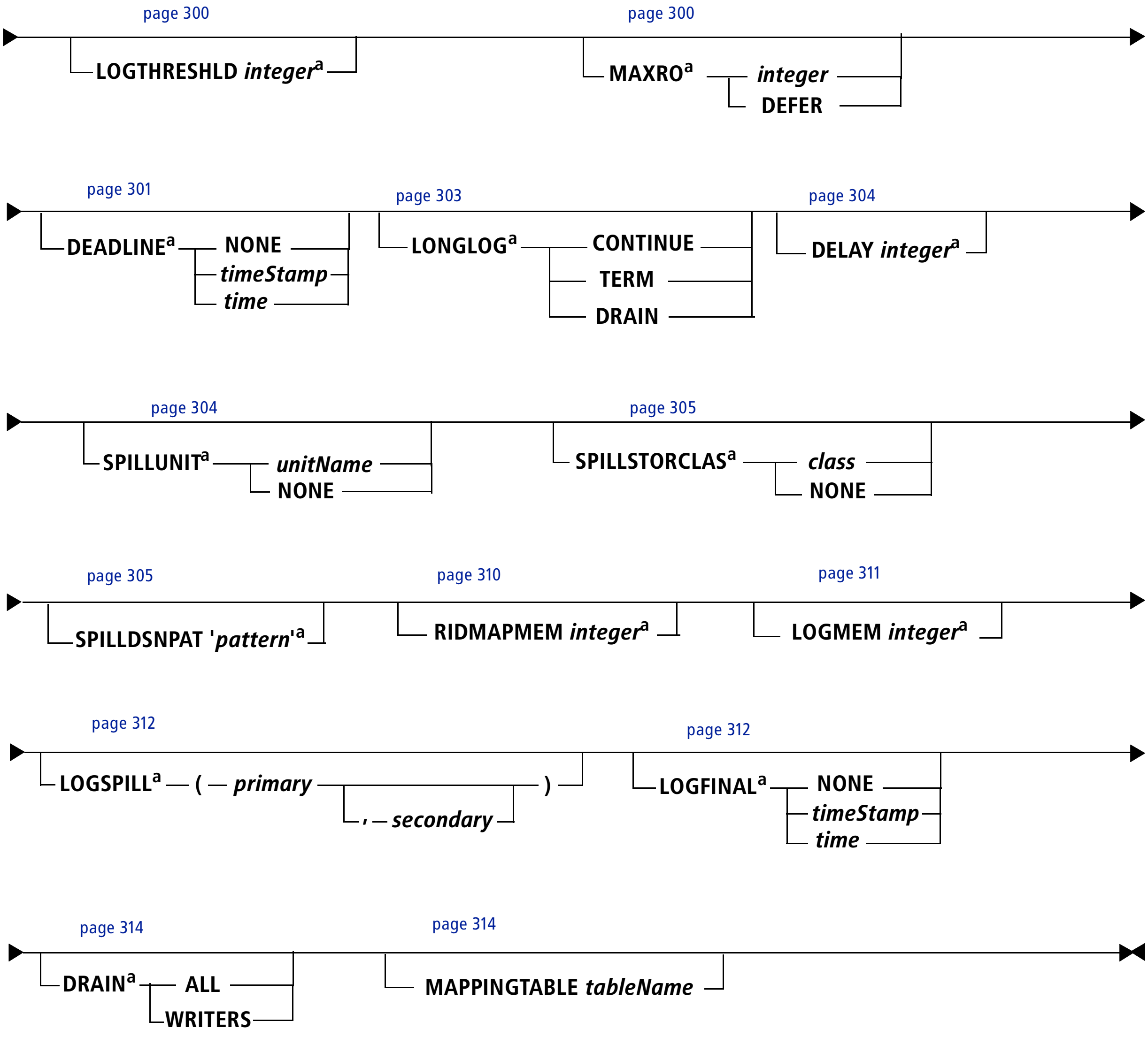
[a] Option has a corresponding installation option.
[b] This option is applicable for a table space reorganization only.
[c] This option is valid for a DSNUTILB reorganization only.

Figure 9 Detail syntax diagrams (part 5 of 5)

⑦ SMS class block detail



⑧ change block detail



[a] Option has a corresponding installation option.

Descriptions of REORG PLUS command options

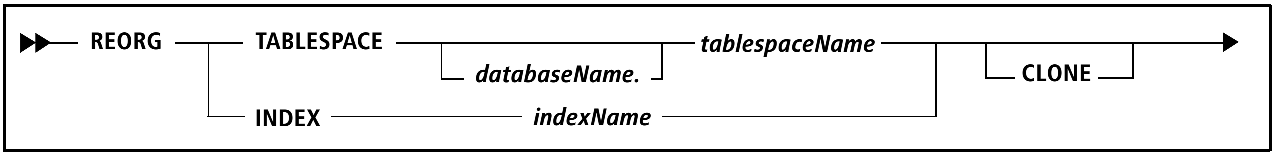
This section describes options in the order in which they appear in the preceding syntax diagrams, which group the options according to these functions:

- basic processing options
- copy options
- statistics options
- selective unload options
- dynamic data set allocation options
- SHRLEVEL CHANGE options

Options that are not applicable to an index-only reorganization are noted as such.

Basic processing options

The basic reorganization processing options control most aspects of REORG PLUS execution.



REORG

REORG is the keyword for the REORG PLUS utility command.

TABLESPACE

This option applies to a table space reorganization only.

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

databaseName

If you do not specify a database name, BMC supplies the value DSNDB04. The database name specifies the database to be reorganized and cannot be DSNDB01, DSNDB06, or DSNDB07.

tablespaceName

The table space name specifies the name of the table space to be reorganized. The table space and associated index spaces must be started in the appropriate status before beginning the REORG PLUS job. For status information, see “Serialization and concurrency” on page 78.

INDEX

This option does not apply to a table space reorganization.

The INDEX option names the index (and thus the corresponding index space) to be reorganized. The index name is the qualified name of the index. If you do not specify the authorization ID qualifier, REORG PLUS uses the DB2® primary authorization ID of the user who is running the REORG PLUS job.

If you specify the INDEX option, REORG PLUS reorganizes only the index. No data in table spaces is reorganized.

Additional consideration
The index space that you are reorganizing must be started in the appropriate status before beginning the REORG PLUS job. For status information, see “Serialization and concurrency” on page 78.

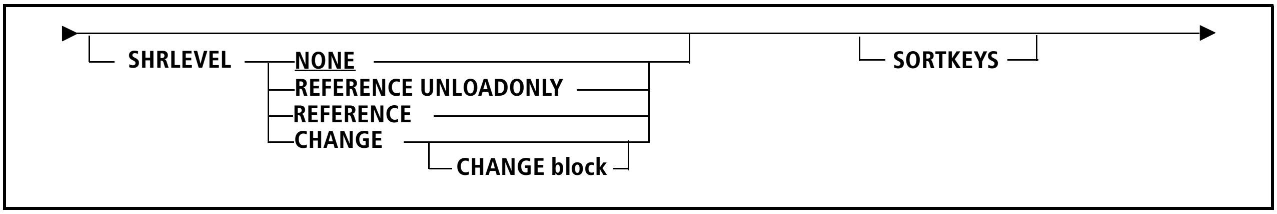
CLONE

This option applies to a DSNUTILB reorganization only.

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. When you specify this option, REORG PLUS invokes DSNUTILB. For more information about DSNUTILB reorganization jobs, see “Reorganization jobs that invoke DSNUTILB” on page 70.

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.

Restriction
You cannot use FASTSWITCH processing when reorganizing clone objects. REORG PLUS changes FASTSWITCH YES to FASTSWITCH NO in this case.



SHRLEVEL

The SHRLEVEL option specifies the level of access that DB2 has to the target spaces during REORG PLUS processing. For important information about using this option, see "SHRLEVEL considerations" on page 96.

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.

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.

Restriction
When invoking DSNUTILB, REORG PLUS issues message BMC50178E and terminates.

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.

For more information, see "SHRLEVEL REFERENCE" on page 97.

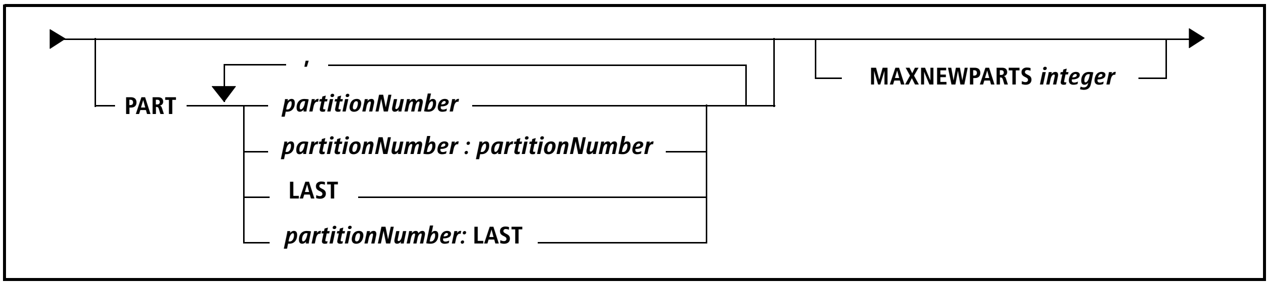
CHANGE

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. For information about the syntax for the additional options to specify with SHRLEVEL CHANGE, see "SHRLEVEL CHANGE options" on page 300. For information about how the function works, see Chapter 7, "Online reorganization."

When invoking DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility for processing.

SORTKEYS

When REORG PLUS invokes DSNUTILB, this option specifies that index keys are to be sorted in parallel. This option is meaningful only when REORG PLUS invokes DSNUTILB. Otherwise, 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.

Specifying partition numbers

Note the following information about how to specify 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.

The following examples illustrate valid PART specifications using partition numbers:

Specification	REORG PLUS reorganizes these partitions
PART 2:4	2, 3, and 4
PART 3:5,7,9:11,13,2	3, 4, 5, 7, 9, 10, 11, 13, and 2
PART 100:104,4096	100, 101, 102, 103, 104, and 4096

Reorganizing a large number of partitions might result in constrained resources. If you encounter this problem, consider specifying smaller groups of partitions in multiple steps. Whether the number of partitions that you have is considered to be a large number depends on your environment. For more information, see "Large number of partitions" on page 105.

DSNUTILB reorganization jobs

For a DSNUTILB reorganization, your partition specification must be a specification that the IBM DB2 REORG utility supports, as follows. If you specify other values, REORG PLUS issues message BMC50178E and terminates.

- For an index reorganization, you can specify only PART *partitionNumber*.
- For a table space reorganization, you can specify one of the following:
  - PART *partitionNumber*
  - PART *partitionNumber*: *partitionNumber*

LAST

When you are reorganizing a partition-by-growth table space, you can specify the last partition either by partition number or by specifying the LAST option. Specifying LAST is useful when you are reorganizing partition-by-growth table spaces because it frees you from having to track which partition number is the last partition before you begin the reorganization. LAST is valid only for partition-by-growth table spaces. If you specify LAST for any other type of table space, REORG PLUS terminates and issues message BMC50173E. For more information about reorganizing partition-by-growth table spaces, see page 118.

The following examples illustrate valid PART specifications using the LAST option:

Specification	REORG PLUS reorganizes these partitions
PART 100:LAST	100 through the last partition
PART LAST	last partition
PART 100,LAST	100 and last

Restriction

For a DSNUTILB reorganization, the LAST option is not valid.

MAXNEWPARTS

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

The MAXNEWPARTS option specifies the following limits:

- the maximum number of partitions that REORG PLUS can add during a reorganization

  Partition extension is also constrained by the DB2 MAXPARTITIONS value with which the table space was created.

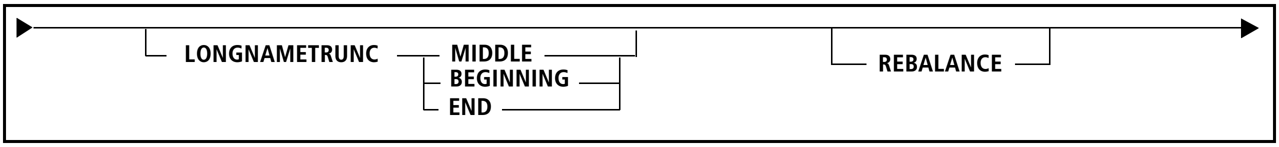
- the maximum number of partitions added by DB2 that REORG PLUS can support (only applies to a SHRLEVEL CHANGE reorganization of a table space or an index defined on the table)

  If DB2 adds more partitions than the value specified for MAXNEWPARTS, REORG PLUS terminates and issues message BMC53025E.

You can specify any integer from 0 through 4096 for MAXNEWPARTS. However, BMC recommends that you specify only the number of new partitions that you expect the table space to require. A smaller MAXNEWPARTS value reduces memory requirements and the number of additional partition-level data sets that you must allocate. For special requirements for copy data sets and SYSREC data sets, see page 331 and page 351.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the MAXNEWPARTS command option in your installation options module by using the MAXNEWPARTS installation option (page 658). REORG PLUS was shipped with a default value of 2 for this option. The command option overrides the default that is in the installation options module.



LONGNAMETRUNC

LONGNAMETRUNC tells REORG PLUS where to truncate names that are longer than the area that is available in REORG PLUS report-style messages.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the LONGNAMETRUNC command option in your installation options module by using the LONGNAMETRUNC installation option. REORG PLUS was shipped with a default value of MIDDLE for this option. The command option overrides the default value that is in the installation options module.

MIDDLE

MIDDLE tells REORG PLUS to truncate outward from the middle of the name. You can shorten this value to M. REORG PLUS indicates the truncated section with the symbols >< as shown in the following example:

```
BMC50521I CREATOR          NAME
BMC50522I LONG_CR><OR_NAME VERY_LONG_TABLE_NAME_><HAN_THIRTY_CHARACTERS
```

BEGINNING

BEGINNING, or B, tells REORG PLUS to truncate from the beginning of the name. REORG PLUS indicates the truncated section with the symbols << as shown in the following example:

```
BMC50521I CREATOR          NAME
BMC50522I <<G_CREATOR_NAME <<_TABLE_NAME_GREATER_THAN_THIRTY_CHARACTERS
```

END

END, or E, tells REORG PLUS to truncate from the end of the name. REORG PLUS indicates the truncated section with the symbols >> as shown in the following example:

```
BMC50521I CREATOR          NAME
BMC50522I LONG_CREATOR_N>> VERY_LONG_TABLE_NAME_GREATER_THAN_THIRTY_C>>
```

REBALANCE

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

For index-controlled and table-controlled partitioned table spaces, you can use the REBALANCE command option to rebalance partitions. In a single execution of REORG PLUS, you can rebalance up to 255 ranges of logically contiguous partitions.

REBALANCE tells REORG PLUS to define new partition boundaries and evenly redistribute rows across the reorganized partitions. When you specify a list of partitions, REORG PLUS rebalances all logically contiguous partitions. REORG PLUS computes limit key breaks based on rebalancing at the record level, which enables REORG PLUS to redistribute the number of rows across partitions.

You can also use REBALANCE with all SHRLEVEL options, including SHRLEVEL CHANGE. However, you must have sufficient unique limit key values to allow REORG PLUS to establish new balanced limit keys. If the number of duplicate keys results in an empty partition, the reorganization fails because REORG PLUS cannot determine the correct limit key.

WARNING

When you use SHRLEVEL NONE, insufficient unique limit key values might cause an error that can only be repaired with a recovery.

Restrictions

The following restrictions apply when you use the REBALANCE option. For more information, including requirements and considerations, see "Partition rebalancing" on page 120.

- When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing. DB2 REORG utility restrictions apply, as described in the IBM documentation.

- You cannot rebalance the partitions of the following table space types:

  — partition-by-growth table spaces

  — XML table spaces

  — table spaces that contain an XML column

  NOTE

  In many cases, you can rebalance table spaces that contain an XML column by including a DDLIN data set in your JCL. For restrictions, see "DDLIN data set" on page 333.

- You cannot perform partition rebalancing on a table space if the table space or an index on the table space has pending DDL changes. REORG PLUS issues message BMC51334E and terminates.

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

- You cannot specify the following combination of options when all partitioning columns are ascending and any column except the last column is nullable:

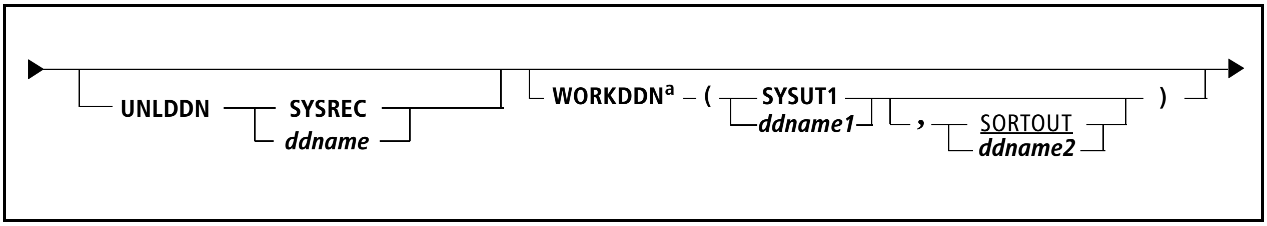
  — SHRLEVEL NONE

  — 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 126.



UNLDDN

This option applies to a table space reorganization only.

UNLDDN allows you to override the default ddname (SYSREC) or ddname prefix for the output data set that contains the unloaded rows to be reorganized. For information about specifying and using the SYSREC data set, see "SYSREC data sets" on page 349.

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. For more information, see "Specifying ddname prefixes" on page 91.

Specifying the default

You can specify the default for the UNLDDN command option in your installation options module by using the UNLDDN installation option (page 683). REORG PLUS was shipped with a default value of SYSREC for this option. The command option overrides the default that is in the installation options module.

WORKDDN

WORKDDN allows you to override the default ddname or ddname prefix of the work data set. SORTOUT is not used by REORG PLUS but is kept for compatibility with the IBM DB2 REORG utility command syntax. For specification guidelines and detailed information about the use of the SYSUT1 work data set, see "SYSUT1 data sets" on page 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.

Multiple data sets

If you use multiple work data sets, the ddname that you specify in this option becomes a prefix. The ddname that you specify in the JCL must have the partition number *nn* appended to this prefix. The length of *nn* can be from one through seven characters, depending on the length of the ddname prefix. The maximum length of the ddname with the prefix must be eight bytes or less. If you are using more than nine work data sets (*nn* is 10 or greater), the prefix that is specified in this option can have a maximum of six characters.

Dynamic data set allocation

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

- acceptable set:

```
BMCRD
BMCRDWK
```

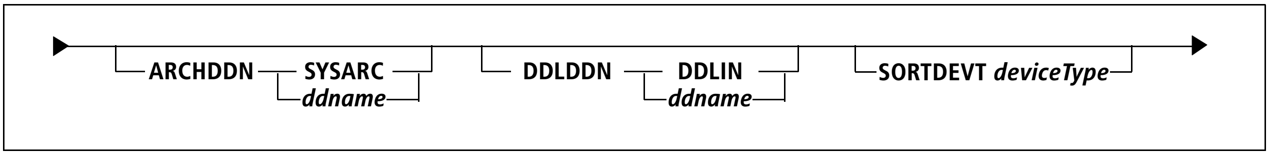
- not acceptable set:

```
BMCRD
BMCRD11
```

When you dynamically allocate SYSUT1 data sets, REORG PLUS appends the data set number to the ddname prefix that you specify. To dynamically allocate more than 99 SYSUT1 data sets, use this option to specify a ddname prefix. The prefix plus the highest data set number must not exceed eight characters. For more information, see “Specifying ddname prefixes” on page 91.

Specifying the default

You can specify the default for the WORKDDN command option in your installation options module by using the WORKDDN installation option (page 686). REORG PLUS was shipped with a default value of SYSUT1 for this option. The command option overrides the default that is in the installation options module.



ARCHDDN

This option applies to a table space reorganization only.

ARCHDDN allows you to override the default ddname of the archive data set (SYSARC). REORG PLUS uses the archive data set to contain rows that it discards during a reorganization. For more information about the SYSARC data set, see "SYSARC data set" on page 342.

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.

REORG PLUS ignores this option when invoking DSNUTILB to reorganize a LOB table space.

Specifying the default

You can specify the default for the ARCHDDN command option in your installation options module by using the ARCHDDN installation option (page 627). 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.

DDLDDN

This option applies to a table space reorganization only.

DDLDDN allows you to override the default ddname of the DDLIN data set. For information about using a DDLIN data set in your REORG PLUS jobs, see page 333.

If you use this command option to override the default name in the installation options module, you must also change the ddname in your JCL.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the DDLDDN command option in your installation options module by using the DDLDDN installation option (page 636). REORG PLUS was shipped with a default value of DDLIN for this option. The command option overrides the default that is in the installation options module.

SORTDEVT

The SORTDEVT option specifies the device type for the sort work files that are allocated dynamically.

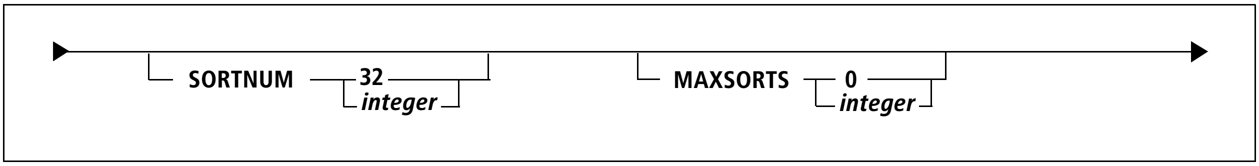
For non-DSNUTILB jobs, if the first parameter of the SORTDEVT installation option is null, this command option overrides the first parameter of the BMCSORT DYNALOC installation option. If the value of the third parameter in the BMCSORT DYNALOC installation option is OFF, specifying a value for the SORTDEVT option turns BMCSORT dynamic allocation on.

DSNUTILB reorganizations

If you are reorganizing a LOB table space, REORG PLUS ignores this option for a DSNUTILB reorganization. In all other cases, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

Specifying the default

You can specify the default for the SORTDEVT command option in your installation options module by using the SORTDEVT installation option (page 673). REORG PLUS was shipped with a default value of (, SYSALLDA) for this option (where the first parameter affects non-DSNUTILB jobs and the second parameter affects DSNUTILB jobs). The command option overrides the default that is in the installation options module for both parameters.



SORTNUM

The SORTNUM option affects the allocation of sort work files in the following cases. You can specify any integer value from 0 through 255.

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility as the number of sort work files to allocate dynamically. For this type of reorganization, the value must be 2 or greater.

All other reorganization jobs
This value is in effect when BMCSORT is allocating your sort work files dynamically. Table 31 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 31 SORTNUM values

Value	Description	Additional considerations
0	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.)	For more information about how this parameter affects dynamic allocation, see “Dynamically allocating SORTWK data sets” on page 339. For more information about the parameter itself, see “DYNALOC installation option” on page 708.
1–32	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.	*Preallocated sort work files* include sort work files that are allocated in your JCL and any sort work files that REORG PLUS dynamically allocates.
33–255	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.	

If the value of the third parameter in the BMCSORT DYNALOC installation option is OFF, specifying a value greater than 0 for the SORTNUM option turns BMCSORT dynamic allocation on and BMCSORT allocates sort work files as needed. For information about when BMCSORT allocates your sort work files dynamically, see the “SORTWK data sets” on page 338.

Specifying the default

You can specify the default for the SORTNUM command option in your installation options module by using the SORTNUM installation option (page 673). REORG PLUS was shipped with a default value of 32 for this option. The command option overrides the default that is in the installation options module.

MAXSORTS

The MAXSORTS option allows you to specify the maximum number of sort tasks that REORG PLUS can run concurrently.

Table 32 describes the effects that MAXSORTS and its relationship with the SMAX installation option have on REORG PLUS processing:

Table 32 Effects of MAXSORTS and SMAX options on REORG PLUS processing

If MAXSORTS value is	And SMAX value is	Then REORG PLUS
0	any	uses the SMAX value
greater than 0	any	uses MAXSORTS and starts only one task per CPU[a]
0	0	multitasks based on the values of the BILDMAX, RORGMAX, and UNLDMAX options

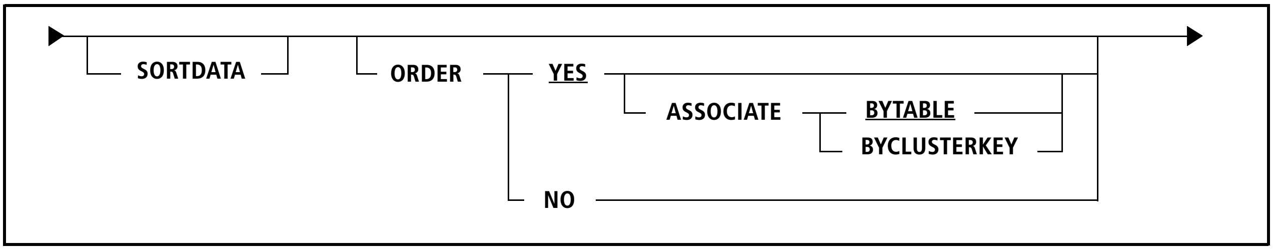
[a] If you want to improve performance by starting more than one task per CPU, specify 0 for MAXSORTS and SMAX and use the multitasking options.

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

- Table 89 on page 542
- “Sort processing options” on page 536
- SMAX installation option on page 672
- “Multitasking processes that invoke BMCSORT” on page 543

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.



SORTDATA

This option applies to a table space reorganization only.

SORTDATA is available only for compatibility with the IBM DB2 REORG utility command, and REORG PLUS treats this option as a comment. REORG PLUS never uses the clustering index to unload the data.

ORDER

The ORDER option tells REORG PLUS how and whether to order the table rows. For performance information about this option, see “ORDER command option” on page 553.

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

Note the following restrictions for ORDER YES:

- When you are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect, REORG PLUS changes ORDER YES to ORDER NO.

- REORG PLUS ignores ORDER YES when all of the following conditions exist:
  - You are performing a two-phase reorganization of a table space with only one table.
  - The table space is not partitioned or is partition-by-growth.
  - No clustering index exists.

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

REORG PLUS terminates when you specify ORDER NO under any of the following conditions:

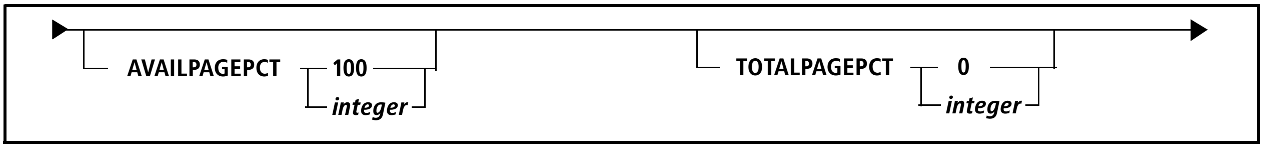
- You are reorganizing a multi-table, segmented table space.

- For a single-phase reorganization, you are performing a SHRLEVEL NONE reorganization.

- For an index-only reorganization, you are performing a SHRLEVEL CHANGE reorganization.

- All of the following conditions apply:

  — the table uses table-controlled partitioning
  — the table has no clustering index
  — you are using a DDLIN data set to alter limit keys or using the REBALANCE command option to rebalance partitions



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 (page 670 and page 181) controls the action that REORG PLUS takes.

- When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the AVAILPAGEPCT command option in your installation options module by using the AVAILPAGEPCT installation option (page 629). 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:

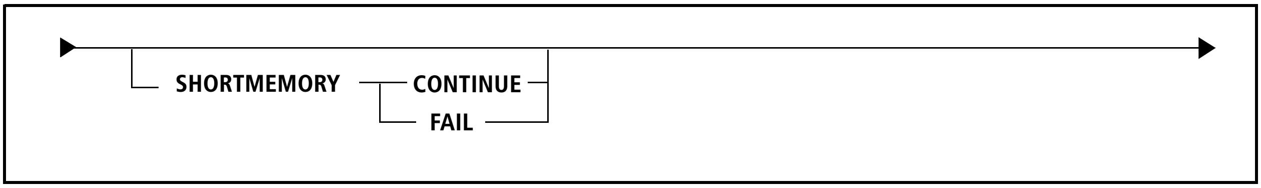
- When you specify values greater than 0 for both TOTALPAGEPCT and AVAILPAGEPCT, REORG PLUS uses the lesser of the two calculated results as the maximum amount of memory for sort processing.

- 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 (page 181) or command option controls the action that REORG PLUS takes.

- When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the TOTALPAGEPCT command option in your installation options module by using the TOTALPAGEPCT installation option (page 681). 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 one of the following memory shortages exists during sort processing:

- The system contains insufficient available pages of memory for REORG PLUS to perform an optimal sort.

- The region contains insufficient memory for REORG PLUS to perform one task with the amount of memory required to perform an optimal sort. However, the region *does* contain at least 1024 KB of memory or the minimum amount of memory specified by the MINSORTMEMORY option. If the region contains at least 1024 KB of memory *and* the amount of memory specified by the MINSORTMEMORY option is available, REORG PLUS uses the greater amount of memory.

Table 33 on page 182 shows the relationship between SHORTMEMORY values and the following conditions:

- memory data obtained from the system
- memory in the region
- value specified for the SMCORE installation option (page 673)
- value specified for the MINSORTMEMORY installation or command option (page 661 and page 183)

Table 33 Action REORG PLUS takes when memory resources are constrained during sort processing

Memory shortage		SHORTMEMORY value	
Location of memory	**Amount of memory is**	**CONTINUE**	**FAIL**
memory in the system	insufficient to run one optimal sort task based on the amount of data to be sorted	REORG PLUS runs one task with 1024 KB of memory or the amount of memory that you specified with MINSORTMEMORY, whichever is greater.	REORG PLUS fails.
	insufficient as specified by MINSORTMEMORY		
virtual memory in the region	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	REORG PLUS runs one task with the available memory.	
	insufficient as specified by MINSORTMEMORY or less than 1024 KB, whichever is greater	REORG PLUS fails.	
	insufficient as specified by the first parameter of SMCORE		

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

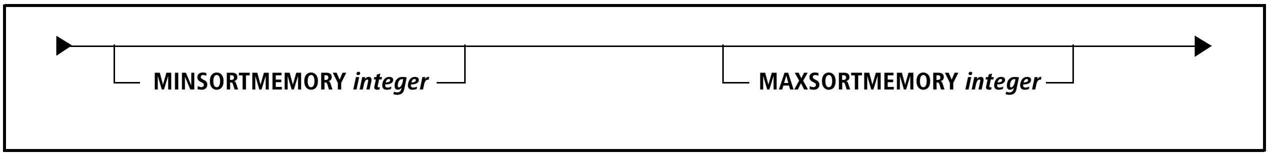
Specifying the default
You can specify the default for the SHORTMEMORY command option in your installation options module by using the SHORTMEMORY installation option (page 670). REORG PLUS was shipped with a default value of CONTINUE for this option. The command option overrides the default that is 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 33.

FAIL

FAIL indicates that, when a memory shortage exists, REORG PLUS should fail. Be aware when specifying FAIL that sufficient memory might exist to sort during the UNLOAD phase of a two-phase reorganization or the REORG phase of a single-phase reorganization. However, because of other system conditions, insufficient available pages might exist during the index build process, which occurs during the RELOAD process (two-phase reorganization) or following the REORG process (single-phase reorganization).



MINSORTMEMORY

The MINSORTMEMORY option specifies the minimum amount of memory, in kilobytes, that REORG PLUS should allocate to each sort task.

A value of 0 tells REORG PLUS to automatically compute the minimum amount of memory that is needed to optimally perform each sort task. In addition to 0, you can specify any number of kilobytes between 1024 and the value that you specify for the MAXSORTMEMORY installation or command option (page 183 and page 659).

NOTE

BMC recommends that you use a value of 0.

For information about how this option interacts with the SMCORE installation option, see "Sort processing options" on page 536.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the MINSORTMEMORY command option in your installation options module by using the MINSORTMEMORY installation option (page 661). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.

MAXSORTMEMORY

The MAXSORTMEMORY option specifies the maximum amount of memory, in kilobytes, that REORG PLUS can allocate to each sort task.

A value of 0 tells REORG PLUS to automatically compute the maximum amount of memory that is needed to optimally perform each sort task. In addition to 0, you can specify any number of kilobytes between the value that you specify for the MINSORTMEMORY installation (page 661) or command option and 2097152.

NOTE

BMC recommends that you use a value of 0.

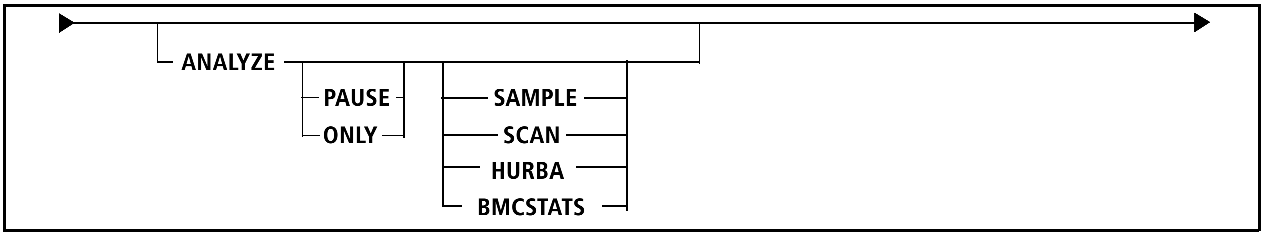
For information about how this option interacts with the SMCORE installation option, see “Sort processing options” on page 536.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the MAXSORTMEMORY command option in your installation options module by using the MAXSORTMEMORY installation option (page 659). 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 sizes for the following data sets:

- unload (SYSREC)
- work (SYSUT1)
- sort (SORTWK)
- full image copy (BMCCPY, BMCCPZ, BMCRCY, and BMCRCZ)
- incremental image copy (BMCICY, BMCICZ, BMCIRY, and BMCIRZ)

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)

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.

NOTE

When reorganizing a LOB table space, REORG PLUS does not sample or scan. Instead, it adds high-used relative byte addresses (HURBAs) from all data sets to estimate copy data set size.

REORG PLUS does not stop the table space and associated index spaces that participate in the reorganization during the ANALYZE phase.

Table 34 on page 186 shows various combinations of ANALYZE options and the effects that they have on how REORG PLUS determines cardinality and average row length.

Table 34 Effects of ANALYZE options on cardinality and average row length estimation

ANALYZE keywords	Cardinality	Average row length
(ANALYZE not specified) ANALYZE ANALYZE PAUSE ANALYZE ONLY	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).	REORG PLUS samples the table space.
ANALYZE SAMPLE ANALYZE PAUSE SAMPLE ANALYZE ONLY SAMPLE	REORG PLUS samples the table space or index space. For an XML table space reorganization, REORG PLUS obtains index cardinality from the DB2 real-time statistics tables.	
ANALYZE SCAN ANALYZE PAUSE SCAN ANALYZE ONLY SCAN	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.	
ANALYZE HURBA	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.	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.
ANALYZE PAUSE HURBA ANALYZE ONLY HURBA	REORG PLUS changes the keywords to ANALYZE PAUSE or ANALYZE ONLY.	
ANALYZE BMCSTATS ANALYZE PAUSE BMCSTATS ANALYZE ONLY BMCSTATS	REORG PLUS uses cardinality from the DASD MANAGER PLUS statistics table.	REORG PLUS uses average row length from the DASD MANAGER PLUS statistics table.

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, but adds HURBAs from all data sets to estimate copy data set size.

- 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 scan. Instead, it adds HURBAs from all data sets to estimate copy data set size.

HURBA

HURBA tells REORG PLUS to use information obtained from the DB2 catalogs and the HURBA to determine a gross estimate for cardinality and the average row length. You can save processing time by specifying HURBA, but you might get less accurate numbers than if you specify another value.

Restrictions

The following restrictions apply to ANALYZE HURBA:

- For the following functions, REORG PLUS changes ANALYZE HURBA to ANALYZE to obtain more accurate numbers:

  — dynamic allocation

  — SHRLEVEL CHANGE

  — ANALYZE PAUSE HURBA or ANALYZE ONLY HURBA

- When invoking DSNUTILB, REORG PLUS ignores this option.

- When you also specify the REBALANCE option, REORG PLUS ignores it.

BMCSTATS

BMCSTATS tells REORG PLUS to use information contained in the DASD MANAGER PLUS statistics tables to determine cardinality and average row length.

If the information in the DASD MANAGER PLUS statistics tables is current, you can save processing time by specifying ANALYZE BMCSTATS. The results will be as accurate as if you had specified ANALYZE SCAN.

If the information in the DASD MANAGER PLUS statistics tables for the object that you are reorganizing is missing or incomplete, REORG PLUS changes BMCSTATS to SAMPLE and continues with the reorganization. To populate the statistics tables, either run the BMCSTATS component of DASD MANAGER PLUS or run a reorganization with BMCSTATS YES.

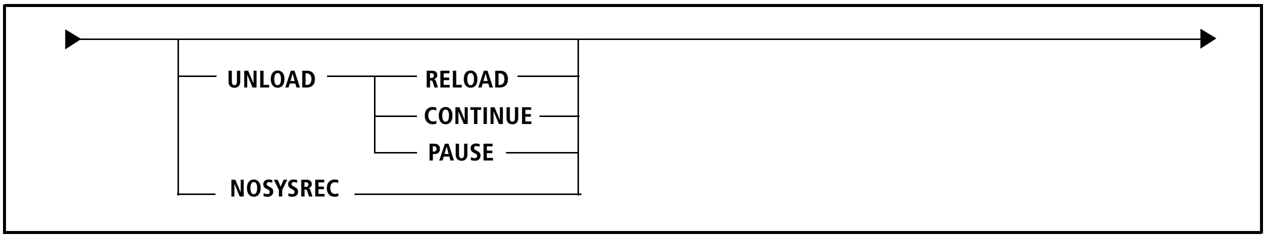
NOTE

The ANALYZE BMCSTATS option is available only if you are using REORG PLUS as a component of the Database Performance for DB2 solution. If you are not, REORG PLUS changes BMCSTATS to SAMPLE and continues with the reorganization.

Restrictions

REORG PLUS ignores this option in the following cases:

- when invoking DSNUTILB
- when reorganizing LOB table spaces



UNLOAD

The UNLOAD option instructs REORG PLUS to continue processing after the UNLOAD phase, to suspend execution after the UNLOAD phase, or to combine the UNLOAD and RELOAD phases into a single processing phase.

RELOAD

UNLOAD RELOAD tells REORG PLUS to combine the UNLOAD and RELOAD phases into a single processing phase named REORG. In this single phase, REORG PLUS provides significant CPU and elapsed time savings over a two-phase reorganization. For performance benefits and considerations when using this single phase, see “REORG phase for a single-phase reorganization” on page 564.

Additional considerations

The following considerations apply to UNLOAD RELOAD:

- If you specify UNLOAD RELOAD and want any VCAT-defined data sets to be deleted and redefined as part of the reorganization, the value of the REDEFINE command or installation option must be YES. You must also provide the necessary IDCAMS commands in the SYSIDCIN data set. For more information, see the REDEFINE option on page 195 and “SYSIDCIN data set” on page 345.

- If you specify UNLOAD RELOAD with ORDER NO, see page 176 for more information about ORDER NO and the various types of reorganizations.

- When invoking DSNUTILB, REORG PLUS passes the value of this option to the IBM DB2 REORG utility as NOSYSREC.

CONTINUE

The CONTINUE option specifies that REORG PLUS continue with a two-phase reorganization process after the data has been unloaded.

Restriction

When you are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect, REORG PLUS changes UNLOAD CONTINUE to UNLOAD RELOAD.

PAUSE

This option does not apply to a SHRLEVEL CHANGE reorganization.

UNLOAD PAUSE instructs REORG PLUS to stop the processing after the data has been unloaded. You can then restart the job in the RELOAD phase. The PAUSE option is useful if you need to redefine data sets during reorganization.

Messages that are displayed at the end of the UNLOAD phase provide the estimated amount of space required to rebuild an index or reload a table space. Key compression of nonleaf pages is not considered when making these estimates.

NOTE

If you are performing a partial reorganization (not reorganizing all partitions of a partitioned table space), do not redefine the nonpartitioned index data sets. REORG PLUS does not rebuild these data sets during a partial reorganization, but only updates them.

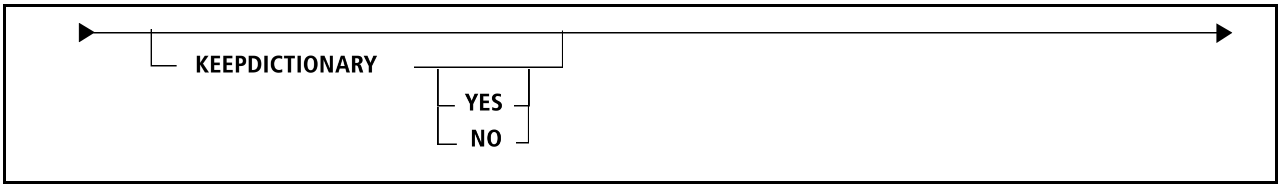
If you specify UNLOAD PAUSE with SHRLEVEL REFERENCE for a partial reorganization with nonpartitioned indexes, copying the nonpartitioned indexes is deferred until the RELOAD phase following the restart.

Restriction

This option is not valid when you are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect. REORG PLUS issues message BMC50115E and terminates.

NOSYSREC

Specifying NOSYSREC is the same as specifying UNLOAD RELOAD.



KEEPDICTIONARY

This option applies to a table space reorganization only.

The KEEPDICTIONARY option tells REORG PLUS whether to keep the existing compression dictionary. If you specify KEEPDICTIONARY without a value, REORG PLUS assumes KEEPDICTIONARY YES. For more information about using compression, see “Table space compression” on page 135.

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.

Specifying the default

You can specify the default for the KEEPDICTIONARY command option in your installation options module by using the KEEPDICTIONARY installation option (page 652). 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 KEEPDICTIONARY 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

If you are reorganizing a LOB table space and specify SHRLEVEL CHANGE, REORG PLUS ignores this option for a DSNUTILB reorganization. In all other cases, REORG PLUS passes the value of this option to the IBM DB2 REORG utility as KEEPDICTIONARY.

Restrictions

REORG PLUS ignores a value of YES and treats the option as if you specified KEEPDICTIONARY NO when either of the following conditions exists:

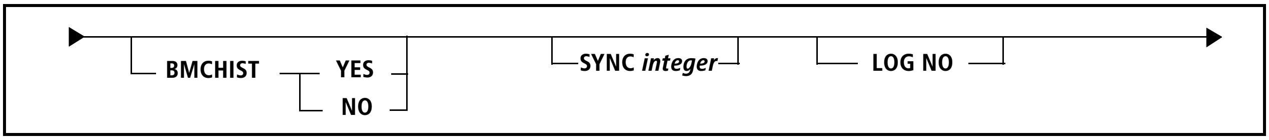
- You are performing partition rebalancing.
- You are reorganizing a partition-by-growth table space.

NO

If you specify KEEPDICTIONARY NO, REORG PLUS builds a new compression dictionary and compresses the data.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.



BMCHIST

This option enables you to choose whether to insert a utility history row into the BMC BMCHIST table when the reorganization successfully completes. This insert occurs in the UTILTERM phase.

You can also use the TERMEXIT option and the TERMEXIT user exit to dynamically control processing of BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS at execution time. For more information, see “TERMEXIT” on page 223 and “Using TERMEXIT to control BMCHIST and statistics updates” on page 795.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the BMCHIST command option in your installation options module by using the BMCHIST installation option (page 630). REORG PLUS was shipped with no default value for this option. The command option overrides the default that is in the installation options module.

YES

If you specify YES, REORG PLUS performs the insert.

NO

If you specify NO, REORG PLUS bypasses the insert.

SYNC

REORG PLUS writes records to the BMCSYNC table that show the number of 1-KB rows that REORG PLUS processed during the UNLOAD and RELOAD phases for a two-phase reorganization, or during the REORG phase for a single-phase reorganization. You can use this information to determine how far the REORG job has progressed.

By default, REORG PLUS writes records only after it has read from or loaded the last row or key in a table space, index, or partition. If you want records written more often, specify an integer with the SYNC option to identify the number of 1-KB rows that you want REORG PLUS to process between writing to the BMCSYNC table.

Additional considerations

Note the following additional information about the SYNC option:

- REORG PLUS records monitoring and restart sync points in the BMCSYNC table as the job progresses. You can issue an SQL statement to query this table to determine how far the reorganization has progressed and the status of objects that you are reorganizing.

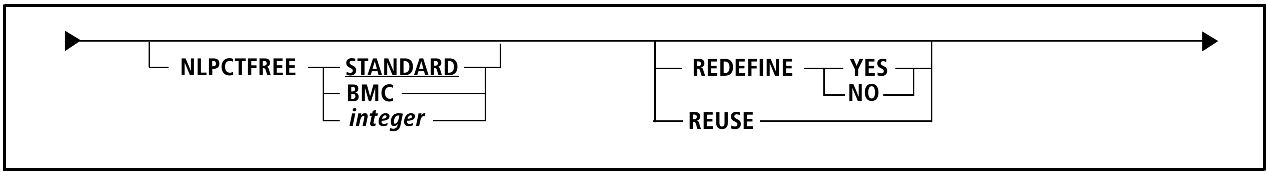
- The SYNC option does not control the sync points that are used to restart REORG PLUS. The restart sync points are established only after the last row or key is loaded in a table space, index, or partition.

- When invoking DSNUTILB, REORG PLUS ignores this option.

LOG NO

LOG NO is not used by REORG PLUS but is provided for compatibility with the IBM DB2 REORG utility command syntax.

When invoking DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility for processing.



NLPCTFREE

NLPCTFREE specifies the percentage of each nonleaf index page to reserve as free space when REORG PLUS rebuilds the indexes.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

STANDARD

The STANDARD option (the default) tells REORG PLUS to use the value that you specified in the DB2 PCTFREE option when you created the index. Like DB2, REORG PLUS leaves up to 10 percent of a nonleaf page free. If you specified a value greater than 10, only 10 percent is left free.

BMC

The BMC option tells REORG PLUS to honor the DB2 PCTFREE value in the DB2 catalog, even if the value is greater than 10.

integer

Specifying an integer identifies the percentage of each nonleaf index page to reserve as free space. You can specify any integer from 0 through 99.

REDEFINE

This option controls whether REORG PLUS deletes and redefines the VSAM data sets for the table space or index space as part of the reorganization. REORG PLUS can redefine both user-defined (VCAT-defined) data sets and data sets that are defined in DB2 storage groups (STOGROUP-defined).

Additional considerations

Note the following additional information about the REDEFINE option:

- When the following conditions exist, you must specify REDEFINE YES:

  —Before the reorganization, an ALTER changed the compression attribute of a participating index.

  —The 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. For more information, see “Staging data sets” on page 98.

- 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 222 and “Using DSRSEXIT to manage VSAM data set redefinition” on page 780.

Specifying the default

You can specify the default for the REDEFINE command option in your installation options module by using the REDEFINE installation option (page 665). 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 set” on page 345.

Storage group considerations

For storage-group-defined objects, the following considerations apply:

- When you specify this option for storage-group-defined data sets, do not provide a SYSIDCIN data set.

- For a storage-group-defined table space or index, REORG PLUS uses the value in DSNZPARMs for the SMS DATACLAS 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 222 and “Using DSRSEXIT to manage VSAM data set redefinition” on page 780.

- 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.

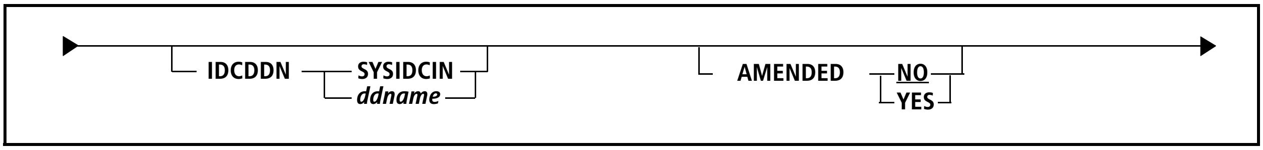
REUSE

This option is used for compatibility with the IBM DB2 REORG utility command syntax. If you specify REUSE and REDEFINE YES, REORG PLUS uses the last keyword that it finds in the command string.

If you specify REUSE, REORG PLUS functions differently depending on the type of reorganization job that you are running:

- When invoking DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility for processing.

- For all other jobs, REORG PLUS functions as if you specified REDEFINE NO.



IDCDDN

The IDCDDN option allows you to override the default ddname for the input data set containing the IDCAMS command statements that REORG PLUS uses to redefine VSAM data sets. The default is SYSIDCIN. For specification guidelines and detailed information about the use of the SYSIDCIN data set, see “SYSIDCIN data set” on page 345.

AMENDED

This option applies to a table space reorganization only.

The AMENDED option specifies that one or more of the tables in the table space were changed by using the BMC Software DATA PACKER *for DB2* AMEND function. The option also specifies whether you want the rows of each table to be re-encoded (using the table’s defined EDITPROC) during the reorganization.

NOTE

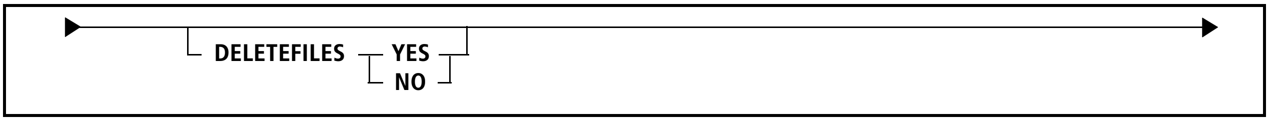
If a table has an index and an EDITPROC, the EDITPROC is normally invoked to decode the row, regardless of the value of this option.

NO

Specifying AMENDED NO, the default prevents REORG PLUS from re-encoding the rows in the table space.

YES

AMENDED YES causes REORG PLUS to invoke any table’s EDITPROC to both decode and then re-encode the rows.



DELETEFILES

DELETEFILES allows you to specify whether you want REORG PLUS to delete all physical sequential data sets whose ddnames match the SYSREC, SYSUT1, and SORTWK ddnames or ddname prefixes after a reorganization.

NOTE

To restart your job during DELETEFILES processing, use RESTART, not RESTART(PHASE).

If you are running REORG PLUS in a worklist environment, REORG PLUS ignores the value that you specified in the DELFILES installation option and processes the job as if you had specified DELFILES=NO. If you want to delete your data sets, you must specify DELETEFILES YES on the command.

Specifying the default

You can specify the default for the DELETEFILES command option in your installation options module by using the DELFILES installation option (page 637). REORG PLUS was shipped with a default value of YES for this option. The command overrides the default that is in the installation options module.

YES

DELETEFILES YES tells REORG PLUS to perform DELETEFILES processing, which depends on whether the job completed successfully.

Successful job completion

For non-DSNUTILB reorganizations, REORG PLUS deletes all physical sequential data sets whose ddnames match the SYSREC, SYSUT1, and SORTWK ddnames or ddname prefixes.

For DSNUTILB reorganizations, REORG PLUS deletes the following data sets:

- the SYSREC and SYSUT1 data sets that were used during the reorganization
- all SYSREC, SYSUT1, and SORTWK data sets allocated in your JCL

Unsuccessful job completion

For non-DSNUTILB reorganizations, if the job does *not* complete successfully, REORG PLUS performs DELETEFILES processing as part of TERMINATE processing in any of the following circumstances:

- You specify TIMEOUT TERM on the command or installation option and a drain timeout occurs.

- You specify ON FAILURE *phase* TERMINATE UTILITY in the command for one of the following phases, and REORG PLUS terminates with an error in that phase:

  — UNLOAD

  — RELOAD

  — LOGAPPLY (includes the LOGFINAL phase)

  — UTILTERM

  If an error occurs in the UTILTERM phase after REORG PLUS has started renaming the data sets, REORG PLUS stops the job instead of terminating it. In this case, REORG PLUS does not delete any image copy data sets or staging VSAM data sets.

- The reorganization is a SHRLEVEL CHANGE reorganization.

NOTE

For a SHRLEVEL CHANGE reorganization, the default is ON FAILURE *phase* TERMINATE UTILITY for most cases before the UTILTERM phase. You must specify ON FAILURE *phase* STOP UTILITY if you do not want REORG PLUS to perform TERMINATE processing. For more information, see Table 35 on page 206.

During DELETEFILES processing for an unsuccessful job, REORG PLUS deletes the following data sets:

- any unregistered full and incremental copy data sets (local and remote) whose ddname matches a copy data set prefix

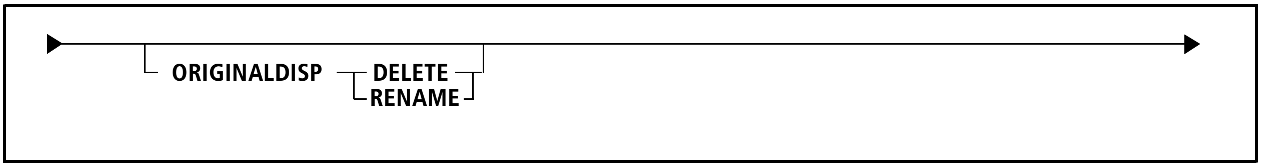
- all physical sequential data sets whose ddnames match the SYSREC, SYSUT1, and SORTWK ddnames or ddname prefixes

- staging VSAM data sets if ORIGINALDISP=DELETE

For DSNUTILB reorganizations, REORG PLUS does not perform DELETEFILES processing for an unsuccessful job.

NO

DELETEFILES NO tells REORG PLUS not to delete any data sets after either a successful or unsuccessful reorganization.



ORIGINALDISP

This option applies to SHRLEVEL REFERENCE and SHRLEVEL CHANGE only.

ORIGINALDISP allows you to specify whether you want REORG PLUS to delete or rename the original data sets after it has renamed the staging data sets and completed the reorganization successfully.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the ORIGINALDISP command option in your installation options module by using the ORIGDISP installation option (page 663). REORG PLUS was shipped with a default value of DELETE for this option. The command overrides the default that is in the installation options module.

DELETE

If you specify this option, REORG PLUS deletes the original data sets.

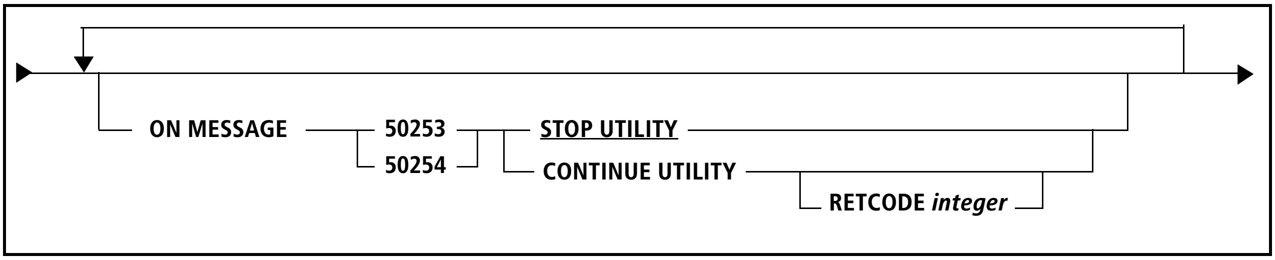
RENAME

If you specify this option, REORG PLUS renames the original data sets to the staging data set names. ORIGINALDISP RENAME enables you to preserve the space initially allocated for the original data sets by renaming them to the staging data set names. They are then ready to be used as the staging data sets in a subsequent reorganization.

REORG PLUS ignores the ORIGINALDISP RENAME option in the following conditions:

- You specify YES for the FASTSWITCH option.
- You specify ZPARM for the FASTSWITCH installation option and the DB2 ZPARM value is YES.

In these cases, 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 98.



ON MESSAGE

This option applies to a table space reorganization only.

ON MESSAGE allows you to specify, for two different error situations, whether REORG PLUS should stop or continue processing when it encounters the error and issues its corresponding error message. If you do not specify ON MESSAGE, the utility stops when it encounters the error. You can repeat this option.

The following message numbers are valid for this option:

- BMC50253 *taskNumber*: DBID *dbid* and PSID *psid* DO NOT MATCH THOSE FOUND IN DATASET *dataSetName*

- BMC50254 *taskNumber*: UNEXPECTED EOF (PAGE=*n*) IN DATASET *dataSetName*

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

STOP UTILITY

STOP UTILITY (the default) tells REORG PLUS to stop the utility when the specified message is issued. The utility ID is left in a restartable state.

CONTINUE UTILITY

CONTINUE UTILITY tells REORG PLUS to continue processing after issuing the corresponding message. Under some fatal conditions, REORG PLUS will not be able to continue. However, if you specify CONTINUE, and REORG PLUS *can* continue, REORG PLUS issues the message number with a suffix of I (indicating that it is an informational message).

NOTE

BMC recommends that you specify CONTINUE only after your utility job has stopped due to an error *and* you have either corrected the problem and restarted the job or determined that you want to restart the job without making any changes.

RETCODE

Use this option to designate the return code that REORG PLUS issues when it encounters the message that you specify with CONTINUE UTILITY. Note the following results for certain values:

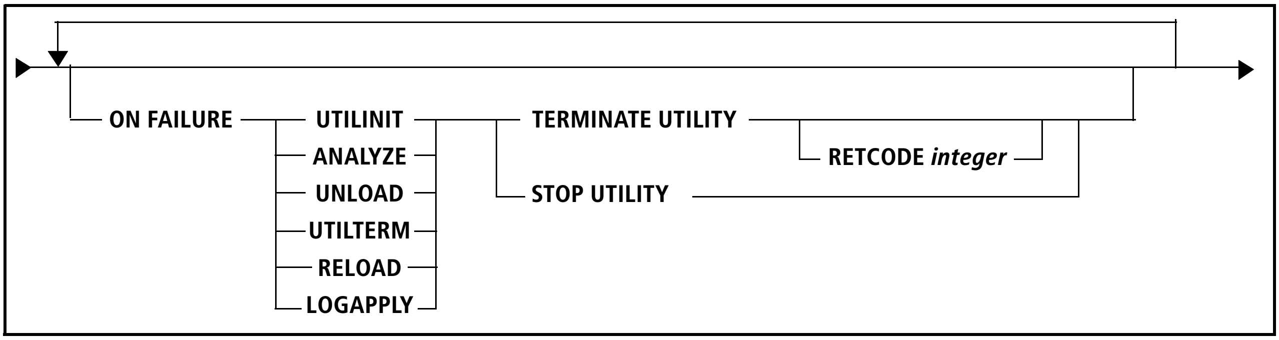
- If you do *not* specify a value for the RETCODE option, REORG PLUS issues the original error return code.

- If you specify 31 or less, REORG PLUS issues the return code.

- If you specify 32 or greater, REORG PLUS issues a user abend code that is equal to the specified integer. Do not specify a value greater than 4095.

NOTE

For a SHRLEVEL CHANGE reorganization, REORG PLUS operates as if you specified ON MESSAGE 50254 CONTINUE UTILITY RETCODE 0 in the ANALYZE phase, even if you do not specify ON MESSAGE.



ON FAILURE

ON FAILURE allows you to recover REORG PLUS from an abnormal termination (when REORG PLUS abends or terminates with a return code that is greater than or equal to eight). You can specify for each phase whether you want REORG PLUS to terminate with a specified return code or stop. If you do not specify ON FAILURE, an abnormal termination stops REORG PLUS.

Additional considerations

The following considerations apply to the ON FAILURE option:

- For a failure in the RELOAD phase of a SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY reorganization, you can only stop the job.

- For a SHRLEVEL CHANGE reorganization, REORG PLUS always functions as if you specified TERMINATE UTILITY.

- When invoking DSNUTILB, REORG PLUS ignores this option.

TERMINATE UTILITY

If you specify ON FAILURE TERMINATE UTILITY and an abend occurs, REORG PLUS cannot honor the TERMINATE option if all of the following conditions exist:

- You are running any type of reorganization *except* SHRLEVEL CHANGE.
- REORG PLUS is unable to start the table space during the UTILTERM phase.
- The abend code is any cancel type ('*x*22').

The ON FAILURE option is valid also for a single-phase reorganization (that is, when UNLOAD RELOAD is in effect). In this case, REORG PLUS combines the UNLOAD and RELOAD phases into one processing phase named REORG. If a failure occurs during unload processing of a single-phase reorganization, REORG PLUS considers the failure to be in the UNLOAD phase. In contrast, if the failure occurs during reload processing, REORG PLUS considers the failure to be in the RELOAD phase.

Table 35 describes processing in each phase when REORG PLUS abnormally terminates and ON FAILURE TERMINATE UTILITY is in effect. For information about the return code that REORG PLUS issues, see page 207.

Table 35 Description of the TERMINATE UTILITY option

Abend in phase	If you specify TERMINATE
UTILINIT ANALYZE	REORG PLUS deletes the row containing the utility ID from the BMCUTIL, BMCSYNC, and BMCDICT tables.
UNLOAD	REORG PLUS deletes the row containing the utility ID from the BMCUTIL, BMCSYNC, and BMCDICT tables. REORG PLUS starts all table and index spaces that are participating in the reorganization. If DELETEFILES YES is in effect, REORG PLUS deletes all of the data sets, any unregistered copy data sets, and (if ORIGINALDISP is DELETE) the staging VSAM data sets.
RELOAD	REORG PLUS deletes the row containing the utility ID from the BMCUTIL, BMCSYNC, and BMCDICT tables. For SHRLEVEL REFERENCE, REORG PLUS starts all table and index spaces taking part in the reorganization. For SHRLEVEL CHANGE, REORG PLUS leaves all objects in their original status. (For SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY, you can only stop REORG PLUS.) If DELETEFILES YES is in effect, REORG PLUS deletes all of the data sets, any unregistered copy data sets, and the staging VSAM data sets (if ORIGINALDISP is DELETE).
LOGAPPLY	REORG PLUS deletes the row containing the utility ID from the BMCUTIL, BMCSYNC, and BMCDICT tables. REORG PLUS leaves all objects in their original status. If DELETEFILES YES is in effect, REORG PLUS deletes all of the data sets, any unregistered copy data sets, and the staging VSAM data sets (if ORIGINALDISP is DELETE). **Note**: The LOGAPPLY phase name applies to both the LOGAPPLY and LOGFINAL phases of a SHRLEVEL CHANGE reorganization.
UTILTERM	Depending on normal UTILTERM processing and the use of the COPY YES option, REORG PLUS might not start the table and index spaces. If DELETEFILES YES is in effect, REORG PLUS deletes all of the data sets, any unregistered copy data sets, and the staging VSAM data sets (if ORIGINALDISP is DELETE). If you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE and the failure occurs during the rename or FASTSWITCH process associated with the staging data sets, REORG PLUS stops the utility instead of terminating it. REORG PLUS leaves the utility ID in a restartable state.

RETCODE

Use this option to designate the return code that REORG PLUS issues when it encounters the message that you specify with TERMINATE UTILITY. Note the following results for certain values:

- If you do *not* specify a value for the RETCODE option, REORG PLUS issues the original error return code.
- If you specify 31 or less, REORG PLUS issues the return code.
- If you specify 32 or greater, REORG PLUS issues a user abend code that is equal to the specified integer. Do not specify a value greater than 4095.

If you specify the TIMEOUT command or installation option with TERM, *rc* and a drain timeout failure occurs, the return code that you specified for TERM overrides the return code from ON FAILURE.

STOP UTILITY

In any phase, when REORG PLUS abnormally terminates and you specify STOP UTILITY for that phase, REORG PLUS stops and leaves the utility ID in a restartable state.

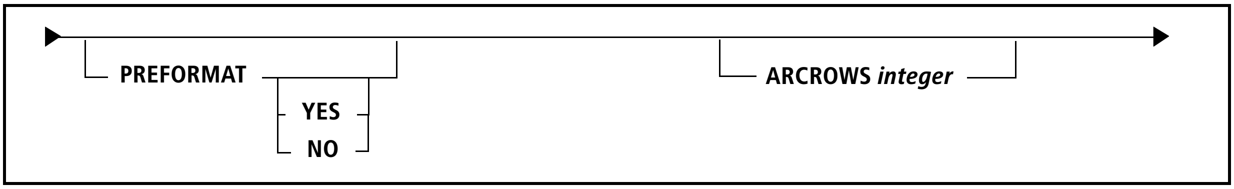
You can also use STOP UTILITY to stop processing and avoid data set renaming when one of the following conditions exists:

- REORG PLUS encounters a key or RID mismatch during the index update process.
- REORG PLUS detects duplicates during the index build process.

NOTE

If you do not use STOP UTILITY and REORG PLUS encounters a key or RID mismatch during index update or detects duplicates during index build, REORG PLUS continues the reorganization and

- completes the LOGAPPLY and LOGFINAL phases
- renames the data sets
- ends with return code 8
- places the corrupted index in RBDP (REBUILD pending) status



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 set” on page 345.

Specifying the default
You can specify the default for the PREFORMAT command option in your installation options module by using the PREFORMAT installation option (page 664). REORG PLUS was shipped with a default value of NO for this option. The command overrides the default that is in the installation options module.

YES

If you specify PREFORMAT YES, REORG PLUS preformats the unused portion of the data set, writing full pages that have been initialized with zeros up to the high-allocated RBA of the table space and index spaces. Preformatting occurs after REORG PLUS reorganizes the data and indexes.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility as PREFORMAT.

NO

If you specify PREFORMAT NO, REORG PLUS does not preformat the unused pages.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

ARCROWS

This option applies to a table space reorganization only.

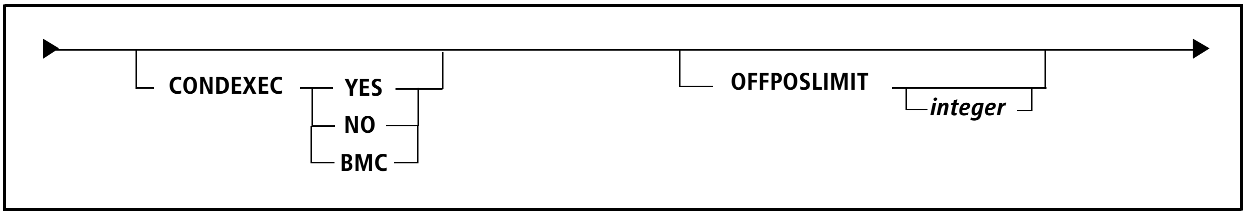
ARCROWS is an estimate of the number of rows that REORG PLUS will discard to a dynamically allocated archive (SYSARC) data set.

If you dynamically allocate the archive data set, REORG PLUS uses the value that you specify for ARCROWS to determine the size of the archive data set. If you do not specify a value for ARCROWS, REORG PLUS allocates enough space to hold all of the rows in the table space.

When you specify a value for both the ARCROWS command option and the SIZEPCT installation or command option, REORG PLUS ignores SIZEPCT for the discard data set and uses only ARCROWS to determine the discard 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.

Specifying the default
You can specify the default for the CONDEXEC command option in your installation options module by using the CONDEXEC installation option (page 631). REORG PLUS was shipped with a default value of NO for this option. The command overrides the default that is in the installation options module.

YES

YES tells REORG PLUS to examine the values of the limit command options (if specified) and the limit installation options in conjunction with the values in the DB2 catalog to determine whether to perform the reorganization. For a description of the limit command options (OFFPOSLIMIT, INDREFLIMIT, and LEAFDISTLIMIT), see page 211, page 213, and page 214. For a description of the limit installation options (OFFPOSLM, INDREFLM, and LEAFDSLM), see page 661, page 649, and page 653.

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:

Return code	Description
1	No calculated value exceeded an exception value. REORG PLUS does not perform a reorganization.
2	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.

For more information, see the following references:

- "Using the DASD MANAGER PLUS exceptions table" on page 130
- "Conditional reorganization" on page 128.

Restrictions:
The following restrictions apply to CONDEXEC BMC:

- This option is available only if you are using REORG PLUS as a component of the Database Performance solution. If you do not have a license for the solution, REORG PLUS ignores the option and continues with the reorganization.
- REORG PLUS terminates if you specify this option when reorganizing a LOB table space.
- When invoking DSNUTILB, REORG PLUS ignores this option.
- If the information in the DASD MANAGER PLUS tables for the object that you are reorganizing is missing or incomplete, REORG PLUS changes BMC to NO and continues with the reorganization.

OFFPOSLIMIT

This option applies to a table space reorganization only.

The OFFPOSLIMIT option allows you to set conditions under which REORG PLUS reorganizes a table space. Valid values are 0 through 100. REORG PLUS also issues a report listing each object it examined, the DB2 catalog values retrieved for each object, and whether each object should be reorganized.

For every table in the table space named in the REORG command, REORG PLUS performs the following calculation. The calculation applies to the specified partitions in SYSIBM.SYSINDEXPART for the table's explicit clustering index.

```
(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 128.

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:

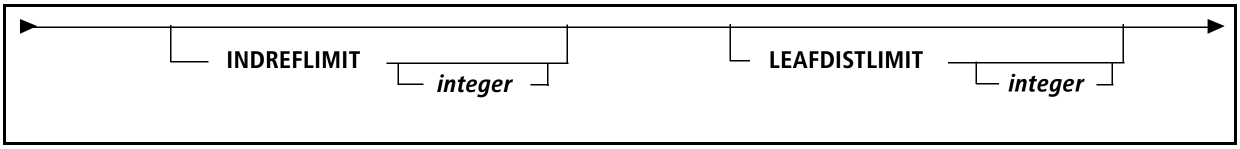
Return code	Description
1	No calculated value exceeded the OFFPOSLIMIT value. REORG PLUS does not perform a reorganization.
2	A calculated value exceeded the OFFPOSLIMIT value. REORG PLUS performs a reorganization. If REORG PLUS issues any return code greater than 2 during processing, that return code 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 (page 661). REORG PLUS was shipped with a default value of 10 for this option. The command overrides the default that is in the installation options module.



INDREFLIMIT

This option applies to a table space reorganization only.

The INDREFLIMIT option allows you to set conditions under which REORG PLUS reorganizes a table space. Valid values are 0 through 100. REORG PLUS also issues a report listing each object it examined, the DB2 catalog values retrieved for each object, and whether each object should be reorganized.

For the table space named in the REORG command, REORG PLUS performs the following calculation. The calculation applies to the specified partitions in SYSIBM.SYSTABLEPART for the table space.

```
(NEARINDREF + FARINDREF) * 100 / CARDF=integer
```

If any calculated integer value exceeds the INDREFLIMIT value, REORG PLUS reorganizes the object.

If you specify the INDREFLIMIT command option without an integer value, REORG PLUS uses the integer value in the corresponding INDREFLM installation option. If you specify CONDEXEC NO or BMC on the REORG command, REORG PLUS ignores the INDREFLIMIT command option. For more information, see “Conditional reorganization” on page 128.

NOTE

You can obtain the report that recommends objects for reorganization *without* performing any reorganizations. To do so, also specify REPORTONLY in the command.

When you specify CONDEXEC YES for either the installation or command option and you specify an INDREFLIMIT value, REORG PLUS issues one of the following return codes:

Return code	Description
1	No calculated value exceeded the INDREFLIMIT value. REORG PLUS does not perform a reorganization.
2	A calculated value exceeded the INDREFLIMIT value. REORG PLUS performs a reorganization. If REORG PLUS issues any return code greater than 2 during processing, that return code supersedes return code 2.

Restriction

If you are reorganizing a LOB table space and specifying this option would cause REORG PLUS to check criteria for conditional execution, REORG PLUS terminates.

Specifying the default

You can specify the default for the INDREFLIMIT command option in your installation options module by using the INDREFLM installation option (page 649). REORG PLUS was shipped with a default value of 10 for this option. The command overrides the default that is in the installation options module.

LEAFDISTLIMIT

This option applies only to an index-only reorganization.

The LEAFDISTLIMIT option allows you to set conditions under which REORG PLUS reorganizes an index. Specify any positive integer (0 or greater). REORG PLUS issues a report that lists each object it examined, the DB2 catalog values retrieved for each object, and whether each object should be reorganized.

For every partition named in the REORG command, REORG PLUS compares the LEAFDISTLIMIT value to the LEAFDIST value in SYSIBM.SYSINDEXPART for the specified index. If any LEAFDIST value exceeds the LEAFDISTLIMIT value, REORG PLUS reorganizes the object.

If you specify the LEAFDISTLIMIT command option without a value, REORG PLUS uses the value in the corresponding LEAFDSLM installation option. If you specify CONDEXEC NO or BMC on the REORG command, REORG PLUS ignores the LEAFDISTLIMIT command option. For more information, see “Conditional reorganization” on page 128.

NOTE

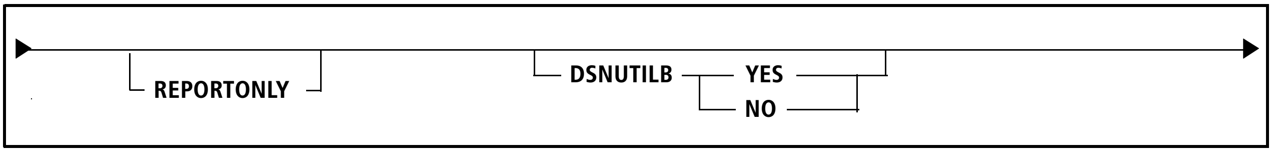
You can obtain the report that recommends objects for reorganization *without* performing any reorganizations. To do so, also specify REPORTONLY on the command.

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:

Return code	Description
1	No calculated value exceeded the LEAFDISTLIMIT value. REORG PLUS does not perform a reorganization.
2	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.

Specifying the default

You can specify the default for the LEAFDISTLIMIT command option in your installation options module by using the LEAFDSLM installation option (page 653). REORG PLUS was shipped with a default value of 200 for this option. The command overrides the default that is in the installation options module.



REPORTONLY

When you specify the REPORTONLY option, REORG PLUS produces a report that lists the tests and results for a conditional reorganization but does not perform the reorganization.

If you specify REPORTONLY along with CONDEXEC YES on the REORG command or in the installation options module, the report lists

- all of the objects that REORG PLUS considered
- the limits that REORG PLUS retrieved for the objects from the DB2 catalog
- whether the objects should be reorganized based on the values of the limit command options (OFFPOSLIMIT, INDREFLIMIT, and LEAFDISTLIMIT), or their corresponding values in the installation options module

If you specify REPORTONLY with CONDEXEC BMC on the REORG command or in the installation options module, the report lists the exceptions that REORG PLUS found in the DASD MANAGER PLUS exceptions table. However, if you specify REPORTONLY with CONDEXEC BMC and you are not using REORG PLUS as part of the Database Performance for DB2 solution (that is, no valid solution password is found), REORG PLUS terminates.

If the value of the CONDEXEC command or installation option is NO, REORG PLUS ignores the REPORTONLY option.

As a result of the REPORTONLY option, REORG PLUS issues one of the following return codes:

Return code	Description
1	No value exceeded the value that you specified in a limit option or no exceptions were found in the DASD MANAGER PLUS exceptions table.
2	A value exceeded the value that you specified in a limit option or an exception was found in the DASD MANGER 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.

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 128.

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 70.

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.

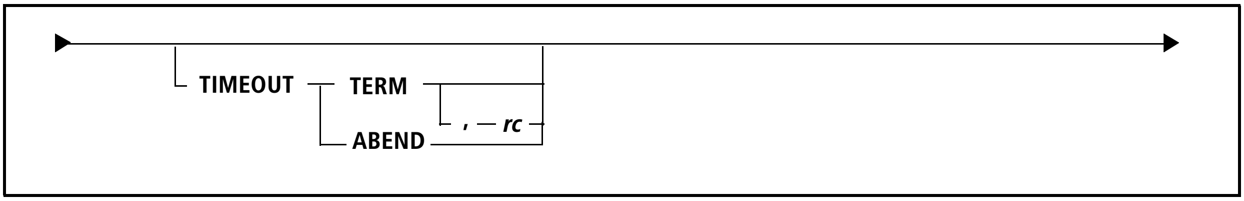
NOTE

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 70.

Specifying the default

You can specify the default for the DSNUTILB command option in your installation options module by using the DSNUTILB installation option (page 642). 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.



TIMEOUT

The TIMEOUT option specifies the action that REORG PLUS should take after it has exhausted all attempts (including retries) to obtain a drain.

The value of the return code depends on what you specify for the TIMEOUT installation and command option and the ON FAILURE command option. Table 36 describes the return code information.

Table 36 Return code hierarchy for the TIMEOUT option

TIMEOUT option	ON FAILURE with RETCODE specified	Return code is returned from
TERM	yes	ON FAILURE The return code is the value that you specified with RETCODE.
TERM	no	TIMEOUT The return code is 8.
TERM, *rc*	does not matter	TIMEOUT The return code is the value that you specified with TERM.
ABEND	does not matter	TIMEOUT The return code is 12.
not specified	no	REORG The return code is 8.

Specifying the default

You can specify the default for the TIMEOUT command option in your installation options module by using the TIMEOUT installation option (page 680). 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.

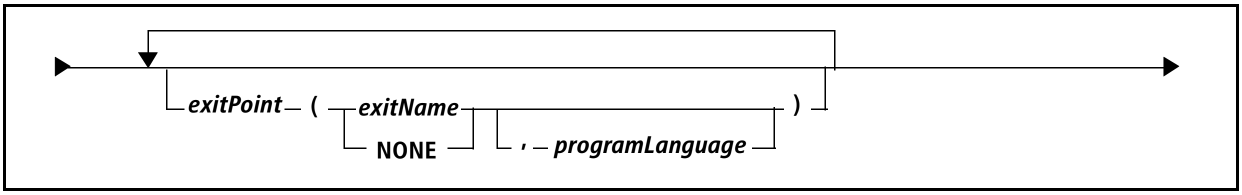
Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

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.



exitPoint

This option enables you to specify the name of the user exit point for which you want to invoke a user-written exit. Table 37 lists the valid values for *exitPoint*, the programming languages that you can specify for each exit, the page on which the description begins for that *exitPoint* value, and the page in Appendix D where you can find the description of the exit.

Table 37 Valid user exit points and supported languages

Valid values for *exitPoint*	Languages allowed	Option description	Exit description
DSNUEXIT	■ assembler The default is assembler (ASM). ■ COBOL II ■ Language Environment COBOL ■ C ■ Language Environment C	page 221	page 735
DSRSEXIT	REXX	page 222	page 780
TERMEXIT	REXX	page 223	page 795

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the program language of a user exit in your installation options module. A program language that you specify with the REORG command overrides the default that is in the installation options module. For more information, see the specific user exits in Appendix A, “REORG PLUS installation options.”

exitName

The *exitName* is the name of the program that you wrote for the exit point.

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 for the specified exit point.

programLanguage

REORG PLUS supports user exits written in one of several languages, depending on the exit point. After the name of your user exit, you can optionally specify the program language in which it was written. If you specify the program language, place a comma between the exit name and the program language.

If you specify an exit name without the language, REORG PLUS assumes that the exit is written in the default language (assembler). Table 37 on page 220 lists the languages that are allowed for each exit point and the defaults. Table 38 displays the value to specify for each language.

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.

Table 38 Program language keywords

Keyword	Specifies an exit written in
ASM	assembler
COBOL2	COBOL II
LE_COBOL	Language Environment COBOL
C	C
LE_C	Language Environment C
REXX	REXX

For more information about creating a user exit, see Appendix C, “REORG PLUS user exits.”

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 details about DSNPAT, see page 289. For details about SPILLDSNPAT, see page 305. For more information about creating a DSNUEXIT user exit and user-defined variables, see Appendix C, “REORG PLUS user exits.”

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the DSNUEXIT command option in your installation options module by using the DSNUEXIT installation option (page 641). 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.

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 set” on page 345.

For details about the DSRSEXIT user exit, see Appendix C, “REORG PLUS user exits.”

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the DSRSEXIT command option in your installation options module by using the DSRSEXIT installation option (page 643). 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.

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 the TERMEXIT user exit, see "Using TERMEXIT to control BMCHIST and statistics updates" on page 795.

NOTE

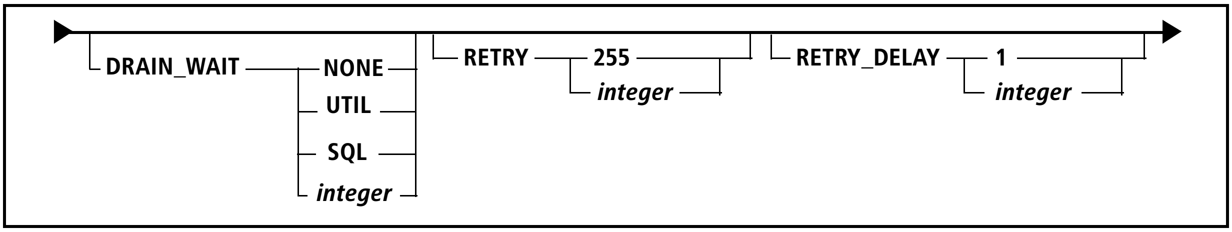
You cannot use a user exit supplied by the TERMEXIT option to override BMCSTATS NO or UPDATEDB2STATS NO to YES.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the TERMEXIT command option in your installation options module by using the TERMEXIT installation option (page 679). REORG PLUS was shipped with no default value for this option. The command overrides the default that is in the installation options module.



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)

When invoking DSNUTILB, REORG PLUS converts the DRAIN_WAIT option based on the DSNZPARM settings, and passes this option to the IBM DB2 REORG utility as DRAIN_WAIT *integer*.

Specifying the default
You can specify the default for the DRAIN_WAIT command option in your installation options module by using the DRNWAIT installation option (page 640). REORG PLUS was shipped with a default value of NONE for this option. The command overrides the default that is in the installation options module.

NONE

NONE means that the drain request issued by REORG PLUS times out immediately if the drain cannot acquire the lock. NONE prevents any application transactions from being queued during the drain process. BMC recommends that you specify NONE in high-transaction environments, such as SAP.

When invoking DSNUTILB, REORG PLUS converts this option to DRAIN_WAIT 1.

UTIL

UTIL tells REORG PLUS to use the standard DB2 utility timeout value defined in DSNZPARMs for your site (IRLMRWT multiplied by UTIMOUT). The wait time applies to each object involved in the reorganization.

When invoking DSNUTILB, REORG PLUS converts this option to DRAIN_WAIT *integer*, where *integer* is the product of SPRTMTOUT multiplied by SPRMUTO.

SQL

If you specify SQL, REORG PLUS uses the standard SQL timeout value (IRLMRWT) as the drain timeout value. The wait time applies to each object involved in the reorganization.

When invoking DSNUTILB, REORG PLUS converts DRAIN_WAIT SQL to DRAIN_WAIT *integer*, where *integer* equals SPRMTOUT.

integer

Specify any integer value from 0 through 1800, as follows:

- 0 is equivalent to the value UTIL.
- 1 through 1800 specifies the number of seconds to wait to obtain the drain for each drain retry before timing out.

RETRY

The RETRY option specifies the maximum number of times that you want REORG PLUS to attempt to obtain a drain before it terminates. The number of attempts can range from 0 through 255.

When invoking DSNUTILB, REORG PLUS ignores this option.

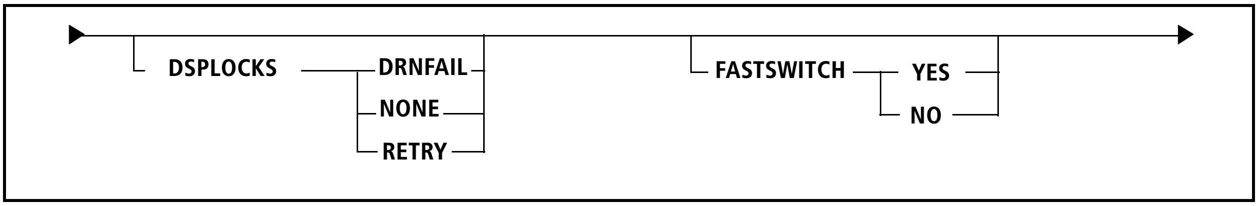
Specifying the default
You can specify the default for the RETRY command option in your installation options module by using the DRNRETRY installation option (page 639). REORG PLUS was shipped with a default value of 255 for this option. The command overrides the default that is in the installation options module.

RETRY_DELAY

After a drain times out, the RETRY_DELAY option specifies the minimum number of seconds that you want REORG PLUS to wait before it tries again to obtain the drain. The number of seconds can range from 1 through 1800.

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the RETRY_DELAY command option in your installation options module by using the DRNDELAY installation option (page 639). REORG PLUS was shipped with a default value of 1 for this option. The command overrides the default that is in the installation options module.



DSPLOCKS

The DSPLOCKS option tells REORG PLUS what action to take regarding displaying claims and locks if a drain attempt times out.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the DSPLOCKS command option in your installation options module by using the DSPLOCKS installation option (page 642). REORG PLUS was shipped with a default value of DRNFAIL for this option. The command overrides the default that is in the installation options module.

DRNFAIL

DRNFAIL tells REORG PLUS to display the claims and locks once, after the final attempt to obtain the drain times out.

NONE

NONE tells REORG PLUS not to display any claims or locks.

RETRY

RETRY tells REORG PLUS to display claims and locks after each drain timeout.

FASTSWITCH

This option applies to SHRLEVEL REFERENCE or SHRLEVEL CHANGE.

The FASTSWITCH option determines the action that REORG PLUS takes in the UTILTERM phase regarding the staging data sets. For more information about FASTSWITCH processing, see “Staging data sets and the FASTSWITCH process” on page 102.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

Specifying the default

You can specify the default for the FASTSWITCH command option in your installation options module by using the FASTSWITCH installation option (page 644). 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.

Restriction

When reorganizing clone objects, REORG PLUS changes FASTSWITCH YES to FASTSWITCH NO.

Additional considerations

Consider the following information before using the FASTSWITCH process:

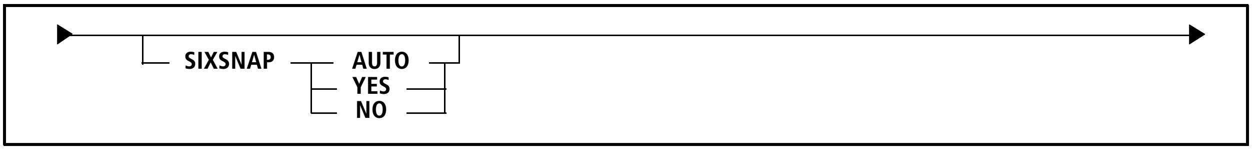
- BMC recommends specifying FASTSWITCH YES when you are reorganizing objects with more than 200 data sets, such as a segmented table space that contains many tables that each have an index.

- 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. For more information, see “Staging data sets and the FASTSWITCH process” on page 102.

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.



SIXSNAP

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

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. 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 when using SIXSNAP, see "Instant Snapshot with nonpartitioned indexes" on page 116. For a list of the supported devices, see the *EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.*

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the SIXSNAP command option in your installation options module by using the SIXSNAP installation option (page 671). REORG PLUS was shipped with a default value of NO for this option. The command overrides the default that is in the installation options module.

AUTO

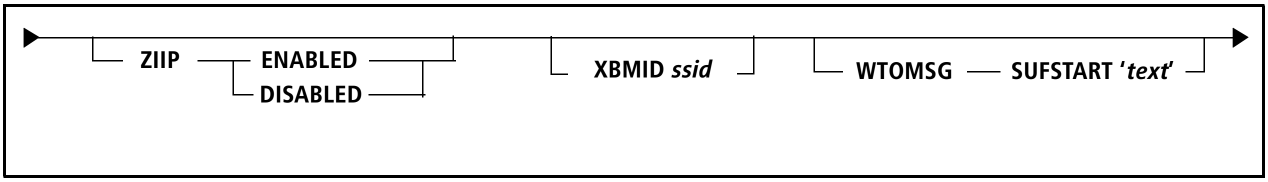
AUTO tells REORG PLUS to first attempt to use Instant Snapshot. If the Instant Snapshot copy fails, REORG PLUS changes SIXSNAP to NO and uses the software-based copy method to recopy the index that could not be copied with Instant Snapshot.

YES

YES tells REORG PLUS to use only Instant Snapshot technology. If the Instant Snapshot copy fails, REORG PLUS terminates.

NO

NO tells REORG PLUS to use the software-based copy method to copy each nonpartitioned index.



ZIIP

The ZIIP option tells REORG PLUS whether to attempt to use IBM® System z® Integrated Information Processors (zIIPs). REORG PLUS can use enclave service request blocks (SRBs) to enable zIIP processing automatically while running jobs. Using zIIP processing can reduce the overall CPU time for REORG PLUS jobs.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the ZIIP command option in your installation options module by using the ZIIP installation option (page 688). REORG PLUS was shipped with a default value of ENABLED for this option. The ZIIP command option overrides the default that is in the installation options module.

ENABLED

ZIIP ENABLED tells REORG PLUS to attempt to offload eligible processing to an available zIIP. If the zIIP is busy or not available, normal processing continues on a general-purpose processor.

To enable and use zIIP processing with REORG PLUS, you must

- have an installed authorized version of XBM or SUF
- start and maintain an XBM subsystem in your environment

NOTE

You can specify a particular XBM subsystem to use by specifying a value for the XBMID installation or command option. For more information, see "XBMID" on page 230 or page 686.

- 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*.

DISABLED

ZIIP DISABLED tells REORG PLUS to not attempt to use zIIP processing.

XBMID

Specify XBMID to identify the appropriate active XBM subsystem when you use either XBM or SUF with REORG PLUS. You must specify an XBM subsystem in either of the following cases:

- You are using a feature that uses the snapshot-processing functions of XBM or SUF.
- You want to use a specific XBM subsystem for zIIP processing.

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.

For more information about using XBM or SUF with REORG PLUS, see “XBM and SUF considerations” on page 140. For more information about XBM and SUF, see the *EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide*.

Snapshot processing

SHRLEVEL CHANGE uses XBM to create a snapshot of the data sets to be reorganized. You can use the XBM Utility Monitor function or MVS console support to dynamically override certain SHRLEVEL CHANGE command options while the reorganization is running. For more information, see “Using XBM to view and dynamically control the log apply process” on page 592.

Both SHRLEVEL CHANGE and SHRLEVEL REFERENCE use the Instant Snapshot technology of XBM or SUF to copy nonpartitioned indexes if you specify YES for the SIXSNAP command or installation option. For more information about the SIXSNAP command option, see “SIXSNAP” on page 228.

zIIP processing

If you specify an XBM subsystem and ZIIP ENABLED is in effect, REORG PLUS attempts to use that subsystem to enable zIIP processing. If that subsystem is not available or not at the correct maintenance level, zIIP processing is not enabled.

If you do not specify an XBM subsystem (either here or with the XBMID installation option), REORG PLUS searches for an XBM subsystem at the appropriate maintenance level to enable zIIP processing.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the XBMID command option in your installation options module by using the XBMID installation option (page 686). The command option overrides the default that is in the installation options module. If the XBMID is not assigned in the installation option, you must specify this command option if you plan to use XBM or SUF.

WTOMSG

This option applies to SHRLEVEL CHANGE only.

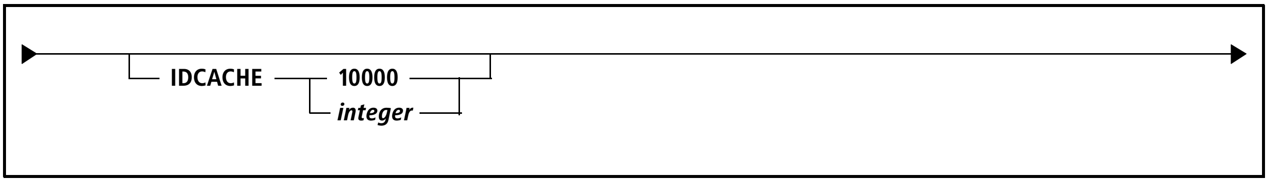
Specify WTOMSG SUFSTART to write message BMC50008I to the MVS™ system log to indicate that the REORG PLUS snapshot initialization has successfully completed. You can use the text of this message to trigger the submission of jobs that you want to run concurrently with the REORG PLUS job.

Specify a text string of up to 50 characters enclosed in single quotes to indicate the message to print in the MVS system log. REORG PLUS truncates strings greater than 50 characters. Quotes cannot appear within the text string. The message has the following format:

```
BMC50008I 'text'
```

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.



IDCACHE

The IDCACHE option allows you to specify the size of the cache that REORG PLUS uses when populating document ID columns. Specify the size as the number of values to reserve. REORG PLUS reserves at least one cache for each unload and log apply task that requires it. The valid values for this option are 1 through 2147483647.

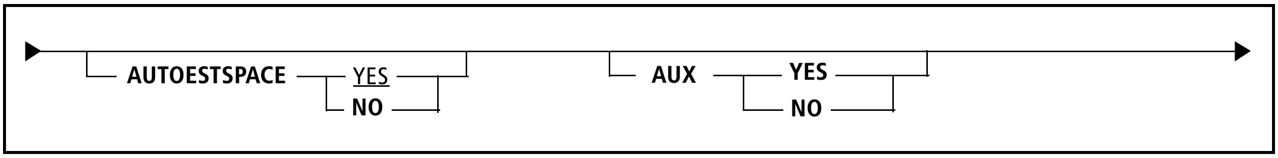
REORG PLUS reserves this cache of numbers in SYSIBM.SYSSEQUENCES. BMC recommends that you use the default value of 10000. Specifying a cache that is too large or too small might cause REORG PLUS to retrieve values that it will not use. Specifying a smaller cache size can also impact performance because REORG PLUS must access and update the DB2 catalog more frequently.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the IDCACHE command option in your installation options module by using the IDCACHE installation option (page 648). REORG PLUS was shipped with a default value of 10000 for this option. The command overrides the default that is in the installation options module.



AUTOESTSPACE

This option applies to a DSNUTILB table space reorganization only.

The AUTOESTSPACE option specifies whether to calculate the optimal hash space size for the table space. REORG PLUS ignores this option if the table space does not contain a table defined as ORGANIZE BY HASH.

Currently, this option applies only when REORG PLUS invokes DSNUTILB. When not invoking DSNUTILB, REORG PLUS ignores this option.

YES

YES (the default) indicates that real-time statistics values are to be used to calculate the hash space.

NO

NO indicates that the HASH SPACE value that is defined on the table is to be used.

AUX

This option applies to a DSNUTILB table space reorganization only.

The AUX option specifies whether to reorganize associated LOB table spaces while reorganizing the base table space.

Currently, this option applies only when REORG PLUS invokes DSNUTILB. When not invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify a default for the AUX command option in your installation options module by using the AUXREORG installation option (page 628). REORG PLUS was shipped with a default value of DEFAULT for this option. The AUX command option overrides the default that is in the installation options module.

YES

YES indicates that associated LOB table spaces are to be reorganized when you specify REORG TABLESPACE on the base table space. If the base table space is partitioned, this option applies only to the LOB table spaces associated with the partitions that are being reorganized.

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 when either of the following conditions exists:

- Participating partitions are in REORP status.
- You are reorganizing a range-partitioned table space and specify REBALANCE.

To prevent your job from terminating when either of these conditions exists, either specify AUXREORG=DEFAULT in your installation options module, or specify AUX YES on the REORG command.

Statistics options

The statistics options allow you to update statistical information in the DASD MANAGER PLUS database statistics tables and in the DB2 catalog. REORG PLUS uses the BMC Common Statistics component to update these statistics.

NOTE

REORG PLUS automatically updates the DB2 real-time statistics tables. For more information, see “DB2 real-time statistics” on page 133.

Reporting options

The Common Statistics component generates a report of the statistics that it updates. You can direct the output of this report in any of the following ways:

- Send the output to a data set that is separate from the REORG PLUS SYSPRINT.

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

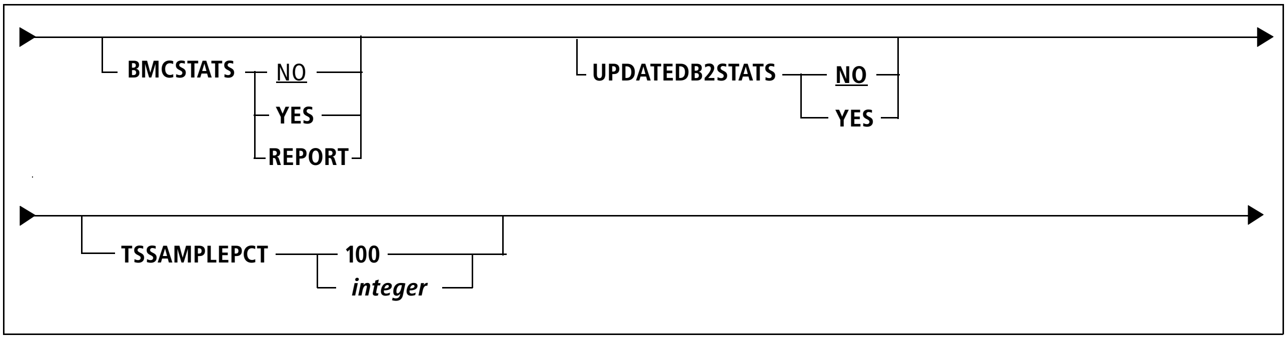
- Send the output to the REORG PLUS SYSPRINT.

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

- Suppress the output.

  To suppress the output, specify `//ASUSRPRT DD DUMMY` in your JCL.

For descriptions of the fields that appear in this report, see the *DASD MANAGER PLUS for DB2 Reference Manual*.



BMCSTATS

The BMCSTATS option tells REORG PLUS whether to use the BMC statistics that it gathers to update the DASD MANAGER PLUS database statistics tables.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

NO

BMCSTATS NO, which is the default, tells REORG PLUS not to update the DASD MANAGER PLUS database statistics tables.

YES

BMCSTATS YES tells REORG PLUS to update the DASD MANAGER PLUS database statistics tables. This option requires that you have the DASD MANAGER PLUS product installed.

Restrictions
The following restrictions apply to BMCSTATS YES:

- REORG PLUS changes BMCSTATS YES to BMCSTATS NO when any of the following conditions exists:

  — you are reorganizing a LOB table space
  — if statistics collection fails in any one of the reload or build tasks
  — on restart, if any participating table space partition was completely loaded or index partition was completely built in the original job

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

Additional considerations

The following additional considerations apply to BMCSTATS YES:

- You must include the BMCPSWD data set in the STEPLIB of your reorganization job.
- For any columns that have a FIELDPROC defined, REORG PLUS gathers statistics for the encoded values from the FIELDPROC. REORG PLUS does not gather statistics for column values that are stored in SYSIBM.SYSFIELDS.
- For multi-column keys, REORG PLUS uses only the first column to update column statistics in the DASD MANAGER PLUS tables.
- The TERMEXIT user exit provides REORG PLUS with user-defined variables that allow you to dynamically control processing of updates to the BMCHIST table and statistics tables during termination processing. You cannot use a TERMEXIT user exit to change BMCSTATS NO to YES. For information about the TERMEXIT user exit, see “Using TERMEXIT to control BMCHIST and statistics updates” on page 795.
- You can control the amount of table space sampling that REORG PLUS does for these statistics. For more information, see “TSSAMPLEPCT” on page 240.

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 236.

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 240.

CLUSTERRATIO

In REORG PLUS versions 9.3.00 and earlier, this option enabled you to specify the method for calculating the CLUSTERRATIO value that was updated in the SYSIBM.SYSINDEXES table of the DB2 catalog when you specified UPDATEDB2STATS YES.

With the statistics changes in REORG PLUS version 10.1.00, REORG PLUS no longer uses this option. REORG PLUS ignores any value that you specify for this option and operates as if you specified CLUSTERRATIO STANDARD.

BMC plans to remove this option in a future version.

UPDATEDB2STATS

UPDATEDB2STATS tells REORG PLUS whether to update statistics in the DB2 catalog. DB2 uses these statistics to determine the access paths that the DB2 optimizer selects.

NO

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

YES

UPDATEDB2STATS YES tells REORG PLUS to update statistics in the DB2 catalog. REORG PLUS uses the BMC Common Statistics component to update access path statistics.

NOTE

REORG PLUS passes the following options to the Common Statistics component. REORG PLUS does not enable you to change these options.

- UPDATEDB2 ACCESSPATH
- HISTORY ACCESSPATH
- FORCEROLLUP N

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:

  — you are reorganizing a LOB table space

  — if statistics collection fails in any one of the reload or build tasks

  — on restart, if any participating table space partition was completely loaded or index partition was completely built in the original job

- REORG PLUS does not collect statistics for the following objects and activities:

  — nonpartitioned indexes when running a partial reorganization

  — activities that occur during the LOGAPPLY phase

  — partitions in a partition-by-growth table space that are added during the reorganization

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 currently use a TERMEXIT user exit to change UPDATEDB2STATS NO to YES. For information about the TERMEXIT user exit, see “Using TERMEXIT to control BMCHIST and statistics updates” on page 795.

- 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."

TSSAMPLEPCT

This option applies to a table space reorganization only.

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

- 1 through 50 tells REORG PLUS to sample the specified percentage of the table space pages.

- 100 tells REORG PLUS to read all table space pages instead of sampling.

NOTE

Values 51 through 99 are not valid.

Restrictions

REORG PLUS ignores the TSSAMPLEPCT option for either of the following types of reorganizations:

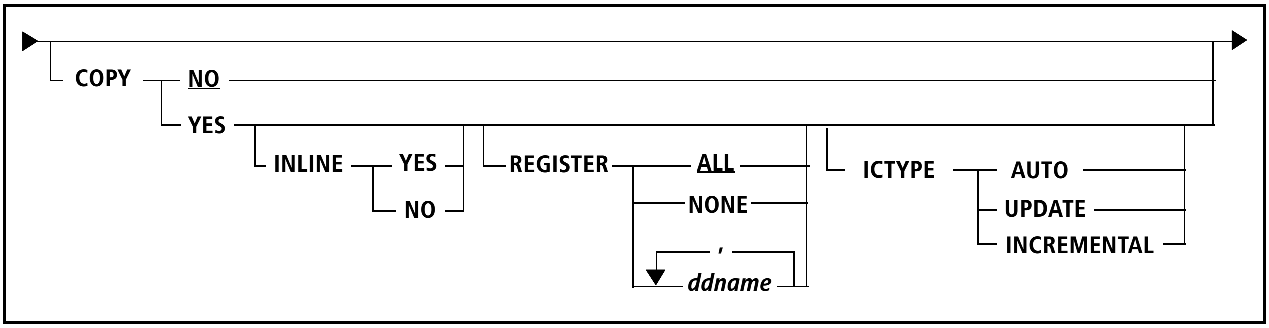
- a DSNUTILB reorganization
- an index reorganization

Specifying the default

You can specify the default for the TSSAMPLEPCT command option in your installation options module by using the TSSAMPLEPCT installation option (page 682). REORG PLUS was shipped with a default value of 100 for this option. The command option overrides the default that is in the installation options module.

Copy options for REORG TABLESPACE

The copy options indicate whether REORG PLUS is to create a copy of the table space and, if so, what type of copy. You can have REORG PLUS create an inline image copy *as* it reloads the table space, or a standard image copy or DSN1COPY-type copy *after* it reloads the table space. Additional options control other aspects of the copy process, including the ability to override the default ddnames of the data sets.



COPY

The COPY option tells REORG PLUS whether or not to produce a copy of the table space or selected partitions. Additional options tell REORG PLUS what type of copy to produce.

NOTE

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.

YES

If you specify COPY YES, REORG PLUS creates a DB2 image copy or DSN1COPY-type copy of a nonpartitioned table space, all partitions of a partitioned table space, or selected partitions of a partitioned table space. For partition-by-growth table spaces, REORG PLUS also copies partitions that it or DB2 adds during the reorganization, as described in “Partition-by-growth table spaces” on page 118.

REORG PLUS creates all copies except inline image copies with system pages at the beginning of the data set (in the same way that the IBM DB2 COPY utility creates image copies when you specify SYSTEMPAGES YES).

For information about allocating data sets for your copies, see "Copy data sets" on page 326.

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.

Restrictions

The following restrictions apply to COPY YES:

- 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.

- If you make *any* copies, BMCCPY must be among them. If you make a BMCRCZ copy, you must also make a BMCRCY copy.

Additional considerations

The following considerations apply to COPY YES:

- If you are running a SHRLEVEL CHANGE reorganization, you might need to make an incremental copy for each corresponding full copy specified, depending on the ICTYPE specification (see "ICTYPE" on page 244). For information about SHRLEVEL CHANGE and the incremental copy data sets, see "Incremental copy data sets" on page 605.

- If the table space was in COPY pending status before the reorganization, REORG PLUS resets the COPY pending status unless you specify REGISTER NONE.

- The difference between an image copy and a DSN1COPY-type copy is one of registration: an image copy is registered, but a DSN1COPY is not.

INLINE

The INLINE option tells REORG PLUS whether to make an inline image copy as it reloads the table space. When invoking DSNUTILB, REORG PLUS passes INLINE to the IBM DB2 REORG utility, which creates inline, single data set copies.

Specifying the default

You can specify the default for the INLINE command option in your installation options module by using the INLINECP installation option (page 649). 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 exist 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 unless a LOB column exists in the table.
- 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.

Restriction

If all of the following conditions exist with INLINE NO, REORG PLUS terminates:

- You specify ICTYPE UPDATE.
- You attempt to create a single image copy for multiple partitions, as in the following scenarios:
  - —You allocate a single copy data set in your JCL but specify multiple partitions.
  - —You specify COPYLVL FULL and COPYSUBSET=YES.

REGISTER

COPY YES REGISTER tells REORG PLUS to register some, none, or all of the full and incremental copies with DB2 in the SYSIBM.SYSCOPY table.

Restrictions

Note the following restrictions on registering copies:

- When invoking DSNUTILB, REORG PLUS ignores this option.

- REORG PLUS does not register any copies if any one is rejected because of a duplicate entry in the SYSIBM.SYSCOPY table.

- For a SHRLEVEL CHANGE reorganization, REORG PLUS forces REGISTER ALL (even if you specify NONE or a list of ddnames) and issues message BMC53008I.

  ALL. REGISTER ALL registers all copies requested in the JCL by ddname.

  NONE. REGISTER NONE does not register any of the copies with DB2 and does not leave the object in COPY pending status.

  ddname. This option enables you to specify by ddname the copy data sets to register. REORG PLUS registers only the copies in the data sets specified by this option. The ddnames that you specify in this option must be present in your JCL.

ICTYPE

This option applies to SHRLEVEL CHANGE only.

Based on the value of this option, REORG PLUS updates the full image copy data sets or creates incremental image copy data sets. You will achieve better performance and reduce the amount of time during which application updates are prevented if REORG PLUS does not have to create incremental copies during the LOGFINAL phase. BMC recommends that you use ICTYPE AUTO to allow REORG PLUS to determine which type of copy is best.

REORG PLUS does not support updating SMS-managed striped data sets for the copy data sets unless you specify ICTYPE AUTO or UPDATE and INLINE YES (or INLINECP=YES).

If your installation allows SMS data sets to go to tape for the SMS classes that you specified for dynamically allocated copy data sets, you must specify ICTYPE INCREMENTAL if *all* of the following statements are true:

- You specify SHRLEVEL CHANGE.
- You specify SMS YES.
- The value of the SMSUNIT command or installation option is NO.
- The value of the INLINE command option or INLINECP installation option is NO.

If you specify SHRLEVEL CHANGE and ICTYPE INCREMENTAL, REORG PLUS behaves as if you specified INLINE NO, regardless of the value that you specified for INLINE.

Specifying the default

You can specify the default for the ICTYPE command option in your installation options module by using the ICTYPE installation option (page 646). REORG PLUS was shipped with a default value of AUTO for this option. The command option, when used with COPY YES, overrides the default that is in the installation options module.

AUTO. This value tells REORG PLUS to determine which type of copy is best.

REORG PLUS updates the full copy data sets if *either* of the following statements is true:

- The value of the INLINE command option or INLINECP installation option is YES.

- All of the full copy data sets are on DASD, and one full copy data set exists for each partition that you are reorganizing.

If neither condition is met, REORG PLUS creates incremental image copies. You must define or dynamically allocate these incremental copy data sets.

When invoking DSNUTILB, REORG PLUS ignores this option and DSNUTILB uses ICTYPE UPDATE.

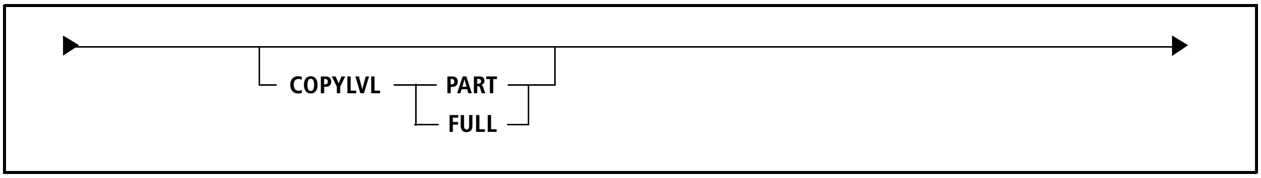
UPDATE. If you specify UPDATE, REORG PLUS always updates the full copy data sets. If you also specify INLINE YES on the command or INLINECP=YES in the installation option, REORG PLUS appends the updated pages to the full copy data sets. When the value of INLINE or INLINECP is YES, the copy data sets can be on tape.

If you specify INLINE NO or INLINECP=NO, REORG PLUS updates the full copy data sets in place. When the value is NO and *either* of the following statements is true, REORG PLUS terminates:

- Any of the data sets are not on DASD.
- 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.

INCREMENTAL. If you specify INCREMENTAL, REORG PLUS creates the incremental copy data sets. You must define or dynamically allocate these data sets. If the data sets do not exist, REORG PLUS terminates.

When invoking DSNUTILB, REORG PLUS changes ICTYPE INCREMENTAL to ICTYPE UPDATE.



COPYLVL

REORG PLUS uses the COPYLVL option only when dynamic allocation is active for copy data sets. COPYLVL tells REORG PLUS how to assign the dynamically allocated image copy data sets in either of the following cases:

- You are reorganizing all partitions of a partitioned table space.
- You are reorganizing a subset of contiguous partitions by using the PART command option and you specify YES for the COPYSUBSET installation option.

Specifying COPYLVL PART can improve performance when reorganizing partitioned objects. However, if you are reorganizing a large number of partitions, consider specifying COPYLVL FULL to avoid encountering memory or data set allocation restrictions. Whether the number of partitions that you have is considered to be a large number of partitions depends on your environment. For more information regarding full copy data sets, see “Copy data sets” on page 326. For more information regarding incremental copy data sets, see “Incremental copy data sets” on page 605.

Specifying the default
You can specify the default for the COPYLVL command option in your installation options module by using the COPYLVL installation option (page 633). 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.

When REORG PLUS invokes DSNUTILB (and you specify COPY YES), DSNUTILB creates a full image copy.

Additional considerations
The following additional considerations apply to COPYLVL PART:

- If you specify a tape device for the UNIT option, REORG PLUS allocates a tape unit for each partition.

- If you are using a GDG name, each partition must have a different GDG base.
- If you are reorganizing a table space that contains more than 99 partitions, use the COPYDDN option (page 248) to specify a ddname prefix for the copy data sets that results in eight characters or less after REORG PLUS appends the highest partition number. If you are also making remote copies, use the RECOVERYDDN option (page 250) in the same way.

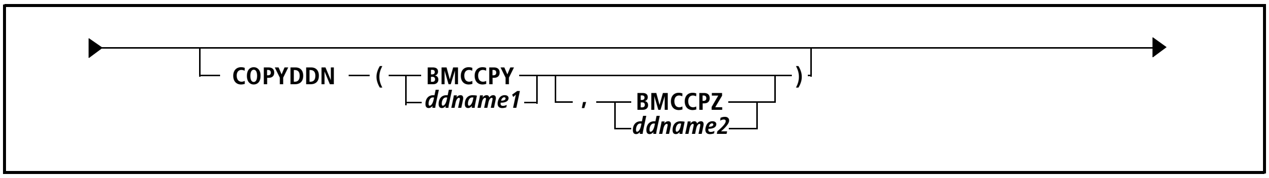
FULL

COPYLVL FULL tells REORG PLUS to allocate a single full copy data set to contain all of the partitions that you are reorganizing. If incremental copy data sets are created, REORG PLUS also allocates a single incremental copy data set to contain all of the partitions that you are reorganizing.

Restrictions

REORG PLUS changes COPYLVL FULL to COPYLVL PART when either of the following conditions exists:

- You specify a subset of partitions, but the value of the COPYSUBSET installation option is NO.
- The value of the COPYSUBSET installation option is YES and you specify a subset of partitions, but they are not contiguous.



COPYDDN

COPYDDN allows you to override the default ddnames or ddname prefixes of the local copy data sets that were specified in your installation options. The ddnames correspond to the data sets that receive a full image copy or DSN1COPY-type copy of the table space or partitions that you are reorganizing.

If you specify *ddname2* for this option, you must either allocate this data set in your JCL, or dynamic allocation must be active for both the primary and backup copy data sets. If you specify *ddname2* and dynamic allocation is active for only the primary copy data set, REORG PLUS terminates because it expects a second copy data set.

If you are registering the copies, *ddname1* will be the DB2 local primary and *ddname2* will be the local backup. For information about specifying and using these data sets, see "Copy data sets" on page 326.

If you use this command option to override the default name in the installation options module, you must also change the ddnames in your JCL.

When REORG PLUS invokes DSNUTILB (and you specify COPY YES), REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

Partition-level copies

If you are making partition-level copies, specify only the ddname prefix (no *nn*) in this option. If you are dynamically allocating copy data sets (and you specify COPYLVL PART), REORG PLUS appends the partition number to the ddname prefix. Specify a prefix that results in eight characters or less after REORG PLUS appends the highest partition number. For more information, see "Specifying ddname prefixes" on page 91.

If you are not dynamically allocating copy data sets, the ddname that you specify in the JCL must have the partition number *nn* appended to this prefix. The length of *nn* can be from one to seven characters, depending on the length of the ddname prefix. The maximum length of the ddname with the prefix must be eight bytes or less. For example, if you are reorganizing partition 157, you could specify ddname1 as BCOPY, and specify BCOPY157 in your JCL.

Dynamic allocation

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

- acceptable set:

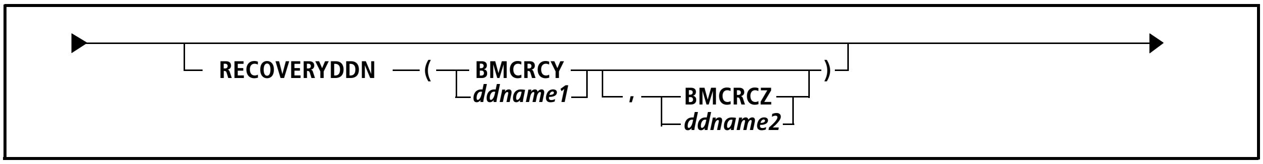
```
BMCRD
BMCRDWK
```

- not acceptable set:

```
BMCRD
BMCRD11
```

Specifying the default

You can specify the default for the COPYDDN command option in your installation options module by using the COPYDDN installation option (page 632). REORG PLUS was shipped with a default value of (BMCCPY, BMCCPZ) 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. The ddnames correspond to the data sets that receive a full image copy or DSN1COPY-type copy of the table space or partitions that you are reorganizing.

If you specify *ddname2* for this option, you must either allocate this data set in your JCL, or dynamic allocation must be active for both the primary and backup remote copy data sets. If you specify *ddname2* and dynamic allocation is active for only the primary copy data set, REORG PLUS terminates because it expects a second copy data set.

If you are registering the copies, *ddname1* will be the DB2 remote primary copy, and *ddname2* will be the remote backup copy. For information about specifying and using these data sets, see "Copy data sets" on page 326.

If you use this command option to override the default name in the installation options module, you must also change the ddnames in your JCL.

When REORG PLUS invokes DSNUTILB (and you specify COPY YES), REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

Partition-level copies

If you are making partition-level copies, specify only the ddname prefix (no *nn*) in this option. If you are dynamically allocating copy data sets (and you specify COPYLVL PART), REORG PLUS appends the partition number to the ddname prefix. Specify a prefix that results in eight characters or less after REORG PLUS appends the highest partition number. For more information, see "Specifying ddname prefixes" on page 91.

If you are not dynamically allocating copy data sets, the ddname that you specify in the JCL must have the partition number *nn* appended to this prefix. The length of *nn* can be from one to seven characters, depending on the length of the ddname prefix. The maximum length of the ddname with the prefix must be eight bytes or less. For example, if you are reorganizing partition 203, you could specify ddname1 as BMCRY, and specify BMCRY203 in your JCL.

Dynamic allocation

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

- acceptable set:

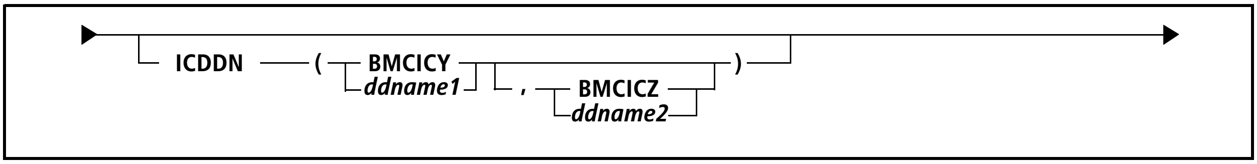
```
BMCRD
BMCRDWK
```

- not acceptable set:

```
BMCRD
BMCRD11
```

Specifying the default

You can specify the default for the RECOVERYDDN command option in your installation options module by using the RCVYDDN installation option (page 665). REORG PLUS was shipped with a default value of (BMCRCY, BMCRCZ) for this option. The command option overrides the default that is in the installation options module.



ICDDN

This option applies to SHRLEVEL CHANGE only.

ICDDN allows you to override the default ddnames or ddname prefixes of the local incremental copy data sets that were specified in your installation options. The ddnames correspond to the data sets that receive an incremental image copy of the table space or partitions that you are reorganizing.

When you register the copies, *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” on page 605.

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. For more information, see “Specifying ddname prefixes” on page 91.

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:

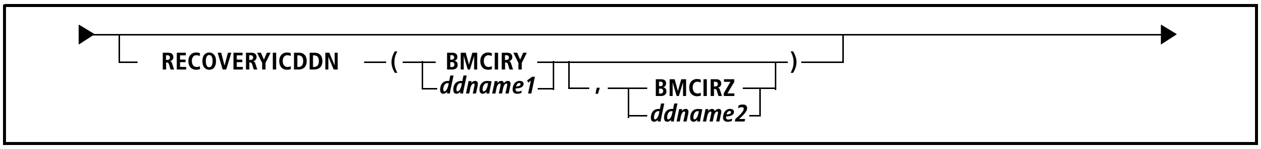
```
BMCRD
BMCRDWK
```

- 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 (page 646). REORG PLUS was shipped with a default value of (BMCICY, BMCICZ) for this option. The command option overrides the default that is in the installation options module.



RECOVERYICDDN

This option applies to SHRLEVEL CHANGE only.

RECOVERYICDDN allows you to override the installation default ddnames corresponding to the data sets that receive a remote incremental image copy or DSN1COPY-type copy of the table space or partitions that you are reorganizing.

When you register the copies, *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” on page 605.

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. For more information, see “Specifying ddname prefixes” on page 91.

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

- acceptable set:

```
BMCRD
BMCRDWK
```

- not acceptable set:

```
BMCRD
BMCRD11
```

Specifying the default

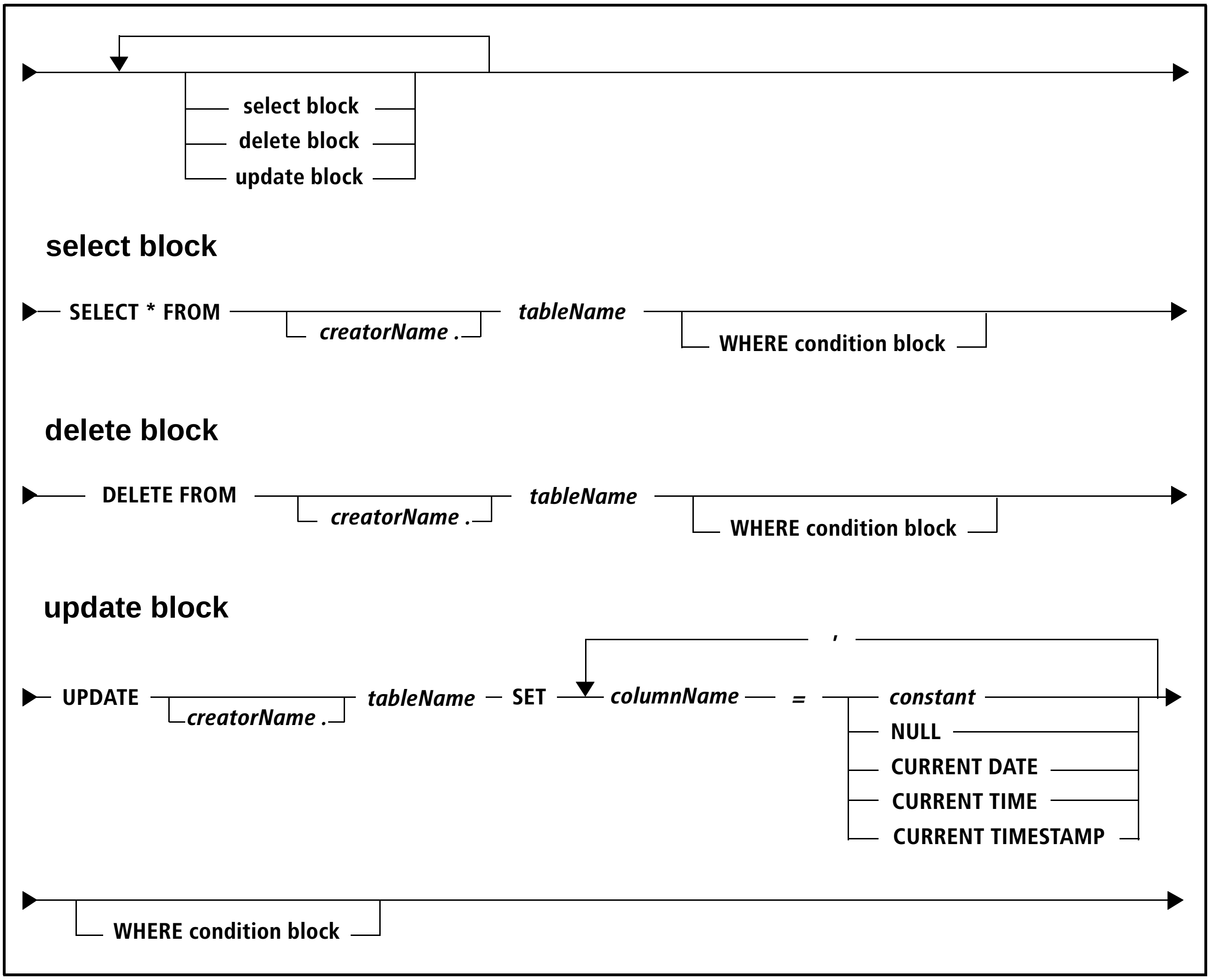
You can specify the default for the RECOVERYICDDN command option in your installation options module by using the RCVICDDN installation option (page 664). REORG PLUS was shipped with a default value of (BMCIRY, BMCIRZ) for this option. The command option overrides the default that is in the installation options module.

Selective unload and update options for REORG TABLESPACE

The SELECT and DELETE options allow you to specify conditions that must be met in order for a row to be unloaded and reloaded during the reorganization. The UPDATE option allows you to reset column values to a constant.

Restrictions
These options do not apply to the following jobs:

- index-only reorganizations
- DSNUTILB jobs
- LOB table space reorganizations
- XML table space reorganizations



DBCS support

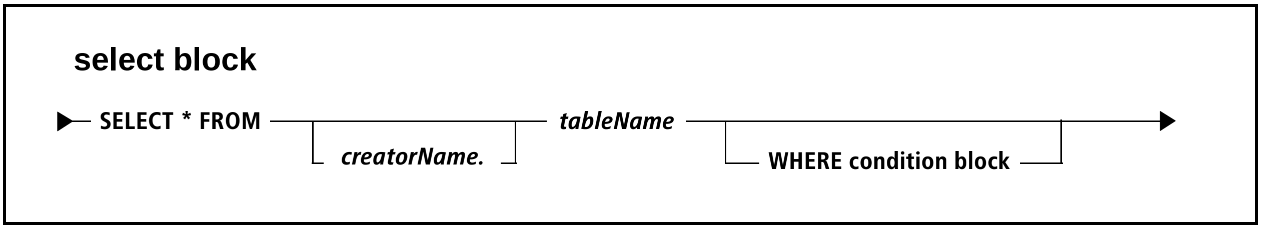
REORG PLUS supports the use of the double-byte character set (DBCS). You can use DBCS characters in DB2 identifiers, such as table or column names, or in the constants specified in your SELECT, DELETE, or UPDATE options. You can also include DBCS characters in DB2 object names, such as database or table 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.



SELECT * FROM

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

The SELECT * FROM option tells REORG PLUS to select rows that match the specified WHERE clause conditions from the named table. REORG PLUS unloads the selected rows and reloads them to the table space. The table name specifies the table to which the select criterion applies. If you do not specify the creator, 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:

- when REORG PLUS invokes DSNUTILB
- when reorganizing LOB or XML table spaces
- when reorganizing table spaces that contain LOB or XML columns

For restrictions on the WHERE clause of your SELECT statement, see "WHERE" on page 264.

Additional considerations

Note the following additional information about the SELECT * FROM option:

- You can specify only one SELECT *or* one DELETE clause per table.

- REORG PLUS deletes rows that are not selected and optionally writes them to the SYSARC archive data set.

  For specification guidelines and detailed information about the use of the SYSARC data set, see "SYSARC data set" on page 342.

- If you are running a SHRLEVEL CHANGE reorganization, see "Considerations for using SELECT or DELETE" on page 602.

WARNING

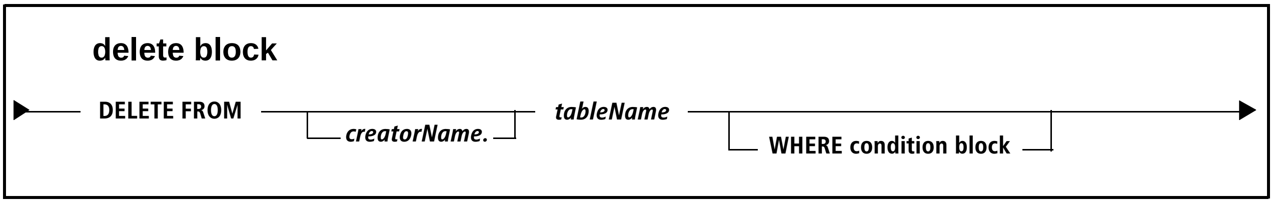
If the rows that you did *not* select are in a table with a primary key, REORG PLUS does not guarantee or maintain any referential integrity of the primary keys after those rows are deleted and does not set the CHECK pending status. You might need to run either the BMC CHECK PLUS for DB2 product or the IBM DB2 CHECK DATA utility on dependent tables to restore referential integrity, or run REORG PLUS on dependent tables to delete dependent rows.

WHERE

The WHERE clause enables you to specify which rows are to be selected. For a detailed description of the WHERE condition, see page 264.

NOTE

When you use WHERE on a partial reorganization, REORG PLUS processes only the specified partitions. If the WHERE condition matches rows in any of the partitions that are not being reorganized, those rows are unaffected. You do not need to specify predicates to limit the WHERE condition to only partitions that you are reorganizing.



DELETE FROM

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

The DELETE FROM option tells REORG PLUS to delete the rows that match the specified WHERE clause conditions from the named table. Rows that REORG PLUS does *not* delete are unloaded and used to reload the table space. The table name specifies the table to which the delete criterion applies. If you do not specify the creator, 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:

- when REORG PLUS invokes DSNUTILB
- when reorganizing LOB or XML table spaces
- when reorganizing table spaces that contain LOB or XML columns

For restrictions on the WHERE clause of your DELETE statement, see “WHERE” on page 264.

Additional considerations
Note the following additional information about the DELETE FROM option:

- You can specify only one SELECT *or* one DELETE clause per table.

- REORG PLUS optionally writes the deleted rows to the SYSARC archive data set.

  For specification guidelines and detailed information about the use of the SYSARC data set, see “SYSARC data set” on page 342.

- If you are running a SHRLEVEL CHANGE reorganization, see “Considerations for using SELECT or DELETE” on page 602.

WARNING

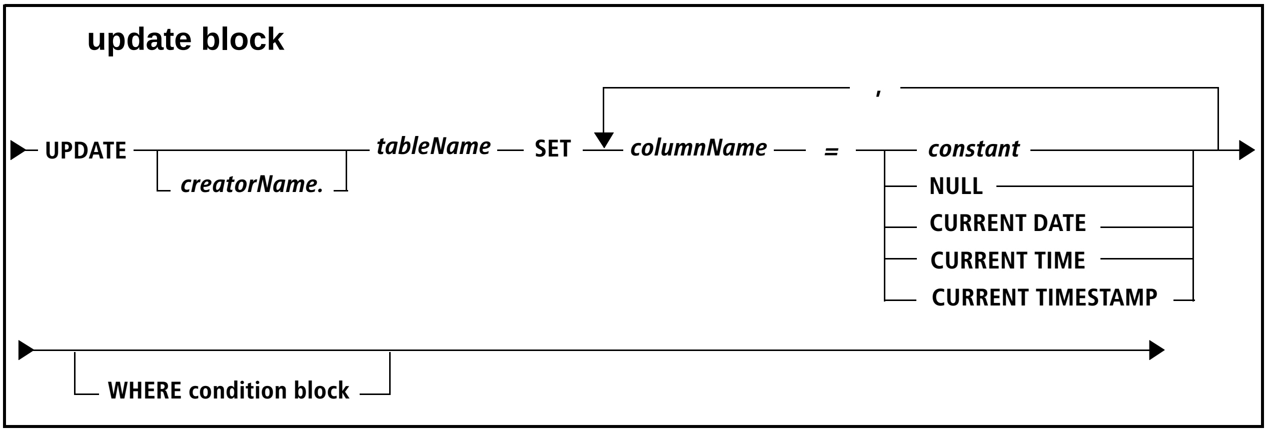
If you are deleting rows from a table with a primary key, REORG PLUS does not guarantee or maintain any referential integrity of the primary keys and does not set the CHECK pending status. You might need to run either the BMC CHECK PLUS product or the IBM DB2 CHECK DATA utility on dependent tables to restore referential integrity, or run REORG PLUS on dependent tables to delete dependent rows.

WHERE

The WHERE clause enables you to specify which rows are to be deleted. For a detailed description of the WHERE condition, see page 264.

NOTE

When you use WHERE on a partial reorganization, REORG PLUS processes only the specified partitions. If the WHERE condition matches rows in any of the partitions that are not being reorganized, those rows are unaffected. You do not need to specify predicates to limit the WHERE condition to only partitions that you are reorganizing.



UPDATE *tableName* SET

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

UPDATE *tableName* SET enables you to reset column values to a constant. The table name specifies the table to which the update criterion applies. If you do not specify the creator, REORG PLUS uses the authorization ID of the user running the reorganization. SET specifies that values in the named columns that match the specified WHERE clause will be set to one of the values listed in Table 39.

Table 39 Values for SET

SET values	Description
constant	updates the column with the specified value Observe the rules for constants in Table 41 on page 267. 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). ***Additional considerations*** The following considerations apply to constants for the SET option: ■ Although not prevented or restricted by DB2, 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.
NULL	resets the column values to null
CURRENT DATE	resets the column values to the current date
CURRENT TIME	resets the column values to the current time
CURRENT TIMESTAMP	resets the column values to the current timestamp

Restrictions

The following restrictions apply to the UPDATE statement:

- REORG PLUS terminates when you specify UPDATE for the following jobs:
  - when REORG PLUS invokes DSNUTILB
  - when reorganizing LOB or XML table spaces
- 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 264.

Additional considerations

Note the following additional information about the UPDATE statement:

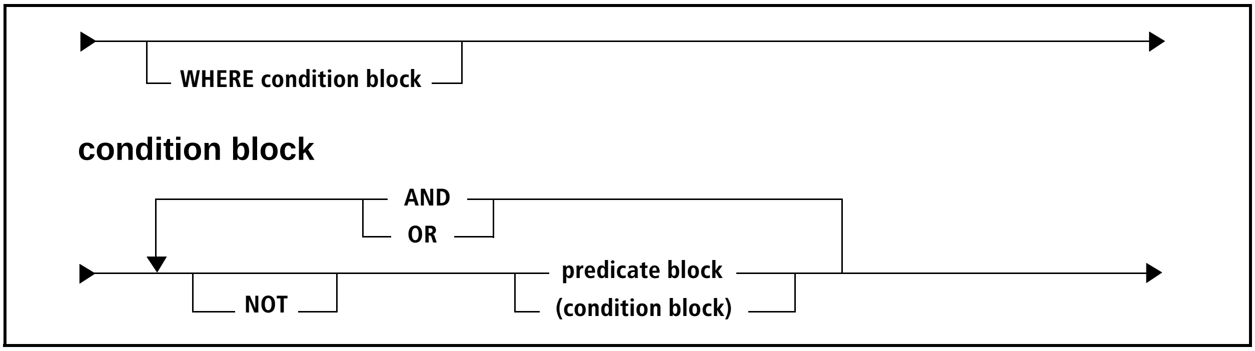
- You can specify only one UPDATE per table.
- During processing, REORG PLUS updates any row change timestamp column when it updates rows.

WHERE

The WHERE clause enables you to specify which rows are to be updated. For a detailed description of the WHERE condition, see page 264.

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. Table 40 shows the result of the WHERE clause when you use the AND, OR, or NOT operator. Predicates are described on page 266.

Table 40 Truth table for AND, OR, NOT

Result of predicate		Result of WHERE condition		
p	**q**	**p AND q**	**p OR q**	**NOT p**
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

Predicates are evaluated left to right, with the following operator precedence:

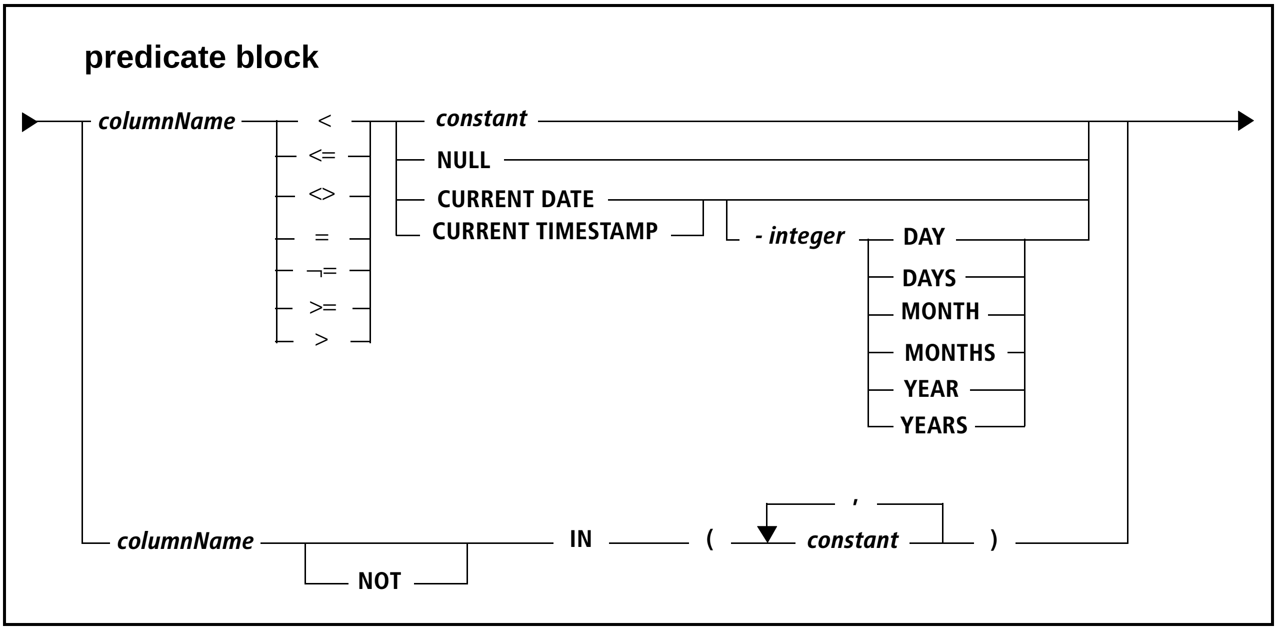
- ( )
- NOT
- AND
- OR

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:

Operator	Meaning
<	less than
< =	less than or equal
< >	not equal
=	equal
¬ =	not equal
> =	greater than or equal
>	greater than
IN	equal to any
NOT IN	not equal to all

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 41 on page 267. 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).

Additional considerations

The following considerations apply to constants in the predicate block:

- Although not prevented or restricted by DB2, 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 41 Rules for constants (part 1 of 2)

Type of constant	Additional details
integer	identical to DB2 SQL integer constants The constant must be within the range of the column’s data type.

decimal	identical to DB2 SQL decimal constants
	The constant must be within the range of the column’s data type.
	Note: To determine whether your site has set the decimal point to a comma, check the DB2 subsystem DSNHDECP setting. To enable REORG PLUS to distinguish comma decimal points from comma separators, include a space after any comma that you use as a separator.
	Example of using the comma as a decimal point:
	`REORG`
	`    SELECT * FROM owner.table`
	`          WHERE SALARY > 100,00`
	Example of using the comma as both a decimal point and a separator:
	`REORG`
	`    SELECT * FROM owner.table`
	`          WHERE SALARY IN (100,00, 200,00, 300,00)`

character string	identical to DB2 SQL strings Use ' ' to denote an empty string. **Note**: REORG PLUS does not validate character data against CCSIDs.
hexadecimal strings	identical to DB2 SQL hexadecimal strings, except that string length is limited to 256 bytes **Notes**: ■ REORG PLUS does not validate hexadecimal string data against CCSIDs. ■ REORG PLUS does not support the hexadecimal string constants BX'xxxx', UX'xxxx', or GX'xxxx'.

Table 41 Rules for constants (part 2 of 2)

Type of constant	Additional details
graphic string	limited to 256 bytes
date / time string	identical to DB2 SQL date / time strings Use only ISO, USA, EUR, and JIS formats.
floating-point	not supported
decimal floating point	
binary strings	
LOBs	
row ID	

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 – *labeled duration*

The CURRENT DATE option describes the current date or optionally describes the current date minus a duration in either days, months, or years. The rules for date / time arithmetic using labeled durations are identical to the DB2 SQL rules given in the DB2 *SQL Reference* manual. The CURRENT DATE is taken from the local time-of-day clock during the UTILINIT phase.

CURRENT TIMESTAMP – *labeled duration*

The CURRENT TIMESTAMP option describes the current timestamp or optionally describes the current timestamp minus a duration in either days, months, or years. The rules for date / time arithmetic using labeled durations are identical to the DB2 SQL rules given in the DB2 *SQL Reference* manual. The CURRENT TIMESTAMP is taken from the local time-of-day clock during the UTILINIT phase.

NOTE

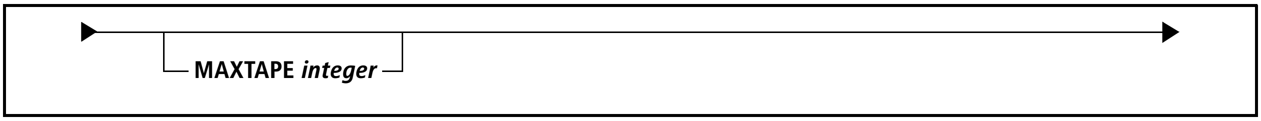
REORG PLUS does not support the WITH TIME ZONE clause for CURRENT TIMESTAMP on a predicate.

Dynamic allocation options

Dynamic allocation options allow you to have REORG PLUS optimally allocate the size and number of work files and copy data sets needed to process your reorganization. Dynamic allocation provides the following benefits:

- eliminates the need for you to include DD statements for the specified file types in your JCL
- eliminates the need for you to calculate data set allocation sizes
- automatically provides an optimal allocation for the highest possible multitasking level
- allows you to use symbolic variables and user-defined variables to assist in creating data set names

The following sections provide the syntax required for enabling and using dynamic data set allocation. For more information, see “Dynamic data set allocation” on page 88.

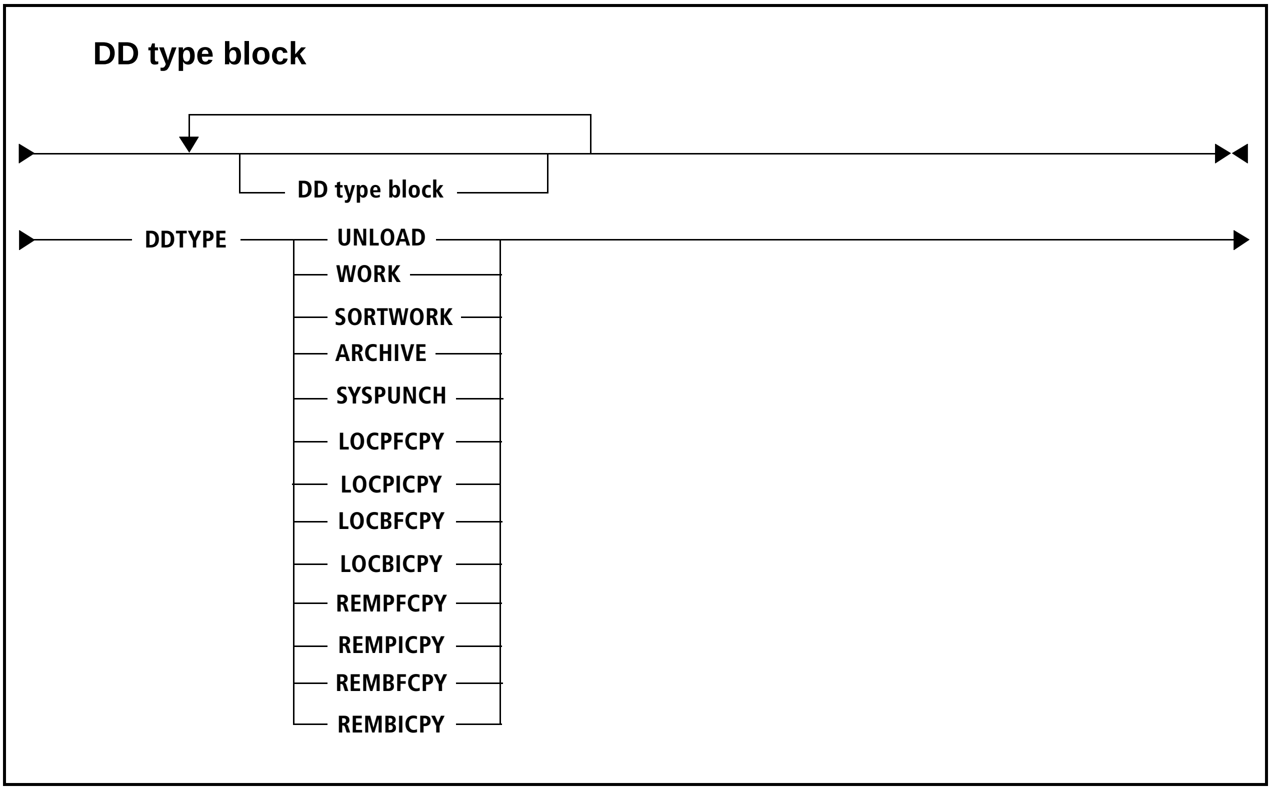


MAXTAPE

The MAXTAPE option allows you to specify a maximum number of tape devices that REORG PLUS will use when dynamically allocating data sets. For details regarding how REORG PLUS handles dynamic data set allocation when it reaches your MAXTAPE limit, see “Reaching the MAXTAPE limit” on page 92.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the MAXTAPE command option in your installation options module by using the MAXTAPE installation option (page 660). 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

This option tells REORG PLUS to dynamically allocate one or more types of data sets, or to override an active dynamic data set allocation value in the installation options module. The DD type block provides additional options that you can use to direct REORG PLUS in various aspects of the dynamic data set allocation process. You can repeat this DD type block for more than one data set type. Table 42 lists the keywords that you can specify for the data set types that you want to allocate dynamically.

Table 42 DDTYPE keywords (part 1 of 2)

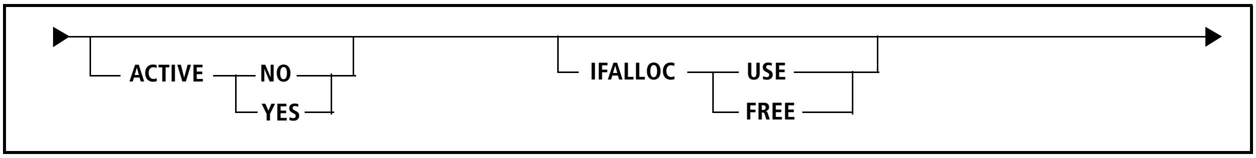
Keyword	Applies specifications for dynamic allocation to
UNLOAD	unload files (SYSREC)
WORK	index data sets (SYSUT1)
SORTWORK	sort work files (SORTWK)
ARCHIVE	archive file, or discard file for DSNUTILB jobs (SYSARC) **Note**: REORG PLUS ignores this DDTYPE option when invoking DSNUTILB to reorganize a LOB table space.

Table 42 DDTYPE keywords (part 2 of 2)

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 (BMCCPY)
LOCPICPY	local primary incremental copy data set (BMCICY)
LOCBFCPY	local backup full copy data set (BMCCPZ)
LOCBICPY	local backup incremental copy data set (BMCICZ)
REMPFCPY	remote primary full copy data set (BMCRCY)
REMPICPY	remote primary incremental copy data set (BMCIRY)
REMBFCPY	remote backup full copy data set (BMCRCZ)
REMBICPY	remote backup incremental copy data set (BMCIRZ)

Specifying the default

You can also specify the DD type in your installation options module by using the DDTYPE installation option (page 690). 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 (page 691). REORG PLUS was shipped with a default value of NO or YES for this option, depending on the DDTYPE value. The command option overrides the default that is in the installation options module.

When running in a worklist environment, REORG PLUS ignores the ACTIVE option in your installation options module. REORG PLUS dynamically allocates your data sets only if the invoking product (DASD MANAGER PLUS for DB2, CATALOG MANAGER for DB2, or CHANGE MANAGER for DB2) supplies the ACTIVE YES syntax.

YES

YES tells REORG PLUS to activate dynamic allocation for the specified data set type. ACTIVE YES must be in effect for REORG PLUS to use the remaining dynamic allocation options.

NOTE

When determining whether to dynamically allocate a data set, REORG PLUS takes into account the values of both the ACTIVE and IFALLOC options for that data set type.

NO

NO tells REORG PLUS to deactivate dynamic allocation for the specified data set type. Use this option if you set dynamic data set allocation active in your installation options but you do not want REORG PLUS to allocate the specified data sets dynamically for a particular processing.

IFALLOC

The IFALLOC option tells REORG PLUS to either free the data sets specified in your JCL or use them and, if necessary, allocate additional data sets.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the IFALLOC command option in your installation options module by using the IFALLOC installation option (page 700). 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:

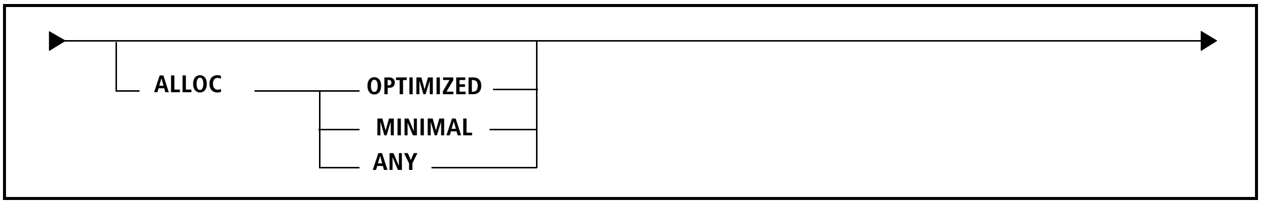
```
SYSU0001
```

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

This option applies only to sort work data sets.

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

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the ALLOC command option in your installation options module by using the ALLOC installation option (page 692). REORG PLUS was shipped with a default value of ANY for this option. The command option overrides the default that is in the installation options module.

OPTIMIZED

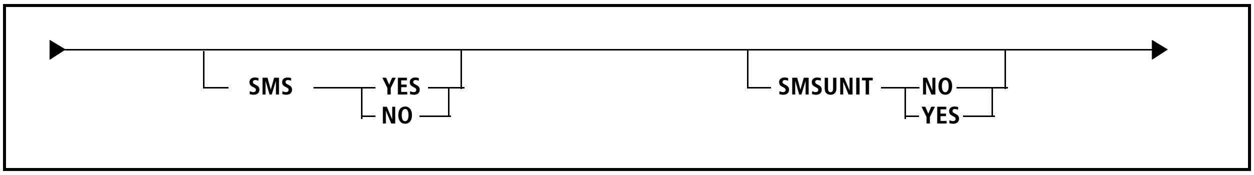
This option tells REORG PLUS to allocate as much DASD as necessary for the best performance results. If REORG PLUS cannot obtain an optimal allocation, processing terminates. For details about how REORG PLUS allocates sort work files for optimized and nonoptimized performance, see page 341.

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 option tells REORG PLUS whether to pass the SMS classes in the SMS allocation parameter list 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 during dynamic allocation.

  The following considerations apply to SMS YES:

  — When you specify SMS YES, REORG PLUS ignores the value that you specify for the MAXEXTSZ.

  — To pass the UNIT value to SMS during dynamic allocation, also specify the SMSUNIT option.

  — When invoking DSNUTILB, REORG PLUS includes the SMS classes in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

- NO tells REORG PLUS not to pass SMS classes during dynamic allocation.

  When you specify SMS NO, REORG PLUS always passes the UNIT value during dynamic allocation.

Specifying the default
You can specify the default for the SMS command option in your installation options module by using the SMS installation option (page 703). REORG PLUS was shipped with a default value of NO for this option. The command option overrides the default that is in the installation options module.

SMSUNIT

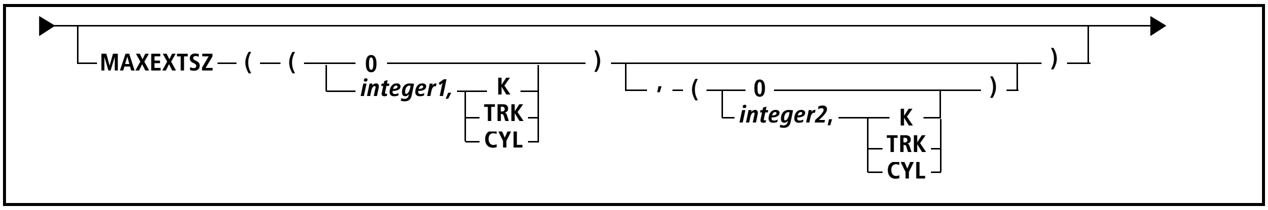
The SMSUNIT option tells REORG PLUS whether to pass the UNIT value in the SMS allocation parameter list to SMS during dynamic allocation. If you do not specify SMS YES, REORG PLUS ignores the SMSUNIT option. REORG PLUS does not modify any other parameters based on this option.

- NO tells REORG PLUS not to pass the value of the UNIT option.
- YES tells REORG PLUS to pass the value of the UNIT option.

  When invoking DSNUTILB, REORG PLUS includes the UNIT option values in the TEMPLATE control statements that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

Specifying the default

You can specify the default for the SMSUNIT command option in your installation options module by using the SMSUNIT installation option (page 703). REORG PLUS was shipped with a default value of NO for this option. The command option overrides the default that is in the installation options module.



MAXEXTSZ

For any extent that REORG PLUS allocates for a dynamically allocated data set, this option enables you to specify the maximum allowable value for the primary space allocation. Because the secondary quantity cannot exceed the primary quantity, MAXEXTSZ also controls the maximum secondary quantity.

Specify MAXEXTSZ by using one of the following values:

- 0 if you do not want to set a limit for space allocation
- an integer for the units specified with the UNIT installation or command option (see page 705 or page 280), with the unit of measure as follows:
  - — K for kilobytes (the default)
  - — TRK for tracks
  - — CYL for cylinders

You can specify values for MAXEXTSZ by using one of the following options:

- If you specify a single value (*integer1*), that value applies to both units (to ensure backward compatibility with previous REORG PLUS versions).

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

  ```
  MAXEXTSZ ((integer1,K),)
  ```
  or
  ```
  MAXEXTSZ (,(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 (see page 704 or page 286). This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

  Specifying two MAXEXTSZ values might be useful if characteristics of the unit that you specify for allocations below the threshold are different from those of the unit that you specify for allocations above the threshold.

Regardless of the amount of space that REORG PLUS determines that it needs, REORG PLUS will not allocate more than your specified MAXEXTSZ limit for either the primary or the secondary quantity.

- If the amount of required space that REORG PLUS calculates is greater than the MAXEXTSZ limit for the *primary* quantity, REORG PLUS uses the secondary extents to hold the remainder of the required primary space.

- If the amount of required space that REORG PLUS calculates cannot be accommodated because of MAXEXTSZ restrictions, the job might terminate with an out-of-space condition on the data set.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes the first parameter of this option as the MAXPRIME value in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For DSNUTILB, the unit of measure is always cylinders.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second parameter of this option as the MAXPRIME value in that template. For information about when REORG PLUS builds a secondary template, see “THRESHLD” on page 286.

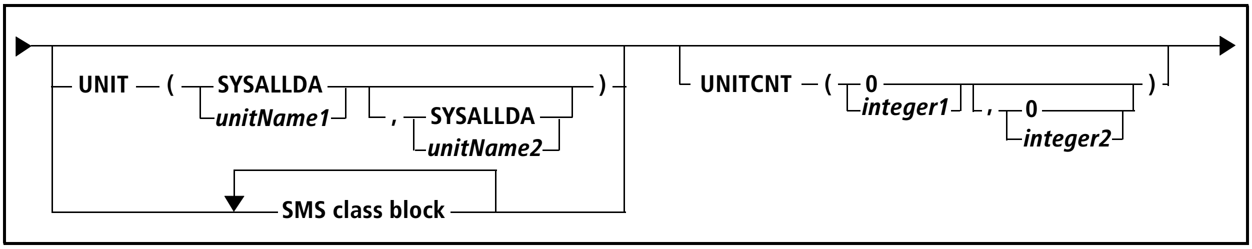
Additional considerations

Note the following additional information about the MAXEXTSZ option:

- The MAXEXTSZ option is not valid for DDTYPE SORTWORK.
- REORG PLUS ignores MAXEXTSZ when you specify SMS YES.
- REORG PLUS checks the value of MAXEXTSZ after applying SIZEPCT to the allocation amount.

Specifying the default

You can specify the default for the MAXEXTSZ command option in your installation options module by using the MAXEXTSZ installation option (page 700). 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.



UNIT

For non-SMS-managed data sets, the UNIT option allows you to specify the primary and secondary unit names that REORG PLUS uses for dynamic data set allocation. The unit names cannot exceed eight characters. Enclose the two unit names within parentheses and separate them with a comma.

For SMS-managed data sets when you specify SMS YES and SMSUNIT YES, this option supplies the unit names that REORG PLUS passes in the SMS allocation parameter list. DSNUTILB reorganization jobs function differently, as described in the DSNUTILB reorganization jobs section.

The THRESHLD installation or command option (page 286) controls which of the specified unit names REORG PLUS selects.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second UNIT parameter in that template. For information about when REORG PLUS builds a secondary template, see “THRESHLD” on page 286.

Specifying the default

You can specify the default for the UNIT command option in your installation options module by using the UNIT installation option (page 705). REORG PLUS was shipped with a default value of (SYSALLDA,SYSALLDA) for this option. The command option overrides the default that is in the installation options module.

UNITCNT

This option enables you to specify the number of devices to allocate when dynamically allocating data sets. Valid values are 0 through 59. A value of 0 tells REORG PLUS to use the system default.

If you specify a second value (*integer2*), REORG PLUS uses this number when the value for THRESHLD (page 286) is exceeded. This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second UNITCNT parameter in that template. For information about when REORG PLUS builds a secondary template, see "THRESHLD" on page 286.

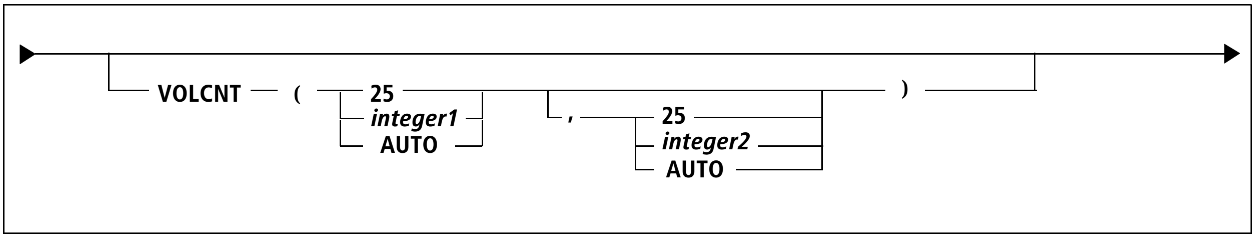
Additional considerations

The following additional considerations apply to the UNITCNT option:

- REORG PLUS ignores this option for sort work files.
- To avoid performance problems, specify only the number of devices that you need.

Specifying the default

You can specify the default for the UNITCNT command option in your installation options module by using the UNITCNT installation option (page 706). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.



VOLCNT

This option enables you to specify the maximum number of volumes to use during dynamic allocation. Valid values are

- 0, which tells REORG PLUS to not specify a volume count for dynamic allocation.
- Integer values 1 through 255 to specify the number of volumes.
- AUTO, which tells REORG PLUS to compute the volume count based on the amount of data, adjusted for the estimated space required (based on, for example, the value of the SIZEPCT installation or command option).

  REORG PLUS computes the volume count by dividing the size estimate by the value specified for the AVGVOLSP installation or command option (see page 693 or page 284).

If you specify a second value (*integer2*), REORG PLUS uses that value when the value for the THRESHLD installation or command option is exceeded (see page 704 or page 286). This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second VOLCNT parameter in that template. For information about when REORG PLUS builds a secondary template, see "THRESHLD" on page 286.

Additional considerations
Note the following additional information about the VOLCNT option:

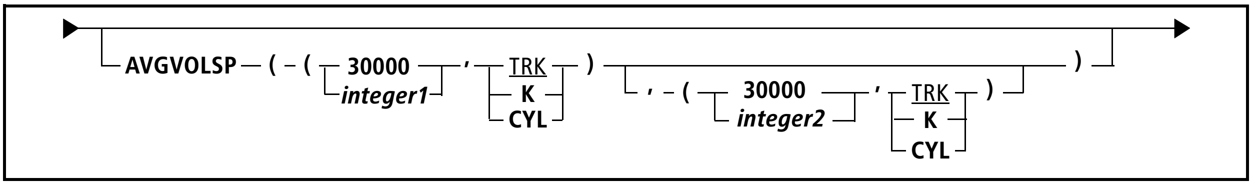
- This option is not available for DDTYPE SORTWORK. If you specify a value for this option for sort work files, REORG PLUS changes the value to 1.

- To avoid performance problems, specify only the number of volumes that you need.

- For an SMS-managed data set, REORG PLUS recommends that you specify a value of 0 if your ACS routines are set up to provide a volume count.

Specifying the default

You can specify the default for the VOLCNT command option in your installation options module by using the VOLCNT installation option (page 707). REORG PLUS was shipped with a default value of 25 for this option. The command option overrides the default that is in the installation options module.



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

REORG PLUS uses AVGVOLSP only when you specify AUTO for the corresponding first or second parameter of the VOLCNT installation or command option.

If you specify a second value (*integer2*), REORG PLUS uses that value when the value for the THRESHLD option is exceeded. This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

Restrictions

The following restrictions apply to the AVGVOLSP option:

- The AVGVOLSP option is not valid for DDTYPE SORTWORK.
- When invoking DSNUTILB, REORG PLUS ignores this option.

Additional considerations

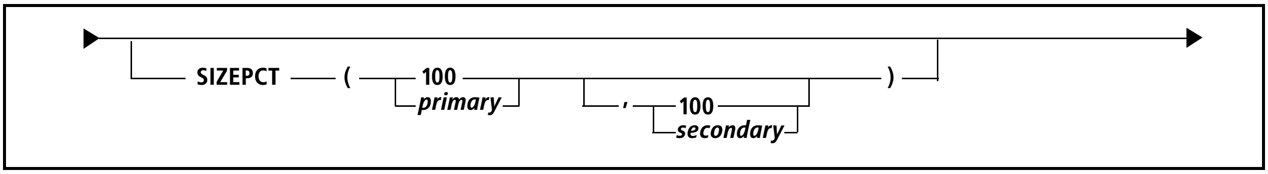
Note the following additional information about the AVGVOLSP option:

- You should not use AVGVOLSP to specify the maximum space on all devices or volumes unless the volumes to be used are empty.

- If you specify a value for AVGVOLSP that is too small, REORG PLUS computes a value for VOLCNT that is too large. However, if you specify a value for this option that is too large, REORG PLUS computes a value for VOLCNT that is too small.

Specifying the default

You can specify the default for the AVGVOLSP command option in your installation options module by using the AVGVOLSP installation option (page 693). REORG PLUS was shipped with a default value of ((30000,TRK),(30000,TRK)) for this option. The command option overrides the default that is in the installation options module.



SIZEPCT

The SIZEPCT option allows you to adjust, by percentages, the allocated data set sizes that REORG PLUS calculates. The values that you specify must be greater than 0.

- *primary* indicates the percentage of the primary data set size calculated by REORG PLUS that you actually want allocated.

- *secondary* indicates the percentage of the secondary data set size calculated by REORG PLUS that you actually want allocated.

  If you specify a secondary size that is greater than the primary size, REORG PLUS changes the secondary value to equal the primary value.

If you are performing a SHRLEVEL CHANGE reorganization that is creating incremental copy data sets, you can use the SIZEPCT option to reduce the size of those data sets. Because REORG PLUS uses the high-used RBA of the table space to determine the primary quantity, the incremental copy data sets can be larger than needed. If you use your estimate of the percentage of pages in the table space that will be updated during the reorganization as the primary SIZEPCT value, less space will be allocated for the data sets.

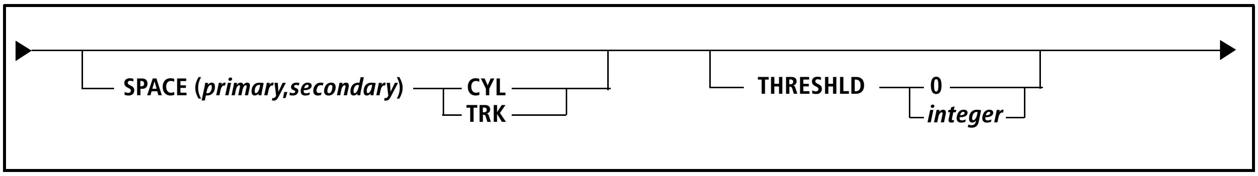
When you specify a value for both the SIZEPCT installation or command option and the ARCROWS command option, REORG PLUS ignores SIZEPCT for the discard data set and uses only ARCROWS to determine the discard data set size.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes the first parameter of this option as the PCTPRIME value in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. If you specify a value greater than 100, REORG PLUS converts it to 100.

Specifying the default

You can specify the default for the SIZEPCT command option in your installation options module by using the SIZEPCT installation option (page 702). REORG PLUS was shipped with a default value of (100,100) for this option for most DDTYPEs and (5,100) for DDTYPE LOCPICPY. The command option overrides the default that is in the installation options module.



SPACE

This option is only valid for a DSNUTILB reorganization.

The SPACE option enables you to override the DSNUTILB space calculation for the dynamically allocated data set. This option also overrides the value for the SIZEPCT option.

When you use this option, you must specify integer values for both the primary and secondary data sets. The numbers that you specify must be greater than 0. Include the unit of measure as follows:

- CYL for cylinders
- TRK for tracks

When not invoking DSNUTILB, REORG PLUS ignores this option.

THRESHLD

The THRESHLD option allows you to specify a threshold value, in kilobytes, above which REORG PLUS applies secondary values to allocated data sets. REORG PLUS tests this threshold for each data set to be allocated. If the size for a particular data set is greater than the threshold, REORG PLUS performs the following tasks:

- When SMS is NO, REORG PLUS uses the secondary values of the following options for the allocated data sets:

  — UNIT (unit name)
  — UNITCNT (unit count)
  — VOLCNT (volume count)
  — AVGVOLSP (average volume space)
  — MAXEXTSZ (extent size)

- 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)
  - if SMSUNIT is YES
    - UNIT (unit name)
    - UNITCNT (unit count)

If you specify 0, or if the threshold is not exceeded, REORG PLUS uses the primary values for these options.

TIP

If you use THRESHLD to send larger data sets to tape, consider setting the MAXTAPE option to limit the number of tapes that you use.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS handles this option differently, depending on the type of data set that you are allocating:

- For copy data sets, REORG PLUS translates this option to a LIMIT value in the TEMPLATE control statement that REORG PLUS builds for the IBM DB2 REORG utility. REORG PLUS also builds the secondary TEMPLATE control statement to which the DB2 REORG utility will switch when the LIMIT is exceeded.

  REORG PLUS assumes that the value that you supply for THRESHLD is in kilobytes. REORG PLUS translates that value to the appropriate value and unit of measure for the LIMIT keyword. Note the following additional information about this value:

  - REORG PLUS rounds down to the nearest whole value.
  - If you specify a value that would cause REORG PLUS to translate to a value less than 1 cylinder, REORG PLUS builds the template with a LIMIT value of 1 CYL.

- For all other data sets, REORG PLUS ignores this option.

Examples

The following examples illustrate how to use THRESHLD for different types of scenarios. In example 1, all allocated data sets for this DDTYPE go to SYSDA:

Figure 10 THRESHLD example 1

```
UNIT(SYSDA)
THRESHLD 0
```

In example 2, allocated data sets with a size greater than 720 MB for this DDTYPE go to unit LARGE. Smaller allocated data sets for this DDTYPE go to unit WORK:

Figure 11 THRESHLD example 2

```
UNIT(WORK,LARGE)
THRESHLD 720000
```

In example 3, data sets for this DDTYPE are SMS-managed. Allocated data sets with a size greater than 72 MB go to STORCLS2, MGMTCLS2, and DATACLS2. Smaller data sets go to the classes specified in the first parameter of each class type.

Figure 12 THRESHLD example 3

```
SMS YES
STORCLAS(STORCLS1,STORCLS2)
MGMTCLAS(MGMTCLS1,MGMTCLS2)
DATACLAS(DATACLS1,DATACLS2)
THRESHLD 72000
```

In example 4, REORG PLUS allocates data sets with the following properties for this DDTYPE:

- Data sets with a size less than or equal to 7200 MB go to unit SYSDA. The maximum volume count is calculated based on an average volume usage of 30000 tracks.

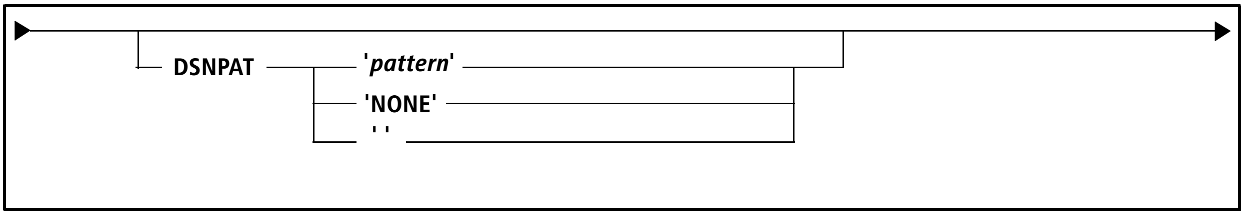
- Data sets with a size greater than 7200 MB go to unit CART with no more than 25 volumes.

Figure 13 THRESHLD example 4

```
VOLCNT(AUTO,25)
AVGVOLSP((30000,TRK),(900000,TRK))
UNIT(SYSDA,CART)
THRESHLD 7200000
```

Specifying the default

You can specify the default for the THRESHLD command option in your installation options module by using the THRESHLD installation option (page 704). 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.



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 43 on page 290) to a valid TEMPLATE variable. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

Specifying the default

You can specify the default for the DSNPAT command option in your installation options module by using the DSNPAT installation option (page 694). REORG PLUS was shipped with a different default value for each DDTYPE. The command option overrides the default that is in the installation options module.

'*pattern*'

Specify the pattern of variables and text, with surrounding single quotes, to use to create data set names.

The pattern that you specify in your DSNPAT option must allow REORG PLUS to generate unique data set names. If REORG PLUS encounters non-unique data set names, processing terminates. Note the following considerations:

- For full and incremental copy data sets, you might need to include additional variables such as &VCAT, &DATEJ, and &TIME4 to generate unique names across multiple reorganizations.

- If you plan to run concurrent REORG PLUS jobs or run multiple REORG PLUS steps in a worklist environment, BMC recommends that you include additional variables in the default pattern to ensure unique names. For example, the following value ensures that REORG PLUS generates unique names:

```
DSNPAT '&UID.&UTILPFX.&DDNAME..D&DATE..T&TIME'
```

Variables

You can use any of the symbolic variables listed in Table 43 to construct your pattern. In addition, you can use text or provide user-defined variables from a user exit. The REORG command option, DSNUEXIT, is described on page 220. User exits are further discussed in Appendix C, "REORG PLUS user exits."

Symbols for numeric variables (such as &DATE, &TIME, &PART, and &PART5) must be prefixed by a national character (alphabetic, #, @, or $). In the following example, the first statement causes an error, but the second is correct:

```
DSNPAT '&DB.&TSIX.&DATE'
DSNPAT '&DB.&TSIX..RP&DATE'
```

Although you can *prefix* a symbolic variable with an alphabetic character, you cannot *append* characters. For example, XX&DB is valid, but &DBXX is invalid. &DB.XX is valid.

User-defined variables must begin with an underscore character, as in _DEPT. User-defined variables are not valid for a DSNUTILB reorganization. For more information, see "DSNUEXIT user-defined variables" on page 736.

Table 43 Symbolic variables for the DSNPAT command option (part 1 of 3)

Symbolic variable	Definition	Length of result[a]	DSNUTILB reorganization
&DATE	current date (in the form *MMDDYY*)	6 bytes	variable passed
&DATEJ	current Julian date (in the form *YYYYDDD*)	7 bytes	variable translated to IBM's &JDATE
&DB	database containing the space for this data set allocation	8 bytes maximum	variable passed
&DDNAME	ddname for this data set allocation	8 bytes maximum	value passed
&GRPNM	DB2 data sharing group name In a non-data sharing environment, GRPNM contains the DB2 SSID.	4 bytes	value passed
&JDATE	current Julian date (in the form *YYDDD*)	5 bytes	job terminated
&JOBNAME	JOB name in the JCL	8 bytes maximum	variable passed
&PART	partition for this data set allocation You can use this variable for any data set. However, REORG PLUS substitutes the partition number for only copy and unload data sets. For all other data sets, REORG PLUS substitutes the value 000 or 0000.	3 bytes for table spaces with 999 partitions or less 4 bytes for table spaces with 1000 through 4096 partitions	variable passed

Table 43 Symbolic variables for the DSNPAT command option (part 2 of 3)

Symbolic variable	Definition	Length of result[a]	DSNUTILB reorganization

&PART5	partition for this data set allocation
	You can use this variable for any data set. However, REORG PLUS substitutes the partition number for only copy and unload data sets. For all other data sets, REORG PLUS substitutes the value 00000.
	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.&DB.&TSIX..P&PART5'`
	REORG PLUS generates the following 5-character partition number for partition 4096:
	`ABC.DSN1.DA.DBNAME.TSNAME.P04096`
		5 bytes for table spaces with 4096 partitions or less	variable passed

&RTYPE	REORG type (TS or IX)	2 bytes maximum	job terminated
&SSID	DB2 subsystem ID	4 bytes	variable passed
&STEPNAME	STEP name in the JCL REORG PLUS ignores PROC names.	8 bytes maximum	variable passed
&TIME	current time (in the form HHMMSS)	6 bytes	variable passed
&TIME4	current time (in the form HHMM)	4 bytes	variable passed
&TSIX	table space or index space specified in your REORG command	8 bytes maximum	variable translated to IBM’s &SN
&USERID or &UID	job user ID You must have a security package to use this variable.	8 bytes maximum	variable passed
&UTIL	BMC utility ID REORG PLUS truncates longer utility IDs to eight characters.	8 bytes maximum	variable translated to IBM’s &UTILID

Table 43 Symbolic variables for the DSNPAT command option (part 3 of 3)

Symbolic variable	Definition	Length of result[a]	DSNUTILB reorganization
&UTILPFX	BMC utility ID prefix	8 bytes maximum	value passed
&UTILSFX	BMC utility ID suffix	8 bytes maximum	value passed
&VCAT	VCATNAME specified in the DB2 catalog for the table space that you are reorganizing; or, if the table space is partitioned, the VCAT name from the first part that you are reorganizing	8 bytes	job terminated

[a] REORG PLUS removes any trailing blanks in the result.

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 44 Valid special delimiter characters for utility ID (part 1 of 2)

Character	Description
.	period
+	plus sign
|	bar
;	semicolon

Table 44 Valid special delimiter characters for utility ID (part 2 of 2)

Character	Description
-	dash
/	slash
¦	broken bar
_	underscore
:	colon
=	equal sign

NOTE

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

Name construction

You can specify any or all nodes of a data set name by using variables or text. For example, the following example generates data set names that contain the ID of the user, the table or index space that is involved in the reorganization, and the name of the reorganization job:

```
DSNPAT '&UID.&TSIX.&JOBNAME'
```

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

```
DSNPAT '&DB.&DDNAME..NEW'
```

In certain cases, a period is required in your pattern as a node delimiter or to indicate the end of a variable name. Table 45 on page 294 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 45 Variable concatenation examples

Task	Code	Result (where &UID=RDAB and _DEPT=DEV)
Concatenate the values of two variables, no node delimiter.	&UID_DEPT	RDABDEV
Make two nodes from the values of two variables.	&UID._DEPT	RDAB.DEV
Concatenate the value of a variable with text, no node delimiter.	&UID.NEW	RDABNEW
Concatenate text with the value of a variable, no node delimiter.	NEW&UID	NEWRDAB
Make two nodes from the value of a variable followed by text.	&UID..NEW	RDAB.NEW
Make two nodes from text followed by the value of a variable.	NEW.&UID	NEW.RDAB

REORG PLUS ignores trailing blanks and null value variables. However, REORG PLUS includes node-delimiting periods in the pattern, regardless of the variable's value, which might result in an invalid data set name. For example, given that &UID=RDAB and the value of _DEPT is null, the following example results in an invalid data set name of RDAB..NEW:

```
&UID._DEPT..NEW
```

GDG names

You can also specify a pattern that contains a generation data group (GDG) name for your dynamically allocated SYSARC, SYSPUNCH, and full and incremental copy data sets. Each DDTYPE must have a different GDG base.

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

If the base does not exist, REORG PLUS creates it for you, using everything in the pattern up to the open parenthesis as the base name. For more information about GDG names and options, see "Generating data set names" on page 89.

The following example shows a valid GDG name:

```
'&UTILPFX.&DDNAME..COPY(+1)'
```

If you are using a substitution variable as the last variable before the open parenthesis, you must include a period before the open parenthesis, as in the following example:

```
'&UTILPFX.&DDNAME.(+1)'
```

If you specify COPYLVL PART on the REORG command, each partition must have a different GDG base. To specify a pattern that includes a partition, the partition must not be in parentheses. The following example shows a valid name:

```
'&UTILPFX.&DDNAME..P&PART.(+1)'
```

You cannot specify a pattern that contains a partitioned data set (PDS) name. The following example shows an invalid name:

```
'&UTILPFX.&DDNAME..(P&PART)'
```

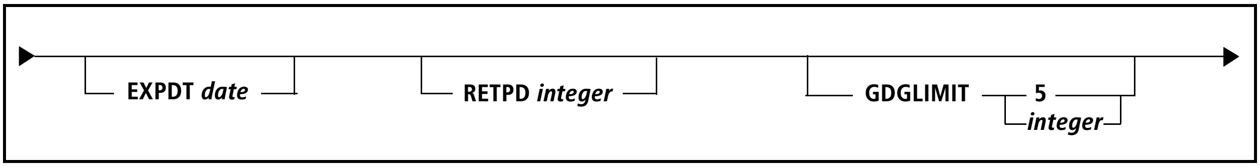
'NONE' or ' '

Specify 'NONE' or ' ' to indicate that you do not want to use any pattern to create data set names during dynamic data set allocation. This option is valid only with sort work data sets.

WARNING

If you specify NONE for SYSREC, SYSUT1, full copy data set, or incremental copy data set patterns, REORG PLUS will not allocate your data sets and will terminate your job.

In the DSNPAT *command* option, you must enclose the keyword NONE in single quotes. However, in the DSNPAT *installation* option, do *not* enclose the keyword NONE in single quotes.



EXPDT

Specify EXPDT to set an expiration date for any SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating. The value of *date* must be in the format *yyyyddd* or *yyyy/ddd*. The variable *yyyy* is the 4-digit year (1900 through 2155), and *ddd* is the 3-digit Julian day (000 through 366).

REORG PLUS ignores this option if you specify it for any of the other data sets that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

Specifying the default

You can specify the default for the EXPDT command option in your installation options module by using the EXPDT installation option (page 698). REORG PLUS was shipped with no default value for this option. The EXPDT command option overrides the RETPD command option and the EXPDT and RETPD (page 702) installation options.

RETPD

Specify RETPD to set a retention period (in days) for any SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating. The value of *integer* must be in the range 0 through 9999.

REORG PLUS ignores this option if you specify it for any *work* data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

Specifying the default

You can specify the default for the RETPD command option in your installation options module by using the RETPD installation option (page 702). REORG PLUS was shipped with no default value for this option. The RETPD command option overrides the default established at installation time for both the RETPD and EXPDT (page 698) installation options. However, if you specify the EXPDT command option, it takes precedence over the RETPD command option.

GDGLIMIT

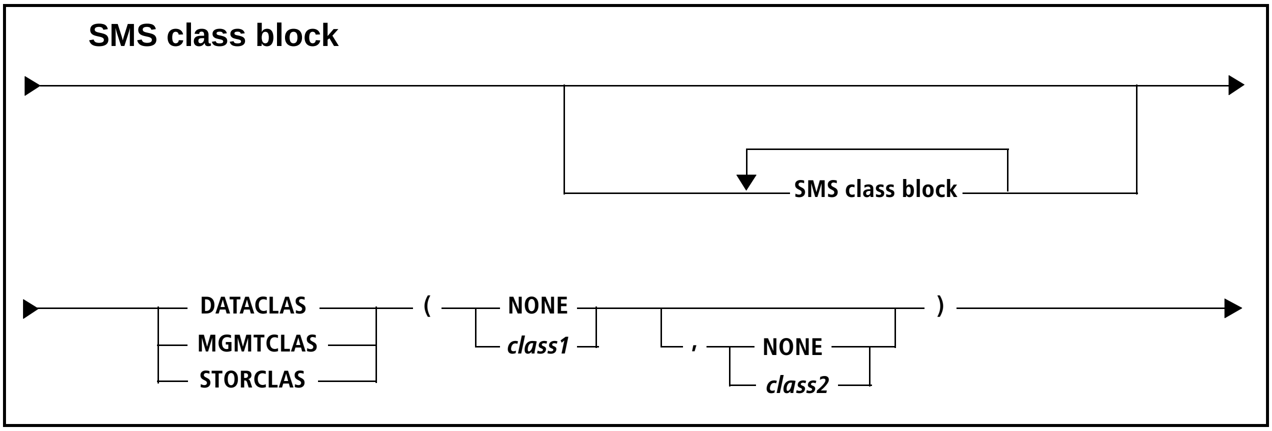
Specify GDGLIMIT to indicate the number of generations to keep for the SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating as a GDG data set. The value of *integer* must be in the range 1 through 255.

REORG PLUS *honors* this option only when creating the GDG base. REORG PLUS *ignores* this option if you specify it for any other data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

Specifying the default

You can specify the default for the GDGLIMIT command option in your installation options module by using the GDGLIMIT installation option (page 699). 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 that REORG PLUS is to use for SMS allocations. You can repeat this block to specify more than one type of SMS class.

The class names for each of the following class types must be enclosed in parentheses and separated by a comma. Use the keyword NONE instead of the class name if you do not want to specify a class for dynamic data set allocation.

The THRESHLD installation option or command option determines how REORG PLUS selects the classes that you specify here. This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS). For more information, see the THRESHLD command option on page 286.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes these classes in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the secondary classes in that template. For information about when REORG PLUS builds a secondary template, see “THRESHLD” on page 286.

Specifying the default

You can specify the default for the SMS class block in your installation options module by using the DATACLAS (page 694), MGMTCLAS (page 701), or STORCLAS (page 704) installation options. REORG PLUS was shipped with default values of (NONE,NONE) for these options. The command options override the default that is in the installation options module.

DATACLAS

DATACLAS indicates the primary and secondary SMS data classes that REORG PLUS uses for allocating the specified data set type. For *class1* and *class2*, specify valid SMS data class names that do not exceed eight characters each.

MGMTCLAS

MGMTCLAS indicates the primary and secondary SMS management classes that REORG PLUS uses for allocating the specified data set type. For *class1* and *class2*, specify valid SMS management class names that do not exceed eight characters each.

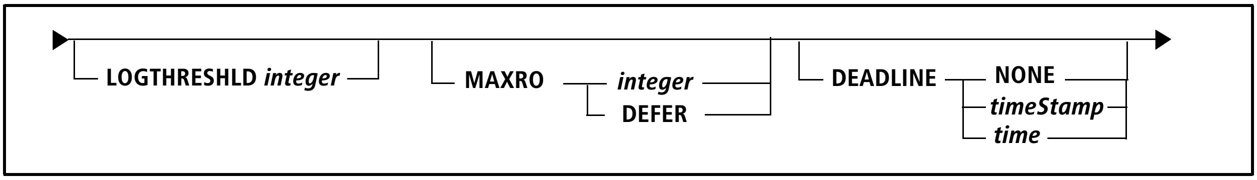
STORCLAS

STORCLAS indicates the primary and secondary SMS storage classes that REORG PLUS uses for allocating the specified data set type. For *class1* and *class2*, specify valid SMS storage class names that do not exceed eight characters each.

SHRLEVEL CHANGE options

You can perform an online reorganization by specifying SHRLEVEL CHANGE, along with its options, on the REORG command. You can specify SHRLEVEL CHANGE for two-phase or single-phase reorganization of a table space or index space. The benefit of SHRLEVEL CHANGE is that the data that you are reorganizing is available for read-write (RW) operations during most of the reorganization processing.

The following section provides the syntax of the options for SHRLEVEL CHANGE. For more information, see Chapter 7, “Online reorganization.”



LOGTHRESHLD

LOGTHRESHLD specifies the number of log records as a positive integer (0 or greater). When REORG PLUS determines that the number of log records remaining to be applied for all of the objects that you are reorganizing is less than or equal to this value, it begins the LOGFINAL phase. The LOGFINAL phase prevents updates to the objects and applies the remaining log records.

You can also use the XBM Utility Monitor function or the MVS operator console to dynamically change the value for this option while the reorganization is in progress. For more information, see “Using XBM to view and dynamically control the log apply process” on page 592.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 586.

Specifying the default
You can specify the default for the LOGTHRESHLD command option in your installation options module by using the LOGTHRSH installation option (page 657). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.

MAXRO

MAXRO tells REORG PLUS when to end the LOGAPPLY phase and begin the LOGFINAL phase.

integer

This integer specifies the maximum number of seconds that REORG PLUS spends applying log records in the LOGFINAL phase. Specify this value as a positive integer (0 or greater). When REORG PLUS estimates that the number of seconds required to apply the remaining log records is less than this value, it ends the LOGAPPLY phase and begins the LOGFINAL phase. The estimated time does not include the time required to produce the incremental copy data sets or the time needed for the UTILTERM phase. In the LOGFINAL phase, REORG PLUS prevents updates to the objects and applies the remaining log records.

DEFER

This value tells REORG PLUS to continue applying log records indefinitely, regardless of the value set in any other SHRLEVEL CHANGE option except DEADLINE, which is still checked. The LOGFINAL phase will not begin until you change this value. If DEADLINE is reached first, the job terminates after issuing message BMC50784E.

You can also use the XBM Utility Monitor function or the MVS operator console to dynamically change the value for this option while the reorganization in progress. For more information, see "Using XBM to view and dynamically control the log apply process" on page 592.

This option interacts with other options to control the log apply process. For information, see "Control of the log apply process" on page 586.

Specifying the default
You can specify the default for the MAXRO command option in your installation options module by using the MAXRO installation option (page 659). REORG PLUS was shipped with a default value of 300 for this option. The command option overrides the default that is in the installation options module.

DEADLINE

DEADLINE specifies the time by which the LOGFINAL phase should finish applying log records. If REORG PLUS determines that the LOGFINAL phase will not finish by the deadline, it terminates the reorganization. When calculating the estimate, REORG PLUS does not include the time required to produce the incremental copy data sets (if creating them) or the time needed for the UTILTERM phase. REORG PLUS does not check the DEADLINE value until the LOGAPPLY phase begins.

If a timestamp or time value is specified on DEADLINE and LOGFINAL, the calculated LOGFINAL timestamp must be less than the calculated DEADLINE timestamp.

You can use the XBM Utility Monitor function or the MVS operator console to dynamically change the DEADLINE values while the reorganization is in progress. For more information, see "Using XBM to view and dynamically control the log apply process" on page 592.

The DEADLINE option interacts with other options to control the log apply process. For information, see "Control of the log apply process" on page 586.

Specifying the default

You can specify the default for the DEADLINE command option in your installation options module by using the DEADLINE installation option (page 636). 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.

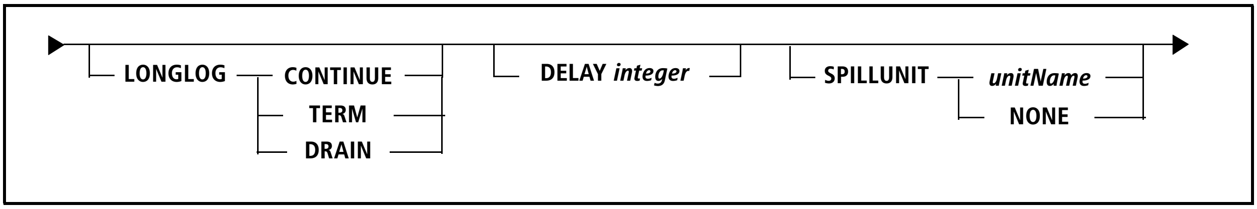
timestamp

Specify a timestamp value to indicate the time by which the LOGFINAL phase should finish applying log records.

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 then uses this calculated timestamp value to determine when the LOGFINAL phase should finish applying log records.



LONGLOG

LONGLOG specifies the action to take if REORG PLUS determines that a longlog condition exists. A longlog condition exists when REORG PLUS determines over a period of time that the DB2 subsystem is generating log records for the objects that you are reorganizing faster than REORG PLUS is applying the records. After detecting that a longlog condition exists, REORG PLUS sends an action write-to-operator (WTO) notification (see page 597) and continues to apply the log records until the delay interval specified on the DELAY option (see page 304) expires. If the longlog condition still exists, REORG PLUS takes the action that you specify, as follows:

CONTINUE

CONTINUE indicates that REORG PLUS is to continue processing in the LOGAPPLY phase.

TERM

TERM indicates that REORG PLUS is to terminate the reorganization.

DRAIN

DRAIN indicates that REORG PLUS is to begin the LOGFINAL phase, which prevents updates to the objects that you are reorganizing and applies the remaining log records.

You can also use the XBM Utility Monitor function or the MVS operator console to dynamically change the value for this option while the reorganization is in progress. For more information, see “Using XBM to view and dynamically control the log apply process” on page 592.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 586.

Specifying the default

You can specify the default for the LONGLOG command option in your installation options module by using the LONGLOG installation option (page 657). REORG PLUS was shipped with a default value of CONTINUE for this option. The command option overrides the default that is in the installation options module.

DELAY

DELAY specifies the number of seconds that are to elapse from the time REORG PLUS detects a longlog condition until it performs the action specified by the LONGLOG option (page 303). If the longlog condition no longer exists at the end of the time period specified by DELAY, the timer is reset. When REORG PLUS detects the next longlog condition, it restarts the timer, using the original DELAY value. The value must be a positive integer (0 or greater).

During the window provided by the DELAY option, you can use the XBM Utility Monitor function or the MVS operator console to change various SHRLEVEL CHANGE options, such as the LONGLOG action or MAXRO. This capability allows you to dynamically change when LOGFINAL processing begins or terminate the reorganization. For more information, see "Using XBM to view and dynamically control the log apply process" on page 592.

This option interacts with other options to control the log apply process. For information, see "Control of the log apply process" on page 586.

Specifying the default
You can specify the default for the DELAY command option in your installation options module by using the DELAY installation option (page 637). 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.

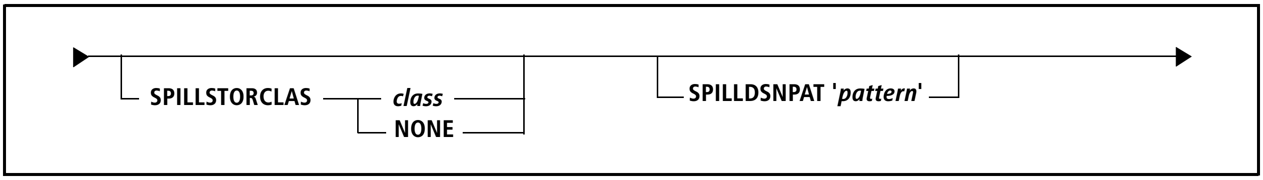
SPILLUNIT

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 601 and "Performance considerations" on page 612.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the SPILLUNIT command option in your installation options module by using the SPILUNIT installation option (page 676). 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 601 and "Performance considerations" on page 612.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the SPILLSTORCLAS command option in your installation options module by using the SPILCLS installation option (page 676). REORG PLUS was shipped with a default value of NONE for this option. The command option overrides the default that is in the installation options module.

SPILLDSNPAT

The SPILLDSNPAT option tells REORG PLUS to use a particular pattern of variables and text to create a prefix for the spill data set names. The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates the spill data sets as they are needed. The spill data sets are VSAM objects.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the SPILLDSNPAT command option in your installation options module by using the SPILDSNP installation option (page 674). REORG PLUS was shipped with a default value of &&UID for this option. The command option overrides the default that is in the installation options module.

'*pattern*'

Specify the pattern of variables and text, with surrounding single quotes, to use to create the data set names. The pattern must resolve to a prefix that is 22 bytes or less in length and not end in a period.

You can use any of the symbolic variables listed in Table 46 to construct your pattern. In addition, you can use text or provide user-defined variables from a user exit. The REORG command option, DSNUEXIT, is described on page 220, and user exits are further discussed in Appendix C, “REORG PLUS user exits.”

Table 46 Symbolic variables for the SPILLDSNPAT command option (part 1 of 2)

Symbolic variable	Definition	Length of result[a]
&DATE	current date (in the form *MMDDYY*)	6 bytes
&DATEJ	current Julian date (in the form *YYYYDDD*)	7 bytes
&DB	database containing the space for this data set allocation	8 bytes maximum
&GRPNM	DB2 data sharing group name **Note**: In a non-data sharing environment, GRPNM contains the DB2 SSID.	4 bytes
&JDATE	current Julian date (in the form *YYDDD*)	5 bytes
&JOBNAME	JOB name in the JCL	8 bytes maximum
&RTYPE	REORG type (TS or IX)	2 bytes maximum
&SSID	DB2 subsystem ID	4 bytes
&STEPNAME	STEP name in the JCL **Note:** REORG PLUS ignores PROC names.	8 bytes maximum
&TIME	current time (in the form *HHMMSS*)	6 bytes
&TIME4	current time (in the form *HHMM*)	4 bytes
&TSIX	table space or index space specified in the REORG command	8 bytes maximum
&USERID or &UID	job user ID **Note:** You must have a security package to use the job user ID variable.	8 bytes maximum
&UTIL	BMC utility ID **Note**: REORG PLUS truncates longer utility IDs to eight characters.	8 bytes maximum
&UTILPFX	BMC utility ID prefix	8 bytes maximum

Table 46 Symbolic variables for the SPILLDSNPAT command option (part 2 of 2)

Symbolic variable	Definition	Length of result[a]
&UTILSFX	BMC utility ID suffix	8 bytes maximum
&VCAT	VCATNAME specified in the DB2 catalog for the table space that you are reorganizing; or, if the table space is partitioned, the VCAT name from the first partition that you are reorganizing	8 bytes

[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" on page 736.

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:

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

If the utility ID is USER1//WORK1, the fields will have the following values:

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

The following special delimiting characters cause REORG PLUS to split the utility ID:

Table 47 Valid special delimiter characters for utility ID (part 1 of 2)

Character	Description
.	period
+	plus sign
|	bar
;	semicolon
-	dash

Table 47 Valid special delimiter characters for utility ID (part 2 of 2)

Character	Description
/	slash
¦	broken bar
_	underscore
:	colon
=	equal sign

NOTE

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. Table 45 on page 294 illustrates concatenation in a data set name pattern and instances in which a period is needed.

NOTE

Concatenation is not affected by a period contained in the value of a user variable.

Table 48 Variable concatenation examples (part 1 of 2)

Task	Code	Result (where &UID=RDAB and _DEPT=DEV)
Concatenate the values of two variables (no node delimiter).	&UID_DEPT	RDABDEV
Make two nodes from the values of two variables.	&UID._DEPT	RDAB.DEV
Concatenate the value of a variable with text (no node delimiter).	&UID.NEW	RDABNEW

Table 48 Variable concatenation examples (part 2 of 2)

Task	Code	Result (where &UID=RDAB and _DEPT=DEV)
Concatenate text with the value of a variable (no node delimiter).	NEW&UID	NEWRDAB
Make two nodes from the value of a variable followed by text.	&UID..NEW	RDAB.NEW
Make two nodes from text followed by the value of a variable.	NEW.&UID	NEW.RDAB

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:

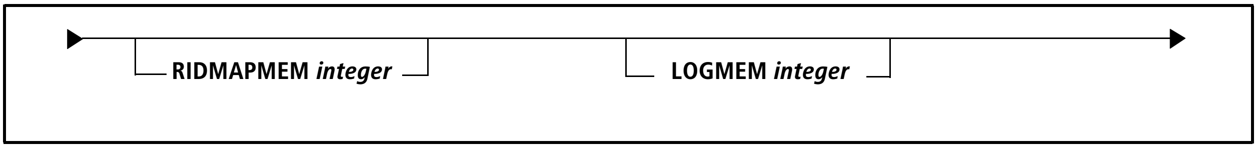
```
SPILLDSNPAT '&DB.&DATE'
```

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.

For information about the interaction of the spill data set options and performance impacts, see "Allocation of spill data sets" on page 601 and "Performance considerations" on page 612.



RIDMAPMEM

The RIDMAPMEM option specifies the amount of data space memory (in kilobytes) that REORG PLUS can use to store the RID translation maps. The number must be 0 or a positive integer. BMC strongly recommends that you use 0.

If you specify 0, REORG PLUS automatically calculates the RIDMAPMEM value for you. To determine the value, REORG PLUS multiplies the value of the installation option RIDMMAXD by the value of the installation option RIDMDSSZ. The result is the maximum amount of storage that REORG PLUS can allocate. However, REORG PLUS uses only as much memory as needed to hold the RID map, up to the calculated value.

If you specify a nonzero value for RIDMAPMEM, REORG PLUS first ensures that the value that you supplied is sufficient to satisfy the minimum storage requirements. If the value is not sufficient, REORG PLUS issues message BMC50887I and changes the value for RIDMAPMEM to the minimum required storage amount.

To calculate the RIDMAPMEM value, use the formula described in “Using formulas for the calculations” on page 613. 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 601
- calculating the data space size, see “Sizing memory for the RID translation map” on page 612
- RIDMMAXD and RIDMDSSZ installation options, see page 667

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default

You can specify the default for the RIDMAPMEM command option in your installation options module by using the RMAPMEM installation option (page 667). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.

LOGMEM

The LOGMEM option specifies the amount of data space memory (in kilobytes) needed to hold the internal structures that REORG PLUS uses to store the log records. The number must be 0 or a positive integer. BMC strongly recommends that you specify 0.

Additional considerations

The following considerations apply to this option:

- When invoking DSNUTILB, REORG PLUS ignores this option.

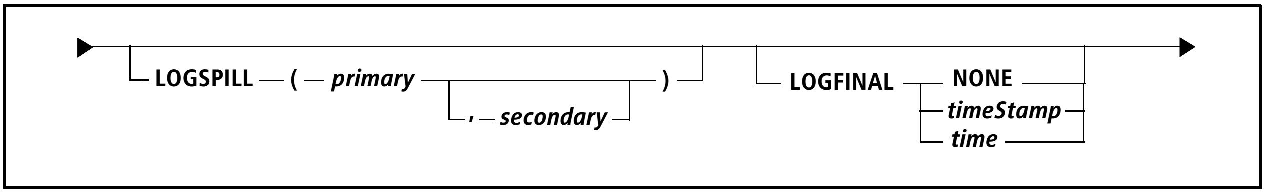
- If you specify 0, REORG PLUS automatically computes the amount of data space memory that it needs. To do so, REORG PLUS uses the formula described in "Using formulas for the calculations" on page 615.

- 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 601. For performance information, see "Sizing memory for log records" on page 614.

Specifying the default

You can specify the default for the LOGMEM command option in your installation options module by using the LOGMEM installation option (page 656). REORG PLUS was shipped with a default value of 0 for this option. The command option overrides the default that is in the installation options module.



LOGSPILL

The LOGSPILL option determines the primary and secondary space allocation (in kilobytes) for the spill data sets that hold the log records. Each number can be a nonzero positive integer.

For information about how the spill options work together to allocate the spill data sets, see "Allocation of spill data sets" on page 601.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the LOGSPILL command option in your installation options module by using the LOGSPIL installation option (page 657). REORG PLUS was shipped with a default value of (20000,10000) for this option. The command option overrides the default that is in the installation options module.

LOGFINAL

The LOGFINAL option allows you to specify the time that you want REORG PLUS to start the LOGFINAL phase. This option enables you to control when updates are prevented to the table space or index that you are reorganizing.

If a timestamp or time value is specified on DEADLINE and LOGFINAL, the calculated LOGFINAL timestamp must be less than the calculated DEADLINE timestamp.

REORG PLUS does not check the LOGFINAL value until the LOGAPPLY phase begins. At that point, if the current time is greater than the time specified on the LOGFINAL option, REORG PLUS immediately begins the LOGFINAL phase.

You can use the XBM Utility Monitor function or the MVS operator console to dynamically change the LOGFINAL values while the reorganization is in progress. For more information, see "Using XBM to view and dynamically control the log apply process" on page 592.

The LOGFINAL option interacts with other options to control the log apply process. For information, see "Control of the log apply process" on page 586.

Restriction
When invoking DSNUTILB, REORG PLUS ignores this option.

Specifying the default
You can specify the default for the LOGFINAL command option in your installation options module by using the LOGFINAL installation option (page 655). 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 options" on page 589.

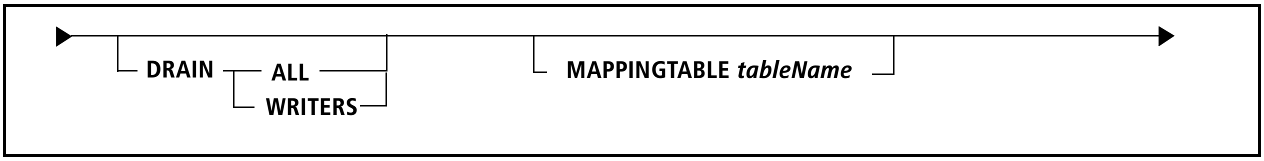
timestamp

When the current timestamp exceeds the value specified on this option, REORG PLUS starts the LOGFINAL phase.

time

This option enables you to specify a time to begin the LOGFINAL phase within the next 24 hours by specifying the time relative to the current time. Specify the time value in *hh:mm:ss* format, using a 24-hour clock.

REORG PLUS replaces the time portion of the current system timestamp with the time specified to create a new timestamp. If the new timestamp value is less than the current system timestamp, REORG PLUS adds one day to the timestamp. REORG PLUS then uses this calculated timestamp value to determine when to start the LOGFINAL phase.



DRAIN

The DRAIN option determines the type of drain that REORG PLUS issues when it enters the LOGFINAL phase, after reaching the MAXRO threshold.

- ALL tells REORG PLUS to drain all readers and writers.

  BMC recommends that you specify DRAIN ALL if either of the following conditions exists:

  — SQL queries might be running that do not commit often enough to allow a drain of the readers to successfully complete in a timely fashion.

  — SQL updaters might be running that require a minimum possible outage, and these updaters might time out if they are all held while awaiting the readers to be drained.

- WRITERS tells REORG PLUS to drain only writers.

Specifying the default
You can specify the default for the DRAIN command option in your installation options module by using the DRAINTYP installation option (page 639). REORG PLUS was shipped with a default value of ALL for this option. The command option overrides the default that is in the installation options module.

MAPPINGTABLE

This option specifies the mapping table that REORG PLUS provides to the IBM DB2 REORG utility when invoking DSNUTILB for SHRLEVEL CHANGE reorganizations. This option is meaningful only when REORG PLUS invokes DSNUTILB. Otherwise, REORG PLUS treats this option as a comment.

Chapter 4

Building and executing REORG PLUS jobs

This chapter presents the following topics:

Building the REORG PLUS job

Building a job for the REORG PLUS product involves creating a set of JCL that includes the following elements:

- a JOB statement (page 316)

- an EXEC statement with the appropriate utility parameters (page 316)

- STEPLIB or JOBLIB DD statements as needed (page 322)

- DD statements as needed for the appropriate number and size of data sets for the data structure that you are reorganizing (page 323)
- REORG PLUS control statements using the appropriate command syntax (page 159)

For examples of REORG PLUS JCL and SYSPRINT output, see Chapter 5, "Examples of REORG PLUS jobs."

JOB statement

Include a REORG PLUS JOB statement that conforms to your site's standards. You can include the REGION parameter on either your JOB statement or your EXEC statement. For recommendations, see "REGION parameter" on page 317.

EXEC statement

The REORG PLUS EXEC statement specifies the module to be run for the REORG PLUS utility. The REORG PLUS module name is ARUUMAIN. The EXEC statement also specifies the utility parameters, which are described in "Utility parameters on the EXEC statement" on page 317.

You can include the REGION parameter on either your EXEC statement or your JOB statement. For recommendations, see "REGION parameter" on page 317.

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

REGION parameter

Include the REGION parameter on either your JOB statement or your EXEC statement to specify the region size (the amount of virtual storage used by the utility). For the best performance, BMC recommends that you specify REGION=0M to allocate the optimal amount available virtual storage to the REORG PLUS job. If your data center does not permit you to specify REGION=0M, specify the amount that allows the most virtual storage both above and below the 16-megabyte line. Using a region size that is less than optimal risks the following potential issues:

- running less efficiently, which could result in additional CPU and elapsed time
- encountering memory failures or jobs that fail when new versions implement changes that require additional memory

NOTE

If you specify a value for REGION other than 0M, ensure that you have an appropriate value set for the MEMLIMIT parameter, either as your site's default SMF option or on your JOB statement or EXEC statement.

BMC makes the following recommendations for the MEMLIMIT option:

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

Utility parameters on the EXEC statement

The REORG PLUS EXEC statement includes the following utility parameters:

- DB2® subsystem ID or group attachment name
- utility ID
- restart parameter
- TSO user ID for notification of progress made on the reorganization
- message level
- installation options module parameter

The following illustration shows the format of the EXEC statement:

```
//stepname EXEC PGM=ARUUMAIN,
//   PARM='ssid,utilityID,restartParameter,userID,MSGLEVEL(n),optionsModule'
```

The REORG PLUS utility parameters are positional. If you do not specify a value for a parameter (thus allowing the default value), and additional parameters follow, you must substitute a comma for that parameter. The comma indicates that you omitted a parameter.

DB2 subsystem identifier (SSID)

This parameter specifies the four-character DB2 subsystem ID where the object resides.

If you do not specify the SSID, REORG PLUS uses the DB2 installation default from the DSNHDECP module. REORG PLUS depends on the application defaults module being named DSNHDECP. If you do not specify an SSID and REORG PLUS cannot find a module named DSNHDECP in your LINKLIST or STEPLIB, REORG PLUS terminates.

Note the following considerations when running REORG PLUS in a data sharing environment:

- REORG PLUS supports the DB2 group attachment name capability. When you supply a group attachment name as the SSID, REORG PLUS uses it to connect all plans. REORG PLUS then determines the actual DB2 SSID from within that group to use for the current reorganization.

- When you are restarting in a data sharing environment, REORG PLUS can use either the same member that was chosen in the original reorganization or any other member in the specified group.

- When both of the following conditions exist, specify a member SSID for your reorganization job instead of a group attachment name. Ensure that the DB2 version of the SSID that you specify corresponds to the version of the DB2 load library that you specify in your STEPLIB.

  — You are using table space compression.

  — The subsystems in your data sharing group are not all at the same DB2 version level.

Utility identifier (utility ID)

This parameter specifies the 1- to 16-character utility ID that gives a unique name to a utility job. If you omit this parameter, REORG PLUS uses the default, *userID.jobName*. Each BMC utility job should have a unique ID.

NOTE

Utility IDs that include special characters might cause REORG PLUS to generate invalid data set names when using dynamic allocation. For more information, see the discussion on page 292 about using the utility ID variable with the DSNPAT option.

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 361.

DSNUTILB reorganizations
After the UTILINIT phase, REORG PLUS passes the restart parameter to DSNUTILB. Restart processing is handled by the IBM DB2 REORG utility.

Blank or not specified

By not specifying a restart parameter, REORG PLUS initiates a new BMC utility job. The utility ID that you specify cannot currently exist in the BMCUTIL table.

RESTART

Specifying this value restarts a BMC utility from the last restart sync point. REORG PLUS takes restart sync points as each phase completes and as the processing of each DB2 object completes. The utility ID must exist in the BMCUTIL table. If the utility ID does not exist in the BMCUTIL table, REORG PLUS ends with return code 8.

RESTART(PHASE)

This value restarts a BMC utility at the beginning of the last incomplete phase. The utility ID must exist in the BMCUTIL table. If the utility ID does not exist in the BMCUTIL table, REORG PLUS ends with return code 8.

NEW

NEW initiates a new BMC utility job or replaces an existing utility ID. Specifying this value keeps you from having to end the utility ID separately and then start a utility specifying a blank restart parameter.

WARNING

Use care when you specify NEW. When you replace or terminate a utility ID for REORG PLUS, you might lose data from the UNLOAD or REORG phase, and you cannot restart the utility. For more information, see "Terminating or canceling a job" on page 366.

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.

For a DSNUTILB reorganization, REORG PLUS also terminates an existing DB2 utility ID.

The minimum JCL that is required when specifying TERM is the STEPLIB to the REORG PLUS load library and the SYSPRINT DD statement.

WARNING

Use care when you specify TERM. When you terminate a utility ID for REORG PLUS, you might lose existing data from the UNLOAD or REORG phase, and you cannot restart the utility. Similar consequences can occur in other BMC utilities. Consult the reference manual for the specific BMC utility. For more information for REORG PLUS, see “Terminating or canceling a job” on page 366.

MAINT

Specifying this value forces MSGLEVEL(1) and causes REORG PLUS to print the following information:

- an options module report that lists the values in the installation options module that you are using for this reorganization
- the values in the DSNHDECP module that REORG PLUS uses
- a summary report of all REORG PLUS fixes that you have applied

When you specify the MAINT parameter, the job ends without affecting any utility that is running.

For this parameter, you need only minimal JCL. Your JCL must include at least a SYSPRINT DD statement and STEPLIB to the REORG PLUS and DB2 load libraries.

User identifier (user ID)

This parameter specifies the TSO user ID that REORG PLUS notifies after it completes each phase and after it completes the command execution.

Message level (MSGLEVEL)

This parameter controls which messages REORG PLUS returns to the user in the SYSPRINT data set. MSGLEVEL(0) returns minimal messages. MSGLEVEL(1) returns additional messages to help you diagnose problems and fine-tune performance.

You can use the MSGLEVEL installation option to set the default value of this parameter. For details, see Appendix A, “REORG PLUS installation options.”

Installation options module

The options module parameter allows you to identify which installation options module to use. If you include this parameter, you must specify the full name of the options module. If you omit this parameter, REORG PLUS uses the default installation options module, ARU$OPTS.

For more information about installation options, see Appendix A, “REORG PLUS installation options.” In addition, the *Utility Products for DB2 Installation Guide* explains how to create multiple installation options modules.

STEPLIB DD statement

The REORG PLUS STEPLIB DD statement must specify the following libraries, unless they are included in your system’s LINKLIST or in a JOBLIB statement:

- load library that contains the files (including the options modules) for the following BMC products and components:

  — REORG PLUS
  — BMCSORT (AUP)
  — Common Statistics (ATS)
  — DB2 Utilities Common Code (D2U)
  — DB2 Solution Common Code (SCC)

- SAS/C transient library, if you specify BMCSTATS YES, BMCSTATS REPORT, or UPDATEDB2STATS YES
- BMCPSWD and ASUBMAIN libraries, if you are updating statistics in the DASD MANAGER PLUS statistics tables
- libraries that contain any DB2 user exits (EDITPROCs, VALIDPROCs, FIELDPROCs, and user-written exit routines)
- DB2 load library

NOTE

When building jobs for co-existence in environments that include DB2 Version 8, the DB2 Version 8 load library must be the first library in the STEPLIB concatenation. For example:

```
//STEPLIB DD
          DD DB2Version8LoadLibrary
          DD DB2Version10LoadLibrary
```

In all other cases, the load library for the most recent DB2 version should be the first library in the concatenation.

All load libraries in the STEPLIB or JOBLIB concatenation must be APF authorized.

REORG PLUS DD statements

REORG PLUS uses data sets specified by ddnames. This section provides specification guidelines, allocation information, and usage notes for each data set that REORG PLUS uses. Use Table 49 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 49 Data set type descriptions and quick command reference (part 1 of 2)

Data set type	Description reference	Default ddname[a]	DDTYPE option keyword (page 270 or page 690)	ddname or prefix installation option	ddname or prefix command option
ALTER statement output	page 338	DDLOUT	NA	NA	NA
archive	page 342	SYSARC	ARCHIVE	ARCHDDN (page 627)	ARCHDDN (page 172)
command input	page 347	SYSIN	NA	NA	NA
copy, full	page 326	■ BMCCPY (local primary) ■ BMCCPZ (local backup) ■ BMCRCY (remote primary) ■ BMCRCZ (remote backup)	■ LOCPFCPY ■ LOCBFCPY ■ REMPFCPY ■ REMBFCPY	■ COPYDDN (primary copies) (page 632) ■ RCVYDDN (remote copies) (page 665)	■ COPYDDN (primary copies) (page 248) ■ RECOVERYDDN (remote copies) (page 250)
copy, incremental	page 605	■ BMCICY (local primary) ■ BMCICZ (local backup) ■ BMCIRY (remote primary) ■ BMCIRZ (remote backup)	■ LOCPICPY ■ LOCBICPY ■ REMPICPY ■ REMBICPY	■ ICDDN (page 646) ■ RCVICDDN (page 664)	■ ICDDN (page 252) ■ RECOVERYICDDN (page 254)
discard (DSNUTILB only)	page 342	SYSARC[b]	ARCHIVE	ARCHDDN[c] (page 627)	ARCHDDN[c] (page 172)
REXX exits library indicator	page 345	SYSEXEC	NA	NA	NA
error	page 345	SYSERR	NA	NA	NA
IDCAMS input	page 345	SYSIDCIN	NA	NA	IDCDDN (page 199)

Table 49 Data set type descriptions and quick command reference (part 2 of 2)

Data set type	Description reference	Default ddname[a]	DDTYPE option keyword (page 270 or page 690)	ddname or prefix installation option	ddname or prefix command option
index work files	page 353	SYSUT1	WORK	WORKDDN[c] (page 686)	WORKDDN[c] (page 170)
input	page 333	DDLIN	NA	DDLDDN (page 636)	DDLDDN (page 172)
LOAD statements (DSNUTILB only)	page 348	SYSPUNCH[b]	SYSPUNCH	NA	NA
message output	page 353	SYSTERM	NA	NA	NA
message output	page 348	SYSPRINT	NA	NA	NA
other	page 357	NA	NA	NA	NA
REXX statement output	page 353	SYSTSPRT	NA	NA	NA
sort message output indicator	page 357	UTPRINT	NA	NA	NA
sort work files	page 338	SORTWK	SORTWORK	NA	NA
statistics report output	page 326	ASUSRPRT	NA	NA	NA
unload	page 349	SYSREC	UNLOAD	UNLDDN[c] (page 683)	UNLDDN[c] (page 169)

[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 (page 690 or page 270).

  NOTE

  Dynamic allocation of the work and copy data sets is required for a DSNUTILB reorganization. For more information, see "Reorganization jobs that invoke DSNUTILB" on page 70.

  — copy, full and incremental (page 326 and page 605)

  — SORTWK (page 338)

  NOTE

  You can have either BMCSORT or REORG PLUS dynamically allocate your sort work data sets. Unless you are running a DSNUTILB reorganization, BMC recommends that you have BMCSORT dynamically allocate these files.

  — SYSREC (page 349)

  — SYSUT1 (page 353)

  For more information about dynamic allocation, see "Dynamic data set allocation" on page 88 and "Dynamic allocation options" on page 269.

- Specify ANALYZE PAUSE or ANALYZE ONLY to have REORG PLUS estimate the space that is needed for your work data sets. For more information about these options, see "ANALYZE option for estimating data set allocation" on page 357.

- 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 the REORG PLUS 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 the REORG PLUS SYSPRINT.

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

- Suppress the output.

  To suppress the output, specify `//ASUSRPRT DD DUMMY` in your JCL.

Copy data sets

REORG PLUS creates one or more output copy data sets when you specify COPY YES. These output data sets contain a DSN1COPY-type copy or image copy of a nonpartitioned table space, all partitions of a partitioned table space, or selected partitions of a partitioned table space.

For a two-phase reorganization, REORG PLUS makes the copies during the RELOAD phase. For a single-phase reorganization, REORG PLUS makes the copies during the REORG phase. REORG PLUS creates all copies except inline image copies with system pages at the beginning of the data set (in the same way that the IBM DB2 COPY utility creates image copies when you specify SYSTEMPAGES YES).

The copy ddnames identify the output data sets that will contain the copy (after the reorganization) either of the table space, of each partition in the table space, or of the specified subset of partitions. The number of copies that REORG PLUS makes when you specify COPY YES depends on the following information:

- (*with dynamic allocation*) the DDTYPE and COPYLVL specifications in your installation and command options

- (*without dynamic allocation*) the existence of the copy ddnames in your JCL

REORG PLUS provides the following default names for each type of copy data set. This book uses the default name to refer to that type of copy data set. You can use a different name by changing it in your installation or command options.

Table 50 Default copy data set names

Default name	Function
BMCCPY*nn*	local primary
BMCCPZ*nn*	local backup
BMCRCY*nn*	remote primary
BMCRCZ*nn*	remote backup

When you are reorganizing either all partitions or a single subset of contiguous partitions of a partitioned table space, you can also specify how REORG PLUS assigns the dynamically allocated full and incremental copy data sets, as described in "COPYLVL" on page 246.

Requirements

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

- BMCCPY is required.
- If you specify BMCRCZ, BMCRCY 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" on page 605.

Overriding the default ddnames

To override the default copy ddnames or ddname prefixes that are specified in your installation options module, use the COPYDDN (page 248) and RECOVERYDDN (page 250) command options.

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 325 to allocate your copy data sets.

NOTE

Dynamic allocation of copy data sets is required for a DSNUTILB reorganization if you also specify COPY YES.

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 51 on page 329, 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 331.

- If you specify DELETEFILES YES and REORG PLUS terminates the job after a failure, REORG PLUS deletes any unregistered full and incremental copy data sets. For more information, see "DELETEFILES" on page 200.

- You can decide when to delete the dynamically allocated copy data sets by specifying an expiration date with the EXPDT option or by specifying a retention period with the RETPD option. For more information about these options, see "EXPDT" on page 296 and "RETPD" on page 296.

- For information about partition-level copies, see "Partition-level copies" on page 330 and "Partition-by-growth table spaces" on page 331.

For more information about dynamic allocation, see "Dynamic data set allocation" on page 88.

Allocating copy data sets in your JCL

If you do not use dynamic allocation, you must allocate the copy data sets in your JCL by specifying DD statements as described in Table 51. Do not specify separate data sets for nonpartitioned, multi-data-set table spaces.

Table 51 DD statements required when allocating copy data sets in your JCL

Table space being copied	Type of reorganization	Copy DD statements required
nonpartitioned	any	Specify one DD statement for each copy type that you want REORG PLUS to make. Do not use *nn* in the ddname.
partitioned, including partition-by-growth	full (entire table space)	■ Specify one DD statement for each copy type that you want REORG PLUS to make. Do not use *nn* in the ddname. *or* ■ For each partition, specify one DD statement for each copy type that you want REORG PLUS to make. Use *nn* for all ddnames, where *nn* matches the partition number.[a] If necessary for partition-by-growth table spaces, specify additional DD statements as discussed in "Partition-by-growth table spaces" on page 331
	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.[b] *or* ■ For each specified partition, specify one DD statement for each copy type that you want REORG PLUS to make. Use *nn* for all ddnames, where *nn* matches the partition number. (The *nn* is not required if you are performing a partial reorganization with only one partition.)[a] If necessary for partition-by-growth table spaces, specify additional DD statements as discussed in "Partition-by-growth table spaces" on page 331

[a] This option is not valid for a DSNUTILB reorganization.

[b] 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 Table 52 to determine the value to use for your data set allocation.

Table 52 Calculations for allocating copy data sets

Table space type	Number of copy data sets	Calculation
nonpartitioned	NA	Multiply the number of pages required for reorganizing by the page size of the table space.
partitioned	single	Multiply the total number of pages for all reorganized partitions by the page size of the table space.
	multiple	For each data set, multiply the number of pages for that partition by the page size of the table space.

Restrictions

Note the following restrictions when allocating copy data sets in your JCL:

- REORG PLUS does not support updating SMS-managed striped data sets for copy data sets if you specify the following options:

  — ICTYPE=AUTO (command or installation option)

  — ICTYPE=UPDATE (command or installation option)

  — INLINE NO (command option)

  — INLINECP=NO (installation option)

- Copy data sets cannot be temporary data sets. For information about how REORG PLUS defines a temporary data set, see "Work file validity and integrity checks" on page 93.

- BMC recommends that you do not specify DISP=MOD for an image copy data set because REORG PLUS does not reset the data set to empty but appends data to any data already present in the file.

Partition-level copies

Note the following additional information when allocating partition-level copies.

Naming partition-level copies

Use the *nn* as part of the ddname only if you are reorganizing a partitioned table space and you want a separate data set for each partition.

JCL rules limit ddnames to eight characters. If you are making partition-level copies, the ddname has two parts: the ddname prefix (minimum of one character) and the partition number *nn* (one to seven characters). The ddname that you specify in the installation options or the command options becomes the ddname prefix.

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 91.

- 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 51 on page 329, plus additional data sets equal to the smaller of the following values:

- MAXPARTITIONS minus the number of partitions in the table space
- MAXNEWPARTS

Although REORG PLUS requires that the additional data sets be allocated, it only uses them if partitions are added to the table space during the reorganization. REORG PLUS deletes unused data sets after the reorganization.

When dynamically allocating these data sets, REORG PLUS uses the DSSIZE value to allocate the data set for each partition. The following example illustrates the additional data set requirements using copy data sets.

Example

The following example illustrates the additional data set requirements using copy data sets. Assume the following scenario:

- You are performing a partial reorganization of a table space with four partitions.
- The value of MAXPARTITIONS is 6.
- Your SYSIN data set contains the following specifications:

```
REORG TABLESPACE databaseName.tableSpaceName
  PART 3:LAST
  MAXNEWPARTS 3
  COPYDDN BMCCPY
```

In this example, REORG PLUS requires that four copy data sets be defined in the JCL or through dynamic allocation:

- one for each of the two partitions that are participating in the reorganization (partitions 3 and LAST)

- two additional for partitions that might be added

  That is, MAXPARTITIONS (6) minus total partitions in the table space (4) which equals 2. This value is smaller than the MAXNEWPARTS value (3).

For more information about partition-by-growth table spaces, see page 118.

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. You can facilitate REORG PLUS multitasking as much as possible by providing a unique full copy data set for each partition that you are reorganizing and ensuring 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 241.

DDLIN data set

The DDLIN input data set contains the SQL ALTER INDEX statements or, for table-controlled partitioning, ALTER TABLE statements, with the new limit key values to use to rebalance partitions. REORG PLUS uses this optional data set only when performing a table space reorganization.

REORG PLUS does not explicitly execute the SQL statements in the DDLIN data set. Instead, it extracts the partition number and the associated limit key values from the statements. REORG PLUS reorganizes the data in the partitions based on the new key values and alters the limit keys after reloading the object. Using the DDLIN data set, you can rebalance up to 255 rebalance groups in a single execution of REORG PLUS.

The data set's attributes must be specified as fixed length (RECFM is F, FB, or FBS), and the record length must be 80 columns (LRECL=80). REORG PLUS uses only columns 1 through 72.

For SHRLEVEL CHANGE and SHRLEVEL REFERENCE, you can use the ALTRFAIL installation option (page 626) to tell REORG PLUS what to do when failures occur during processing of the ALTER statements in your DDLIN data set.

Guidelines for using the DDLIN data set

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

- REORG PLUS ignores any ALTER INDEX or ALTER TABLE statements that refer to an index that is not part of the reorganization.

- If you specify the same partition number on more than one ALTER INDEX or ALTER TABLE statement, REORG PLUS uses the limit key value from the last statement that it found.

- Character, hexadecimal, and graphic string constants are limited to a length of 256 bytes.

- REORG PLUS does not support columns that use a FIELDPROC in SQL ALTER INDEX or ALTER TABLE statements in the DDLIN data set.

Conditions that cause REORG PLUS to terminate

REORG PLUS terminates when the following conditions exist:

- You specify any of the following items in the DDLIN data set:

  — the same partition number more than once on the same ALTER INDEX or ALTER TABLE statement, as shown in the following example:

```
ALTER INDEX USER1.TBL1INX
              PART  1 VALUES (X'11'),
              PART  2 VALUES (X'22'),
              PART  2 VALUES (X'11'),
              PART  3 VALUES (X'33'),
              PART  4 VALUES (X'44')
             ;
```

  — any of the following constants as a limit key value:

    - floating-point
    - decimal floating-point
    - XML
    - LOB
    - row ID
    - graphic types
    - binary strings

  — a limit key constant that spans a line

  — an invalid limit key value

  — a limit key value in VARGRAPHIC format ('G' or 'N' in front of a double-byte string)

- You are using the DDLIN data set to rebalance a table space that has pending DDL changes, or an index on that table space has pending DDL changes.

- You are using the DDLIN data set to rebalance an XML table space or partition-by-growth table space.
- 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, whether or not that object is part of the reorganization

WARNING

Do not issue an ALTER statement outside of REORG PLUS to alter the limit keys of an object if that object currently is participating in a reorganization that is waiting to be restarted.

Sample ALTER INDEX statements and resulting messages

In the following example, the DDLIN data set contains three ALTER INDEX statements. The first and third statements alter nine partitions of index USER1.TBL1INX, which is an index on table 1. The second statement alters three partitions of index USER1.TBL2INX, which is an index on table 2. The user specified to use the data set on the reorganization of the table space that contains table 1.

The statements in the DDLIN data set are as follows:

```
ALTER INDEX USER1.TBL1INX
              PART  1 VALUES (X'11'),
              PART  2 VALUES (X'22'),
              PART  3 VALUES (X'33'),
              PART  4 VALUES (X'44')
             ;
ALTER INDEX USER1.TBL2INX
              PART 1 VALUES (X'0B'),
              PART 2 VALUES (X'0C'),
              PART 3 VALUES (X'0D')
             ;
 ALTER INDEX USER1.TBL1INX
              PART  5 VALUES (X'55'),
              PART  6 VALUES (X'56'),
              PART  7 VALUES (X'67'),
              PART  8 VALUES (X'78'),
              PART  9 VALUES (X'89')
             ;
```

REORG PLUS processed the ALTER statements successfully and sent the following messages to SYSPRINT:

```
BMC51291I A DDLIN DATASET HAS BEEN FOUND AND CONTAINS THE FOLLOWING STATEMENTS:

 BMC50102I    ALTER INDEX USER1.TBL1INX
 BMC50102I                  PART  1 VALUES (X'11'),
 BMC50102I                  PART  2 VALUES (X'22'),
 BMC50102I                  PART  3 VALUES (X'33'),
 BMC50102I                  PART  4 VALUES (X'44')
 BMC50102I    ;
 BMC50102I    ALTER INDEX USER1.TBL2INX
 BMC50102I                  PART 1 VALUES (X'0B'),
 BMC50102I                  PART 2 VALUES (X'0C'),
 BMC50102I                  PART 3 VALUES (X'0D')
 BMC50102I    ;
 BMC50102I    ALTER INDEX USER1.TBL1INX
 BMC50102I                  PART  5 VALUES (X'55'),
 BMC50102I                  PART  6 VALUES (X'56'),
 BMC50102I                  PART  7 VALUES (X'67'),
 BMC50102I                  PART  8 VALUES (X'78'),
 BMC50102I                  PART  9 VALUES (X'89')
 BMC50102I    ;
 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
BMC51232I ALTER STATEMENT 10 WILL BE PROCESSED
BMC51293I 10 ALTER STATEMENT(S) WILL BE PROCESSED FROM THE DDLIN FILE
```

DDLOUT data sets

The DDLOUT output data set contains all SQL ALTER statements that REORG PLUS executes (in order of execution) during partition rebalancing. REORG PLUS opens this optional data set only if it executes an ALTER.

REORG PLUS opens the data set with the following attributes, regardless of any attributes that you specify:

- LRECL=80
- RECFM=FB
- BLKSIZE=27920

REORG PLUS writes the DDL to columns 1 through 72, and writes EBCDIC blanks to columns 73 through 80. This format enables you to use this data set as DDLIN input in a subsequent REORG PLUS job.

You can produce only a report of the ALTER statements, without completing the reorganization, by completing the following steps:

1. Include a DDLOUT DD statement in your JCL.

2. Specify the following options on your REORG command:

   — SHRLEVEL CHANGE
   — REBALANCE
   — MAXRO DEFER

3. When REORG PLUS issues message BMC50829I, indicating that the LOGAPPLY phase is starting, cancel the job.

   REORG PLUS backs out the reorganization, but leaves the information in the DDLOUT data set intact.

SORTWK data sets

SORTWK data sets are the work files that BMCSORT uses. For a single-phase reorganization, BMCSORT uses the SORTWK data sets as work files in the REORG phase. For a two-phase reorganization, BMCSORT uses the SORTWK data sets as follows:

- in the UNLOAD phase to sort the unloaded table space rows (if you do not specify ORDER NO)

- in the RELOAD phase to sort the non-data-sorting indexes (and the clustering indexes if you specify ORDER NO)

REORG PLUS does not require SORTWK data sets when the following conditions exist. 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 the following conditions exist 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 341.

Dynamically allocating SORTWK data sets

Dynamic allocation takes place when any of the following actions occurs:

- You specifically request dynamic allocation through command or installation options.
- BMCSORT determines that it needs more sort work space than other allocation methods provide (allocated in your JCL, dynamically allocated by REORG PLUS, or both).

Several factors affect this dynamic allocation, as described in the following sections.

REORG PLUS allocation (ACTIVE YES)

If REORG PLUS dynamic allocation is active for sort work data sets, REORG PLUS calculates the optimal file size and number of files to allocate. REORG PLUS allocates at least 12 and up to 99 sort work data sets, depending on the total space required. REORG PLUS dynamic allocation is active when you specify ACTIVE YES for DDTYPE SORTWORK.

NOTE

Unless you are running a DSNUTILB reorganization, BMC recommends that you use BMCSORT to dynamically allocate your SORTWK data sets.

Note the following considerations:

- If you specify IFALLOC USE, REORG PLUS allocates sort work data sets, if needed, in addition to the ones specified in your JCL.

- When REORG PLUS dynamic allocation is active for sort work data sets, any values that you specify for SORTNUM, SORTDEVT, and the BMCSORT DYNALOC installation option are not used for allocation unless BMCSORT determines that it needs more sort work data sets than REORG PLUS allocated. In this case, BMCSORT uses these options, but *only for allocation of the additional data sets*.

- REORG PLUS deletes the sort work files when you specify DELETEFILES YES and the reorganization ends successfully. For more information, see "DELETEFILES" on page 200.

BMCSORT allocation

If BMCSORT allocation is active (see Table 53 on page 341), BMCSORT dynamically allocates your sort work files in the following cases:

- if REORG PLUS dynamic allocation is not enabled for sort work files (ACTIVE NO) and you have not specified SORTWK DD statements in your JCL

- in special cases when REORG PLUS dynamic allocation is enabled (see page 340)

- in special cases when you have allocated sort work data sets in your JCL (see "Allocating SORTWK data sets in your JCL" on page 341)

Table 53 on page 341 describes how BMCSORT allocates sort work files based on values for the SORTDEVT and SORTNUM options and the BMCSORT DYNALOC installation option.

Table 53 Factors that affect BMCSORT dynamic allocation of sort work data sets

SORTDEVT and SORTNUM	Third parameter of BMCSORT DYNALOC	Results
SORTDEVT specified *or* SORTNUM *n* specified (where *n* is greater than 0)	ON or OFF If the value is OFF, specifying a value greater than 0 for SORTNUM or specifying SORTDEVT changes this value to ON.	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.
SORTDEVT not specified *and* SORTNUM 0	ON	BMCSORT allocates the number of data sets that it determines are needed, up to 32 per sort task.
	OFF	BMCSORT does not allocate any sort work data sets and attempts to perform sort processing in memory.

Allocating SORTWK data sets in your JCL

Unless you specify ACTIVE YES IFALLOC FREE for REORG PLUS dynamic allocation of sort work data sets, REORG PLUS uses any SORTWK DD statements that you specify in your JCL.

To allocate SORTWK data sets in your JCL, determine the space needed for these data sets by specifying ANALYZE PAUSE or ANALYZE ONLY. The options tell REORG PLUS to provide an estimate of the space needed for your work data sets. For more information, see "ANALYZE option for estimating data set allocation" on page 357.

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 65535 tracks on a single volume.

- When you specify SORTWK DD statements in your JCL, BMCSORT dynamically allocates additional sort work data sets when both of the following conditions exist:

  — BMCSORT determines that it needs more sort work space than you have allocated.

  — The number of started sort tasks is such that at least one sort work data set can be allocated to each sort task.

SYSARC data set

REORG PLUS uses the SYSARC data set in one of the following ways:

- For native REORG PLUS jobs, SYSARC is an archive data set that contains discarded rows. REORG PLUS discards these rows when performing one of the following processes during a table space reorganization:

  — a SELECT or DELETE operation

  — rebalance of a table space that is LARGE (either by definition or default) or defined with DSSIZE, and where the last partition key has been altered

- 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 information about SYSARC data sets:

- The archive data set is required if 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 has *any* of the following attributes:

    - LARGE (either by definition or default)
    - defined with DSSIZE
    - 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 reload the discarded rows in the SYSARC data set by using the FORMAT BMC option of any currently supported version of LOADPLUS for DB2. The table that you are loading must have the same definition as the table from which the archive rows originated. For more information, see the *LOADPLUS for DB2 Reference Manual*.

Dynamically allocating SYSARC data sets

REORG PLUS allocates the SYSARC data set only when performing the following processes:

- processing a SELECT or DELETE statement
- rebalancing the last partition of a table space that is LARGE or was defined with DSSIZE specified, or uses table-controlled partitioning

REORG PLUS uses information from the ANALYZE phase to analyze the requirements for the current execution and calculates the optimal file size and number of files to allocate.

Note the following additional information about dynamically allocating SYSARC data sets:

- You can use the ARCROWS command option to control the size of a dynamically allocated archive data set. For more information, see "ARCROWS" on page 209.

- You can decide when to delete the dynamically allocated archive (SYSARC) data set by specifying an expiration date with the EXPDT option or by specifying a retention period with the RETPD option. For more information about these options, see "EXPDT" on page 296 and "RETPD" on page 296.

Allocating SYSARC data sets in your JCL

The following restrictions apply to the SYSARC data set:

- To accommodate restarting REORG PLUS any time after the UNLOAD phase for a two-phase reorganization, the SYSARC should not be a temporary data set. For information about how REORG PLUS treats temporary data sets, see "Work file validity and integrity checks" on page 93.

- BMC recommends that you do not use either of the following DD specifications for the SYSARC data set:

  — DD DUMMY

  — DSN=NULLFILE

- Do not use the same SYSARC data set for two different jobs. If you specify DISP=MOD for an existing SYSARC data set, REORG PLUS treats it as if you had specified DISP=OLD. Therefore, you will lose the data that already exists in the data set.

- If you restart a job using the same SYSARC data set (DISP=MOD) as you used for the initial run, and the data set had an expiration date or retention period, respond **U** to the following message to continue processing:

```
IEC507D REPLY 'U'-USE OR 'M'-UNLOAD
```

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes the dynamic allocation information for the optional SYSARC data set in the TEMPLATE control statement that REORG PLUS builds for the discard data set for the IBM DB2 REORG utility. This data set contains the rows that are discarded during the reorganization.

DB2 REORG requires the discard data set when certain conditions exist. If you do not enable dynamic allocation for this data set and DB2 REORG needs the data set, the reorganization terminates. For information about this data set, including requirements, see the DISCARD and SYSDISC information in the documentation for the DB2 REORG utility.

Allocation

To allocate this data set, you must enable dynamic allocation for the ARCHIVE DDTYPE. If you include a SYSARC DD statement in your JCL for a DSNUTILB reorganization, REORG PLUS ignores it.

Considerations

The following considerations apply to the SYSARC data set for DSNUTILB reorganization jobs:

- When invoking DSNUTILB to reorganize a LOB table space, REORG PLUS does not allocate this data set.

- The ARCROWS option is available for the SYSARC data set when you run a native REORG PLUS job, but is not available for a DSNUTILB job.

- You can decide when to delete this data set by specifying an expiration date with the EXPDT option, or by specifying a retention period with the RETPD option. For more information about these options, see “EXPDT” on page 296 and “RETPD” on page 296.

SYSERR data set

REORG PLUS recommends the SYSERR data set only if you specify SHRLEVEL CHANGE. This data set contains diagnostic messages that REORG PLUS might produce in error situations when running a SHRLEVEL CHANGE reorganization.

NOTE

BMC recommends that you do not depend on the format of this proprietary data set; this format is subject to change without notice.

SYSEXEC data set

REORG PLUS always requires the SYSEXEC data set for REXX exits. This DD statement specifies the library concatenation where REXX exits reside.

SYSIDCIN data set

SYSIDCIN is the input data set containing your IDCAMS command statements. REORG PLUS uses these statements to delete and redefine user-defined (VCAT-defined) data sets or the staging data sets for your VCAT-defined data sets. REORG PLUS issues the commands that the SYSIDCIN data set contains. You are responsible for command specifications and results.

REORG PLUS requires the SYSIDCIN data set if the value of the REDEFINE option is YES and one of the following conditions exists:

- You want to delete and redefine your existing VCAT-defined VSAM data sets as part of the reorganization for SHRLEVEL NONE (the default) or SHRLEVEL REFERENCE UNLOADONLY.

- You want to delete and define the staging data sets for your VCAT-defined VSAM data sets as part of the reorganization for SHRLEVEL REFERENCE or SHRLEVEL CHANGE.

If you omit the SYSIDCIN data set and specify REDEFINE YES, REORG PLUS performs the following actions, depending on the value of the PREFORMAT command or installation option:

- If PREFORMAT=NO, REORG PLUS changes the value of REDEFINE to NO and continues with the reorganization (without deleting and defining the data sets).

- If PREFORMAT=YES, REORG PLUS terminates the reorganization and issues message BMC50391E.

The UTILINIT phase reads, parses, and performs minimal verification checks on the commands in this data set. REORG PLUS issues the commands for each object during reload processing.

You still have the option of deleting and redefining VCAT-defined data sets when you specify UNLOAD PAUSE on the command. However, when you specify UNLOAD CONTINUE or UNLOAD RELOAD, the only way to delete and redefine VCAT-defined data sets as part of the reorganization is to also specify REDEFINE YES and provide the SYSIDCIN data set that contains the necessary IDCAMS command statements.

SYSIDCIN can be either a single or concatenated list of sequential data sets, partitioned data set members, or both. The data set must be specified as fixed length with blocked records (RECFM=FB), and the record length must be 80 bytes (LRECL=80). Only columns 1 through 72 are used.

For important restart information, see "Recovering from a failure" on page 368.

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 361.

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 346, 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 346, 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 98.

To define the staging data sets and avoid definition errors, provide a DELETE statement followed by a SET MAXCC=0 command (in case the delete process fails) before each DEFINE statement. REORG PLUS processes this set of statements (DELETE, SET MAXCC, DEFINE) for each object as the RELOAD phase begins for that object. REORG PLUS ignores any command that references one of the original VCAT-defined data sets and issues message BMC50604I.

SYSIN data set

REORG PLUS always requires the SYSIN data set. This input data set contains the REORG command. The UTILINIT phase reads, parses, and verifies the REORG command that is provided in this data set. The data set's attributes must be specified as fixed length (RECFM is F, FB, or FBS), and the record length must be 80 columns (LRECL=80). REORG PLUS uses only columns 1 through 72.

SYSPRINT data set

REORG PLUS always requires the SYSPRINT data set. This output data set contains REORG PLUS messages.

NOTE

BMC recommends that you do not depend on the content and format of this data set (for example, as input to user-defined processes). Message content and format are subject to change without notice.

REORG PLUS overrides any data control block attributes that you specify in your JCL with DCB=(RECFM=VBA,LRECL=137,BLKSIZE=141).

Note the following considerations if you direct SYSPRINT to a tape or disk data set:

- The DSN messages from DB2, such as those from a QUIESCE utility, are lost.
- BMC recommends that you do not specify 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 REORG PLUS displays and how to change the message level, see “Message level (MSGLEVEL)” on page 321.

SYSPUNCH data set

When invoking DSNUTILB, REORG PLUS includes the dynamic allocation information for the optional SYSPUNCH data set in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. This data set contains LOAD control statements for the rows that are discarded and written to the discard file during the reorganization.

DB2 REORG requires this data set when certain conditions exist. If you do not enable dynamic allocation for this data set and DB2 REORG determines that it needs the data set, the reorganization terminates. For more information about this data set, see the documentation for the DB2 REORG utility.

Allocation

To allocate this data set, you must enable dynamic allocation for the SYSPUNCH DDTYPE. If you include a SYSPUNCH DD statement in your JCL, REORG PLUS ignores it.

For non-DSNUTILB reorganizations, REORG PLUS does not allocate this data set.

Considerations

The following considerations apply to the SYSPUNCH data set for DSNUTILB reorganization jobs:

- When invoking DSNUTILB to reorganize a LOB table space, REORG PLUS does not allocate this data set.

- You can decide when to delete this data set by specifying an expiration date with the EXPDT option, or by specifying a retention period with the RETPD option. For more information about these options, see “EXPDT” on page 296 and “RETPD” on page 296.

SYSREC data sets

The SYSREC data set will contain the table space’s unloaded rows.

NOTE

BMC recommends that you do not depend on the format of this proprietary data set; this format is subject to change without notice.

After loading the table space, REORG PLUS deallocates the data set to free the device. Table 54 lists the circumstances under which REORG PLUS requires a SYSREC data set.

Table 54 SYSREC usage (part 1 of 2)

Type of reorganization	SYSREC requirement
single-phase table space reorganization when you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE	not used References to single-phase reorganizations in this section are for SHRLEVEL NONE only.

Table 54 SYSREC usage (part 2 of 2)

Type of reorganization	SYSREC requirement
two-phase table space reorganization	required REORG PLUS uses this data set to pass information between the UNLOAD and RELOAD phases.
single-phase table space reorganization when you specify SHRLEVEL NONE **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 361.	optional REORG PLUS writes information to this data set during the REORG phase for restart purposes only.

For information about performance improvements for this data set, see page 531.

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 325 to allocate your SYSREC data sets and determine the appropriate size for the data sets. If you do not use dynamic allocation, you must allocate the SYSREC data sets in your JCL by specifying DD statements.

NOTE

For a DSNUTILB reorganization, you must have REORG PLUS dynamically allocate your SYSREC data sets.

Number of SYSREC data sets to allocate

Use the information in Table 55 to determine how many SYSREC data sets to allocate. Partition-by-growth table spaces have special requirements as described after the table.

Table 55 Number of SYSREC data sets to allocate

Reorganization type	Object type	Number of SYSREC data sets
Any	Nonpartitioned table space	One
Two-phase	Partitioned table space	One or multiple
Single-phase	Partitioned table space	Multiple or none
Any	Partition-by-growth table space	See the section that follows

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 page 118.

Naming conventions for partition-level SYSREC data sets

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

The ddname that you specify in the JCL must have the partition number *nn* appended to this prefix. The *nn* must match the partition that you are reorganizing, and you must allow sufficient bytes for the partition number to be added to the prefix and still have a valid ddname (eight characters or less).

Dynamically allocating SYSREC data sets

REORG PLUS uses information from the ANALYZE phase to analyze the requirements for the current execution and calculates the optimal file size and number of files to allocate.

Note the following information when dynamically allocating SYSREC data sets:

- When you use dynamic allocation of SYSREC data sets during partition rebalancing, REORG PLUS cannot predict how much data movement will occur between rebalanced partitions. Therefore, REORG PLUS might inadequately size the SYSREC data set for a partition that has a large amount of data rebalanced into it. BMC recommends that you code the required SYSREC*nn* 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 SYSREC*nn* data sets. This consideration does not apply when you use the REBALANCE command option (page 167) to rebalance partitions.

- REORG PLUS deletes the SYSREC data sets when you specify DELETEFILES YES and the reorganization ends successfully. For more information, see “DELETEFILES” on page 200.

- If you plan to have REORG PLUS allocate more than 99 SYSREC data sets, use the UNLDDN option to specify a ddname prefix that results in eight characters or less after REORG PLUS appends the highest data set number. For more information, see “Specifying ddname prefixes” on page 91.

Allocating SYSREC data sets in your JCL

Note the following restrictions when allocating SYSREC data sets in your JCL:

- To enable REORG PLUS to restart any time after the UNLOAD phase for a two-phase reorganization, SYSREC cannot be a temporary data set. For information about how REORG PLUS defines temporary data sets, see “Work file validity and integrity checks” on page 93.

- 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 set

Use the SYSTERM data set if you are using REORG PLUS as part of the Database Performance solution. This output data set contains DASD MANAGER PLUS SQL and C runtime error messages.

SYSTSPRT data set

REORG PLUS always requires the SYSTSPRT data set for REXX exits. REXX routes all output from the REXX 'SAY' statements to this data set. It is usually coded as a SYSOUT DD statement.

SYSUT1 data sets

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

NOTE

BMC recommends that you do 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. Table 56 on page 354 lists the circumstances under which REORG PLUS requires the SYSUT1 data set.

Table 56 SYSUT1 usage

Reorganization type	SYSUT1 requirement
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 361.
two-phase index reorganization	required
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 325 to allocate your SYSUT1 data sets and determine the appropriate size for the data sets. BMC recommends that you dynamically allocate your SYSUT1 data sets. If you do not use dynamic allocation, you must specify DD statements in your JCL for the SYSUT1 data sets.

NOTE

For a DSNUTILB reorganization, you must have REORG PLUS dynamically allocate your SYSUT1 data sets.

Number of SYSUT1 data sets to allocate

Use the information in Table 57 to determine how many SYSUT1 data sets to allocate.

Table 57 Number of SYSUT1 data sets to allocate

Type of reorganization	Number of SYSUT1 data sets
Any table space reorganization	One data set for all indexes or one data set for each participating index
Two-phase index reorganization	One data set
Single-phase index reorganization	One data set or no data sets

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 532.

If you specify more than one SYSUT1 data set, specify a DD statement named SYSUT1*nn* 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 (specifying SHRLEVEL REFERENCE), 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.

Note the following information when dynamically allocating SYSUT1 data sets:

- REORG PLUS deletes the SYSUT1 data sets when you specify DELETEFILES YES and the reorganization ends successfully. For more information, see "DELETEFILES" on page 200.

- If you plan to have REORG PLUS allocate more than 99 SYSUT1 data sets, use the WORKDDN option to specify a ddname prefix that results in eight characters or less after REORG PLUS appends the highest data set number. For more information, see “Specifying ddname prefixes” on page 91.

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 if you are running REORG PLUS in WARN mode.

- To enable REORG PLUS to restart any time after the UNLOAD phase for a two-phase reorganization, the SYSUT1 cannot be a temporary data set. For information about how REORG PLUS defines temporary data sets, see “Work file validity and integrity checks” on page 93.

- 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 once 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 in the JCL, and so on. This assignment of indexes to data sets by key length can assist you in allocating the sizes of your data sets and allows you to place data sets on different devices when needed.

  Due to this assignment, you should allocate the space for the participating index with the largest key length in the first SYSUT1 DD statement in your JCL, space for the index with the second-largest key length in the second SYSUT1 DD statement, and so on.

UTPRINT data set

REORG PLUS always requires the UTPRINT data set if sorting is necessary. The presence of this data set tells REORG PLUS to report sort messages. However, the actual messages for each sort process appear in separate SYS*nnnnn* data sets, where *nnnnn* is a system-assigned sequential number. You cannot specify a sequential or partitioned data set for UTPRINT. UTPRINT supports only SYSOUT data sets.

WARNING

JES3 users should be aware of a limitation within JES3 that does not allow concurrent tasks to share SYSOUT data sets. (For a full description of this limitation, see IBM APAR OY23946.) This limitation means that you cannot use additional sort routine reporting DDs (other than UTPRINT) if they are defined as JES3 SYSOUT data sets and when REORG PLUS is multitasking its sort activity. If you attempt to use an unsupported DD, you risk S1FB abends when concurrent sort tasks are running. JES3 version 4.2.1 users should also refer to IBM APARs OW00111 and OY63725.

Other data sets

Because it uses BMCSORT, REORG PLUS ignores any traditional sort routine DD statements (such as $ORTPARM and DFSPARM) that you specify.

ANALYZE option for estimating data set allocation

If you specify ANALYZE PAUSE or ANALYZE ONLY, REORG PLUS gathers information about the objects that you are reorganizing. In addition to cardinality and average row size, the ANALYZE phase provides estimated data set sizes for the following data sets:

- unload (SYSREC)
- work (SYSUT1)
- sort (SORTWK)
- full image copy (BMCCPY, BMCCPZ, BMCRCY, and BMCRCZ)
- incremental image copy (BMCICY, BMCICZ, BMCIRY, and BMCIRZ)

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

REORG PLUS writes these statistics to the SYSPRINT data set. For information about the other statistical information messages that the ANALYZE phase issues, see “ANALYZE messages” on page 528.

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 188.

As an alternative to using ANALYZE PAUSE or ONLY to estimate sizes for data set allocation, you can have REORG PLUS dynamically allocate your data sets for you. To use dynamic allocation, specify ANALYZE (without PAUSE or ONLY). You must also have dynamic data set allocation active, either in your installation options or with the DDTYPE command option.

If you do not use the PAUSE or ONLY keywords with ANALYZE, REORG PLUS also gathers the information described in this section. However, instead of pausing or stopping, REORG PLUS continues processing. If dynamic allocation is enabled, REORG PLUS uses the ANALYZE phase information to dynamically allocate your data sets. In this case, the ANALYZE phase does not write the statistics to SYSPRINT.

For both optimum and minimum sort work file estimates, REORG PLUS uses the largest index to determine estimates. The optimum value is either the space required to sort the largest task (the task unloading the most data) or the space required to sort the largest index, whichever is greater. The space required for the task that unloads the most data can always be determined by sampling.

Table 58 details the space estimates provided for both table space and index reorganizations and provides estimates for both single and multiple SYSUT1 and SYSREC data sets. Refer to the specification guidelines for each data set provided in the preceding pages to determine whether to specify single or multiple SYSUT1 and SYSREC data sets.

Table 58 Estimates provided by the ANALYZE option (part 1 of 2)

Data sets for which estimates are provided	Reorganization type		Information provided
	Table space	Index	
single SYSREC data set	yes	not applicable	provides an estimate for all table space reorganizations except for a single-phase reorganization of a partitioned table space
multiple SYSREC data sets	yes	not applicable	provides an estimate for each partition that you are reorganizing in a partitioned table space only
single SYSUT1 data set	yes	yes	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.

Table 58 Estimates provided by the ANALYZE option (part 2 of 2)

Data sets for which estimates are provided	Reorganization type		Information provided
	Table space	Index	
multiple SYSUT1 data sets	yes	not applicable	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.
SORTWK data sets	yes	yes	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.
single full or incremental image copy data set (BMCCPY, BMCRCY, BMCICY, and so on)	yes	not applicable	provides an estimate for single copy data sets when you are performing ■ any full table space reorganization ■ a partial reorganization in which the partitions are specified as a contiguous subset and COPYSUBSET is YES
multiple full or incremental image copy data sets (BMCCPY*nn*, BMCRCY*nn*, BMCICY*nn*, and so on)	yes	not applicable	provides an estimate for each partition that you are reorganizing in a partitioned table space only

REORG PLUS provides the estimated information in table format. Messages BMC51260I and multiple BMC51263I messages provide the estimates. A separate BMC51263I message for each data set provides the following information:

- data set name
- number of kilobytes
- primary and secondary 3380 cylinder quantities
- primary and secondary 3390 cylinder quantities
- index name, where applicable

NOTE

For several reasons, including rows that contain VARCHAR columns and tables that contain EDITPROCs, ANALYZE might report a secondary quantity for SYSREC that is too large. The reason is that the primary quantity is based on the average row length, and the secondary quantity is based on the maximum row length from the DB2 catalog. In this instance, BMC recommends that you provide a secondary quantity of approximately 25 percent of the primary quantity.

Using ANALYZE with compressed table spaces

REORG PLUS uses the compressed row length to determine the size of the SYSREC and SORTWK data sets whenever possible.

SYSREC data set

REORG PLUS estimates the size of the SYSREC data set in the following manner:

- For compressed table spaces, REORG PLUS uses the average compressed row length.
- For noncompressed table spaces, REORG PLUS always uses the actual row length.
- For a multi-table table space, REORG PLUS averages the row length for the various tables.

Table 59 describes whether REORG PLUS uses compressed or expanded rows when KEEPDICTIONARY is in effect.

Table 59 Record size for SYSREC data sets

Type of reorganization	KEEPDICTIONARY value	Row length used
single phase	YES	compressed
	NO	expanded
two phase	YES	compressed
	NO	compressed

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 323. Ensure that all required libraries are available and APF-authorized as described in “STEPLIB DD statement” on page 322.

Restarting REORG PLUS

For a single-phase SHRLEVEL REFERENCE reorganization, you can restart REORG PLUS from a failure during any phase due to the nondestructive nature of this type of reorganization.

For a SHRLEVEL CHANGE (single- or two-phase) reorganization, you cannot restart any time before the beginning of the UTILTERM phase, unless you are restarting after ANALYZE PAUSE. Until UTILTERM begins, all user updates are made to the original data sets, which the reorganization has not yet changed. The data sets are exactly as they were before you ran the reorganization. After UTILTERM begins, restart works the same as it does for any other type of reorganization. For more information about restarting a SHRLEVEL CHANGE reorganization, see “Restart considerations for a SHRLEVEL CHANGE reorganization” on page 609.

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 SYSREC*nn* and SYSUT1*nn* data sets are present and are defined as cataloged data sets. Dynamically allocated SYSREC*nn*, SYSUT1*nn*, full copy, and incremental copy data sets are automatically reallocated by REORG PLUS on restart.

You cannot make structural changes to objects, such as altering a column from VARCHAR to CHAR, before restarting a failed reorganization. REORG PLUS relies on the object structure to remain unchanged between restart and the previous run. If you change the structure between runs, REORG PLUS might issue a user abend 3200 with reason code 5, or produce unpredictable results.

Specifying the RESTART and RESTART(PHASE) options

If a reorganization fails, correct the problem and restart the reorganization either with RESTART or RESTART(PHASE). REORG PLUS issues messages as it unloads, reloads, or rebuilds each DB2 object. The BMCSYNC table contains an entry for each DB2 object involved in the reorganization and its current status.

Specify RESTART without (PHASE) to restart REORG PLUS from the last restart sync point. REORG PLUS takes restart sync points as each phase completes and as the processing of each DB2 object completes. The utility ID must exist in the BMCUTIL table.

NOTE

The SYNC option (on the REORG command) controls the frequency with which rows are updated in the BMCSYNC table. SYNC does not control sync points that REORG PLUS uses when restarting. During the reload processing, restart sync points are established only after the last row is loaded in a table space, index, or partition.

Specify RESTART(PHASE) to restart REORG PLUS at the beginning of the last incomplete phase. The utility ID must exist in the BMCUTIL table.

Restart considerations and restrictions

This section describes considerations and restrictions that you should be familiar with before you restart a REORG PLUS job. For detailed instructions about dealing with a failure during the reorganization, see Table 61 on page 367.

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:

- 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 Table 60 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 60 Recopying of data sets for restart processing

		Are data sets recopied?	
Type of reorganization	**Phase for restart**	**RESTART**	**RESTART(PHASE)**
single-phase reorganization	REORG phase	no	yes
two-phase reorganization	UNLOAD phase	no	yes
	RELOAD phase	no	no

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 the CLONE option when restarting a reorganization.

DELETEFILES

To restart your job during DELETEFILES processing, specify RESTART without (PHASE).

DSNUTILB reorganization

When restarting a DSNUTILB reorganization job, REORG PLUS passes the RESTART or RESTART(PHASE) parameter that you specified to DSNUTILB for processing.

Dynamic allocation

On restart, REORG PLUS automatically reallocates dynamically allocated data sets.

If you change any dynamic data set allocation option on restart and the change results in different ddnames or a different number of DDs than the original option had, you can receive an error. If you need to change the number of SYSREC or SYSUT1 data sets, you must resubmit the job with a parameter of NEW.

You cannot change the value for the ACTIVE option on any restart. To change the value of other dynamic data set allocation options, specify RESTART(PHASE).

Inline image copies

If you restart a SHRLEVEL NONE or SHRLEVEL REFERENCE table space reorganization job, REORG PLUS changes the value of the INLINE command to NO if *all* of the following statements are true:

- The table space is partitioned.
- You have a single image copy data set.
- At least one (but not all) of the partitions was reloaded before the failure.

This change occurs regardless of the value you specified for the INLINE command or the INLINECP installation option.

Terminating or canceling a job

If you want to end the utility immediately and want the ability to restart your job, cancel the job by using the MVS or TSO CANCEL command.

If you want to end the job and do not intend to restart, terminate the reorganization job by performing one of the following actions. If the job is currently running, it terminates at the next sync point.

- Delete the corresponding rows from the BMCUTIL, BMCSYNC, and BMCDICT tables.

- Specify TERM on the restart parameter of the utility. If you terminate a job, you cannot restart it. For more information, see “TERM” on page 320.

Recovering the DB2 object after terminating or canceling a job

After you terminate or cancel a reorganization job, you might need to perform steps to recover your DB2 objects. Table 61 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 61 Recovering objects after terminating or canceling jobs (part 1 of 2)

Phase in which you terminated or canceled the job	SHRLEVEL specified	Recovery steps
terminated or canceled in UTILINIT or UNLOAD	any	All objects are usable and no additional steps are required. **Note**: 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.
DSNUTILB	SHRLEVEL NONE, SHRLEVEL REFERENCE, or SHRLEVEL CHANGE	For information about any recovery required, see the appropriate IBM DB2 documentation.
terminated in RELOAD or REORG	SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY	Recover any unusable objects.[a]
	SHRLEVEL REFERENCE or SHRLEVEL CHANGE	Restart the objects in their original statuses to make them usable.
canceled in RELOAD or REORG	SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY	■ If you want the ability to restart the job: A. Reset the STATUS column in the BMCUTIL table to S (for Stopped). B. Restart the job. ■ If you do not plan to restart the job, recover any unusable objects.[a]
	SHRLEVEL REFERENCE or SHRLEVEL CHANGE	Restart the objects in their original statuses to make them usable.

Table 61 Recovering objects after terminating or canceling jobs (part 2 of 2)

Phase in which you terminated or canceled the job	SHRLEVEL specified	Recovery steps
canceled or terminated in UTILTERM	SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY	All objects are usable and no additional steps are required.
	SHRLEVEL REFERENCE or SHRLEVEL CHANGE	BMC recommends that you restart your job rather than attempting to rename the staging data sets manually.

[a] Use the highest log RBA or LRSN listed in message BMC51281I or BMC51282I to recover TORBA. For its method of specifying the RBA or LRSN value when performing a RECOVER TORBA, see the documentation for your recovery software.

Recovering from a failure

Table 62 provides instructions for recovering after a reorganization job fails. The first column identifies the processing phases and, if applicable, the type of processing within the phase. The second column provides the steps to take when a reorganization job fails during that stage of processing.

Table 62 Recovering from a reorganization failure (part 1 of 2)

Phase in which job fails	Recovery steps
UTILINIT	Resubmit the job.
ANALYZE	Correct the problem that caused the failure and restart the job with RESTART.
UNLOAD	■ Correct the problem that caused the failure and restart the job with RESTART. ■ For SHRLEVEL CHANGE, resubmit the job.
RELOAD - redefinition of VSAM data sets, reload and index build processing	■ Correct the problem that caused the failure and restart the job with RESTART. ■ If the IDCAMS DEFINE failed, manually define the data set. ■ If you want to start your job at the beginning of the RELOAD phase, specify RESTART(PHASE). ■ For SHRLEVEL CHANGE, resubmit the job.
RELOAD - copy processing	■ Correct the problem that caused the failure and restart the job with RESTART. ■ For SHRLEVEL CHANGE, resubmit the job.

Table 62 Recovering from a reorganization failure (part 2 of 2)

Phase in which job fails	Recovery steps
REORG	■ For SHRLEVEL REFERENCE, correct the problem that caused the failure and restart the job with RESTART. ■ For SHRLEVEL NONE, take the appropriate action: — If you specified a SYSREC data set for a table space reorganization, or a SYSUT1 data set for an index reorganization, correct the problem that caused the failure and restart the job with RESTART. — If you did not specify a SYSREC data set for a table space reorganization, or a SYSUT1 data set for an index reorganization, and if REORG PLUS did not dynamically allocate these data sets, you will probably not be able to restart the job, but you can attempt to do so with RESTART. If you receive message BMC50009S indicating that REORG PLUS is unable to restart, recover any objects in recovery pending state, manually delete any dynamically allocated data sets, and resubmit the job. ■ For SHRLEVEL CHANGE, correct the problem and resubmit the job.
LOGAPPLY	For SHRLEVEL CHANGE only, resubmit the job.
LOGFINAL	For SHRLEVEL CHANGE only, resubmit the job.
UTILTERM	Correct the problem that caused the failure and restart the job with RESTART. Note the following additional information: ■ For SHRLEVEL REFERENCE and SHRLEVEL CHANGE, if the failure occurred while processing limit-key ALTER statements, the value of the ALTRFAIL installation option determines the action that REORG PLUS takes. This option applies to ALTER statements processed either from your DDLIN data set or as a result of the REBALANCE option. ■ For SHRLEVEL CHANGE, REORG PLUS backs out the reorganization and leaves the spaces in their original statuses when the following conditions exist: — The failure occurred while REORG PLUS was setting restrictive statuses. — Conditions exist that might cause a restarted reorganization to create invalid indexes. If you cannot restart the job, see "Not completing in the UTILTERM phase" on page 370 for SHRLEVEL REFERENCE and SHRLEVEL CHANGE.

Not completing 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:

- If you are using a DDLIN data set, see "Using a DDLIN data set."
- If you are *not* using a DDLIN data set and want to complete the job manually, see "Completing manually when no DDLIN data set is involved."
- If you want to go back to the starting point (no DDLIN data set or the DDLIN ALTER statement failed), see "Backing out the reorganization" on page 371.

Using a DDLIN data set

If you are using a DDLIN data set, look for message BMC51297I in SYSPRINT. If the message is displayed, the ALTER statement finished and you must complete the reorganization by issuing a DB2 START command with ACCESS(FORCE).

If message BMC51297I does not appear in SYSPRINT, the ALTER statement did not finish. You cannot complete the reorganization and must go back to the starting point. If you specified ALTRFAIL=TERM in your installation options, REORG PLUS performs the back out for you. For more information, see "Backing out the reorganization" on page 371.

Completing manually when no DDLIN data set is involved

If you are *not* using a DDLIN data set and want to complete the reorganization manually, first determine whether the FASTSWITCH or rename processing (whichever you are using) completed. The appearance of message BMC50895I in SYSPRINT indicates that the FASTSWITCH or rename processing completed.

FASTSWITCH or rename processing completed

If the FASTSWITCH or rename process completed, perform the following steps:

1. Issue a DB2 START command with ACCESS (UT) for the objects that you are reorganizing.
2. Remove the appropriate pending flags, such as RECP.
3. Run the MODIFY utility to remove SYSCOPY entries for prior utilities and image copies.

4 Take a full image copy.

5 Issue a DB2 START command to start your object in its original status.

FASTSWITCH or rename processing did not complete

If the FASTSWITCH or rename process did not complete, perform the following steps:

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

2 Complete the renaming operations manually:

- If you are using the rename process, rename the data sets that REORG PLUS did not finish renaming. For more information, see “Staging data sets and the rename process” on page 100.

- 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 102.

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

4 Remove the appropriate pending flags, such as RECP.

5 Run the MODIFY utility to remove SYSCOPY entries for prior utilities and image copies.

6 Take a full image copy.

7 Issue a DB2 START command to start your object in its original status.

Backing out the reorganization

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

- You are *not* using a DDLIN data set and want to back out the reorganization.
- You *are* using a DDLIN data set, the ALTER statement did not complete, and the value of the ALTRFAIL installation option is RCVRPEND.

NOTE

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

- if the failure occurs during processing of limit-key ALTER statements (either in your DDLIN data set or as a result of the REBALANCE option) and the value of the ALTRFAIL installation option is TERM
- for SHRLEVEL CHANGE, if the failure occurs while REORG PLUS is setting restrictive statuses, and a restarted reorganization might create invalid indexes

The steps to follow depend on whether you are using the rename process or the FASTSWITCH process.

Backing out when using the rename process

To go back to the starting point, perform the following steps:

1. Back out the renames that REORG PLUS has already completed.

   For more information, see “Staging data sets and the rename process” on page 100.

2. Issue a DB2 START command with ACCESS (FORCE) to start your object with its original status.

Backing out when using the FASTSWITCH process

Perform the following steps, depending on whether the FASTSWITCH process completed.

- If the FASTSWITCH process completed, as indicated by message BMC50895I in SYSPRINT, complete the following steps:

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

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

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

- If the FASTSWITCH process did not complete, issue a DB2 START command with ACCESS (FORCE) for the objects that you are reorganizing.

Chapter 5

Examples of REORG PLUS jobs

This chapter presents the following reorganization examples:

Overview

This chapter presents several examples of jobs that were run by using the REORG PLUS product. Each example includes the following information:

- a description of the reorganization job
- the REORG PLUS job stream
- the SYSPRINT from the job

Some examples include additional output that might be useful to understand a particular feature.

All of these examples have the following common properties:

- These examples use a single BMC product load library, assuming that this library contains the following files and libraries:

  — REORG PLUS load files

  — common component load files that REORG PLUS requires

  — SAS/C transient library (required for statistics processing)

- When data sets are allocated in the JCL, these examples use minimal data set space allocations.

- The value for the FILECHK installation option is set to WARN. This value prevents the jobs from failing when they encounter temporary data sets.

You can find copies of the JCL for these examples in member ARUEX*nn* in the *HLQ*.*LLQ*CNTL installation data set (where *nn* is the number of the example, *HLQ* is the high-level qualifier specified during installation, and *LLQ* is the low-level qualifier or prefix set during installation).

For syntax details, see Chapter 3, "Syntax of the REORG command." For information about JCL statement requirements, see Chapter 4, "Building and executing REORG PLUS jobs."

Use Table 63 to locate an example with a specific reorganization, process, or object type, or one that uses a specific option. Use the chapter table of contents to find the example.

Table 63 Cross-reference of examples by function (part 1 of 3)

Function	Examples
Reorganization type	
full (entire table space/all partitions)	1, 2, 7, 8, 9, 10-13
partial (selected partitions)	3, 5, 6
index only	4
two phase	8, 10-12
DSNUTILB reorganization	10
Object type	
partitioned table space	2, 3, 5, 6, 12-14
segmented table space	1, 7-10
partition-by-growth table space	11
VCAT-defined table space	1

Table 63 Cross-reference of examples by function (part 2 of 3)

Function	Examples
clustering index	1-3, 5-7, 11-13
nonpartitioned secondary index	2-6, 8, 12-14
data-partitioned secondary index	14
index created with DEFER YES	1
index containing keys with random ordering	10
ASUSRPRT data set	3, 14
ASUSRPRT DD DUMMY	2, 6
DDLIN data set	12
DDLOUT data set	14
SYSIDCIN data set	1
Command option	
ACTIVE NO	1, 2, 9, 12
ACTIVE YES	3, 9, 10
ANALYZE	12, 13
ANALYZE ONLY	7
ANALYZE SAMPLE	11, 14
ANALYZE SCAN	2
ARCHDDN	8
BMCSTATS YES	2, 3, 5, 6, 11, 14
COPY YES	1, 2, 3, 8-14
COPYDDN	2, 8, 10
COPYLVL	2, 3, 10, 11
DDTYPE	1-6, 9-14
DEADLINE	12
DELAY	12-14
DELETEFILES NO	1
DSNPAT	2-6, 9-14
FASTSWITCH	12-14
IFALLOC	10
ICTYPE	14
INLINE YES	10
KEEPDICTIONARY NO	11
LOGMEM	13
LOGSPILL	12-14
LOGTHRESHLD	14
LONGLOG	13
MAXEXTSZ	10

Table 63 Cross-reference of examples by function (part 3 of 3)

Function	Examples
MAXRO	12, 13
MAXNEWPARTS	11
ON FAILURE	9
ORDER NO	7, 8
ORDER YES	7, 9-14
PART	3, 5, 6
PREFORMAT YES	10, 11, 14
REBALANCE	14
RECOVERYDDN	8, 10
REDEFINE NO	3, 4, 8, 10
REDEFINE YES	1
REGISTER	8, 11
REORG INDEX	4
RIDMAPMEM	13
SELECT (selective unload)	8
SET	2
SHRLEVEL NONE	10, 11
SHRLEVEL REFERENCE	2, 3
SHRLEVEL CHANGE	12-14
SIZEPCT	12-14
SORTDATA	9, 10
SORTDEVT	10
SORTNUM	10
SPILLDSNPAT	12-14
SPILLSTORCLAS	12-14
SPILLUNIT	12-14
THRESHLD	10
UNLOAD PAUSE	5, 6
UNIT	10-14
UNITCNT	10
UNLOAD CONTINUE	7, 8, 10-12
UPDATE	2
UPDATEDB2STATS YES	1, 5, 6, 10, 11, 14
VOLCNT	10
WHERE clause	2, 8
XBMID	12-14

Example 1: VCAT-defined segmented table space with SYSIDCIN

In this example, REORG PLUS reorganizes a segmented VCAT-defined table space. The table space has a clustering index that was defined using the DB2® CREATE INDEX DEFER command. The REORG PLUS job populates the clustering index as part of the reorganization.

REORG PLUS issues message BMC50482I in the SYSPRINT when it creates the index. Message BMC51276I indicates how many keys were loaded into the index. Because this example uses a VCAT-defined table space, a SYSIDCIN DD statement and its contents are included in the JCL.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 64 describes the key command options and DD statements for this job.

Table 64 Example 1 key command options and DD statements (part 1 of 2)

Command options and DD statements used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
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 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.
UPDATEDB2STATS YES	requests that statistics be updated in the DB2 catalog The SYSPRINT shows the messages that the Common Statistics component displays for the statistics being updated in the DB2 catalog.
DELETEFILES NO	overrides the default and requests that REORG PLUS not delete the work files upon completion of the reorganization
DDTYPE ... ACTIVE NO	overrides the default and requests that REORG PLUS not enable dynamic allocation for the specified data sets
//SYSIDCIN	contains the IDCAMS command statements used to redefine VCAT-defined data sets

Table 64 Example 1 key command options and DD statements (part 2 of 2)

Command options and DD statements used in JCL	Description
//SYSREC	contains the rows that you are reorganizing

//BMCCPY
//BMCCPZ
//BMCRCY
//BMCRCZ
	the 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 when COPY YES is specified.

Figure 14 shows JCL for example 1.

Figure 14 JCL for example 1 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  SEGMENTED TABLESPACE, VCAT DEFINED, INDEX CREATED WITH DEFER    *
//*  YES. REORG TABLESPACE WITH REDEFINE YES AND SYSIDCIN IS USED,    *
//*  4 COPIES (TWO LOCALS & TWO REMOTES),                             *
//*  REGISTER ALL COPIES (DEFAULT), UPDATE DB2 CATALOG WITH STATS     *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG  EXEC PGM=ARUUMAIN,
//             PARM='DEDR,ARU01,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//*
//SYSIDCIN  DD  *
 DELETE -
   (DEDRCAT.DSNDBC.ARUDB001.TS00111.I0001.A001) -
   PURGE -
   CLUSTER -
   CATALOG(DEDRCAT)
 DELETE -
   (DEDRCAT.DSNDBC.ARUDB001.TS001.I0001.A001) -
   PURGE -
   CLUSTER -
   CATALOG(DEDRCAT)
 DEFINE CLUSTER( -
                NAME(DEDRCAT.DSNDBC.ARUDB001.TS001.I0001.A001) -
                LINEAR     -
                REUSE -
                VOLUMES(ARU382 ARU4C4 ARU112) -
```

Figure 14 JCL for example 1 (part 2 of 2)

```
                CYLINDERS (1 1) -
                SHAREOPTIONS(3 3) -
               ) -
        DATA   ( -
                NAME(DEDRCAT.DSNDBD.ARUDB001.TS001.I0001.A001) -
               ) -
        CATALOG(DEDRCAT)
 DEFINE CLUSTER( -
                NAME(DEDRCAT.DSNDBC.ARUDB001.TS00111.I0001.A001) -
                LINEAR -
                REUSE -
                VOLUMES(ARU382 ARU4C4 ARU112) -
                CYLINDERS (10 5) -
                SHAREOPTIONS(3 3) -
               ) -
        DATA   ( -
                NAME(DEDRCAT.DSNDBD.ARUDB001.TS00111.I0001.A001) -
               ) -
        CATALOG(DEDRCAT)
/*
//SYSREC    DD DSN=ARU.EXMPL01.SYSREC,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//*
//BMCCPY    DD DSN=ARU.EXMPL01.BMCCPY,
//             UNIT=WORK,SPACE=(CYL,(5,5)),
//             DISP=(,CATLG)
//BMCCPZ    DD DSN=ARU.EXMPL01.BMCCPZ,
//             UNIT=WORK,SPACE=(CYL,(5,5)),
//             DISP=(,CATLG)
//BMCRCY    DD DSN=ARU.EXMPL01.BMCRCY,
//             UNIT=WORK,SPACE=(CYL,(5,5)),
//             DISP=(,CATLG)
//BMCRCZ    DD DSN=ARU.EXMPL01.BMCRCZ,
//             UNIT=WORK,SPACE=(CYL,(5,5)),
//             DISP=(,CATLG)
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB001.TS001
COPY YES
UPDATEDB2STATS YES
REDEFINE YES
DELETEFILES NO
DDTYPE UNLOAD ACTIVE NO
DDTYPE WORK ACTIVE NO
DDTYPE LOCPFCPY ACTIVE NO
/*
```

Figure 15 shows the SYSPRINT output for example 1.

Figure 15 SYSPRINT for example 1 (part 1 of 6)

```
                               *****  B M C    R E O R G    P L U S    F O R    D B 2    V10R1.00  *****
                                             (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
               REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50001I UTILITY EXECUTION STARTING    1/27/2011   13:16:55 ...
BMC50002I UTILITY ID = 'ARU01'.  DB2 SUBSYSTEM ID = 'DEDR'.  OPTION MODULE = 'ARU$OPTS'.
BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
BMC50471I REGION=0M,BELOW 16M=8832K,ABOVE 16M=1407496K,IEFUSI=NO,CPUS=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

BMC50471I REORG PLUS FOR DB2--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.03.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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
```

Figure 15 SYSPRINT for example 1 (part 2 of 6)

```
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO
```

Figure 15 SYSPRINT for example 1 (part 3 of 6)

```
BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DEDR
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
```

Figure 15 SYSPRINT for example 1 (part 4 of 6)

```
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'
BMC50471I ...EXCEPTIONS  ='ASU101.EXCEPTIONS2'

BMC50102I REORG TABLESPACE ARUDB001.TS001
BMC50102I COPY YES
BMC50102I UPDATEDB2STATS YES
BMC50102I REDEFINE YES
BMC50102I DELETEFILES NO
BMC50102I DDTYPE UNLOAD ACTIVE NO
BMC50102I DDTYPE WORK ACTIVE NO
BMC50102I DDTYPE LOCPFCPY ACTIVE NO

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 1: SAMPLING STATISTICS: PART=0,TP=619,SP=94,SR=1350,AVGR=173,SD=65,SE=0,AVGF=1436,SD=65,SE=74,EP=19
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB001.TS001.I0001.A001'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 8860  AVG SORTWK ROW LENGTH = 173  AVG UNLOAD ROW LENGTH = 173
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0011 = 8860  AVG SORTWK ROW LENGTH = 173  AVG UNLOAD ROW LENGTH = 173
BMC51264I UNLOAD WILL READ 617 DATA PAGES FROM SPACE 'ARUDB001.TS001'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGAEXM1  &STEPNAME   BMCREORG  &DB         ARUDB001  &TSIX       TS001     &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       131655    &SSID       DEDR      &UTIL       ARU01
BMC50425I &UTILPFX    ARU01     &UTILSFX              &DATE8      01272011  &GRPNM      DEDR      &VCAT       DEDRCAT
BMC50425I &TIME4      1316      &DATEJ      2011027   &JDATE      11027

BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT
BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50474I BELOW 16M = 8220K, ABOVE 16M = 1398356K, CPUS = 3
BMC50479I TOTAL PAGES: 2325749, ALLOWED: 0; AVAILABLE PAGES: 347659, ALLOWED: 347659
BMC51302I MAX TASKS = 1, MAX PARTITIONS PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 10000, I/O WAITS = 2 ,DDNAME = SYS00015
BMC51272I UNLOAD STATISTICS:  10000 ROWS UNLOADED FROM SPACE 'ARUDB001.TS001', 0 ROWS DISCARDED, 0 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'079D49ED0000' IS THE HIGHEST LOGRBA FOR SPACE 'ARUDB001.TS001'
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB001.TS001.I0001.A001'
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB001.TS00111.I0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 10000, I/O WAITS = 5 ,DDNAME = SYS00022
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB001.TS001.I0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 10000, I/O WAITS = 6 ,DDNAME = SYS00018
BMC50481I 1: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:02
BMC50476I DDNAME = SYSREC, I/OS = 9, I/O WAITS = 4, RDB LOCK WAITS = 0
BMC51276I BUILD  STATISTICS:  10000 KEYS LOADED INTO INDEX 'ARU.TS00111'
BMC51275I RELOAD STATISTICS:  10000 ROWS LOADED INTO SPACE 'ARUDB001.TS001'
BMC50476I DDNAME = BMCCPY, I/OS = 7, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCCPZ, I/OS = 7, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCY, I/OS = 7, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCZ, I/OS = 7, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 620 PAGES COPIED TO DATASET = 'ARU.EXMPL01.BMCCPY'
BMC50376I 620 PAGES COPIED TO DATASET = 'ARU.EXMPL01.BMCCPZ'
BMC50376I 620 PAGES COPIED TO DATASET = 'ARU.EXMPL01.BMCRCY'
BMC50376I 620 PAGES COPIED TO DATASET = 'ARU.EXMPL01.BMCRCZ'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I REORG PHASE COMPLETE.  ELAPSED TIME = 00:00:04

BMCSTATS V10.1.0 REPORT FOR DEDR V910              TIME 2011-01-27-13.17.22.735814                                            1


  INDEX --------------- ARU.TS00111
    ASSOCIATIONS
      TABLESPACE ------ ARUDB001.TS001
      TABLE ----------- ARU.TS0011
    HISTORY
      SAVESTATS--------------------N  UPDATEDB2--------------------Y  UTILCODE----------------(NULL)  LOCATION------------------DEDR
      SAMPLING---------------------N
    ATTRIBUTES
      PIECESIZE--------------2097152  TYPE TS----------------(BLANK)  UNIQUERULE-------------------D  KEYLENGTH-------------------51
      SUBPAGES---------------------0  INDEXTYPE--------------------2  CLUSTERING-------------------Y  COLCOUNT---------------------2
      INDEXSPACE-------------TS00111  PGSIZE-----------------------4  EXTTYPE----------------(BLANK)  COMPRESS---------------------N
      COLNAME----------------EMPNAME
```

Figure 15 SYSPRINT for example 1 (part 5 of 6)

```
    STATISTICS
      FIRSTKEYCARD-----------------1  FULLKEYCARD--------------10000  NACTIVE--------------------174  TBCARDF------------------10000
      NUMNONLEAF-------------------4  NLEVELS----------------------3  CLUSTERRATIO---------------100  NLEAF----------------------167
      AVGKEYLEN-------------------51  REORGSPACE------------------15  REORGSPACE_KB--------------720  REORGNLEVELS-----------------3
      HIGH2K-----X'00D1C1C3D8E4C5D3'  HIGH2K (CHAR)---------.JACQUEL  LOW2K------X'00D1C1C3D8E4C5D3'  LOW2K (CHAR)----------.JACQUEL
      DATAREPEAT-----------------556
    ALLOCATION
      SPACE----------------------150  SPACE_KB------------------7200  NUMDATASETS------------------1

  INDEXPART ----------- ARU.TS00111
    ASSOCIATIONS
      TABLESPACE ------ ARUDB001.TS001
      TABLE ----------- ARU.TS0011
    HISTORY
      SAVESTATS--------------------N  UPDATEDB2--------------------Y  UTILCODE----------------(NULL)  LOCATION----------------(NULL)
      SAMPLING---------------------N
    ATTRIBUTES
      PQTY------------------------10  SQTY-------------------------5  PCTFREE---------------------10  FREEPAGE---------------------0
      ALLOCUNIT--------------------C  STORTYPE---------------------E  NLAVGKEYLEN-----------------51  VCATNAME---------------DEDRCAT
      STORNAME---------------DEDRCAT
    STATISTICS
      FIRSTKEYCARD-----------------1  FULLKEYCARD--------------10000  CARD---------------------10000  NACTIVE--------------------174
      CLUSTERRATIO---------------100  NUMNONLEAF-------------------4  NLEVELS----------------------3  FAROFFPOS--------------------0
      NEAROFFPOS------------------61  LEAFDIST---------------------1  NLEAF----------------------167  FREE------------------------78
      FULL-------------------------0  AVGKEYLEN-------------------51
      LEAFNEAR---------------------3  LEAFFAR----------------------0  PSEUDO_DEL_RIDS--------------0  REORGSPACE------------------15
      REORGSPACE_KB--------------720  REORGNLEVELS-----------------3  PCTUSED---------------------10  PQTYROWS----------------106080
      DATAREPEAT-----------------556
    ALLOCATION
      IPREFIX----------------------I  SPACE----------------------150  SPACE_KB------------------7200  NUMDATASETS------------------1
      EXTENTS----------------------1  VOLCOUNT---------------------1  DEVTYPE-------------------3390  VOLUME------------------ARU246

  COLUMN -------------- EMPNAME
    ASSOCIATIONS
      TABLESPACE ------ ARUDB001.TS001
      TABLE ----------- ARU.TS0011
    ATTRIBUTES
      COLNO------------------------3  COLTYPE-------------------CHAR  LENGTH----------------------40  NULLS------------------------Y
    STATISTICS
      COLCARD----------------------1  COLNULLS---------------------0
      COLAVG----------------------41  COLMIN----------------------41  COLMAX----------------------41
      HIGH2K-----X'00D1C1C3D8E4C5D3'  HIGH2K (CHAR)---------.JACQUEL  LOW2K------X'00D1C1C3D8E4C5D3'  LOW2K (CHAR)----------.JACQUEL

      MOST FREQUENT VALUES
BMCSTATS V10.1.0 REPORT FOR DEDR V910              TIME 2011-01-27-13.17.22.735814                                            2

       FREQUENCY-------------1.000000     COLVAL-X'00D1C1C3D8E4C5D3E8D540D2C1E8C540C8C1E6D2C9D5E2C6E2E2E2E2E2E2E2E2E2E2E2E2E2E2E2'
                                          (CHAR)-----------------------------------------.JACQUELYN KAYE HAWKINSFSSSSSSSSSSSSSSSSS
BMCSTATS V10.1.0 REPORT FOR DEDR V910              TIME 2011-01-27-13.17.22.735814                                            1


TABLESPACE ------------ ARUDB001.TS001
    HISTORY
      SAVESTATS--------------------N  UPDATEDB2--------------------Y  UTILCODE----------------(NULL)  LOCATION------------------DEDR
      SAMPLING---------------------N
    ATTRIBUTES
      PARTITIONS-------------------0  NTABLES----------------------1  PGSIZE-----------------------4  SEGSIZE---------------------64
      DSSIZE-----------------------0
      MAXROWS--------------------255  TYPE-------------------(BLANK)  ENCODING---------------------E  MAXPARTITIONS----------------0
      LOB--------------------------N  XML--------------------------N  MEMBER_CLUSTER---------(BLANK)
    STATISTICS
      NACTIVE--------------------619  REORGSPACE------------------60  REORGSPACE_KB-------------2880  ROWMAXFOUND----------------179
      ROWMINFOUND----------------179  ROWAVG---------------------173
    ALLOCATION
      SPACE-----------------------60  SPACE_KB------------------2880  NUMDATASETS------------------1

  TABLEPART ----------- ARUDB001.TS001
    HISTORY
      SAVESTATS--------------------N  UPDATEDB2--------------------Y  LOCATION----------------(NULL)  UTILCODE----------------(NULL)
      SAMPLING---------------------N
    ATTRIBUTES
      PQTY-------------------------1  SQTY-------------------------1  PCTFREE---------------------20  FREEPAGE---------------------9
      ALLOCUNIT--------------------C  STORTYPE---------------------E  COMPRESS---------------(BLANK)  TRACKMOD---------------(BLANK)
      FORMAT---------------------RRF  VCATNAME---------------DEDRCAT  STORNAME----------------------
```

Figure 15 SYSPRINT for example 1 (part 6 of 6)

```
    STATISTICS
      CARD---------------------10000  NACTIVE--------------------619  NPAGES---------------------556  ROWAVG---------------------173
      ROWMAXFOUND----------------179  ROWMINFOUND----------------179  AVGNONCOMPROWLEN----------(-1)  DIRTY------------------------0
      FULL-------------------------0  FARINDREF--------------------0  NEARINDREF-------------------0  PERCACTIVE------------------70
      PCTPAGES--------------------89  PERCDROP---------------------0  REORGSPACE------------------60  REORGSPACE_KB-------------2880
      PCTUSED--------------------100  PQTYROWS------------------2088
    ALLOCATION
      IPREFIX----------------------I  SPACE-----------------------60  SPACE_KB------------------2880  NUMDATASETS------------------1
      EXTENTS----------------------4  VOLCOUNT---------------------1  DEVTYPE-------------------3390  VOLUME------------------ARU246

  TABLE --------------- ARU.TS0011
    ASSOCIATIONS
      TABLESPACE ------ ARUDB001.TS001
    HISTORY
      SAVESTATS--------------------N  UPDATEDB2--------------------Y  UTILCODE----------------(NULL)  LOCATION----------------(NULL)
    STATISTICS
      CARD---------------------10000  NPAGES---------------------556  SPACE_KB---------------------0  AVGNONCOMPROWLEN----------(-1)
      ROWAVG---------------------173  ROWMAXFOUND----------------179  ROWMINFOUND----------------179  INDREF-----------------------0
      PCTPAGES--------------------89  PCTROWCOMP-------------------0
BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Example 2: Partitioned table space, single-phase reorganization with dynamic data set allocation

Both cases in this example reorganize a 16-partition, storage-group-defined table space. The table has three indexes: a clustering index, a unique nonpartitioned secondary index, and a non-unique nonpartitioned secondary index.

Both cases combine unload and reload processing into a single REORG phase. One of the differences between the two cases, however, is the data availability during the job as specified by the SHRLEVEL option:

- Case 1 uses SHRLEVEL NONE (the default). Therefore, the data is unavailable during the job.

- Case 2 specifies SHRLEVEL REFERENCE, which allows the data to be available in read-only (RO) status during the job. REORG PLUS messages BMC50890I and BMC50891I detail the rename and delete activity associated with the staging data sets.

In case 1, dynamic data set allocation by REORG PLUS is inactive, although BMCSORT dynamically allocates the sort work files. With the exception of sort work files, dynamic allocation is inactivated using the DDTYPE *ddtype* ACTIVE NO command option, overriding the ACTIVE YES installation option values.

Case 2 also illustrates dynamic data set allocation by REORG PLUS for unload, work, and sort work data sets and the full copy data set. The dynamic data set allocation report (messages BMC50445I through BMC50448I) shows the data set allocations.

In both cases, REORG PLUS creates the inline copy data set while reloading the table space. Table 65 describes the key command options and DD statements for this job.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 65 Key command options and DD statements used in example 2 (part 1 of 2)

Command options and DD statements used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
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 COPYLVL PART	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 //ASUSRPRT DD DUMMY. This DD statement tells REORG PLUS not to produce a report of the statistics.
SHRLEVEL NONE SHRLEVEL REFERENCE	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 (RP) status during the job.
ANALYZE SCAN	calculates optimal data set sizes for dynamic data set allocation (used in case 2 only)

Table 65 Key command options and DD statements used in example 2 (part 2 of 2)

Command options and DD statements used in JCL	Description

//SYSREC01
:
:
//SYSREC16
	data sets that contain the rows that you are reorganizing
	This example specifies multiple SYSREC*nn* 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.

//SYSUT101 //SYSUT102	In case 1, two SYSUT1*nn* data sets are allocated for the two nonpartitioned secondary index In case 2, no SYSUT1*nn* data sets are specified in the JCL because dynamic data set allocation is active.
//FULLCP	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)

Figure 16 shows the JCL for case 1 of example 2.

Figure 16 JCL for example 2, case 1 (part 1 of 3)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  PARTITIONED TABLESPACE, 16 PARTS, 3 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='DEDR,EXMPL02A,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
```

Figure 16 JCL for example 2, case 1 (part 2 of 3)

```
//SYSUDUMP  DD  SYSOUT=*
//ASUSRPRT  DD  DUMMY
//*
//SYSREC01  DD DSN=ARU.EXMPL02A.SYSREC01,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC02  DD DSN=ARU.EXMPL02A.SYSREC02,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC03  DD DSN=ARU.EXMPL02A.SYSREC03,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC04  DD DSN=ARU.EXMPL02A.SYSREC04,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC05  DD DSN=ARU.EXMPL02A.SYSREC05,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC06  DD DSN=ARU.EXMPL02A.SYSREC06,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC07  DD DSN=ARU.EXMPL02A.SYSREC07,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC08  DD DSN=ARU.EXMPL02A.SYSREC08,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC09  DD DSN=ARU.EXMPL02A.SYSREC09,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC10  DD DSN=ARU.EXMPL02A.SYSREC10,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC11  DD DSN=ARU.EXMPL02A.SYSREC11,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC12  DD DSN=ARU.EXMPL02A.SYSREC12,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC13  DD DSN=ARU.EXMPL02A.SYSREC13,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC14  DD DSN=ARU.EXMPL02A.SYSREC14,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC15  DD DSN=ARU.EXMPL02A.SYSREC15,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
//SYSREC16  DD DSN=ARU.EXMPL02A.SYSREC16,
//          UNIT=WORK,SPACE=(CYL,(20,10)),
//          DISP=(MOD,CATLG,CATLG)
```

Figure 16 JCL for example 2, case 1 (part 3 of 3)

```
//*
//SYSUT101  DD DSN=ARU.EXMPL02A.SYSUT101,
//          UNIT=WORK,SPACE=(CYL,(20,20)),
//          DISP=(MOD,CATLG,CATLG)
//SYSUT102  DD  DSN=ARU.EXMPL02A.SYSUT102,
//          UNIT=WORK,SPACE=(CYL,(20,20)),
//          DISP=(MOD,CATLG,CATLG)
//*
//FULLCP    DD DSN=ARU.EXMPL02A.FULLCPY,
//          UNIT=CART,
//          DISP=(,CATLG)
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB007.TS007
UPDATE ARU.TS0071 SET EMPDEPT = 'DP' WHERE EMPNO = '011110009'
COPY YES
COPYLVL FULL
COPYDDN(FULLCP)
BMCSTATS YES
DDTYPE UNLOAD ACTIVE NO
DDTYPE WORK ACTIVE NO
DDTYPE LOCPFCPY ACTIVE NO
/*
```

Figure 17 shows the JCL for case 2 of example 2.

Figure 17 JCL for example 2, case 2 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  PARTITIONED TABLESPACE, 16 PARTS, 3 INDEXS, STOGROUP DEFINED     *
//*  -CLUSTERING INDEX, UNIQUE                                        *
//*  -SECONDARY INDEX, UNIQUE                                         *
//*  -SECONDARY INDEX, NON-UNIQUE                                     *
//*                                                                   *
//*  REORG TABLESPACE USING SINGLE PHASE REORG (DEFAULT),             *
//*  USING SHRLEVEL REFERENCE OPTION,                                 *
//*  USING DYNAMIC WORKFILE ALLOCATIONS FOR WORK AND COPY,            *
//*  DELETE WORK FILES VIA THE DELETEFILES YES OPTION (DEFAULT),      *
//*  REALLOCATE SPACE VIA THE REDEFINE YES OPTION (DEFAULT),          *
//*  MAKE IMAGE COPY ON DASD OF ALL PARTS,                            *
//*  UPDATE DASD MANAGER STATS.                                       *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='DEDR,EXMPL02B,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
```

Figure 17 JCL for example 2, case 2 (part 2 of 2)

```
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//ASUSRPRT  DD  DUMMY
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB007.TS007
      SHRLEVEL REFERENCE
      COPY YES
      COPYLVL PART
      ANALYZE SCAN
      BMCSTATS YES
      DDTYPE WORK
        DSNPAT 'ARU.&UTILPFX.&DDNAME..A&PART5'
      DDTYPE LOCPFCPY
        DSNPAT 'ARU.&UTILPFX.&DDNAME.(+1)'
/*
```

Figure 18 shows the SYSPRINT output for case 1 of example 2.

Figure 18 SYSPRINT for example 2, case 1 (part 1 of 8)

```
                                *****  B M C   R E O R G   P L U S   F O R   D B 2   V10R1.00  *****
                                              (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
                REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
 BMC50001I UTILITY EXECUTION STARTING    1/27/2011   13:48:10 ...
 BMC50002I UTILITY ID = 'EXMPL02A'.  DB2 SUBSYSTEM ID = 'DEDR'.  OPTION MODULE = 'ARU$OPTS'.
 BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
 BMC50471I REGION=0M,BELOW 16M=8824K,ABOVE 16M=1409420K,IEFUSI=NO,CPUS=3
 BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

 BMC50471I REORG PLUS FOR DB2--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I SOLUTION COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMCSORT ENGINE--V02.03.01
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMC STATS API--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT

 BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
 BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
 BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
 BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
 BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
 BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
 BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
 BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
 BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
 BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
 BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
 BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
 BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
 BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
 BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
 BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
 BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
 BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
 BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
 BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
 BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
```

Figure 18 SYSPRINT for example 2, case 1 (part 2 of 8)

```
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
```

Figure 18 SYSPRINT for example 2, case 1 (part 3 of 8)

```
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DEDR
BMC50471I CHARACTER SET           = ALPHANUM
BMC50471I DATE FORMAT             = USA
BMC50471I TIME FORMAT             = USA
```

Figure 18 SYSPRINT for example 2, case 1 (part 4 of 8)

```
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'
BMC50471I ...EXCEPTIONS  ='ASU101.EXCEPTIONS2'

BMC50102I REORG TABLESPACE ARUDB007.TS007
BMC50102I UPDATE ARU.TS0071 SET EMPDEPT = 'DP' WHERE EMPNO = '011110009'
BMC50102I COPY YES
BMC50102I COPYLVL FULL
BMC50102I COPYDDN(FULLCP)
BMC50102I BMCSTATS YES
BMC50102I DDTYPE UNLOAD ACTIVE NO
BMC50102I DDTYPE WORK ACTIVE NO
BMC50102I DDTYPE LOCPFCPY ACTIVE NO

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 16: SAMPLING STATISTICS: PART=1,TP=94,SP=92,SR=1610,AVGR=101,SD=65,SE=0,AVGF=1750,SD=65,SE=147,EP=36
BMC50482I 16: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001'
BMC51301I 15: SAMPLING STATISTICS: PART=2,TP=3,SP=1,SR=10,AVGR=133,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC51301I 14: SAMPLING STATISTICS: PART=3,TP=3,SP=1,SR=10,AVGR=159,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC50482I 15: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002'
BMC51301I 16: SAMPLING STATISTICS: PART=4,TP=3,SP=1,SR=10,AVGR=112,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC50482I 14: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003'
BMC50482I 16: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004'
BMC50482I 13: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005'
BMC50482I 15: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A006'
BMC50482I 14: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A007'
BMC50482I 12: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A008'
BMC50482I 16: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A009'
BMC50482I 13: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A010'
BMC51301I 14: SAMPLING STATISTICS: PART=13,TP=3,SP=1,SR=10,AVGR=112,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC50482I 11: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A011'
BMC50482I 15: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A012'
BMC51301I 12: SAMPLING STATISTICS: PART=14,TP=3,SP=1,SR=10,AVGR=133,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC51301I 16: SAMPLING STATISTICS: PART=16,TP=23,SP=21,SR=414,AVGR=101,SD=65,SE=0,AVGF=1971,SD=65,SE=303,EP=6
BMC50482I 14: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A013'
BMC50482I 16: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A016'
BMC51301I 10: SAMPLING STATISTICS: PART=15,TP=3,SP=1,SR=10,AVGR=159,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC50482I 12: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A014'
BMC50482I 10: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A015'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 2084  AVG SORTWK ROW LENGTH = 102  AVG UNLOAD ROW LENGTH = 102
BMC50484I ESTIMATED CARDINALITY OF PART 0001 = 1610  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I ESTIMATED CARDINALITY OF PART 0002 = 10  AVG SORTWK ROW LENGTH = 133  AVG UNLOAD ROW LENGTH = 133
BMC50484I ESTIMATED CARDINALITY OF PART 0003 = 10  AVG SORTWK ROW LENGTH = 159  AVG UNLOAD ROW LENGTH = 159
BMC50484I ESTIMATED CARDINALITY OF PART 0004 = 10  AVG SORTWK ROW LENGTH = 112  AVG UNLOAD ROW LENGTH = 112
BMC50484I ESTIMATED CARDINALITY OF PART 0005 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
```

Figure 18 SYSPRINT for example 2, case 1 (part 5 of 8)

```
BMC50484I ESTIMATED CARDINALITY OF PART 0006 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I ESTIMATED CARDINALITY OF PART 0007 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I ESTIMATED CARDINALITY OF PART 0008 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I ESTIMATED CARDINALITY OF PART 0009 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I ESTIMATED CARDINALITY OF PART 0010 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I ESTIMATED CARDINALITY OF PART 0011 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I ESTIMATED CARDINALITY OF PART 0012 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I ESTIMATED CARDINALITY OF PART 0013 = 10  AVG SORTWK ROW LENGTH = 112  AVG UNLOAD ROW LENGTH = 112
BMC50484I ESTIMATED CARDINALITY OF PART 0014 = 10  AVG SORTWK ROW LENGTH = 133  AVG UNLOAD ROW LENGTH = 133
BMC50484I ESTIMATED CARDINALITY OF PART 0015 = 10  AVG SORTWK ROW LENGTH = 159  AVG UNLOAD ROW LENGTH = 159
BMC50484I ESTIMATED CARDINALITY OF PART 0016 = 414  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC51264I UNLOAD WILL READ 119 DATA PAGES FROM SPACE 'ARUDB007.TS007'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGAEXM2  &STEPNAME   BMCREORG  &DB         ARUDB007  &TSIX       TS007     &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       134810    &SSID       DEDR      &UTIL       EXMPL02A
BMC50425I &UTILPFX    EXMPL02A  &UTILSFX              &DATE8      01272011  &GRPNM      DEDR      &VCAT       DEDRCAT
BMC50425I &TIME4      1348      &DATEJ      2011027   &JDATE      11027

BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT
BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50474I BELOW 16M = 8136K, ABOVE 16M = 1388412K, CPUS = 3
BMC50479I TOTAL PAGES: 2258012, ALLOWED: 0; AVAILABLE PAGES: 277810, ALLOWED: 277810
BMC51302I MAX TASKS = 9, MAX PARTITIONS PER TASK = 2, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 2
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 5: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 2: PARTITION =    6, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00030
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 6
BMC51281I UNLOAD STATISTICS:  X'079D4B7275B3' IS THE HIGHEST LOGRBA FOR PARTITION 6
BMC50477I 2: PARTITION =   16, ROWS/KEYS = 414, I/O WAITS = 1 ,DDNAME = SYS00031
BMC50486I 3: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC51271I UNLOAD STATISTICS:  414 ROWS/KEYS UNLOADED FROM PARTITION 16
BMC51281I UNLOAD STATISTICS:  X'079D4B9236DC' IS THE HIGHEST LOGRBA FOR PARTITION 16
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50477I 5: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00032
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 4
BMC51281I UNLOAD STATISTICS:  X'079D4B8208F2' IS THE HIGHEST LOGRBA FOR PARTITION 4
BMC50477I 5: PARTITION =    9, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00033
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 9
BMC51281I UNLOAD STATISTICS:  X'079D4B72DA8B' IS THE HIGHEST LOGRBA FOR PARTITION 9
BMC50481I 5: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50477I 3: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00035
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 2
BMC51281I UNLOAD STATISTICS:  X'079D4B81B494' IS THE HIGHEST LOGRBA FOR PARTITION 2
BMC50486I 7: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 3: PARTITION =    7, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00037
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 7
BMC51281I UNLOAD STATISTICS:  X'079D4B729779' IS THE HIGHEST LOGRBA FOR PARTITION 7
BMC50481I 3: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 1 ,DDNAME = SYS00039
BMC50486I 4: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 7: PARTITION =   11, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00044
BMC50486I 9: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 4: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00047
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A016'
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 3
BMC51281I UNLOAD STATISTICS:  X'079D4B81D53F' IS THE HIGHEST LOGRBA FOR PARTITION 3
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 11
BMC51281I UNLOAD STATISTICS:  X'079D4B731E11' IS THE HIGHEST LOGRBA FOR PARTITION 11
BMC51271I UNLOAD STATISTICS:  1610 ROWS/KEYS UNLOADED FROM PARTITION 1
BMC51281I UNLOAD STATISTICS:  X'079D4B8FD457' IS THE HIGHEST LOGRBA FOR PARTITION 1
BMC50486I 6: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 4: PARTITION =    8, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00048
BMC50477I 9: PARTITION =   12, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00051
BMC50477I 1: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00050
BMC50477I 7: PARTITION =   14, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00049
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 8
BMC51281I UNLOAD STATISTICS:  X'079D4B72B902' IS THE HIGHEST LOGRBA FOR PARTITION 8
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 12
BMC51281I UNLOAD STATISTICS:  X'079D4B734000' IS THE HIGHEST LOGRBA FOR PARTITION 12
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 14
BMC51281I UNLOAD STATISTICS:  X'079D4B91F6DE' IS THE HIGHEST LOGRBA FOR PARTITION 14
```

Figure 18 SYSPRINT for example 2, case 1 (part 6 of 8)

```
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 5
BMC51281I UNLOAD STATISTICS:  X'079D4B7253ED' IS THE HIGHEST LOGRBA FOR PARTITION 5
BMC50481I 4: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:01
BMC50481I 7: SORT COMPLETE.  ELAPSED TIME = 00:00:01
BMC50486I 8: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 6: PARTITION =   10, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00058
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 10
BMC51281I UNLOAD STATISTICS:  X'079D4B72FC14' IS THE HIGHEST LOGRBA FOR PARTITION 10
BMC50477I 6: PARTITION =   13, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00061
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 13
BMC51281I UNLOAD STATISTICS:  X'079D4B91D32A' IS THE HIGHEST LOGRBA FOR PARTITION 13
BMC50481I 6: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50477I 8: PARTITION =   15, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00064
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 15
BMC51281I UNLOAD STATISTICS:  X'079D4B9217A3' IS THE HIGHEST LOGRBA FOR PARTITION 15
BMC51272I UNLOAD STATISTICS:  2084 ROWS UNLOADED FROM SPACE 'ARUDB007.TS007', 0 ROWS DISCARDED, 1 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'079D4B9236DC' IS THE HIGHEST LOGRBA FOR SPACE 'ARUDB007.TS007'
BMC50481I 8: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003'
BMC50482I 5: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A004'
BMC50477I 5: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00063
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A014'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001'
BMC50482I 5: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004'
BMC50477I 5: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00043
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A016'
BMC50477I 2: PARTITION =   16, ROWS/KEYS = 414, I/O WAITS = 5 ,DDNAME = SYS00067
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 4
BMC50482I 2: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A016'
BMC50477I 2: PARTITION =   16, ROWS/KEYS = 414, I/O WAITS = 3 ,DDNAME = SYS00040
BMC51274I RELOAD STATISTICS:  414 ROWS/KEYS LOADED INTO PARTITION 16
BMC50482I 9: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A012'
BMC50477I 9: PARTITION =   12, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00070
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A013'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A015'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A012'
BMC50482I 4: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A003'
BMC50477I 4: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00088
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 5 ,DDNAME = SYS00090
BMC50482I 9: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A012'
BMC50477I 9: PARTITION =   12, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00099
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 12
BMC50481I 9: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:04
BMC50482I 4: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003'
BMC50477I 4: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00069
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 3
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 3 ,DDNAME = SYS00072
BMC51274I RELOAD STATISTICS:  1610 ROWS/KEYS LOADED INTO PARTITION 1
BMC50482I 7: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A014'
BMC50477I 7: PARTITION =   14, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00092
BMC50482I 5: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A009'
BMC50477I 5: PARTITION =    9, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00096
BMC50482I 7: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A014'
BMC50477I 7: PARTITION =   14, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00071
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 14
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A006'
BMC50477I 2: PARTITION =    6, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00098
BMC50482I 6: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A013'
BMC50477I 6: PARTITION =   13, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00101
BMC50482I 8: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A015'
BMC50477I 8: PARTITION =   15, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00102
BMC50482I 6: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A013'
BMC50477I 6: PARTITION =   13, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00084
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 13
BMC50482I 8: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A015'
BMC50477I 8: PARTITION =   15, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00087
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 15
BMC50481I 8: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:05
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A009'
BMC50482I 5: RELOAD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A009'
BMC50477I 5: PARTITION =    9, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00113
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 9
BMC50481I 5: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:07
```

Figure 18 SYSPRINT for example 2, case 1 (part 7 of 8)

```
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A006'
BMC50482I 2: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A006'
BMC50477I 2: PARTITION =    6, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00118
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 6
BMC50481I 2: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:08
BMC50482I 4: BUILD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A008'
BMC50477I 4: PARTITION =    8, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00116
BMC50482I 6: BUILD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A010'
BMC50477I 6: PARTITION =   10, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00121
BMC50482I 7: BUILD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A011'
BMC50477I 7: PARTITION =   11, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00117
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A005'
BMC50477I 1: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00115
BMC50482I 3: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A002'
BMC50477I 3: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00123
BMC50482I 3: RELOAD COMPLETE.  ELAPSED TIME = 00:00:07  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002'
BMC50477I 3: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00046
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 2
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A011'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A008'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A010'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005'
BMC50482I 7: RELOAD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A011'
BMC50477I 7: PARTITION =   11, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00133
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 11
BMC50481I 7: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:09
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005'
BMC50477I 1: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00134
BMC50482I 4: RELOAD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A008'
BMC50477I 4: PARTITION =    8, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00135
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 5
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 8
BMC50481I 1: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:09
BMC50482I 6: RELOAD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A010'
BMC50477I 6: PARTITION =   10, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00136
BMC50481I 4: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:09
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 10
BMC50481I 6: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:09
BMC50482I 3: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A007'
BMC50477I 3: PARTITION =    7, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00138
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A007'
BMC50482I 3: RELOAD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A007'
BMC50477I 3: PARTITION =    7, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00141
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 7
BMC50481I 3: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:10
BMC50476I DDNAME = SYSREC01, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC02, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC03, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC04, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC05, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC06, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC07, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC08, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC09, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC10, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC11, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC12, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC13, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC14, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC15, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC16, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT101, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT102, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51276I BUILD  STATISTICS:  2084 KEYS LOADED INTO INDEX 'ARU.TS00711'
BMC51275I RELOAD STATISTICS:  2084 ROWS LOADED INTO SPACE 'ARUDB007.TS007'
BMC50474I BELOW 16M = 8108K, ABOVE 16M = 1386660K, CPUS = 3
BMC50479I TOTAL PAGES: 2319265, ALLOWED: 0; AVAILABLE PAGES: 339550, ALLOWED: 339550
BMC51303I MAX TASKS = 2, INDEXES PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 2
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50476I DDNAME = SYSUT101, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50476I DDNAME = SYSUT102, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00712.I0001.A001'
```

Figure 18 SYSPRINT for example 2, case 1 (part 8 of 8)

```
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 2084, I/O WAITS = 5 ,DDNAME = SYS00147
BMC51276I BUILD  STATISTICS:  2084 KEYS LOADED INTO INDEX 'ARU.TS00712'
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00713.I0001.A001'
BMC50477I 2: PARTITION =    0, ROWS/KEYS = 2084, I/O WAITS = 5 ,DDNAME = SYS00149
BMC51276I BUILD  STATISTICS:  2084 KEYS LOADED INTO INDEX 'ARU.TS00713'
BMC50476I DDNAME = FULLCP, I/OS = 2, I/O WAITS = 1, RDB LOCK WAITS = 2
BMC50376I 132 PAGES COPIED TO DATASET = 'ARU.EXMPL02A.FULLCPY'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT102', DSNAME = 'ARU.EXMPL02A.SYSUT102'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT101', DSNAME = 'ARU.EXMPL02A.SYSUT101'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC16', DSNAME = 'ARU.EXMPL02A.SYSREC16'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC15', DSNAME = 'ARU.EXMPL02A.SYSREC15'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC14', DSNAME = 'ARU.EXMPL02A.SYSREC14'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC13', DSNAME = 'ARU.EXMPL02A.SYSREC13'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC12', DSNAME = 'ARU.EXMPL02A.SYSREC12'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC11', DSNAME = 'ARU.EXMPL02A.SYSREC11'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC10', DSNAME = 'ARU.EXMPL02A.SYSREC10'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC09', DSNAME = 'ARU.EXMPL02A.SYSREC09'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC08', DSNAME = 'ARU.EXMPL02A.SYSREC08'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC07', DSNAME = 'ARU.EXMPL02A.SYSREC07'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC06', DSNAME = 'ARU.EXMPL02A.SYSREC06'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC05', DSNAME = 'ARU.EXMPL02A.SYSREC05'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC04', DSNAME = 'ARU.EXMPL02A.SYSREC04'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC03', DSNAME = 'ARU.EXMPL02A.SYSREC03'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC02', DSNAME = 'ARU.EXMPL02A.SYSREC02'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC01', DSNAME = 'ARU.EXMPL02A.SYSREC01'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I REORG PHASE COMPLETE.  ELAPSED TIME = 00:00:17

BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Figure 19 shows the SYSPRINT output for case 2 of example 2.

Figure 19 SYSPRINT for example 2, case 2 (part 1 of 13)

```
                              *****  B M C    R E O R G    P L U S    F O R    D B 2    V10R1.00  *****
                                            (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
                REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50001I UTILITY EXECUTION STARTING    1/27/2011    13:58:38 ...
BMC50002I UTILITY ID = 'EXMPL02B'.  DB2 SUBSYSTEM ID = 'DEDR'.  OPTION MODULE = 'ARU$OPTS'.
BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
BMC50471I REGION=0M,BELOW 16M=8852K,ABOVE 16M=1410552K,IEFUSI=NO,CPUS=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

BMC50471I REORG PLUS FOR DB2--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.03.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                              INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                         INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                             INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                    IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                            IXONEX=NO                               SMCORE=(OK,OK)
BMC50471I AUXREORG=DEFAULT                          IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                          KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                              LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                               LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                                 LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                                 LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                               LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                   LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                              LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                             LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                             LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                             MAXNEWPARTS=2                           TAPEDISP=DELETE
```

Figure 19 SYSPRINT for example 2, case 2 (part 2 of 13)

```
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
```

Figure 19 SYSPRINT for example 2, case 2 (part 3 of 13)

```
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
```

Figure 19 SYSPRINT for example 2, case 2 (part 4 of 13)

```
BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DEDR
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'
BMC50471I ...EXCEPTIONS  ='ASU101.EXCEPTIONS2'

BMC50102I REORG TABLESPACE ARUDB007.TS007
BMC50102I       SHRLEVEL REFERENCE
BMC50102I       COPY YES
BMC50102I       COPYLVL PART
BMC50102I       ANALYZE SCAN
BMC50102I       BMCSTATS YES
BMC50102I       DDTYPE WORK
BMC50102I         DSNPAT 'ARU.&UTILPFX.&DDNAME..A&PART5'
BMC50102I       DDTYPE LOCPFCPY
BMC50102I         DSNPAT 'ARU.&UTILPFX.&DDNAME.(+1)'

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:02

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50482I 17: SCAN COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00713.I0001.A001'
BMC51301I 17: SAMPLING STATISTICS: PART=2,TP=3,SP=1,SR=10,AVGR=133,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC51301I 16: SAMPLING STATISTICS: PART=1,TP=94,SP=92,SR=1610,AVGR=101,SD=65,SE=0,AVGF=1750,SD=65,SE=147,EP=36
BMC50482I 16: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001'
BMC50482I 17: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002'
BMC51301I 15: SAMPLING STATISTICS: PART=3,TP=3,SP=1,SR=10,AVGR=159,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC51301I 14: SAMPLING STATISTICS: PART=4,TP=3,SP=1,SR=10,AVGR=112,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC50482I 15: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003'
BMC50482I 14: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004'
BMC50482I 17: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A006'
BMC50482I 16: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005'
BMC50482I 15: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A008'
BMC50482I 13: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A007'
BMC50482I 14: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A009'
BMC50482I 17: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A011'
BMC50482I 12: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A010'
BMC51301I 11: SAMPLING STATISTICS: PART=13,TP=3,SP=1,SR=10,AVGR=112,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC51301I 13: SAMPLING STATISTICS: PART=15,TP=3,SP=1,SR=10,AVGR=159,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC51301I 15: SAMPLING STATISTICS: PART=14,TP=3,SP=1,SR=10,AVGR=133,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC50482I 16: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A012'
BMC50482I 11: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A013'
BMC50482I 13: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A015'
BMC51301I 14: SAMPLING STATISTICS: PART=16,TP=23,SP=21,SR=414,AVGR=101,SD=65,SE=0,AVGF=1971,SD=65,SE=303,EP=6
```

Figure 19 SYSPRINT for example 2, case 2 (part 5 of 13)

```
BMC50482I 15: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A014'
BMC50482I 14: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A016'
BMC51266I CARDINALITY OF SPACE = 2084  AVG SORTWK ROW LENGTH = 102  AVG UNLOAD ROW LENGTH = 102
BMC50484I CARDINALITY OF PART 0001 = 1610  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I CARDINALITY OF PART 0002 = 10  AVG SORTWK ROW LENGTH = 133  AVG UNLOAD ROW LENGTH = 133
BMC50484I CARDINALITY OF PART 0003 = 10  AVG SORTWK ROW LENGTH = 159  AVG UNLOAD ROW LENGTH = 159
BMC50484I CARDINALITY OF PART 0004 = 10  AVG SORTWK ROW LENGTH = 112  AVG UNLOAD ROW LENGTH = 112
BMC50484I CARDINALITY OF PART 0005 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0006 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0007 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0008 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0009 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0010 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0011 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0012 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I CARDINALITY OF PART 0013 = 10  AVG SORTWK ROW LENGTH = 112  AVG UNLOAD ROW LENGTH = 112
BMC50484I CARDINALITY OF PART 0014 = 10  AVG SORTWK ROW LENGTH = 133  AVG UNLOAD ROW LENGTH = 133
BMC50484I CARDINALITY OF PART 0015 = 10  AVG SORTWK ROW LENGTH = 159  AVG UNLOAD ROW LENGTH = 159
BMC50484I CARDINALITY OF PART 0016 = 414  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC51264I UNLOAD WILL READ 119 DATA PAGES FROM SPACE 'ARUDB007.TS007'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGBEXM2  &STEPNAME   BMCREORG  &DB         ARUDB007  &TSIX       TS007     &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       135838    &SSID       DEDR      &UTIL       EXMPL02B
BMC50425I &UTILPFX    EXMPL02B  &UTILSFX              &DATE8      01272011  &GRPNM      DEDR      &VCAT       DEDRCAT
BMC50425I &TIME4      1358      &DATEJ      2011027   &JDATE      11027

BMC50877I POINT OF CONSISTENCY ESTABLISHED AT RBA/LRSN = 079D4BF6B67C
BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                       UNIT OR                      KBYTES   KBYTES     ALOC      ALOC
BMC50447I DDNAME   DSNAME                                       DATACLAS MGMTCLAS STORCLAS      PRI      SEC      PRI       SEC

BMC50448I BMCCPY01 ARU.EXMPL02B.BMCCPY01.G0013V00               SYSALLDA                        376       37        8         1  TRK
BMC50448I BMCCPY02 ARU.EXMPL02B.BMCCPY02.G0013V00               SYSALLDA                         96        9        2         1  TRK
BMC50448I BMCCPY03 ARU.EXMPL02B.BMCCPY03.G0013V00               SYSALLDA                         96        9        2         1  TRK
BMC50448I BMCCPY04 ARU.EXMPL02B.BMCCPY04.G0013V00               SYSALLDA                         96        9        2         1  TRK
BMC50448I BMCCPY05 ARU.EXMPL02B.BMCCPY05.G0013V00               SYSALLDA                          8        0        1         0  TRK
BMC50448I BMCCPY06 ARU.EXMPL02B.BMCCPY06.G0013V00               SYSALLDA                          8        0        1         0  TRK
BMC50448I BMCCPY07 ARU.EXMPL02B.BMCCPY07.G0013V00               SYSALLDA                          8        0        1         0  TRK
BMC50448I BMCCPY08 ARU.EXMPL02B.BMCCPY08.G0013V00               SYSALLDA                          8        0        1         0  TRK
BMC50448I BMCCPY09 ARU.EXMPL02B.BMCCPY09.G0013V00               SYSALLDA                          8        0        1         0  TRK
BMC50448I BMCCPY10 ARU.EXMPL02B.BMCCPY10.G0013V00               SYSALLDA                          8        0        1         0  TRK
BMC50448I BMCCPY11 ARU.EXMPL02B.BMCCPY11.G0013V00               SYSALLDA                          8        0        1         0  TRK
BMC50448I BMCCPY12 ARU.EXMPL02B.BMCCPY12.G0013V00               SYSALLDA                          8        0        1         0  TRK
BMC50448I BMCCPY13 ARU.EXMPL02B.BMCCPY13.G0013V00               SYSALLDA                         96        9        2         1  TRK
BMC50448I BMCCPY14 ARU.EXMPL02B.BMCCPY14.G0013V00               SYSALLDA                         96        9        2         1  TRK
BMC50448I BMCCPY15 ARU.EXMPL02B.BMCCPY15.G0013V00               SYSALLDA                         96        9        2         1  TRK
BMC50448I BMCCPY16 ARU.EXMPL02B.BMCCPY16.G0013V00               SYSALLDA                         92        9        2         1  TRK
BMC50448I SYSUT101 ARU.EXMPL02B.SYSUT101.A00000                 SYSALLDA                         57        6        2         1  TRK
BMC50448I SYSUT102 ARU.EXMPL02B.SYSUT102.A00000                 SYSALLDA                         43        5        1         1  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50474I BELOW 16M = 8212K, ABOVE 16M = 1399636K, CPUS = 3
BMC50479I TOTAL PAGES: 2171890, ALLOWED: 0; AVAILABLE PAGES: 189202, ALLOWED: 189202
BMC51302I MAX TASKS = 9, MAX PARTITIONS PER TASK = 2, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 2
BMC50486I 8: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 9: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 8: PARTITION =   15, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00029
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 15
BMC51281I UNLOAD STATISTICS:  X'079D4BEF6E58' IS THE HIGHEST LOGRBA FOR PARTITION 15
BMC50481I 8: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50486I 4: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 9: PARTITION =   12, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00030
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 12
BMC51281I UNLOAD STATISTICS:  X'079D4BD7EE79' IS THE HIGHEST LOGRBA FOR PARTITION 12
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A015'
BMC50486I 5: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 4: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00033
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 3
BMC51281I UNLOAD STATISTICS:  X'079D4BE0717F' IS THE HIGHEST LOGRBA FOR PARTITION 3
BMC50477I 4: PARTITION =    8, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00034
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 8
BMC51281I UNLOAD STATISTICS:  X'079D4BD767F4' IS THE HIGHEST LOGRBA FOR PARTITION 8
```

Figure 19 SYSPRINT for example 2, case 2 (part 6 of 13)

```
BMC50481I 4: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 5: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00038
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 4
BMC51281I UNLOAD STATISTICS:  X'079D4BE0A80A' IS THE HIGHEST LOGRBA FOR PARTITION 4
BMC50482I 8: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A015'
BMC50477I 8: PARTITION =   15, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00037
BMC50477I 5: PARTITION =    9, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00041
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 9
BMC51281I UNLOAD STATISTICS:  X'079D4BD789BA' IS THE HIGHEST LOGRBA FOR PARTITION 9
BMC50482I 8: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A015'
BMC50477I 8: PARTITION =   15, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00032
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 15
BMC50476I DDNAME = BMCCPY15, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY15.G0013V00'
BMC50481I 5: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50481I 8: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:01
BMC50486I 3: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 1 ,DDNAME = SYS00045
BMC50482I 9: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A012'
BMC50477I 9: PARTITION =   12, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00042
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A003'
BMC51271I UNLOAD STATISTICS:  1610 ROWS/KEYS UNLOADED FROM PARTITION 1
BMC51281I UNLOAD STATISTICS:  X'079D4BECDDCA' IS THE HIGHEST LOGRBA FOR PARTITION 1
BMC50477I 1: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00047
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 5
BMC51281I UNLOAD STATISTICS:  X'079D4BD70359' IS THE HIGHEST LOGRBA FOR PARTITION 5
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50486I 7: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 3: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00051
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 2
BMC51281I UNLOAD STATISTICS:  X'079D4BE05077' IS THE HIGHEST LOGRBA FOR PARTITION 2
BMC50486I 6: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A004'
BMC50477I 3: PARTITION =    7, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00054
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 7
BMC51281I UNLOAD STATISTICS:  X'079D4BD7466B' IS THE HIGHEST LOGRBA FOR PARTITION 7
BMC50481I 3: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50477I 7: PARTITION =   11, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00057
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 11
BMC51281I UNLOAD STATISTICS:  X'079D4BD7CCCC' IS THE HIGHEST LOGRBA FOR PARTITION 11
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A012'
BMC50477I 7: PARTITION =   14, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00062
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 14
BMC51281I UNLOAD STATISTICS:  X'079D4BEF4928' IS THE HIGHEST LOGRBA FOR PARTITION 14
BMC50482I 9: RELOAD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A012'
BMC50477I 9: PARTITION =   12, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00059
BMC50477I 6: PARTITION =   10, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00064
BMC50481I 7: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 12
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 10
BMC51281I UNLOAD STATISTICS:  X'079D4BD7AB43' IS THE HIGHEST LOGRBA FOR PARTITION 10
BMC50476I DDNAME = BMCCPY12, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY12.G0013V00'
BMC50481I 9: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:02
BMC50477I 6: PARTITION =   13, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00068
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 13
BMC51281I UNLOAD STATISTICS:  X'079D4BEF2656' IS THE HIGHEST LOGRBA FOR PARTITION 13
BMC50481I 6: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 4: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A003'
BMC50477I 4: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00066
BMC50482I 4: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A003'
BMC50477I 4: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00046
BMC50477I 2: PARTITION =    6, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00076
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 3
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 6
BMC51281I UNLOAD STATISTICS:  X'079D4BD724E2' IS THE HIGHEST LOGRBA FOR PARTITION 6
BMC50476I DDNAME = BMCCPY03, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY03.G0013V00'
BMC50477I 2: PARTITION =   16, ROWS/KEYS = 414, I/O WAITS = 1 ,DDNAME = SYS00079
BMC51271I UNLOAD STATISTICS:  414 ROWS/KEYS UNLOADED FROM PARTITION 16
BMC51281I UNLOAD STATISTICS:  X'079D4BEF8D7B' IS THE HIGHEST LOGRBA FOR PARTITION 16
BMC51272I UNLOAD STATISTICS:  2084 ROWS UNLOADED FROM SPACE 'ARUDB007.TS007', 0 ROWS DISCARDED, 0 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'079D4BEF8D7B' IS THE HIGHEST LOGRBA FOR SPACE 'ARUDB007.TS007'
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:00
```

Figure 19 SYSPRINT for example 2, case 2 (part 7 of 13)

```
BMC50482I 5: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A004'
BMC50477I 5: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00077
BMC50482I 5: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A004'
BMC50477I 5: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00053
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 4
BMC50476I DDNAME = BMCCPY04, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY04.G0013V00'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A002'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A001'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A014'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A013'
BMC50482I 4: BUILD COMPLETE.  ELAPSED TIME = 00:00:04  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A008'
BMC50477I 4: PARTITION =    8, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00144
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A016'
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:04  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 5 ,DDNAME = SYS00161
BMC50482I 5: BUILD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A009'
BMC50477I 5: PARTITION =    9, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00159
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:07  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 3 ,DDNAME = SYS00099
BMC50482I 3: BUILD COMPLETE.  ELAPSED TIME = 00:00:04  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A002'
BMC50477I 3: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00162
BMC51274I RELOAD STATISTICS:  1610 ROWS/KEYS LOADED INTO PARTITION 1
BMC50476I DDNAME = BMCCPY01, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50482I 3: RELOAD COMPLETE.  ELAPSED TIME = 00:00:06  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A002'
BMC50477I 3: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00091
BMC50376I 65 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY01.G0013V00'
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 2
BMC50476I DDNAME = BMCCPY02, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY02.G0013V00'
BMC50482I 7: BUILD COMPLETE.  ELAPSED TIME = 00:00:04  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A014'
BMC50477I 7: PARTITION =   14, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00171
BMC50482I 7: RELOAD COMPLETE.  ELAPSED TIME = 00:00:07  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A014'
BMC50477I 7: PARTITION =   14, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00106
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 14
BMC50476I DDNAME = BMCCPY14, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY14.G0013V00'
BMC50482I 6: BUILD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A013'
BMC50477I 6: PARTITION =   13, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00196
BMC50482I 6: RELOAD COMPLETE.  ELAPSED TIME = 00:00:09  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A013'
BMC50477I 6: PARTITION =   13, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00136
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 13
BMC50476I DDNAME = BMCCPY13, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY13.G0013V00'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A008'
BMC50482I 4: RELOAD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A008'
BMC50477I 4: PARTITION =    8, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00209
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A016'
BMC50477I 2: PARTITION =   16, ROWS/KEYS = 414, I/O WAITS = 5 ,DDNAME = SYS00210
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 8
BMC50476I DDNAME = BMCCPY08, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY08.G0013V00'
BMC50482I 2: RELOAD COMPLETE.  ELAPSED TIME = 00:00:10  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A016'
BMC50477I 2: PARTITION =   16, ROWS/KEYS = 414, I/O WAITS = 3 ,DDNAME = SYS00153
BMC51274I RELOAD STATISTICS:  414 ROWS/KEYS LOADED INTO PARTITION 16
BMC50481I 4: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:13
BMC50476I DDNAME = BMCCPY16, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 19 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY16.G0013V00'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A009'
BMC50482I 5: RELOAD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A009'
BMC50477I 5: PARTITION =    9, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00224
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 9
BMC50476I DDNAME = BMCCPY09, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY09.G0013V00'
BMC50481I 5: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:14
BMC50482I 3: BUILD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A007'
BMC50477I 3: PARTITION =    7, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00233
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A005'
BMC50477I 1: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00235
BMC50482I 7: BUILD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A011'
BMC50477I 7: PARTITION =   11, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00239
BMC50482I 6: BUILD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A010'
BMC50477I 6: PARTITION =   10, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00251
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A006'
BMC50477I 2: PARTITION =    6, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00263
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A007'
```

Figure 19 SYSPRINT for example 2, case 2 (part 8 of 13)

```
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A005'
BMC50482I 3: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A007'
BMC50477I 3: PARTITION =    7, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00285
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 7
BMC50476I DDNAME = BMCCPY07, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY07.G0013V00'
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A005'
BMC50477I 1: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00286
BMC50481I 3: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:18
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 5
BMC50476I DDNAME = BMCCPY05, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY05.G0013V00'
BMC50481I 1: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:18
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A011'
BMC50482I 7: RELOAD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A011'
BMC50477I 7: PARTITION =   11, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00290
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 11
BMC50476I DDNAME = BMCCPY11, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY11.G0013V00'
BMC50481I 7: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:18
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A010'
BMC50482I 6: RELOAD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A010'
BMC50477I 6: PARTITION =   10, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00297
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 10
BMC50476I DDNAME = BMCCPY10, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY10.G0013V00'
BMC50481I 6: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:19
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A006'
BMC50482I 2: RELOAD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A006'
BMC50477I 2: PARTITION =    6, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00301
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 6
BMC50476I DDNAME = BMCCPY06, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL02B.BMCCPY06.G0013V00'
BMC50481I 2: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:20
BMC50476I DDNAME = SYSUT101, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT102, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51276I BUILD  STATISTICS:  2084 KEYS LOADED INTO INDEX 'ARU.TS00711'
BMC51275I RELOAD STATISTICS:  2084 ROWS LOADED INTO SPACE 'ARUDB007.TS007'
BMC50474I BELOW 16M = 8156K, ABOVE 16M = 1393144K, CPUS = 3
BMC50479I TOTAL PAGES: 2152979, ALLOWED: 0; AVAILABLE PAGES: 171618, ALLOWED: 171618
BMC51303I MAX TASKS = 2, INDEXES PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 2
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50476I DDNAME = SYSUT102, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50476I DDNAME = SYSUT101, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:01
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00713.I0001.A001'
BMC50477I 2: PARTITION =    0, ROWS/KEYS = 2084, I/O WAITS = 5 ,DDNAME = SYS00323
BMC51276I BUILD  STATISTICS:  2084 KEYS LOADED INTO INDEX 'ARU.TS00713'
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00712.I0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 2084, I/O WAITS = 5 ,DDNAME = SYS00325
BMC51276I BUILD  STATISTICS:  2084 KEYS LOADED INTO INDEX 'ARU.TS00712'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT101', DSNAME = 'ARU.EXMPL02B.SYSUT101.A00000'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT102', DSNAME = 'ARU.EXMPL02B.SYSUT102.A00000'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I REORG PHASE COMPLETE.  ELAPSED TIME = 00:00:30

BMC50894I RENAME PROCESS STARTING AT   1/27/2011  13:59:16
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A001'...
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A003'...
BMC50890I 2: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A002'...
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A004'...
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A005'...
BMC50890I 6: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A006'...
BMC50891I 3: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A003' IS RENAMED
BMC50890I 30: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A014'...
BMC50890I 29: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A013'...
BMC50890I 7: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A007'...
BMC50890I 25: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A009'...
BMC50890I 28: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A012'...
BMC50890I 22: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A006'...
BMC50891I 1: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A001' IS RENAMED
BMC50891I 2: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A002' IS RENAMED
BMC50890I 17: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A001'...
BMC50890I 10: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A010'...
```

Figure 19 SYSPRINT for example 2, case 2 (part 9 of 13)

```
BMC50890I 27: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A011'...
BMC50890I 8: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A008'...
BMC50890I 23: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A007'...
BMC50890I 13: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A013'...
BMC50890I 9: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A009'...
BMC50890I 12: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A012'...
BMC50890I 21: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A005'...
BMC50890I 11: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A011'...
BMC50890I 16: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A016'...
BMC50890I 26: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A010'...
BMC50890I 15: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A015'...
BMC50890I 24: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A008'...
BMC50890I 19: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A003'...
BMC50890I 14: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A014'...
BMC50890I 18: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A002'...
BMC50891I 4: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A004' IS RENAMED
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003'...
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001'...
BMC50890I 20: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A004'...
BMC50890I 2: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002'...
BMC50891I 5: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A005' IS RENAMED
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004'...
BMC50891I 6: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A006' IS RENAMED
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005'...
BMC50891I 22: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A006' IS RENAMED
BMC50890I 6: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A006'...
BMC50891I 13: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A013' IS RENAMED
BMC50891I 15: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A015' IS RENAMED
BMC50891I 7: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A007' IS RENAMED
BMC50891I 29: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A013' IS RENAMED
BMC50891I 17: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A001' IS RENAMED
BMC50891I 9: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A009' IS RENAMED
BMC50891I 18: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A002' IS RENAMED
BMC50891I 28: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A012' IS RENAMED
BMC50891I 8: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A008' IS RENAMED
BMC50891I 14: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A014' IS RENAMED
BMC50891I 21: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A005' IS RENAMED
BMC50891I 11: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A011' IS RENAMED
BMC50891I 23: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A007' IS RENAMED
BMC50891I 19: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A003' IS RENAMED
BMC50891I 24: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A008' IS RENAMED
BMC50891I 27: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A011' IS RENAMED
BMC50891I 25: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A009' IS RENAMED
BMC50891I 30: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A014' IS RENAMED
BMC50890I 21: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A005'...
BMC50890I 7: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A007'...
BMC50890I 27: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A011'...
BMC50890I 18: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A002'...
BMC50890I 19: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A003'...
BMC50890I 25: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A009'...
BMC50890I 22: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A006'...
BMC50890I 24: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A008'...
BMC50890I 23: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A007'...
BMC50890I 28: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A012'...
BMC50890I 15: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A015'...
BMC50890I 11: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A011'...
BMC50890I 14: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A014'...
BMC50890I 13: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A013'...
BMC50890I 8: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A008'...
BMC50890I 17: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A001'...
BMC50890I 9: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A009'...
BMC50890I 29: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A013'...
BMC50891I 1: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001' IS RENAMED
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A001'...
BMC50891I 3: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003' IS RENAMED
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A003'...
BMC50891I 16: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A016' IS RENAMED
BMC50891I 26: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A010' IS RENAMED
BMC50891I 12: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A012' IS RENAMED
BMC50891I 10: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A010' IS RENAMED
BMC50891I 20: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50890I 10: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A010'...
BMC50890I 30: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A014'...
BMC50890I 16: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A016'...
BMC50890I 12: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A012'...
BMC50890I 26: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A010'...
```

Figure 19 SYSPRINT for example 2, case 2 (part 10 of 13)

```
BMC50891I 4: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004' IS RENAMED
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A004'...
BMC50891I 2: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002' IS RENAMED
BMC50890I 20: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A004'...
BMC50890I 2: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A002'...
BMC50891I 5: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005' IS RENAMED
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A005'...
BMC50891I 3: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A003' IS RENAMED
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A003'...
BMC50891I 1: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A001' IS RENAMED
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A001'...
BMC50891I 6: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A006' IS RENAMED
BMC50890I 6: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A006'...
BMC50891I 4: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A004' IS RENAMED
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A004'...
BMC50891I 25: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A009' IS RENAMED
BMC50890I 25: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A009'...
BMC50891I 22: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A006' IS RENAMED
BMC50890I 22: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A006'...
BMC50891I 21: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A005' IS RENAMED
BMC50890I 21: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A005'...
BMC50891I 9: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A009' IS RENAMED
BMC50890I 9: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A009'...
BMC50891I 18: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A002' IS RENAMED
BMC50890I 18: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A002'...
BMC50891I 19: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A003' IS RENAMED
BMC50890I 19: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A003'...
BMC50891I 11: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A011' IS RENAMED
BMC50890I 11: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A011'...
BMC50891I 14: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A014' IS RENAMED
BMC50890I 14: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A014'...
BMC50891I 8: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A008' IS RENAMED
BMC50890I 8: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A008'...
BMC50891I 2: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A002' IS RENAMED
BMC50890I 2: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A002'...
BMC50891I 29: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A013' IS RENAMED
BMC50890I 29: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A013'...
BMC50891I 12: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A012' IS RENAMED
BMC50890I 12: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A012'...
BMC50891I 23: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A007' IS RENAMED
BMC50890I 23: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A007'...
BMC50891I 13: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A013' IS RENAMED
BMC50890I 13: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A013'...
BMC50891I 7: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A007' IS RENAMED
BMC50890I 7: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A007'...
BMC50891I 28: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A012' IS RENAMED
BMC50890I 28: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A012'...
BMC50891I 10: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A010' IS RENAMED
BMC50890I 10: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A010'...
BMC50891I 5: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A005' IS RENAMED
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A005'...
BMC50891I 16: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A016' IS RENAMED
BMC50890I 16: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A016'...
BMC50891I 15: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A015' IS RENAMED
BMC50890I 15: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A015'...
BMC50891I 24: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A008' IS RENAMED
BMC50890I 24: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A008'...
BMC50891I 30: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A014' IS RENAMED
BMC50890I 30: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A014'...
BMC50891I 17: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A001' IS RENAMED
BMC50890I 17: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A001'...
BMC50891I 26: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A010' IS RENAMED
BMC50890I 26: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A010'...
BMC50891I 20: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50890I 20: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A004'...
BMC50891I 27: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A011' IS RENAMED
BMC50890I 27: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A011'...
BMC50891I 12: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A012' IS RENAMED
BMC50890I 12: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A012'...
BMC50891I 8: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A008' IS RENAMED
BMC50890I 8: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A008'...
BMC50891I 29: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A013' IS RENAMED
BMC50890I 29: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A013'...
BMC50891I 14: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A014' IS RENAMED
BMC50890I 14: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A014'...
BMC50891I 11: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A011' IS RENAMED
```

Figure 19 SYSPRINT for example 2, case 2 (part 11 of 13)

```
BMC50890I 11: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A011'...
BMC50891I 21: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A005' IS RENAMED
BMC50890I 21: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A005'...
BMC50891I 6: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A006' IS RENAMED
BMC50890I 6: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A006'...
BMC50891I 16: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A016' IS RENAMED
BMC50890I 16: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A016'...
BMC50891I 9: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A009' IS RENAMED
BMC50890I 9: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A009'...
BMC50891I 22: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A006' IS RENAMED
BMC50890I 22: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A006'...
BMC50891I 4: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A004' IS RENAMED
BMC50891I 25: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A009' IS RENAMED
BMC50890I 25: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A009'...
BMC50891I 23: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A007' IS RENAMED
BMC50890I 23: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A007'...
BMC50891I 19: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A003' IS RENAMED
BMC50890I 19: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A003'...
BMC50891I 7: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A007' IS RENAMED
BMC50890I 7: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A007'...
BMC50891I 28: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A012' IS RENAMED
BMC50890I 28: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A012'...
BMC50891I 18: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A002' IS RENAMED
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A015'...
BMC50890I 18: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A002'...
BMC50891I 27: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A011' IS RENAMED
BMC50890I 27: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A011'...
BMC50891I 24: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A008' IS RENAMED
BMC50890I 24: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A008'...
BMC50891I 3: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A003' IS RENAMED
BMC50891I 1: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A001' IS RENAMED
BMC50891I 5: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A005' IS RENAMED
BMC50891I 2: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A002' IS RENAMED
BMC50891I 30: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A014' IS RENAMED
BMC50890I 30: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A014'...
BMC50891I 17: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A001' IS RENAMED
BMC50890I 17: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A001'...
BMC50891I 20: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50890I 20: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A004'...
BMC50891I 10: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A010' IS RENAMED
BMC50890I 10: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A010'...
BMC50891I 13: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A013' IS RENAMED
BMC50890I 13: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A013'...
BMC50891I 26: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A010' IS RENAMED
BMC50890I 26: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A010'...
BMC50891I 15: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A015' IS RENAMED
BMC50890I 15: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A015'...
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00713.I0001.A001'...
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A016'...
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00712.I0001.A001'...
BMC50891I 19: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A003' IS RENAMED
BMC50891I 4: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A015' IS RENAMED
BMC50891I 5: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00713.I0001.A001' IS RENAMED
BMC50891I 16: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A016' IS RENAMED
BMC50891I 21: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A005' IS RENAMED
BMC50891I 14: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A014' IS RENAMED
BMC50891I 3: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A016' IS RENAMED
BMC50891I 25: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A009' IS RENAMED
BMC50891I 23: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A007' IS RENAMED
BMC50891I 12: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A012' IS RENAMED
BMC50891I 11: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A011' IS RENAMED
BMC50891I 13: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A013' IS RENAMED
BMC50891I 24: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A008' IS RENAMED
BMC50891I 26: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A010' IS RENAMED
BMC50891I 22: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A006' IS RENAMED
BMC50891I 15: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A015' IS RENAMED
BMC50891I 7: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A007' IS RENAMED
BMC50891I 8: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A008' IS RENAMED
BMC50891I 9: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A009' IS RENAMED
BMC50891I 10: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A010' IS RENAMED
BMC50891I 17: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A001' IS RENAMED
BMC50891I 18: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A002' IS RENAMED
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00713.I0001.A001'...
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A016'...
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A015'...
BMC50891I 1: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00712.I0001.A001' IS RENAMED
```

Figure 19 SYSPRINT for example 2, case 2 (part 12 of 13)

```
BMC50891I 6: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A006' IS RENAMED
BMC50891I 27: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A011' IS RENAMED
BMC50891I 20: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50891I 28: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A012' IS RENAMED
BMC50891I 29: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A013' IS RENAMED
BMC50891I 30: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A014' IS RENAMED
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00712.I0001.A001'...
BMC50891I 4: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A015' IS RENAMED
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A015'...
BMC50891I 5: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00713.I0001.A001' IS RENAMED
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00713.I0001.A001'...
BMC50891I 3: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A016' IS RENAMED
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A016'...
BMC50891I 1: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00712.I0001.A001' IS RENAMED
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00712.I0001.A001'...
BMC50891I 4: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A015' IS RENAMED
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A015'...
BMC50891I 3: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A016' IS RENAMED
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A016'...
BMC50891I 5: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00713.I0001.A001' IS RENAMED
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00713.I0001.A001'...
BMC50891I 1: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00712.I0001.A001' IS RENAMED
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00712.I0001.A001'...
BMC50891I 4: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A015' IS RENAMED
BMC50891I 3: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A016' IS RENAMED
BMC50891I 5: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00713.I0001.A001' IS RENAMED
BMC50891I 1: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00712.I0001.A001' IS RENAMED
BMC50895I RENAME PROCESS COMPLETE.  ELAPSED TIME =  00:00:04

BMC50890I 22: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A006'...
BMC50890I 2: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A002'...
BMC50890I 23: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A007'...
BMC50890I 19: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A003'...
BMC50890I 8: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A008'...
BMC50890I 25: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A009'...
BMC50890I 4: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A004'...
BMC50890I 24: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A008'...
BMC50890I 7: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A007'...
BMC50890I 29: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A013'...
BMC50890I 9: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A009'...
BMC50890I 11: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A011'...
BMC50890I 16: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A016'...
BMC50890I 6: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A006'...
BMC50890I 17: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A001'...
BMC50890I 13: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A013'...
BMC50890I 21: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A005'...
BMC50890I 28: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A012'...
BMC50890I 26: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A010'...
BMC50890I 30: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A014'...
BMC50890I 10: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A010'...
BMC50890I 12: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A012'...
BMC50890I 27: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A011'...
BMC50890I 20: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A004'...
BMC50890I 1: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A001'...
BMC50890I 5: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A005'...
BMC50890I 15: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A015'...
BMC50890I 18: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A002'...
BMC50890I 3: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A003'...
BMC50890I 14: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A014'...
BMC50891I 2: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A002' IS DELETED
BMC50890I 2: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A015'...
BMC50891I 24: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A008' IS DELETED
BMC50891I 19: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A003' IS DELETED
BMC50891I 23: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A007' IS DELETED
BMC50891I 22: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A006' IS DELETED
BMC50891I 25: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A009' IS DELETED
BMC50890I 24: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A016'...
BMC50891I 8: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A008' IS DELETED
BMC50890I 19: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00712.I0001.A001'...
BMC50890I 23: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00713.I0001.A001'...
BMC50891I 6: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A006' IS DELETED
BMC50891I 4: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A004' IS DELETED
BMC50891I 11: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A011' IS DELETED
BMC50891I 26: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A010' IS DELETED
BMC50891I 29: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A013' IS DELETED
BMC50891I 27: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A011' IS DELETED
```

Figure 19 SYSPRINT for example 2, case 2 (part 13 of 13)

```
BMC50891I 9: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A009' IS DELETED
BMC50891I 16: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A016' IS DELETED
BMC50891I 28: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A012' IS DELETED
BMC50891I 7: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A007' IS DELETED
BMC50891I 20: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A004' IS DELETED
BMC50891I 1: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A001' IS DELETED
BMC50891I 12: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A012' IS DELETED
BMC50891I 13: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A013' IS DELETED
BMC50891I 30: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A014' IS DELETED
BMC50891I 17: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A001' IS DELETED
BMC50891I 21: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A005' IS DELETED
BMC50891I 10: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A010' IS DELETED
BMC50891I 3: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A003' IS DELETED
BMC50891I 15: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A015' IS DELETED
BMC50891I 5: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A005' IS DELETED
BMC50891I 18: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A002' IS DELETED
BMC50891I 14: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A014' IS DELETED
BMC50891I 24: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A016' IS DELETED
BMC50891I 2: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A015' IS DELETED
BMC50891I 23: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00713.I0001.A001' IS DELETED
BMC50891I 19: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00712.I0001.A001' IS DELETED

BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

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

This example illustrates a partial reorganization of a 16-partition storage-group-defined table space. The table has three indexes: the clustering index, a unique nonpartitioned secondary index (NPSI), and a non-unique nonpartitioned secondary index.

In case 1, REORG PLUS reorganizes partitions 1 through 5, 8, and 16. In case 2, only partitions 1 through 5 are reorganized. This case illustrates creating a single copy of a subset of partitions. The following attributes of this job enable this copy:

- A contiguous subset of partitions is specified with the PART option.
- COPYLVL FULL is specified.
- A different options module, JG9$OPTS, is specified on the EXEC statement. This options module changes the default value for the COPYSUBSET installation option to YES.

Another difference between the two cases is the data availability during the job as specified by the SHRLEVEL option:

- Case 1 uses the default, SHRLEVEL NONE. Therefore, the data is unavailable during the job.

- Case 2 specifies SHRLEVEL REFERENCE, which allows the data to be available in read-only status during the job.

  REORG PLUS messages BMC50890I and BMC50891I detail the rename and delete activity associated with the staging data sets.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 66 describes the key command options for this job.

Table 66 Key command options used in example 3 (part 1 of 2)

Command options used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
PART 1:5, 8, 16 PART 1:5	reorganizes partitions 1 through 5, 8, and 16 (case 1) reorganizes contiguous partitions 1-5 to enable a single copy of a subset of partitions (case 2) For case 2, COPYSUBSET=YES is also set in the installation options for these copies.
COPYLVL FULL	requests a full image copy for a partitioned table space This option is required to create a copy of the subset of partitions in case 2.
COPY YES	creates DB2 image copies of table space partitions 1 through 5, 8, and 16 (case 1) creates a single image copy of table space partitions 1 through 5 (case 2) 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.
REDEFINE NO REDEFINE YES (default)	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). In case 2, the default of REDEFINE YES tells REORG PLUS to delete and redefine the VSAM data sets containing the DB2 objects.
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 a DD statement for the ASUSRPRT data set. This DD statement tells REORG PLUS not to print the statistics from the Common Statistics component to SYSPRINT, but to send them to a separate data set. Figure 24 on page 427 shows a portion of this ASUSRPRT data set for case 2.

Table 66 Key command options used in example 3 (part 2 of 2)

Command options used in JCL	Description
SHRLEVEL NONE SHRLEVEL REFERENCE	In case 1, the default SHRLEVEL NONE is used. 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 status during the job.
DDTYPE LOCBFCPY ACTIVE YES DDTYPE REMPFCPY ACTIVE YES DDTYPE REMBFCPY ACTIVE YES	activates dynamic allocation for the local backup copy and the remote copies, overriding the installation defaults

Figure 20 shows the JCL for case 1 of example 3.

Figure 20 JCL for example 3, case 1 (part 1 of 2)

```
//        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='DEDR,EXMPL03A,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//ASUSRPRT  DD  SYSOUT=*
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB007.TS007 PART 1:5, 8, 16
      COPY YES
      REDEFINE NO
      BMCSTATS YES
      DDTYPE UNLOAD
        DSNPAT 'ARU.EXMPL03A.&DDNAME'
      DDTYPE WORK
        DSNPAT 'ARU.EXMPL03A.&DDNAME'
```

Figure 20 JCL for example 3, case 1 (part 2 of 2)

```
      DDTYPE LOCPFCPY
        DSNPAT 'ARU.EXMPL03A.&DDNAME'
      DDTYPE LOCBFCPY ACTIVE YES
        DSNPAT 'ARU.EXMPL03A.&DDNAME'
      DDTYPE REMPFCPY ACTIVE YES
        DSNPAT 'ARU.EXMPL03A.&DDNAME'
      DDTYPE REMBFCPY ACTIVE YES
        DSNPAT 'ARU.EXMPL03A.&DDNAME'
/*
```

Figure 21 shows the JCL for case 2 of example 3.

Figure 21 JCL for example 3, case 2 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*   PARTITIONED TABLESPACE, 16 PARTS, 3 INDEXES, STOGROUP-DEFINED   *
//*   -CLUSTERING INDEX, UNIQUE                                       *
//*   -SECONDARY INDEX, UNIQUE                                        *
//*   -SECONDARY INDEX, NONUNIQUE                                     *
//*                                                                   *
//*   REORG PARTS 1-5,          SHRLEVEL REFERENCE                    *
//*   ONE SYSREC PER PART,                                            *
//*   REDEFINE SPACE (USING REDEFINE YES OPTION),                     *
//*   MAKE 4 COPIES OF EACH PART REORGED (LOCAL AND REMOTE COPIES),   *
//*   ONE IMAGE COPY DATASET FOR ALL PARTS (COPYSUBSET=YES).          *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='DEDR,EXMPL03B,NEW,,MSGLEVEL(1),JG9$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//ASUSRPRT  DD  SYSOUT=*
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB007.TS007 PART 1:5
      SHRLEVEL REFERENCE
      COPY YES
      COPYLVL FULL
      BMCSTATS YES
      DDTYPE WORK
        DSNPAT 'ARU.EXMPL03B.&DDNAME'
      DDTYPE LOCPFCPY
        DSNPAT 'ARU.EXMPL03B.&DDNAME'
```

Figure 21 JCL for example 3, case 2 (part 2 of 2)

```
      DDTYPE LOCBFCPY ACTIVE YES
        DSNPAT 'ARU.EXMPL03B.&DDNAME'
      DDTYPE REMPFCPY ACTIVE YES
        DSNPAT 'ARU.EXMPL03B.&DDNAME'
      DDTYPE REMBFCPY ACTIVE YES
        DSNPAT 'ARU.EXMPL03B.&DDNAME'
/*
```

Figure 22 shows the SYSPRINT output for case 1 of example 3.

Figure 22 SYSPRINT for example 3, case 1 (part 1 of 7)

```
                               *****  B M C   R E O R G   P L U S   F O R   D B 2   V10R1.00  *****
                                             (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
               REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50001I UTILITY EXECUTION STARTING    1/27/2011    14:07:43 ...
BMC50002I UTILITY ID = 'EXMPL03A'.  DB2 SUBSYSTEM ID = 'DEDR'.  OPTION MODULE = 'ARU$OPTS'.
BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
BMC50471I REGION=0M,BELOW 16M=8852K,ABOVE 16M=1410084K,IEFUSI=NO,CPUS=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

BMC50471I REORG PLUS FOR DB2--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.03.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
```

Figure 22 SYSPRINT for example 3, case 1 (part 2 of 7)

```
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
```

Figure 22 SYSPRINT for example 3, case 1 (part 3 of 7)

```
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DEDR
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)
```

Figure 22 SYSPRINT for example 3, case 1 (part 4 of 7)

```
BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'
BMC50471I ...EXCEPTIONS  ='ASU101.EXCEPTIONS2'

BMC50102I REORG TABLESPACE ARUDB007.TS007 PART 1:5, 8, 16
BMC50102I       COPY YES
BMC50102I       REDEFINE NO
BMC50102I       BMCSTATS YES
BMC50102I       DDTYPE UNLOAD
BMC50102I         DSNPAT 'ARU.EXMPL03A.&DDNAME'
BMC50102I       DDTYPE WORK
BMC50102I         DSNPAT 'ARU.EXMPL03A.&DDNAME'
BMC50102I       DDTYPE LOCPFCPY
BMC50102I         DSNPAT 'ARU.EXMPL03A.&DDNAME'
BMC50102I       DDTYPE LOCBFCPY ACTIVE YES
BMC50102I         DSNPAT 'ARU.EXMPL03A.&DDNAME'
BMC50102I       DDTYPE REMPFCPY ACTIVE YES
BMC50102I         DSNPAT 'ARU.EXMPL03A.&DDNAME'
BMC50102I       DDTYPE REMBFCPY ACTIVE YES
BMC50102I         DSNPAT 'ARU.EXMPL03A.&DDNAME'

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 7: SAMPLING STATISTICS: PART=1,TP=94,SP=92,SR=1610,AVGR=101,SD=65,SE=0,AVGF=1750,SD=65,SE=147,EP=36
BMC50482I 7: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001'
BMC51301I 6: SAMPLING STATISTICS: PART=2,TP=3,SP=1,SR=10,AVGR=133,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC50482I 6: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002'
BMC51301I 7: SAMPLING STATISTICS: PART=4,TP=3,SP=1,SR=10,AVGR=112,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC51301I 5: SAMPLING STATISTICS: PART=3,TP=3,SP=1,SR=10,AVGR=159,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC50482I 6: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005'
BMC50482I 7: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004'
BMC50482I 5: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003'
BMC50482I 4: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A008'
BMC51301I 3: SAMPLING STATISTICS: PART=16,TP=23,SP=21,SR=404,AVGR=101,SD=65,SE=0,AVGF=1923,SD=65,SE=295,EP=6
BMC50482I 3: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A016'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 2044  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I ESTIMATED CARDINALITY OF PART 0001 = 1610  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I ESTIMATED CARDINALITY OF PART 0002 = 10  AVG SORTWK ROW LENGTH = 133  AVG UNLOAD ROW LENGTH = 133
BMC50484I ESTIMATED CARDINALITY OF PART 0003 = 10  AVG SORTWK ROW LENGTH = 159  AVG UNLOAD ROW LENGTH = 159
BMC50484I ESTIMATED CARDINALITY OF PART 0004 = 10  AVG SORTWK ROW LENGTH = 112  AVG UNLOAD ROW LENGTH = 112
BMC50484I ESTIMATED CARDINALITY OF PART 0005 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I ESTIMATED CARDINALITY OF PART 0008 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC50484I ESTIMATED CARDINALITY OF PART 0016 = 404  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC51264I UNLOAD WILL READ 116 DATA PAGES FROM SPACE 'ARUDB007.TS007'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGAEXM3  &STEPNAME   BMCREORG  &DB         ARUDB007  &TSIX       TS007     &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       140743    &SSID       DEDR      &UTIL       EXMPL03A
BMC50425I &UTILPFX    EXMPL03A  &UTILSFX              &DATE8      01272011  &GRPNM      DEDR      &VCAT       DEDRCAT
BMC50425I &TIME4      1407      &DATEJ      2011027   &JDATE      11027

BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                       UNIT OR                      KBYTES   KBYTES      ALOC      ALOC
BMC50447I DDNAME   DSNAME                                       DATACLAS MGMTCLAS STORCLAS      PRI      SEC       PRI       SEC

BMC50448I BMCCPY01 ARU.EXMPL03A.BMCCPY01                        SYSALLDA                        376       37         8         1  TRK
```

Figure 22 SYSPRINT for example 3, case 1 (part 5 of 7)

```
BMC50448I BMCCPY02 ARU.EXMPL03A.BMCCPY02                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCCPY03 ARU.EXMPL03A.BMCCPY03                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCCPY04 ARU.EXMPL03A.BMCCPY04                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCCPY05 ARU.EXMPL03A.BMCCPY05                      SYSALLDA                   8       0       1        0  TRK
BMC50448I BMCCPY08 ARU.EXMPL03A.BMCCPY08                      SYSALLDA                   8       0       1        0  TRK
BMC50448I BMCCPY16 ARU.EXMPL03A.BMCCPY16                      SYSALLDA                  92       9       2        1  TRK
BMC50448I BMCCPZ01 ARU.EXMPL03A.BMCCPZ01                      SYSALLDA                 376      37       8        1  TRK
BMC50448I BMCCPZ02 ARU.EXMPL03A.BMCCPZ02                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCCPZ03 ARU.EXMPL03A.BMCCPZ03                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCCPZ04 ARU.EXMPL03A.BMCCPZ04                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCCPZ05 ARU.EXMPL03A.BMCCPZ05                      SYSALLDA                   8       0       1        0  TRK
BMC50448I BMCCPZ08 ARU.EXMPL03A.BMCCPZ08                      SYSALLDA                   8       0       1        0  TRK
BMC50448I BMCCPZ16 ARU.EXMPL03A.BMCCPZ16                      SYSALLDA                  92       9       2        1  TRK
BMC50448I BMCRCY01 ARU.EXMPL03A.BMCRCY01                      SYSALLDA                 376      37       8        1  TRK
BMC50448I BMCRCY02 ARU.EXMPL03A.BMCRCY02                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCRCY03 ARU.EXMPL03A.BMCRCY03                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCRCY04 ARU.EXMPL03A.BMCRCY04                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCRCY05 ARU.EXMPL03A.BMCRCY05                      SYSALLDA                   8       0       1        0  TRK
BMC50448I BMCRCY08 ARU.EXMPL03A.BMCRCY08                      SYSALLDA                   8       0       1        0  TRK
BMC50448I BMCRCY16 ARU.EXMPL03A.BMCRCY16                      SYSALLDA                  92       9       2        1  TRK
BMC50448I BMCRCZ01 ARU.EXMPL03A.BMCRCZ01                      SYSALLDA                 376      37       8        1  TRK
BMC50448I BMCRCZ02 ARU.EXMPL03A.BMCRCZ02                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCRCZ03 ARU.EXMPL03A.BMCRCZ03                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCRCZ04 ARU.EXMPL03A.BMCRCZ04                      SYSALLDA                  96       9       2        1  TRK
BMC50448I BMCRCZ05 ARU.EXMPL03A.BMCRCZ05                      SYSALLDA                   8       0       1        0  TRK
BMC50448I BMCRCZ08 ARU.EXMPL03A.BMCRCZ08                      SYSALLDA                   8       0       1        0  TRK
BMC50448I BMCRCZ16 ARU.EXMPL03A.BMCRCZ16                      SYSALLDA                  92       9       2        1  TRK
BMC50448I SYSREC01 ARU.EXMPL03A.SYSREC01                      SYSALLDA                 227      57       5        2  TRK
BMC50448I SYSREC02 ARU.EXMPL03A.SYSREC02                      SYSALLDA                   2       1       1        1  TRK
BMC50448I SYSREC03 ARU.EXMPL03A.SYSREC03                      SYSALLDA                   2       1       1        1  TRK
BMC50448I SYSREC04 ARU.EXMPL03A.SYSREC04                      SYSALLDA                   2       1       1        1  TRK
BMC50448I SYSREC05 ARU.EXMPL03A.SYSREC05                      SYSALLDA                   1       1       1        1  TRK
BMC50448I SYSREC08 ARU.EXMPL03A.SYSREC08                      SYSALLDA                   1       1       1        1  TRK
BMC50448I SYSREC16 ARU.EXMPL03A.SYSREC16                      SYSALLDA                  57      15       2        1  TRK
BMC50448I SYSUT101 ARU.EXMPL03A.SYSUT101                      SYSALLDA                  56       6       2        1  TRK
BMC50448I SYSUT102 ARU.EXMPL03A.SYSUT102                      SYSALLDA                  42       5       1        1  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50509I STATISTICS COLLECTION FOR INDEX SPACE ARU.TS00712 BYPASSED DUE TO PARTIAL TABLE SPACE REORG
BMC50509I STATISTICS COLLECTION FOR INDEX SPACE ARU.TS00713 BYPASSED DUE TO PARTIAL TABLE SPACE REORG
BMC50474I BELOW 16M = 8200K, ABOVE 16M = 1394988K, CPUS = 3
BMC50479I TOTAL PAGES: 2306339, ALLOWED: 0; AVAILABLE PAGES: 300086, ALLOWED: 300086
BMC51302I MAX TASKS = 7, MAX PARTITIONS PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 4: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 1 ,DDNAME = SYS00045
BMC51271I UNLOAD STATISTICS:  1610 ROWS/KEYS UNLOADED FROM PARTITION 1
BMC51281I UNLOAD STATISTICS:  X'079D4C71B457' IS THE HIGHEST LOGRBA FOR PARTITION 1
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001'
BMC50486I 3: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 4: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00047
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 4
BMC51281I UNLOAD STATISTICS:  X'079D4C65C94B' IS THE HIGHEST LOGRBA FOR PARTITION 4
BMC50481I 4: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 5 ,DDNAME = SYS00048
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 3 ,DDNAME = SYS00046
BMC51274I RELOAD STATISTICS:  1610 ROWS/KEYS LOADED INTO PARTITION 1
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004'
BMC50477I 3: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00050
BMC50476I DDNAME = BMCCPY01, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCCPZ01, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCY01, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCZ01, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 65 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPY01'
BMC50376I 65 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPZ01'
BMC50376I 65 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCY01'
BMC50376I 65 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCZ01'
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 3
BMC51281I UNLOAD STATISTICS:  X'079D4C65A990' IS THE HIGHEST LOGRBA FOR PARTITION 3
BMC50481I 3: SORT COMPLETE.  ELAPSED TIME = 00:00:01
BMC50481I 1: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:02
BMC50486I 5: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
```

Figure 22 SYSPRINT for example 3, case 1 (part 6 of 7)

```
BMC50482I 4: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A004'
BMC50477I 4: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00051
BMC50482I 4: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004'
BMC50477I 4: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00049
BMC50477I 2: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00053
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 4
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 2
BMC51281I UNLOAD STATISTICS:  X'079D4C65885D' IS THE HIGHEST LOGRBA FOR PARTITION 2
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:01
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003'
BMC50476I DDNAME = BMCCPY04, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCCPZ04, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCY04, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCZ04, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPY04'
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPZ04'
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCY04'
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCZ04'
BMC50481I 4: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:03
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002'
BMC50486I 7: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 5: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00056
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 5
BMC51281I UNLOAD STATISTICS:  X'079D4C5A367E' IS THE HIGHEST LOGRBA FOR PARTITION 5
BMC50482I 3: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A003'
BMC50477I 3: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00055
BMC50482I 3: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003'
BMC50477I 3: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00052
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 3
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A002'
BMC50477I 2: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00057
BMC50476I DDNAME = BMCCPY03, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCCPZ03, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCY03, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCZ03, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPY03'
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPZ03'
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCY03'
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCZ03'
BMC50482I 2: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002'
BMC50477I 2: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00054
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 2
BMC50486I 6: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50482I 5: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A005'
BMC50477I 5: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00058
BMC50481I 3: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:03
BMC50476I DDNAME = BMCCPY02, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCCPZ02, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCY02, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCZ02, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPY02'
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPZ02'
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCY02'
BMC50376I 4 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCZ02'
BMC50477I 7: PARTITION =   16, ROWS/KEYS = 414, I/O WAITS = 1 ,DDNAME = SYS00059
BMC50481I 2: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:04
BMC51271I UNLOAD STATISTICS:  414 ROWS/KEYS UNLOADED FROM PARTITION 16
BMC51281I UNLOAD STATISTICS:  X'079D4C7446DC' IS THE HIGHEST LOGRBA FOR PARTITION 16
BMC50481I 7: SORT COMPLETE.  ELAPSED TIME = 00:00:01
BMC50477I 6: PARTITION =    8, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00061
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005'
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 8
BMC51281I UNLOAD STATISTICS:  X'079D4C5A9B56' IS THE HIGHEST LOGRBA FOR PARTITION 8
BMC50482I 5: RELOAD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005'
BMC50477I 5: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00060
BMC51272I UNLOAD STATISTICS:  2054 ROWS UNLOADED FROM SPACE 'ARUDB007.TS007', 0 ROWS DISCARDED, 0 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'079D4C7446DC' IS THE HIGHEST LOGRBA FOR SPACE 'ARUDB007.TS007'
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 5
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A016'
BMC50476I DDNAME = BMCCPY05, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCCPZ05, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCY05, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCZ05, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPY05'
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPZ05'
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCY05'
```

Figure 22 SYSPRINT for example 3, case 1 (part 7 of 7)

```
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCZ05'
BMC50481I 5: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:05
BMC50482I 6: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A008'
BMC50477I 6: PARTITION =    8, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00063
BMC50482I 7: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A016'
BMC50477I 7: PARTITION =   16, ROWS/KEYS = 414, I/O WAITS = 5 ,DDNAME = SYS00064
BMC50482I 7: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A016'
BMC50477I 7: PARTITION =   16, ROWS/KEYS = 414, I/O WAITS = 3 ,DDNAME = SYS00062
BMC51274I RELOAD STATISTICS:  414 ROWS/KEYS LOADED INTO PARTITION 16
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A008'
BMC50476I DDNAME = BMCCPY16, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCCPZ16, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCY16, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCZ16, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 19 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPY16'
BMC50376I 19 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPZ16'
BMC50376I 19 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCY16'
BMC50376I 19 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCZ16'
BMC50482I 6: RELOAD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A008'
BMC50477I 6: PARTITION =    8, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00065
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 8
BMC50481I 7: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:05
BMC50476I DDNAME = BMCCPY08, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCCPZ08, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCY08, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCZ08, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPY08'
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCCPZ08'
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCY08'
BMC50376I 3 PAGES COPIED TO DATASET = 'ARU.EXMPL03A.BMCRCZ08'
BMC50481I 6: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:06
BMC50476I DDNAME = SYSREC01, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC02, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC03, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC04, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC05, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC08, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC16, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT101, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT102, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51276I BUILD  STATISTICS:  2054 KEYS LOADED INTO INDEX 'ARU.TS00711'
BMC51275I RELOAD STATISTICS:  2054 ROWS LOADED INTO SPACE 'ARUDB007.TS007'
BMC50474I BELOW 16M = 7948K, ABOVE 16M = 1392760K, CPUS = 3
BMC50479I TOTAL PAGES: 2307306, ALLOWED: 0; AVAILABLE PAGES: 301928, ALLOWED: 301928
BMC51303I MAX TASKS = 2, INDEXES PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 2
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50476I DDNAME = SYSUT101, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 1: UPDATE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00712.I0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 2054, I/O WAITS = 10 ,DDNAME = SYS00068
BMC51277I BUILD  STATISTICS:  2054 KEY-RID PAIRS UPDATED IN INDEX 'ARU.TS00712'
BMC50476I DDNAME = SYSUT102, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 2: UPDATE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS00713.I0001.A001'
BMC50477I 2: PARTITION =    0, ROWS/KEYS = 2054, I/O WAITS = 7 ,DDNAME = SYS00069
BMC51277I BUILD  STATISTICS:  2054 KEY-RID PAIRS UPDATED IN INDEX 'ARU.TS00713'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT102', DSNAME = 'ARU.EXMPL03A.SYSUT102'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT101', DSNAME = 'ARU.EXMPL03A.SYSUT101'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC16', DSNAME = 'ARU.EXMPL03A.SYSREC16'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC08', DSNAME = 'ARU.EXMPL03A.SYSREC08'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC05', DSNAME = 'ARU.EXMPL03A.SYSREC05'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC04', DSNAME = 'ARU.EXMPL03A.SYSREC04'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC03', DSNAME = 'ARU.EXMPL03A.SYSREC03'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC02', DSNAME = 'ARU.EXMPL03A.SYSREC02'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC01', DSNAME = 'ARU.EXMPL03A.SYSREC01'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I REORG PHASE COMPLETE.  ELAPSED TIME = 00:00:17

BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Figure 23 shows the SYSPRINT output for case 2 of example 3.

Figure 23 SYSPRINT for example 3, case 2 (part 1 of 7)

```
                               *****  B M C   R E O R G   P L U S   F O R   D B 2   V10R1.00  *****
                                             (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
               REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
 BMC50001I UTILITY EXECUTION STARTING    1/27/2011   14:14:06 ...
 BMC50002I UTILITY ID = 'EXMPL03B'.  DB2 SUBSYSTEM ID = 'DEDR'.  OPTION MODULE = 'JG9$OPTS'.
 BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
 BMC50471I REGION=0M,BELOW 16M=8860K,ABOVE 16M=1410260K,IEFUSI=NO,CPUS=3
 BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

 BMC50471I REORG PLUS FOR DB2--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I SOLUTION COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMCSORT ENGINE--V02.03.01
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMC STATS API--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT

 BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
 BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
 BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
 BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
 BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
 BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
 BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
 BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
 BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
 BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
 BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
 BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
 BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
 BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
 BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
 BMC50471I COPYSUBSET=YES                          LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
 BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
 BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
 BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
 BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
 BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
 BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
 BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
 BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
 BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
 BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
 BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
 BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
 BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
 BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
 BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
 BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
 BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
 BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
 BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
 BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
 BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
 BMC50471I IDCACHE=10000                           SCPYMAX=8

 BMC50471I PLAN=ARUQA

 BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
 BMC50470I ACTIVE   = YES                                YES                                  NO
 BMC50470I IFALLOC  = USE                                USE                                  USE
 BMC50470I ALLOC    = N/A                                N/A                                  ANY
 BMC50470I SMS      = NO                                 NO                                   NO
 BMC50470I SMSUNIT  = NO                                 NO                                   NO
 BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
 BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
 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
```

Figure 23 SYSPRINT for example 3, case 2 (part 2 of 7)

```
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO
```

Figure 23 SYSPRINT for example 3, case 2 (part 3 of 7)

```
BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DEDR
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
```

Figure 23 SYSPRINT for example 3, case 2 (part 4 of 7)

```
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'
BMC50471I ...EXCEPTIONS  ='ASU101.EXCEPTIONS2'

BMC50102I REORG TABLESPACE ARUDB007.TS007 PART 1:5
BMC50102I       SHRLEVEL REFERENCE
BMC50102I       COPY YES
BMC50102I       COPYLVL FULL
BMC50102I       BMCSTATS YES
BMC50102I       DDTYPE WORK
BMC50102I         DSNPAT 'ARU.EXMPL03B.&DDNAME'
BMC50102I       DDTYPE LOCPFCPY
BMC50102I         DSNPAT 'ARU.EXMPL03B.&DDNAME'
BMC50102I       DDTYPE LOCBFCPY ACTIVE YES
BMC50102I         DSNPAT 'ARU.EXMPL03B.&DDNAME'
BMC50102I       DDTYPE REMPFCPY ACTIVE YES
BMC50102I         DSNPAT 'ARU.EXMPL03B.&DDNAME'
BMC50102I       DDTYPE REMBFCPY ACTIVE YES
BMC50102I         DSNPAT 'ARU.EXMPL03B.&DDNAME'

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:02

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 5: SAMPLING STATISTICS: PART=1,TP=64,SP=62,SR=1610,AVGR=101,SD=65,SE=0,AVGF=2596,SD=65,SE=110,EP=6
BMC50482I 5: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001'
BMC51301I 4: SAMPLING STATISTICS: PART=2,TP=3,SP=1,SR=10,AVGR=133,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC50482I 4: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002'
BMC51301I 3: SAMPLING STATISTICS: PART=3,TP=3,SP=1,SR=10,AVGR=159,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC51301I 5: SAMPLING STATISTICS: PART=4,TP=3,SP=1,SR=10,AVGR=112,SD=64,SE=0,AVGF=1000,SD=64,SE=0,EP=0
BMC50482I 3: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003'
BMC50482I 5: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004'
BMC50482I 2: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 1640  AVG SORTWK ROW LENGTH = 102  AVG UNLOAD ROW LENGTH = 102
BMC50484I ESTIMATED CARDINALITY OF PART 0001 = 1610  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I ESTIMATED CARDINALITY OF PART 0002 = 10  AVG SORTWK ROW LENGTH = 133  AVG UNLOAD ROW LENGTH = 133
BMC50484I ESTIMATED CARDINALITY OF PART 0003 = 10  AVG SORTWK ROW LENGTH = 159  AVG UNLOAD ROW LENGTH = 159
BMC50484I ESTIMATED CARDINALITY OF PART 0004 = 10  AVG SORTWK ROW LENGTH = 112  AVG UNLOAD ROW LENGTH = 112
BMC50484I ESTIMATED CARDINALITY OF PART 0005 = 0  AVG SORTWK ROW LENGTH = 0  AVG UNLOAD ROW LENGTH = 0
BMC51264I UNLOAD WILL READ 65 DATA PAGES FROM SPACE 'ARUDB007.TS007'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGBEXM3  &STEPNAME   BMCREORG  &DB         ARUDB007  &TSIX       TS007     &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       141406    &SSID       DEDR      &UTIL       EXMPL03B
BMC50425I &UTILPFX    EXMPL03B  &UTILSFX              &DATE8      01272011  &GRPNM      DEDR      &VCAT       DEDRCAT
BMC50425I &TIME4      1414      &DATEJ      2011027   &JDATE      11027

BMC50877I POINT OF CONSISTENCY ESTABLISHED AT RBA/LRSN = 079D4C965F46
BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                       UNIT OR                      KBYTES   KBYTES     ALOC      ALOC
BMC50447I DDNAME   DSNAME                                       DATACLAS MGMTCLAS STORCLAS      PRI      SEC      PRI       SEC

BMC50448I BMCCPY   ARU.EXMPL03B.BMCCPY                          SYSALLDA                        300       30        7         1  TRK
BMC50448I BMCCPZ   ARU.EXMPL03B.BMCCPZ                          SYSALLDA                        300       30        7         1  TRK
BMC50448I BMCRCY   ARU.EXMPL03B.BMCRCY                          SYSALLDA                        300       30        7         1  TRK
BMC50448I BMCRCZ   ARU.EXMPL03B.BMCRCZ                          SYSALLDA                        300       30        7         1  TRK
BMC50448I SYSUT101 ARU.EXMPL03B.SYSUT101                        SYSALLDA                         45        5        1         1  TRK
BMC50448I SYSUT102 ARU.EXMPL03B.SYSUT102                        SYSALLDA                         34        4        1         1  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50509I STATISTICS COLLECTION FOR INDEX SPACE ARU.TS00712 BYPASSED DUE TO PARTIAL TABLE SPACE REORG
BMC50509I STATISTICS COLLECTION FOR INDEX SPACE ARU.TS00713 BYPASSED DUE TO PARTIAL TABLE SPACE REORG
BMC50474I BELOW 16M = 8216K, ABOVE 16M = 1399752K, CPUS = 3
BMC50388I 18: COPY OF SECONDARY INDEX: 'DEDRCAT.DSNDBD.ARUDB007.TS00712.I0001.A001' STARTED
BMC50388I 17: COPY OF SECONDARY INDEX: 'DEDRCAT.DSNDBD.ARUDB007.TS00713.I0001.A001' STARTED
BMC50389I 18: COPY OF SECONDARY INDEX: 'DEDRCAT.DSNDBD.ARUDB007.TS00712.I0001.A001' COMPLETE
BMC50389I 17: COPY OF SECONDARY INDEX: 'DEDRCAT.DSNDBD.ARUDB007.TS00713.I0001.A001' COMPLETE
BMC50479I TOTAL PAGES: 2318650, ALLOWED: 0; AVAILABLE PAGES: 314618, ALLOWED: 314618
BMC51302I MAX TASKS = 5, MAX PARTITIONS PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50486I 3: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 3: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00040
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 3
BMC51281I UNLOAD STATISTICS:  X'079D4C7780D2' IS THE HIGHEST LOGRBA FOR PARTITION 3
BMC50481I 3: SORT COMPLETE.  ELAPSED TIME = 00:00:00
```

Figure 23 SYSPRINT for example 3, case 2 (part 5 of 7)

```
BMC50486I 5: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 2: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00042
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 2
BMC51281I UNLOAD STATISTICS:  X'079D4C7780D2' IS THE HIGHEST LOGRBA FOR PARTITION 2
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50486I 4: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 5: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 1 ,DDNAME = SYS00047
BMC51271I UNLOAD STATISTICS:  0 ROWS/KEYS UNLOADED FROM PARTITION 5
BMC51281I UNLOAD STATISTICS:  X'079D4C7780D2' IS THE HIGHEST LOGRBA FOR PARTITION 5
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A002'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A003'
BMC50477I 4: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 1 ,DDNAME = SYS00049
BMC51271I UNLOAD STATISTICS:  10 ROWS/KEYS UNLOADED FROM PARTITION 4
BMC51281I UNLOAD STATISTICS:  X'079D4C7780D2' IS THE HIGHEST LOGRBA FOR PARTITION 4
BMC50481I 4: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 1 ,DDNAME = SYS00057
BMC51271I UNLOAD STATISTICS:  1610 ROWS/KEYS UNLOADED FROM PARTITION 1
BMC51281I UNLOAD STATISTICS:  X'079D4C7780D2' IS THE HIGHEST LOGRBA FOR PARTITION 1
BMC51272I UNLOAD STATISTICS:  1640 ROWS UNLOADED FROM SPACE 'ARUDB007.TS007', 0 ROWS DISCARDED, 0 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'079D4C7780D2' IS THE HIGHEST LOGRBA FOR SPACE 'ARUDB007.TS007'
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A004'
BMC50482I 5: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A005'
BMC50477I 5: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 5 ,DDNAME = SYS00055
BMC50482I 3: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A003'
BMC50477I 3: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00063
BMC50482I 3: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A003'
BMC50477I 3: PARTITION =    3, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00045
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 3
BMC50481I 3: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:02
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A002'
BMC50477I 2: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00073
BMC50482I 2: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A002'
BMC50477I 2: PARTITION =    2, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00048
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 2
BMC50481I 2: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:03
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A005'
BMC50482I 5: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A005'
BMC50477I 5: PARTITION =    5, ROWS/KEYS = 0, I/O WAITS = 3 ,DDNAME = SYS00087
BMC51274I RELOAD STATISTICS:  0 ROWS/KEYS LOADED INTO PARTITION 5
BMC50481I 5: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:03
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A001'
BMC50482I 4: BUILD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A004'
BMC50477I 4: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 5 ,DDNAME = SYS00095
BMC50482I 4: RELOAD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A004'
BMC50477I 4: PARTITION =    4, ROWS/KEYS = 10, I/O WAITS = 3 ,DDNAME = SYS00064
BMC51274I RELOAD STATISTICS:  10 ROWS/KEYS LOADED INTO PARTITION 4
BMC50481I 4: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:04
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 5 ,DDNAME = SYS00103
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1610, I/O WAITS = 3 ,DDNAME = SYS00090
BMC51274I RELOAD STATISTICS:  1610 ROWS/KEYS LOADED INTO PARTITION 1
BMC50481I 1: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:05
BMC50476I DDNAME = SYSUT101, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT102, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51276I BUILD  STATISTICS:  1640 KEYS LOADED INTO INDEX 'ARU.TS00711'
BMC51275I RELOAD STATISTICS:  1640 ROWS LOADED INTO SPACE 'ARUDB007.TS007'
BMC50474I BELOW 16M = 8148K, ABOVE 16M = 1398528K, CPUS = 3
BMC50479I TOTAL PAGES: 2316063, ALLOWED: 0; AVAILABLE PAGES: 314325, ALLOWED: 314325
BMC51303I MAX TASKS = 2, INDEXES PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 2
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50476I DDNAME = SYSUT101, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 1: UPDATE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00712.I0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 1640, I/O WAITS = 6 ,DDNAME = SYS00106
BMC51277I BUILD  STATISTICS:  1640 KEY-RID PAIRS UPDATED IN INDEX 'ARU.TS00712'
BMC50476I DDNAME = SYSUT102, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 2: UPDATE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.BMCDBD.ARUDB007.TS00713.I0001.A001'
BMC50477I 2: PARTITION =    0, ROWS/KEYS = 1640, I/O WAITS = 5 ,DDNAME = SYS00107
BMC51277I BUILD  STATISTICS:  1640 KEY-RID PAIRS UPDATED IN INDEX 'ARU.TS00713'
BMC50476I DDNAME = BMCCPY, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCCPZ, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
```

Figure 23 SYSPRINT for example 3, case 2 (part 6 of 7)

```
BMC50476I DDNAME = BMCRCY, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCZ, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 80 PAGES COPIED TO DATASET = 'ARU.EXMPL03B.BMCCPY'
BMC50376I 80 PAGES COPIED TO DATASET = 'ARU.EXMPL03B.BMCCPZ'
BMC50376I 80 PAGES COPIED TO DATASET = 'ARU.EXMPL03B.BMCRCY'
BMC50376I 80 PAGES COPIED TO DATASET = 'ARU.EXMPL03B.BMCRCZ'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT102', DSNAME = 'ARU.EXMPL03B.SYSUT102'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT101', DSNAME = 'ARU.EXMPL03B.SYSUT101'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I REORG PHASE COMPLETE.  ELAPSED TIME = 00:00:12

BMC50894I RENAME PROCESS STARTING AT   1/27/2011  14:14:22
BMC50890I 12: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00713.I0001.A001'...
BMC50890I 10: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A005'...
BMC50890I 9: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A004'...
BMC50890I 7: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A002'...
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A005'...
BMC50890I 6: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A001'...
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A003'...
BMC50890I 2: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A002'...
BMC50891I 12: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00713.I0001.A001' IS RENAMED
BMC50890I 8: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A003'...
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A001'...
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A004'...
BMC50890I 11: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00712.I0001.A001'...
BMC50891I 10: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A005' IS RENAMED
BMC50891I 9: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50891I 7: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A002' IS RENAMED
BMC50890I 12: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00713.I0001.A001'...
BMC50890I 9: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A004'...
BMC50890I 10: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A005'...
BMC50891I 3: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A003' IS RENAMED
BMC50890I 7: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A002'...
BMC50891I 4: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A004' IS RENAMED
BMC50891I 8: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A003' IS RENAMED
BMC50891I 11: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00712.I0001.A001' IS RENAMED
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003'...
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004'...
BMC50891I 2: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A002' IS RENAMED
BMC50891I 1: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A001' IS RENAMED
BMC50891I 5: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS007.I0001.A005' IS RENAMED
BMC50891I 6: DATASET 'DEDRCAT.DSNDBC.ARUDB007.TS00711.I0001.A001' IS RENAMED
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001'...
BMC50890I 11: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00712.I0001.A001'...
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005'...
BMC50890I 8: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A003'...
BMC50890I 2: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002'...
BMC50890I 6: ATTEMPTING TO RENAME DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A001'...
BMC50891I 9: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50890I 9: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A004'...
BMC50891I 10: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A005' IS RENAMED
BMC50890I 10: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A005'...
BMC50891I 12: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00713.I0001.A001' IS RENAMED
BMC50890I 12: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00713.I0001.A001'...
BMC50891I 4: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A004' IS RENAMED
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A004'...
BMC50891I 1: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A001' IS RENAMED
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A001'...
BMC50891I 8: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A003' IS RENAMED
BMC50890I 8: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A003'...
BMC50891I 2: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A002' IS RENAMED
BMC50890I 2: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A002'...
BMC50891I 3: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A003' IS RENAMED
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A003'...
BMC50891I 7: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A002' IS RENAMED
BMC50890I 7: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A002'...
BMC50891I 11: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00712.I0001.A001' IS RENAMED
BMC50890I 11: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00712.I0001.A001'...
BMC50891I 5: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS007.I0001.A005' IS RENAMED
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A005'...
BMC50891I 9: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50890I 9: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A004'...
BMC50891I 8: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A003' IS RENAMED
BMC50890I 8: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A003'...
BMC50891I 11: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00712.I0001.A001' IS RENAMED
BMC50890I 11: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00712.I0001.A001'...
```

Figure 23 SYSPRINT for example 3, case 2 (part 7 of 7)

```
BMC50891I 4: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A004' IS RENAMED
BMC50890I 4: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A004'...
BMC50891I 12: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00713.I0001.A001' IS RENAMED
BMC50890I 12: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00713.I0001.A001'...
BMC50891I 7: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A002' IS RENAMED
BMC50890I 7: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A002'...
BMC50891I 3: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A003' IS RENAMED
BMC50890I 3: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A003'...
BMC50891I 2: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A002' IS RENAMED
BMC50890I 2: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A002'...
BMC50891I 6: DATASET 'DEDRCAT.DSNDBD.ARUDB007.TS00711.I0001.A001' IS RENAMED
BMC50890I 6: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A001'...
BMC50891I 1: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A001' IS RENAMED
BMC50890I 1: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A001'...
BMC50891I 10: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A005' IS RENAMED
BMC50890I 10: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A005'...
BMC50891I 5: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS007.I0001.A005' IS RENAMED
BMC50890I 5: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A005'...
BMC50891I 6: DATASET 'DEDRCAT.BMCDBC.ARUDB007.TS00711.I0001.A001' IS RENAMED
BMC50890I 6: ATTEMPTING TO RENAME DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A001'...
BMC50891I 11: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00712.I0001.A001' IS RENAMED
BMC50891I 10: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A005' IS RENAMED
BMC50891I 2: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A002' IS RENAMED
BMC50891I 9: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A004' IS RENAMED
BMC50891I 8: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A003' IS RENAMED
BMC50891I 4: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A004' IS RENAMED
BMC50891I 7: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A002' IS RENAMED
BMC50891I 3: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A003' IS RENAMED
BMC50891I 1: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A001' IS RENAMED
BMC50891I 12: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00713.I0001.A001' IS RENAMED
BMC50891I 5: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS007.I0001.A005' IS RENAMED
BMC50891I 6: DATASET 'DEDRCAT.BMCDBD.ARUDB007.TS00711.I0001.A001' IS RENAMED
BMC50895I RENAME PROCESS COMPLETE.  ELAPSED TIME =  00:00:01

BMC50890I 7: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A002'...
BMC50890I 10: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A005'...
BMC50890I 2: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A002'...
BMC50890I 8: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A003'...
BMC50890I 9: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A004'...
BMC50890I 1: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A001'...
BMC50890I 3: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A003'...
BMC50890I 5: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A005'...
BMC50890I 6: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A001'...
BMC50890I 4: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A004'...
BMC50890I 12: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00713.I0001.A001'...
BMC50890I 11: ATTEMPTING TO DELETE DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00712.I0001.A001'...
BMC50891I 7: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A002' IS DELETED
BMC50891I 2: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A002' IS DELETED
BMC50891I 10: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A005' IS DELETED
BMC50891I 8: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A003' IS DELETED
BMC50891I 9: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A004' IS DELETED
BMC50891I 5: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A005' IS DELETED
BMC50891I 3: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A003' IS DELETED
BMC50891I 1: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A001' IS DELETED
BMC50891I 6: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00711.I0001.A001' IS DELETED
BMC50891I 4: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS007.I0001.A004' IS DELETED
BMC50891I 12: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00713.I0001.A001' IS DELETED
BMC50891I 11: DATASET 'DEDRCAT.OLDDBC.ARUDB007.TS00712.I0001.A001' IS DELETED

BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Figure 24 on page 427 shows a portion of the statistics report sent to ASUSRPRT for case 2 of example 3.

This figure shows one set of statistics for each object type. Ellipses (...) at the beginning of a line indicate the omitted statistics sections.

Figure 24 ASUSRPRT for example 3, case 2 (part 1 of 2)

```
BMCSTATS V10.1.0 REPORT FOR DEDR V910              TIME 2011-01-27-14.43.27.423218                                            1


  INDEX --------------- ARU.TS00711
    ASSOCIATIONS
      TABLESPACE ------ ARUDB007.TS007
      TABLE ----------- ARU.TS0071
    HISTORY
      SAVESTATS--------------------Y  UPDATEDB2--------------------N  UTILCODE----------------(NULL)  LOCATION------------------DEDR
      SAMPLING---------------------N
    ATTRIBUTES
      PIECESIZE--------------------0  TYPE TS----------------(BLANK)  UNIQUERULE-------------------U  KEYLENGTH-------------------34
      SUBPAGES---------------------0  INDEXTYPE--------------------2  CLUSTERING-------------------Y  COLCOUNT---------------------3
      INDEXSPACE-------------TS00711  PGSIZE-----------------------4  EXTTYPE----------------(BLANK)  COMPRESS---------------------N
      COLNAME------------EMPHIREDATE
    STATISTICS
      FIRSTKEYCARD----------------89  FULLKEYCARD---------------2084  NACTIVE--------------------105  TBCARDF-------------------2084
      NUMNONLEAF------------------16  NLEVELS----------------------2  CLUSTERRATIO---------------100  NLEAF-----------------------31
      AVGKEYLEN-------------------34  REORGSPACE-----------------212  REORGSPACE_KB------------10176  REORGNLEVELS-----------------2
      HIGH2K-----------X'0022000101'  HIGH2K (CHAR)------------.....  LOW2K------------X'0014690308'  LOW2K (CHAR)-------------.....
      DATAREPEAT------------------77
    ALLOCATION
      SPACE----------------------212  SPACE_KB-----------------10176

  INDEXPART ----------- ARU.TS00711 PART 001
    ASSOCIATIONS
      TABLESPACE ------ ARUDB007.TS007
      TABLE ----------- ARU.TS0071
    HISTORY
      SAVESTATS--------------------Y  UPDATEDB2--------------------N  UTILCODE----------------(NULL)  LOCATION----------------(NULL)
      SAMPLING---------------------N
    ATTRIBUTES
      PQTY-------------------------1  SQTY-------------------------1  PCTFREE---------------------15  FREEPAGE---------------------9
      ALLOCUNIT--------------------C  STORTYPE---------------------I  NLAVGKEYLEN-----------------15  VCATNAME---------------DEDRCAT
      STORNAME--------------ARULARSG
    STATISTICS
      FIRSTKEYCARD----------------17  FULLKEYCARD---------------1610  CARD----------------------1610  NACTIVE---------------------26
      CLUSTERRATIO---------------100  NUMNONLEAF-------------------1  NLEVELS----------------------2  FAROFFPOS--------------------0
      NEAROFFPOS-------------------6  LEAFDIST--------------------15  NLEAF-----------------------20  FREE------------------------17
      FULL-------------------------0  AVGKEYLEN-------------------34
      LEAFNEAR---------------------3  LEAFFAR----------------------0  PSEUDO_DEL_RIDS--------------0  REORGSPACE------------------15
      REORGSPACE_KB--------------720  REORGNLEVELS-----------------2  PCTUSED--------------------100  PQTYROWS-----------------13114
      DATAREPEAT------------------56
    ALLOCATION
      IPREFIX----------------------I  SPACE-----------------------15  SPACE_KB-------------------720  NUMDATASETS------------------1
      EXTENTS----------------------1  VOLCOUNT---------------------1  DEVTYPE-------------------3390  VOLUME------------------TM0002
...

  COLUMN -------------- EMPHIREDATE
    ASSOCIATIONS
      TABLESPACE ------ ARUDB007.TS007
      TABLE ----------- ARU.TS0071
    ATTRIBUTES
      COLNO------------------------6  COLTYPE-------------------DATE  LENGTH-----------------------4  NULLS------------------------Y
      SCALE------------------------0
    STATISTICS
      COLCARD---------------------89  COLNULLS---------------------0
BMCSTATS V10.1.0 REPORT FOR DEDR V910              TIME 2011-01-24-15.43.27.423218                                            8

      COLAVG-----------------------5  COLMIN-----------------------5  COLMAX-----------------------5
      HIGH2K-----------X'0022000101'  HIGH2K (CHAR)------------.....  LOW2K------------X'0014690308'  LOW2K (CHAR)-------------.....

      MOST FREQUENT VALUES
      FREQUENCY-------------0.383877  COLVAL-----------X'0014881209'
                                      (CHAR)-------------------..h..
      FREQUENCY-------------0.195777  COLVAL-----------X'0022000101'
                                      (CHAR)-------------------.....
      FREQUENCY-------------0.191939  COLVAL-----------X'0014030101'
                                      (CHAR)-------------------.....
      FREQUENCY-------------0.191939  COLVAL-----------X'0014690308'
                                      (CHAR)-------------------.....
      FREQUENCY-------------0.191939  COLVAL-----------X'0014721112'
                                      (CHAR)-------------------.....
      FREQUENCY-------------0.191939  COLVAL-----------X'0014820101'
                                      (CHAR)-------------------..b..
```

Figure 24 ASUSRPRT for example 3, case 2 (part 2 of 2)

```
      FREQUENCY-------------0.191939  COLVAL-----------X'0014820707'
                                      (CHAR)-------------------..b..
      FREQUENCY-------------0.191939  COLVAL-----------X'0014850131'
                                      (CHAR)-------------------..e..
      FREQUENCY-------------0.191939  COLVAL-----------X'0022000228'
                                      (CHAR)-------------------.....
      FREQUENCY-------------0.000960  COLVAL-----------X'0019900324'
                                      (CHAR)-------------------.....
...

  COLUMN -------------- EMPHIREDATE PART 005
    ATTRIBUTES
      COLNO------------------------6  COLTYPE-------------------DATE  LENGTH-----------------------4  NULLS------------------------Y
      SCALE------------------------0
    STATISTICS
      COLCARD----------------------0  COLNULLS---------------------0
      COLAVG-----------------------5  COLMIN-----------------------5  COLMAX-----------------------5
      HIGH2K---------------------X''  HIGH2K (CHAR)-----------(NULL)  LOW2K----------------------X''  LOW2K (CHAR)------------(NULL)
      HIGHK----------------------X''  HIGHK (CHAR)------------(NULL)  LOWK-----------------------X''  LOWK (CHAR)-------------(NULL)
...
BMCSTATS V10.1.0 REPORT FOR DEDR V910              TIME 2011-01-24-15.43.27.423218                                             1


TABLESPACE ------------ ARUDB007.TS007
    HISTORY
      SAVESTATS--------------------Y  UPDATEDB2--------------------N  UTILCODE----------------(NULL)  LOCATION------------------DEDR
      SAMPLING---------------------N
    ATTRIBUTES
      PARTITIONS------------------16  NTABLES----------------------1  PGSIZE-----------------------4  SEGSIZE----------------------0
      DSSIZE-----------------------0
      MAXROWS--------------------255  TYPE-------------------(BLANK)  ENCODING---------------------E  MAXPARTITIONS----------------0
      LOB--------------------------N  XML--------------------------N  MEMBER_CLUSTER---------(BLANK)
    STATISTICS
      NACTIVE--------------------116  REORGSPACE-----------------213  REORGSPACE_KB------------10224  ROWMAXFOUND----------------165
      ROWMINFOUND----------------107  ROWAVG---------------------103
    ALLOCATION
      SPACE----------------------235  SPACE_KB-----------------11280

  TABLEPART ----------- ARUDB007.TS007    PART 001
    HISTORY
      SAVESTATS--------------------Y  UPDATEDB2--------------------N  LOCATION----------------(NULL)  UTILCODE----------------(NULL)
      SAMPLING---------------------N
    ATTRIBUTES
      PQTY-------------------------2  SQTY-------------------------1  PCTFREE---------------------20  FREEPAGE---------------------9
      ALLOCUNIT--------------------C  STORTYPE---------------------I  COMPRESS---------------(BLANK)  TRACKMOD---------------(BLANK)
      FORMAT---------------------RRF  VCATNAME---------------DEDRCAT  STORNAME--------------ARULARSG
    STATISTICS
      CARD----------------------1610  NACTIVE---------------------64  NPAGES----------------------56  ROWAVG---------------------102
      ROWMAXFOUND----------------118  ROWMINFOUND----------------107  AVGNONCOMPROWLEN----------(-1)  DIRTY------------------------0
      FULL-------------------------0  FARINDREF--------------------0  NEARINDREF-------------------0  PERCACTIVE------------------65
      PCTPAGES--------------------87  PERCDROP---------------------0  REORGSPACE------------------15  REORGSPACE_KB--------------720
      PCTUSED---------------------50  PQTYROWS------------------9367
    ALLOCATION
      IPREFIX----------------------I  SPACE-----------------------30  SPACE_KB------------------1440  NUMDATASETS------------------1
      EXTENTS----------------------1  VOLCOUNT---------------------1  DEVTYPE-------------------3390  VOLUME------------------TM0002
...

  TABLE --------------- ARU.TS0071
    ASSOCIATIONS
      TABLESPACE ------ ARUDB007.TS007
    HISTORY
      SAVESTATS--------------------Y  UPDATEDB2--------------------N  UTILCODE----------------(NULL)  LOCATION----------------(NULL)
    STATISTICS
      CARD----------------------2084  NPAGES----------------------77  SPACE_KB-----------------11280  AVGNONCOMPROWLEN----------(-1)
      ROWAVG---------------------103  ROWMAXFOUND----------------165  ROWMINFOUND----------------107  INDREF-----------------------0
      PCTPAGES--------------------66  PCTROWCOMP-------------------0
```

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.

Table 67 describes the key command options for this job.

Table 67 Key command options used in example 4

Command options used in JCL	Description
REORG INDEX	specifies that the index named in the statement is to be reorganized
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.

Figure 25 shows the JCL for example 4.

Figure 25 JCL for example 4 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  SEGMENTED TABLESPACE, ONE INDEX (NONUNIQUE, NONCLUSTERED)        *
//*                                                                   *
//*  REORG INDEX ONLY                                                 *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='DHV,EXMPL04,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//*
//SYSIN     DD  *
REORG INDEX ARU.TS00211
REDEFINE NO
```

Figure 25 JCL for example 4 (part 2 of 2)

```
DDTYPE WORK
  DSNPAT 'ARU.EXMPL04.&DDNAME'
/*
```

Figure 26 shows the SYSPRINT output for example 4.

Figure 26 SYSPRINT for example 4 (part 1 of 4)

```
                               *****  B M C   R E O R G   P L U S   F O R   D B 2   V10R1.00  *****
                                             (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
               REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50001I UTILITY EXECUTION STARTING    1/27/2011   14:19:33 ...
BMC50002I UTILITY ID = 'EXMPL04'.  DB2 SUBSYSTEM ID = 'DHV'.  OPTION MODULE = 'ARU$OPTS'.
BMC50024I CONNECTED TO SUBSYSTEM 'DHV2'
BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
BMC50471I REGION=0M,BELOW 16M=8852K,ABOVE 16M=1410276K,IEFUSI=NO,CPUS=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

BMC50471I REORG PLUS FOR DB2--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.03.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA
```

Figure 26 SYSPRINT for example 4 (part 2 of 4)

```
BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
```

Figure 26 SYSPRINT for example 4 (part 3 of 4)

```
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DHV
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
```

Figure 26 SYSPRINT for example 4 (part 4 of 4)

```
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'

BMC50102I REORG INDEX ARU.TS00211
BMC50102I REDEFINE NO
BMC50102I DDTYPE WORK
BMC50102I   DSNPAT 'ARU.EXMPL04.&DDNAME'

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:02

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.ARUDB002.TS00211.I0001.A001'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 10000
BMC51264I UNLOAD WILL READ 74 DATA PAGES FROM SPACE 'ARUDB002.TS00211'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGAEXM4  &STEPNAME   BMCREORG  &DB         ARUDB002  &TSIX       TS00211   &RTYPE      IX
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       141933    &SSID       DHV2      &UTIL       EXMPL04
BMC50425I &UTILPFX    EXMPL04   &UTILSFX              &DATE8      01272011  &GRPNM      DHV       &VCAT       DSNDHV
BMC50425I &TIME4      1419      &DATEJ      2011027   &JDATE      11027

BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                       UNIT OR                      KBYTES   KBYTES     ALOC      ALOC
BMC50447I DDNAME   DSNAME                                       DATACLAS MGMTCLAS STORCLAS      PRI      SEC      PRI       SEC

BMC50448I SYSUT1   ARU.EXMPL04.SYSUT1                           SYSALLDA                        196       20        4         1  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50474I BELOW 16M = 8460K, ABOVE 16M = 1403660K, CPUS = 3
BMC50479I TOTAL PAGES: 2302985, ALLOWED: 0; AVAILABLE PAGES: 305500, ALLOWED: 305500
BMC51302I MAX TASKS = 1, MAX PARTITIONS PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 10000, I/O WAITS = 1 ,DDNAME = SYS00004
BMC51278I UNLOAD STATISTICS:  10000 KEYS UNLOADED FROM INDEX 'ARU.TS00211'
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.ARUDB002.TS00211.I0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 10000, I/O WAITS = 5 ,DDNAME = SYS00006
BMC50481I 1: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:01
BMC50476I DDNAME = SYSUT1, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51276I BUILD  STATISTICS:  10000 KEYS LOADED INTO INDEX 'ARU.TS00211'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT1  ', DSNAME = 'ARU.EXMPL04.SYSUT1'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I REORG PHASE COMPLETE.  ELAPSED TIME = 00:00:06

BMC50501I DB2 OBJECT STATISTICS
BMC50525I     INDEX   ARU.TS00211
BMC50526I             FIRST KEY COLUMN = EMPHIREDATE
BMC50527I             CLUSTER = N   UNIQUE = D   COMPRESS = N   PAGESIZE = 4K   KEYLEN =   14   COLCOUNT = 3
BMC50528I             PART   NACTIVE       CARD   EXTENTS   LEVELS
BMC50529I                0        76      10000         1        2
BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Example 5: Partial reorganization with UNLOAD PAUSE

In this example, REORG PLUS reorganizes partition 1 of a partitioned table space. The table has one clustering index and two nonpartitioned secondary indexes.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 68 describes the key command options for this job.

Table 68 Key command options used in example 5

Command options used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
PART 01	limits the reorganization to partition 1
UNLOAD PAUSE	instructs REORG PLUS to pause after the UNLOAD phase The index space taking part in the reorganization will be stopped at the end of unload processing. REORG PLUS completes with return code 4.
BMCSTATS YES UPDATEDB2STATS YES	tells REORG PLUS to update the statistics in the DASD MANAGER PLUS database statistics tables and the DB2 catalog REORG PLUS does not save the statistics in this case, because the job is paused.

Figure 27 shows the JCL for example 5.

Figure 27 JCL for example 5 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  PARTITIONED TABLESPACE, ONE CLUSTERING INDEX, TWO SECONDARY      *
//*  INDEXES REORG UNLOAD PAUSE PART 01 OF PARTITIONED TABLESPACE.    *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='DEDR,EXMPL05,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
```

Figure 27 JCL for example 5 (part 2 of 2)

```
//*
//SYSIN     DD   *
REORG TABLESPACE ARUDB032.TS032 PART 01
UNLOAD PAUSE
BMCSTATS YES
UPDATEDB2STATS YES
DDTYPE UNLOAD
  DSNPAT 'ARU.EXMPL05.&DDNAME'
DDTYPE WORK
  DSNPAT 'ARU.EXMPL05.&DDNAME'
/*
```

Figure 28 shows the SYSPRINT output for example 5.

Figure 28 SYSPRINT for example 5 (part 1 of 5)

```
                                *****  B M C   R E O R G   P L U S   F O R   D B 2   V10R1.00  *****
                                              (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
                REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
 BMC50001I UTILITY EXECUTION STARTING    1/27/2011   14:24:12 ...
 BMC50002I UTILITY ID = 'EXMPL05'.  DB2 SUBSYSTEM ID = 'DEDR'.  OPTION MODULE = 'ARU$OPTS'.
 BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
 BMC50471I REGION=0M,BELOW 16M=8852K,ABOVE 16M=1409032K,IEFUSI=NO,CPUS=3
 BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

 BMC50471I REORG PLUS FOR DB2--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I SOLUTION COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMCSORT ENGINE--V02.03.01
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMC STATS API--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT

 BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
 BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
 BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
 BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
 BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
 BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
 BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
 BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
 BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
 BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
 BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
 BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
 BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
 BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
 BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
 BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
 BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
 BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
 BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
 BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
 BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
 BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
 BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
 BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
 BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
 BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
 BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
 BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
 BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
 BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
 BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
```

Figure 28 SYSPRINT for example 5 (part 2 of 5)

```
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO
```

Figure 28 SYSPRINT for example 5 (part 3 of 5)

```
BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                = 910
BMC50471I SUBSYSTEM DEFAULT      = DEDR
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)
```

Figure 28 SYSPRINT for example 5 (part 4 of 5)

```
BMC50471I ASCII CCSID             = (819,65534,65534)
BMC50471I UNICODE CCSID           = (367,1208,1200)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'
BMC50471I ...EXCEPTIONS  ='ASU101.EXCEPTIONS2'

BMC50102I REORG TABLESPACE ARUDB032.TS032 PART 01
BMC50102I UNLOAD PAUSE
BMC50102I BMCSTATS YES
BMC50102I UPDATEDB2STATS YES
BMC50102I DDTYPE UNLOAD
BMC50102I   DSNPAT 'ARU.EXMPL05.&DDNAME'
BMC50102I DDTYPE WORK
BMC50102I   DSNPAT 'ARU.EXMPL05.&DDNAME'

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 1: SAMPLING STATISTICS: PART=1,TP=6,SP=4,SR=100,AVGR=101,SD=65,SE=0,AVGF=2500,SD=64,SE=346,EP=0
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB032.TS032.I0001.A001'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 100  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC50484I ESTIMATED CARDINALITY OF PART 0001 = 100  AVG SORTWK ROW LENGTH = 101  AVG UNLOAD ROW LENGTH = 101
BMC51264I UNLOAD WILL READ 4 DATA PAGES FROM SPACE 'ARUDB032.TS032'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGAEXM5  &STEPNAME   BMCREORG  &DB         ARUDB032  &TSIX       TS032     &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       142412    &SSID       DEDR      &UTIL       EXMPL05
BMC50425I &UTILPFX    EXMPL05   &UTILSFX              &DATE8      01272011  &GRPNM      DEDR      &VCAT       DEDRCAT
BMC50425I &TIME4      1424      &DATEJ      2011027   &JDATE      11027

BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                       UNIT OR                      KBYTES   KBYTES     ALOC      ALOC
BMC50447I DDNAME   DSNAME                                       DATACLAS MGMTCLAS STORCLAS      PRI      SEC      PRI       SEC

BMC50448I SYSREC01 ARU.EXMPL05.SYSREC01                         SYSALLDA                         11        3        1         1  TRK
BMC50448I SYSUT101 ARU.EXMPL05.SYSUT101                         SYSALLDA                          3        1        1         1  TRK
BMC50448I SYSUT102 ARU.EXMPL05.SYSUT102                         SYSALLDA                          2        1        1         1  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50509I STATISTICS COLLECTION FOR INDEX SPACE ARU.TS03212 BYPASSED DUE TO PARTIAL TABLE SPACE REORG
BMC50509I STATISTICS COLLECTION FOR INDEX SPACE ARU.TS03213 BYPASSED DUE TO PARTIAL TABLE SPACE REORG
BMC50474I BELOW 16M = 8236K, ABOVE 16M = 1398480K, CPUS = 3
BMC50479I TOTAL PAGES: 2318414, ALLOWED: 0; AVAILABLE PAGES: 332708, ALLOWED: 332708
BMC51302I MAX TASKS = 1, MAX PARTITIONS PER TASK = 0, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 100, I/O WAITS = 1 ,DDNAME = SYS00007
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50481I 1: UNLOAD TASK COMPLETE.  ELAPSED TIME = 00:00:00
BMC50476I DDNAME = SYSREC01, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT101, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT102, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51271I UNLOAD STATISTICS:  100 ROWS/KEYS UNLOADED FROM PARTITION 1
BMC51281I UNLOAD STATISTICS:  X'079D4CB0E9F8' IS THE HIGHEST LOGRBA FOR PARTITION 1
BMC51286I RELOADING OF DATASET 'DEDRCAT.DSNDBD.ARUDB032.TS032.I0001.A001' WILL REQUIRE 6 PAGES
BMC51290I RELOADING OF DATASET 'DEDRCAT.DSNDBD.ARUDB032.TS03211.I0001.A001' WILL REQUIRE 4 PAGES (APPROX.)
BMC51290I RELOADING OF DATASET 'DEDRCAT.DSNDBD.ARUDB032.TS03212.I0001.A001' WILL REQUIRE 4 PAGES (APPROX.)
```

Figure 28 SYSPRINT for example 5 (part 5 of 5)

```
BMC51288I RELOADING OF DATASET 'DEDRCAT.DSNDBD.ARUDB032.TS03213.I0001.A001' MAY REQUIRE 4 PAGES
BMC51272I UNLOAD STATISTICS:  100 ROWS UNLOADED FROM SPACE 'ARUDB032.TS032', 0 ROWS DISCARDED, 0 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'079D4CB0E9F8' IS THE HIGHEST LOGRBA FOR SPACE 'ARUDB032.TS032'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I UNLOAD PHASE COMPLETE.  ELAPSED TIME = 00:00:04

BMC50005I REORG COMMAND EXECUTION PAUSING BEFORE PHASE = 'RELOAD'
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 4
```

Example 6: Restart of a paused REORG 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.

Table 69 describes the key command options for this job.

Table 69 Key command options used in example 6

Command options used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
PART 01	limits the reorganization to partition 1
BMCSTATS YES UPDATEDB2STATS YES	tells REORG PLUS to update statistics in the DASD MANAGER PLUS database statistics tables and the DB2 catalog The JCL for this example also includes the DD statement //ASUSRPRT DD DUMMY. This DD statement tells REORG PLUS not to produce a report of the statistics.

Figure 29 on page 440 shows the JCL for example 6.

Figure 29 JCL for example 6

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  PARTITIONED TABLESPACE, ONE CLUSTERING INDEX, TWO SECONDARY      *
//*  INDEXES.                                                         *
//*  RESTART FROM UNLOAD PAUSE OF PART 01 - UTILID MUST BE THE SAME   *
//*                                                                   *
//*  NOTE: UNLOAD PAUSE OPTION IN REORG COMMAND DOES NOT HAVE TO BE   *
//*        DELETED IN RESTART JOB.  ALSO, NOTE MESSAGE FROM           *
//*        UPDATEDB2STATS YES.                                        *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='DEDR,EXMPL05,RESTART,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//ASUSRPRT  DD  DUMMY
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB032.TS032 PART 01
UNLOAD PAUSE
BMCSTATS YES
UPDATEDB2STATS YES
DDTYPE UNLOAD
  DSNPAT 'ARU.EXMPL05.&DDNAME'
DDTYPE WORK
  DSNPAT 'ARU.EXMPL05.&DDNAME'
/*
```

Figure 30 shows the SYSPRINT output for example 6.

Figure 30 SYSPRINT for example 6 (part 1 of 5)

```
                               *****  B M C   R E O R G   P L U S   F O R   D B 2   V10R1.00  *****
                                             (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
               REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
 BMC50001I UTILITY EXECUTION STARTING    1/27/2011   14:29:11 ...
 BMC50002I UTILITY ID = 'EXMPL05'.  DB2 SUBSYSTEM ID = 'DEDR'.  OPTION MODULE = 'ARU$OPTS'.
 BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
 BMC50471I REGION=0M,BELOW 16M=8864K,ABOVE 16M=1410452K,IEFUSI=NO,CPUS=3
 BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

 BMC50471I REORG PLUS FOR DB2--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I SOLUTION COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMCSORT ENGINE--V02.03.01
 BMC50471I   NO MAINTENANCE TO REPORT
```

Figure 30 SYSPRINT for example 6 (part 2 of 5)

```
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(OK,OK)
BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
```

Figure 30 SYSPRINT for example 6 (part 3 of 5)

```
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))             ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                         (NONE,NONE)                           (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                         (NONE,NONE)                           (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                         (NONE,NONE)                           (NONE,NONE)
BMC50470I THRESHLD = 0                                   0                                     0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                       ((0,K),(0,K))                         ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                   5                                     5
BMC50470I GDGEMPTY = NO                                  NO                                    NO
BMC50470I GDGSCRAT = NO                                  NO                                    NO

BMC50470I DDTYPE   = LOCBFCPY                            LOCBICPY                              REMPFCPY
BMC50470I ACTIVE   = NO                                  NO                                    NO
BMC50470I IFALLOC  = USE                                 USE                                   USE
BMC50470I ALLOC    = N/A                                 N/A                                   N/A
BMC50470I SMS      = NO                                  NO                                    NO
BMC50470I SMSUNIT  = NO                                  NO                                    NO
BMC50470I SIZEPCT  = (100,100)                           (100,100)                             (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                 (SYSALLDA,SYSALLDA)                   (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                               (0,0)                                 (0,0)
BMC50470I VOLCNT   = (25,25)                             (25,25)                               (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))             ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                         (NONE,NONE)                           (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                         (NONE,NONE)                           (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                         (NONE,NONE)                           (NONE,NONE)
BMC50470I THRESHLD = 0                                   0                                     0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                       ((0,K),(0,K))                         ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                   5                                     5
BMC50470I GDGEMPTY = NO                                  NO                                    NO
BMC50470I GDGSCRAT = NO                                  NO                                    NO

BMC50470I DDTYPE   = REMPICPY                            REMBFCPY                              REMBICPY
BMC50470I ACTIVE   = NO                                  NO                                    NO
BMC50470I IFALLOC  = USE                                 USE                                   USE
BMC50470I ALLOC    = N/A                                 N/A
BMC50470I SMS      = NO                                  NO                                    NO
BMC50470I SMSUNIT  = NO                                  NO                                    NO
BMC50470I SIZEPCT  = (100,100)                           (100,100)                             (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                 (SYSALLDA,SYSALLDA)                   (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                               (0,0)                                 (0,0)
BMC50470I VOLCNT   = (25,25)                             (25,25)                               (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))             ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                         (NONE,NONE)                           (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                         (NONE,NONE)                           (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                         (NONE,NONE)                           (NONE,NONE)
BMC50470I THRESHLD = 0                                   0                                     0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                       ((0,K),(0,K))                         ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                   5                                     5
BMC50470I GDGEMPTY = NO                                  NO                                    NO
BMC50470I GDGSCRAT = NO                                  NO                                    NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
```

Figure 30 SYSPRINT for example 6 (part 4 of 5)

```
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DEDR
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'
BMC50471I ...EXCEPTIONS  ='ASU101.EXCEPTIONS2'
BMC50003I RESTARTING UTILITY: COMMAND = 'REORG', PHASE = 'RELOAD'

BMC50102I REORG TABLESPACE ARUDB032.TS032 PART 01
BMC50102I UNLOAD PAUSE
BMC50102I BMCSTATS YES
BMC50102I UPDATEDB2STATS YES
BMC50102I DDTYPE UNLOAD
BMC50102I   DSNPAT 'ARU.EXMPL05.&DDNAME'
BMC50102I DDTYPE WORK
BMC50102I   DSNPAT 'ARU.EXMPL05.&DDNAME'

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50425I &JOBNAME    JRGAEXM6  &STEPNAME   BMCREORG  &DB         ARUDB032  &TSIX       TS032     &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       142911    &SSID       DEDR      &UTIL       EXMPL05
BMC50425I &UTILPFX    EXMPL05   &UTILSFX              &DATE8      01272011  &GRPNM      DEDR      &VCAT       DEDRCAT
BMC50425I &TIME4      1429      &DATEJ      2011027   &JDATE      11027
```

Figure 30 SYSPRINT for example 6 (part 5 of 5)

```
BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                   UNIT OR                        KBYTES   KBYTES     ALOC       ALOC
BMC50447I DDNAME   DSNAME                                   DATACLAS MGMTCLAS STORCLAS        PRI      SEC      PRI        SEC

BMC50448I SYSREC01 ARU.EXMPL05.SYSREC01                     SYSALLDA                           11        3        1          1  TRK
BMC50448I SYSUT101 ARU.EXMPL05.SYSUT101                     SYSALLDA                            3        1        1          1  TRK
BMC50448I SYSUT102 ARU.EXMPL05.SYSUT102                     SYSALLDA                            2        1        1          1  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50509I STATISTICS COLLECTION FOR INDEX SPACE ARU.TS03212 BYPASSED DUE TO PARTIAL TABLE SPACE REORG
BMC50509I STATISTICS COLLECTION FOR INDEX SPACE ARU.TS03213 BYPASSED DUE TO PARTIAL TABLE SPACE REORG
BMC50474I BELOW 16M = 8220K, ABOVE 16M = 1399692K, CPUS = 3
BMC50479I TOTAL PAGES: 2274278, ALLOWED: 0; AVAILABLE PAGES: 294203, ALLOWED: 294203
BMC51303I MAX TASKS = 2, INDEXES PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 2
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50482I 0: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB032.TS03211.I0001.A001'
BMC50477I 0: PARTITION =    1, ROWS/KEYS = 100, I/O WAITS = 5 ,DDNAME = SYS00008
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB032.TS032.I0001.A001'
BMC50477I 0: PARTITION =    1, ROWS/KEYS = 100, I/O WAITS = 3 ,DDNAME = SYS00005
BMC51274I RELOAD STATISTICS:  100 ROWS/KEYS LOADED INTO PARTITION 1
BMC50476I DDNAME = SYSREC01, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC51276I BUILD  STATISTICS:  100 KEYS LOADED INTO INDEX 'ARU.TS03211'
BMC51275I RELOAD STATISTICS:  100 ROWS LOADED INTO SPACE 'ARUDB032.TS032'
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50476I DDNAME = SYSUT101, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50476I DDNAME = SYSUT102, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 1: UPDATE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB032.TS03212.I0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 100, I/O WAITS = 2 ,DDNAME = SYS00011
BMC51277I BUILD  STATISTICS:  100 KEY-RID PAIRS UPDATED IN INDEX 'ARU.TS03212'
BMC50482I 2: UPDATE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.ARUDB032.TS03213.I0001.A001'
BMC50477I 2: PARTITION =    0, ROWS/KEYS = 100, I/O WAITS = 2 ,DDNAME = SYS00012
BMC51277I BUILD  STATISTICS:  100 KEY-RID PAIRS UPDATED IN INDEX 'ARU.TS03213'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT102', DSNAME = 'ARU.EXMPL05.SYSUT102'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT101', DSNAME = 'ARU.EXMPL05.SYSUT101'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC01', DSNAME = 'ARU.EXMPL05.SYSREC01'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I RELOAD PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50387W IMAGE COPY REQUIRED.  TABLE SPACE STATE SET TO "COPY PENDING"
BMC278098I AGGREGATE STATISTICS NOT PRODUCED FOR INDEX ARU.TS03211
BMC278098I AGGREGATE STATISTICS NOT PRODUCED FOR TABLESPACE ARUDB032.TS032
BMC278098I AGGREGATE STATISTICS NOT PRODUCED FOR TABLE ARU.TS0321
BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 4
```

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 SYSUT1*nn* data sets based on index key length. The index with the largest key length will be assigned to the first SYSUT1*nn* data set, the index with the second largest key length will be assigned to the second SYSUT1*nn*, and so on.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 70 describes the key command options for these jobs.

Table 70 Key command options used in example 7

Command options used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
UNLOAD CONTINUE	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.
ORDER NO ORDER YES	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.
ANALYZE ONLY	terminates utility execution after the ANALYZE phase completes, and removes utility ID entries from the BMC Software BMCUTIL and BMCSYNC tables

Figure 31 shows the JCL for case 1 of example 7.

Figure 31 JCL for example 7, case 1 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//* SEGMENTED TABLESPACE, 3 INDEXES (ONE CLUSTERED, TWO NONCLUSTERED) *
//*                                                                   *
//* CASE 1: REORG TABLESPACE, ORDER NO, ANALYZE ONLY.                 *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='DHV,EXMPL07A,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
```

Figure 31 JCL for example 7, case 1 (part 2 of 2)

```
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB250.TS250
UNLOAD CONTINUE
ORDER NO
ANALYZE ONLY
/*
```

Figure 32 shows the JCL for case 2 of example 7.

Figure 32 JCL for example 7, case 2

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//* SEGMENTED TABLESPACE, 3 INDEXES (ONE CLUSTERED, TWO NONCLUSTERED) *
//*                                                                   *
//* CASE 2: REORG TABLESPACE, ORDER YES, ANALYZE ONLY.                *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='DHV,EXMPL07B,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB250.TS250
ORDER YES
ANALYZE ONLY
/*
```

Figure 33 shows the SYSPRINT output for case 1 of example 7.

Figure 33 SYSPRINT for example 7, case 1 (part 1 of 5)

```
                               *****  B M C   R E O R G   P L U S   F O R   D B 2   V10R1.00  *****
                                             (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
               REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
 BMC50001I UTILITY EXECUTION STARTING    1/27/2011   14:52:25 ...
 BMC50002I UTILITY ID = 'EXMPL07A'.  DB2 SUBSYSTEM ID = 'DHV'.  OPTION MODULE = 'ARU$OPTS'.
 BMC50024I CONNECTED TO SUBSYSTEM 'DHV2'
 BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
 BMC50471I REGION=0M,BELOW 16M=8852K,ABOVE 16M=1409996K,IEFUSI=NO,CPUS=3
 BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

 BMC50471I REORG PLUS FOR DB2--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
```

Figure 33 SYSPRINT for example 7, case 1 (part 2 of 5)

```
BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.03.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(OK,OK)
BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
```

Figure 33 SYSPRINT for example 7, case 1 (part 3 of 5)

```
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
```

Figure 33 SYSPRINT for example 7, case 1 (part 4 of 5)

```
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DHV
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'

BMC50102I REORG TABLESPACE ARUDB250.TS250
BMC50102I UNLOAD CONTINUE
BMC50102I ORDER NO
BMC50102I ANALYZE ONLY

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:05

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 1: SAMPLING STATISTICS: PART=0,TP=13,SP=11,SR=333,AVGR=104,SD=65,SE=0,AVGF=3027,SD=64,SE=184,EP=0
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.ARUDB250.TS250.I0001.A001'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 333  AVG SORTWK ROW LENGTH = 104  AVG UNLOAD ROW LENGTH = 104
BMC50484I ESTIMATED CARDINALITY OF TABLE TS2501 = 333  AVG SORTWK ROW LENGTH = 104  AVG UNLOAD ROW LENGTH = 104
BMC51264I UNLOAD WILL READ 11 DATA PAGES FROM SPACE 'ARUDB250.TS250'
```

Figure 33 SYSPRINT for example 7, case 1 (part 5 of 5)

```
BMC51260I REORG PLUS DASD REQUIREMENT ESTIMATES
BMC51261I                           3380 CYLS     3390 CYLS
BMC51262I DDNAME          KBYTES    PRI    SEC    PRI    SEC   INDEX

BMC51263I SYSREC              37      1      1      1      1

BMC51263I SYSUT101            24      1      1      1      1   ARU.TS25012
BMC51263I SYSUT102            16      1      1      1      1   ARU.TS25013
BMC51263I SYSUT103            14      1      1      1      1   ARU.TS25011

BMC51263I SYSUT1              72      1      1      1      1

BMC51263I SORTWK (OPT)        72      1      1      1      1
BMC51263I SORTWK (MIN)        24      1      1      1      1

BMC51263I BMCCPY              52      1      1      1      1


BMC51263I BMCCPZ              52      1      1      1      1


BMC51263I BMCRCY              52      1      1      1      1


BMC51263I BMCRCZ              52      1      1      1      1

BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Figure 34 shows the SYSPRINT output for case 2 of example 7.

Figure 34 SYSPRINT for example 7, case 2 (part 1 of 5)

```
                               *****  B M C   R E O R G   P L U S   F O R   D B 2    V10R1.00  *****
                                             (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
               REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50001I UTILITY EXECUTION STARTING    1/27/2011   15:02:21 ...
BMC50002I UTILITY ID = 'EXMPL07B'.  DB2 SUBSYSTEM ID = 'DHV'.  OPTION MODULE = 'ARU$OPTS'.
BMC50024I CONNECTED TO SUBSYSTEM 'DHV2'
BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
BMC50471I REGION=0M,BELOW 16M=8864K,ABOVE 16M=1408868K,IEFUSI=NO,CPUS=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

BMC50471I REORG PLUS FOR DB2--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.03.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
```

Figure 34 SYSPRINT for example 7, case 2 (part 2 of 5)

```
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                            TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                                TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                          TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                                TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                        TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                          TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                               TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                              TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                             TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                          UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                              UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                             UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                 UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                  UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                             UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                               WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                         WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                               WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                                XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                             XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                           ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                              WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                 YES                                  NO
BMC50470I IFALLOC  = USE                                 USE                                  USE
BMC50470I ALLOC    = N/A                                 N/A                                  ANY
BMC50470I SMS      = NO                                  NO                                   NO
BMC50470I SMSUNIT  = NO                                  NO                                   NO
BMC50470I SIZEPCT  = (100,100)                           (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                 (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                         (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                         (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                         (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                   0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                       ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                 N/A                                  N/A
BMC50470I RETPD    = N/A                                 N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                 N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                 N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                 N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                             LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                  YES                                  YES
BMC50470I IFALLOC  = USE                                 USE                                  USE
BMC50470I ALLOC    = N/A                                 N/A                                  N/A
BMC50470I SMS      = NO                                  NO                                   NO
BMC50470I SMSUNIT  = NO                                  NO                                   NO
BMC50470I SIZEPCT  = (100,100)                           (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                 (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                               (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                             (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))           ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                         (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                         (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                         (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                   0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                       ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                   5                                    5
BMC50470I GDGEMPTY = NO                                  NO                                   NO
BMC50470I GDGSCRAT = NO                                  NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                            LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                  NO                                   NO
BMC50470I IFALLOC  = USE                                 USE                                  USE
BMC50470I ALLOC    = N/A                                 N/A                                  N/A
BMC50470I SMS      = NO                                  NO                                   NO
BMC50470I SMSUNIT  = NO                                  NO                                   NO
BMC50470I SIZEPCT  = (100,100)                           (100,100)                            (100,100)
```

Figure 34 SYSPRINT for example 7, case 2 (part 3 of 5)

```
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
```

Figure 34 SYSPRINT for example 7, case 2 (part 4 of 5)

```
BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DHV
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'

BMC50102I REORG TABLESPACE ARUDB250.TS250
BMC50102I ORDER YES
BMC50102I ANALYZE ONLY

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:10

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 1: SAMPLING STATISTICS: PART=0,TP=13,SP=11,SR=333,AVGR=104,SD=65,SE=0,AVGF=3027,SD=64,SE=184,EP=0
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.ARUDB250.TS250.I0001.A001'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 333  AVG SORTWK ROW LENGTH = 104  AVG UNLOAD ROW LENGTH = 104
BMC50484I ESTIMATED CARDINALITY OF TABLE TS2501 = 333  AVG SORTWK ROW LENGTH = 104  AVG UNLOAD ROW LENGTH = 104
BMC51264I UNLOAD WILL READ 11 DATA PAGES FROM SPACE 'ARUDB250.TS250'

BMC51260I REORG PLUS DASD REQUIREMENT ESTIMATES
BMC51261I                           3380 CYLS     3390 CYLS
BMC51262I DDNAME          KBYTES    PRI    SEC    PRI    SEC   INDEX

BMC51263I SYSREC              49      1      1      1      1

BMC51263I SYSUT101            24      1      1      1      1   ARU.TS25012
BMC51263I SYSUT102            16      1      1      1      1   ARU.TS25013

BMC51263I SYSUT1              48      1      1      1      1

BMC51263I SORTWK (OPT)        49      1      1      1      1
BMC51263I SORTWK (MIN)        49      1      1      1      1

BMC51263I BMCCPY              52      1      1      1      1


BMC51263I BMCCPZ              52      1      1      1      1


BMC51263I BMCRCY              52      1      1      1      1


BMC51263I BMCRCZ              52      1      1      1      1
```

Figure 34 SYSPRINT for example 7, case 2 (part 5 of 5)

```
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Example 8: Selective unload with discards to archive data set

In this example, REORG PLUS reorganizes a segmented table space containing only one table. The table has one nonpartitioned secondary index.

Although the installation options enable dynamic allocation for the unload, work, and primary copy data sets, they also specify IFALLOC USE for these data sets. Therefore, the job does not have to specify ACTIVE NO to have REORG PLUS use the data sets allocated in the JCL.

In the SYSPRINT output, message BMC50477I (issued during the UNLOAD phase) indicates the total number of rows in the table space before reorganization. Message BMC51272I (issued at the end of the UNLOAD phase) indicates the number of rows selected to be unloaded, number of rows discarded, and number of rows updated.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 71 describes the key command options and DD statements for this job.

Table 71 Command options and DD statements used in example 8 (part 1 of 2)

Command options and DD statements used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
UNLOAD CONTINUE	tells REORG PLUS to continue the reorganization with two-phase processing after the UNLOAD phase has unloaded the data
COPY YES	creates a DB2 image copy of the table space REORG PLUS makes four copies of the table space after reorganizing it.
REGISTER (ARUCPY1, ARUCPY3)	specifies that only two of the four copies, ARUCPY1 and ARUCPY3, are to be registered in SYSIBM.SYSCOPY

Table 71 Command options and DD statements used in example 8 (part 2 of 2)

Command options and DD statements used in JCL	Description
COPYDDN (ARUCPY1, ARUCPY2)	■ 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 ■ overrides the default names of the BMCCPY and BMCCPZ
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
//ARUCPY2
//ARUCPY3
//ARUCPY4
	ddnames that are used for the data sets that receive full image copies of the table space that you are reorganizing

//ARUARC	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

Figure 35 shows the JCL for example 8.

Figure 35 JCL for example 8 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*   SEGMENTED TABLESPACE, ONE TABLE, ONE INDEX                      *
//*   REORG TABLESPACE RETAINING ONLY SELECTED ROWS                   *
//*   ALL DELETED ROWS GO TO THE ARCHIVE DATASET                      *
//*   MAKE 4 COPIES (TWO LOCALS, TWO REMOTES)                         *
```

Figure 35 JCL for example 8 (part 2 of 2)

```
//*   REGISTER ONLY THE PRIMARY COPIES.                               *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='DHV,EXMPL08,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//*
//SYSREC    DD DSN=ARU.EXMPL08.SYSREC,
//          DISP=(MOD,CATLG,CATLG),
//          UNIT=WORK,SPACE=(CYL,(30,10))
//*
//ARUCPY1   DD DSN=ARU.EXMPL08.ARUCPY1,
//          DISP=(,CATLG),
//          SPACE=(CYL,(1,1),RLSE),UNIT=WORK
//ARUCPY2   DD DSN=ARU.EXMPL08.ARUCPY2,
//          DISP=(,CATLG),
//          SPACE=(CYL,(1,1),RLSE),UNIT=WORK
//ARUCPY3   DD DSN=ARU.EXMPL08.ARUCPY3,
//          DISP=(,CATLG),
//          SPACE=(CYL,(1,1),RLSE),UNIT=WORK
//ARUCPY4   DD DSN=ARU.EXMPL08.ARUCPY4,
//          DISP=(,CATLG),
//          SPACE=(CYL,(1,1),RLSE),UNIT=WORK
//*
//ARUARC    DD DSN=ARU.EXMPL08.ARUARC,
//          DISP=(MOD,CATLG,CATLG),
//          UNIT=WORK,SPACE=(CYL,(10,10))
//*
//SYSUT1    DD DSN=ARU.EXMPL08.SYSUT1,
//          DISP=(MOD,CATLG,CATLG),
//          UNIT=WORK,SPACE=(CYL,(1,1),RLSE)
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB210.TS210
UNLOAD CONTINUE
COPY YES REGISTER (ARUCPY1, ARUCPY3)
COPYDDN (ARUCPY1, ARUCPY2)
RECOVERYDDN (ARUCPY3, ARUCPY4)
ARCHDDN (ARUARC)
REDEFINE NO
ORDER NO
SELECT * FROM ARU.TS2101 WHERE
         (COL1_CHAR = 'JEREMY' AND COL2_DECIMAL = 111.0000)
/*
```

Figure 36 shows the SYSPRINT output for example 8.

Figure 36 SYSPRINT for example 8 (part 1 of 5)

```
                               *****  B M C   R E O R G   P L U S   F O R   D B 2    V10R1.00  *****
                                             (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
               REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50001I UTILITY EXECUTION STARTING    1/27/2011   15:39:40 ...
BMC50002I UTILITY ID = 'EXMPL08'.  DB2 SUBSYSTEM ID = 'DHV'.  OPTION MODULE = 'ARU$OPTS'.
BMC50024I CONNECTED TO SUBSYSTEM 'DHV2'
BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
BMC50471I REGION=0M,BELOW 16M=8832K,ABOVE 16M=1409420K,IEFUSI=NO,CPUS=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

BMC50471I REORG PLUS FOR DB2--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.03.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(OK,OK)
BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                                 WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                    YES                                  NO
BMC50470I IFALLOC  = USE                                    USE                                  USE
BMC50470I ALLOC    = N/A                                    N/A                                  ANY
BMC50470I SMS      = NO                                     NO                                   NO
BMC50470I SMSUNIT  = NO                                     NO                                   NO
BMC50470I SIZEPCT  = (100,100)                              (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                    (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                                  (0,0)                                N/A
BMC50470I VOLCNT   = (25,25)                                (25,25)                              N/A
```

Figure 36 SYSPRINT for example 8 (part 2 of 5)

```
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            N/A
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
```

Figure 36 SYSPRINT for example 8 (part 3 of 5)

```
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DHV
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
```

Figure 36 SYSPRINT for example 8 (part 4 of 5)

```
BMC50471I ...COLDIST    ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP   ='ATS101.RS_STOGROUP'

BMC50102I REORG TABLESPACE ARUDB210.TS210
BMC50102I UNLOAD CONTINUE
BMC50102I COPY YES REGISTER (ARUCPY1, ARUCPY3)
BMC50102I COPYDDN (ARUCPY1, ARUCPY2)
BMC50102I RECOVERYDDN (ARUCPY3, ARUCPY4)
BMC50102I ARCHDDN (ARUARC)
BMC50102I REDEFINE NO
BMC50102I ORDER NO
BMC50102I SELECT * FROM ARU.TS2101 WHERE
BMC50102I          (COL1_CHAR = 'JEREMY' AND COL2_DECIMAL = 111.0000)

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:03

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 1: SAMPLING STATISTICS: PART=0,TP=37,SP=35,SR=1266,AVGR=66,SD=65,SE=0,AVGF=3617,SD=65,SE=354,EP=8
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.ARUDB210.TS210.I0001.A001'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 1266  AVG SORTWK ROW LENGTH = 66  AVG UNLOAD ROW LENGTH = 66
BMC50484I ESTIMATED CARDINALITY OF TABLE TS2101 = 1266  AVG SORTWK ROW LENGTH = 66  AVG UNLOAD ROW LENGTH = 66
BMC51264I UNLOAD WILL READ 35 DATA PAGES FROM SPACE 'ARUDB210.TS210'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGAEXM8  &STEPNAME   BMCREORG  &DB         ARUDB210  &TSIX       TS210     &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       153940    &SSID       DHV2      &UTIL       EXMPL08
BMC50425I &UTILPFX    EXMPL08   &UTILSFX              &DATE8      01272011  &GRPNM      DHV       &VCAT       DSNDHV
BMC50425I &TIME4      1539      &DATEJ      2011027   &JDATE      11027

BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT
BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50474I BELOW 16M = 8428K, ABOVE 16M = 1401904K, CPUS = 3
BMC50479I TOTAL PAGES: 2117209, ALLOWED: 0; AVAILABLE PAGES: 245302, ALLOWED: 245302
BMC51302I MAX TASKS = 1, MAX PARTITIONS PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 1266, I/O WAITS = 1 ,DDNAME = SYS00009
BMC50481I 1: UNLOAD TASK COMPLETE.  ELAPSED TIME = 00:00:00
BMC50476I DDNAME = SYSREC, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = ARUARC, I/OS = 3, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT1, I/OS = 2, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51286I RELOADING OF DATASET 'DSNDHV.DSNDBD.ARUDB210.TS210.I0001.A001' WILL REQUIRE 8 PAGES
BMC51288I RELOADING OF DATASET 'DSNDHV.DSNDBD.ARUDB210.TS21011.I0001.A001' MAY REQUIRE 6 PAGES
BMC51272I UNLOAD STATISTICS:  204 ROWS UNLOADED FROM SPACE 'ARUDB210.TS210', 1062 ROWS DISCARDED, 0 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'C73EB0D6CD17' IS THE HIGHEST LOGRBA FOR SPACE 'ARUDB210.TS210'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I UNLOAD PHASE COMPLETE.  ELAPSED TIME = 00:00:06

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50474I BELOW 16M = 8396K, ABOVE 16M = 1401636K, CPUS = 3
BMC50479I TOTAL PAGES: 2122013, ALLOWED: 0; AVAILABLE PAGES: 246047, ALLOWED: 246047
BMC51303I MAX TASKS = 1, INDEXES PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50375I INLINE COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.ARUDB210.TS210.I0001.A001'
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DSNDHV.DSNDBD.ARUDB210.TS210.I0001.A001'
BMC50477I 0: PARTITION =    0, ROWS/KEYS = 204, I/O WAITS = 2 ,DDNAME = SYS00010
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50476I DDNAME = SYSREC, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC51275I RELOAD STATISTICS:  204 ROWS LOADED INTO SPACE 'ARUDB210.TS210'
BMC50476I DDNAME = SYSUT1, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.ARUDB210.TS21011.I0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 204, I/O WAITS = 5 ,DDNAME = SYS00013
BMC51276I BUILD  STATISTICS:  204 KEYS LOADED INTO INDEX 'ARU.TS21011'
BMC50476I DDNAME = ARUCPY1, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = ARUCPY2, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = ARUCPY3, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = ARUCPY4, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 9 PAGES COPIED TO DATASET = 'ARU.EXMPL08.ARUCPY1'
BMC50376I 9 PAGES COPIED TO DATASET = 'ARU.EXMPL08.ARUCPY2'
BMC50376I 9 PAGES COPIED TO DATASET = 'ARU.EXMPL08.ARUCPY3'
BMC50376I 9 PAGES COPIED TO DATASET = 'ARU.EXMPL08.ARUCPY4'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT1  ', DSNAME = 'ARU.EXMPL08.SYSUT1'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC  ', DSNAME = 'ARU.EXMPL08.SYSREC'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I RELOAD PHASE COMPLETE.  ELAPSED TIME = 00:00:03
```

Figure 36 SYSPRINT for example 8 (part 5 of 5)

```
BMC50501I DB2 OBJECT STATISTICS
BMC50515I TABLESPACE  ARUDB210.TS210  PARTS    = 0         TABLES   = 1     SEGSIZE = 4     DSSIZE =    0G    PAGESIZE = 4K
BMC50516I             PART  NACTIVE       CARD   EXTENTS   DBCARD PCOMP   KSAVED  PSAVED
BMC50517I                0        8        204         1        0     0        0       0
BMC50518I   TABLE     ARU.TS2101
BMC50519I             ROWAVG    NPAGES       CARD   PCTPAGES
BMC50520I                 66         5        204         62
BMC50525I     INDEX   ARU.TS21011
BMC50526I             FIRST KEY COLUMN = COL1_CHAR
BMC50527I             CLUSTER = N   UNIQUE = D   COMPRESS = N   PAGESIZE = 4K   KEYLEN =   20   COLCOUNT = 3
BMC50528I             PART   NACTIVE       CARD   EXTENTS   LEVELS
BMC50529I                0         5        204         1        2
BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Example 9: ON FAILURE with a user-specified return code

This example shows a full table space reorganization of a segmented table space that contains multiple tables. The job fails with a space failure (abend D37) because the SYSREC data set has insufficient space. The reorganization terminates with return code 31 as specified with the ON FAILURE TERMINATE option. The job is not restartable because of the ON FAILURE TERMINATE option, but it can be resubmitted from the beginning at a later time. For more information about the ON FAILURE option, see page 205.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 72 describes the key command options and DD statements for this job.

Table 72 Key command options and DD statements used in example 9 (part 1 of 2)

Command options and DD statements used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
COPY YES	tells REORG PLUS to create a DB2 image copy of the table space
REDEFINE YES	tells REORG PLUS to delete and redefine the VSAM data sets containing the DB2 objects
ORDER YES	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.
SORTDATA	enables this job to be compatible if run with the IBM DB2 REORG utility

Table 72 Key command options and DD statements used in example 9 (part 2 of 2)

Command options and DD statements used in JCL	Description
ON FAILURE UNLOAD TERMINATE UTILITY RETCODE 31	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.
DDTYPE LOCBFCPY ACTIVE YES DDTYPE REMPFCPY ACTIVE YES DDTYPE REMBFCPY ACTIVE YES	activates dynamic allocation for the local backup copy and the remote copies, overriding the installation defaults
DSNPAT	specifies data set names for dynamic allocation that override the default patterns
//SYSREC	contains the rows that you are reorganizing The SYSREC data set is allocated with only one track (SPACE=(TRK,(1))) causing the job to abend during the UNLOAD phase.

Figure 37 shows the JCL for example 9.

Figure 37 JCL for example 9 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*   SEGMENTED TABLESPACE, MULTIPLE TABLES,                          *
//*   REORG ENTIRE TABLESPACE, ALLOCATE ONLY ONE TRACK FOR THE SYSREC *
//*   TO FORCE REORG ABEND W/ D37 DURING THE UNLOAD PHASE.            *
//*                                                                   *
//*   *******************************************************         *
//*   ON FAILURE TESTING                                              *
//*      ON FAILURE UNLOAD TERMINATE UTILITY RETCODE 31               *
//*   NOTE: WITH ON FAILURE UNLOAD TERMINATE UTILITY RETCODE          *
//*         SPECIFIED IN REORG, REORG STARTED ALL TABLESPACES         *
//*         AND TERMED THE UTILID WITH A RETURN CODE 31.              *
//*         IN THIS CASE, RESTART IS NOT ALLOWED.                     *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BMCREORG EXEC PGM=ARUUMAIN,
//             PARM='DHV,EXMPL09,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//*
```

Figure 37 JCL for example 9 (part 2 of 2)

```
//SYSREC    DD  DSN=ARU.EXMPL09.SYSREC,
//            DISP=(MOD,CATLG,CATLG),
//            UNIT=3390,SPACE=(TRK,(1,0))
//*
//SYSIN     DD  *
REORG TABLESPACE ARUDB006.TS006
COPY YES
ORDER YES
SORTDATA
ON FAILURE UNLOAD TERMINATE UTILITY RETCODE 31
DDTYPE UNLOAD ACTIVE NO
DDTYPE WORK
  DSNPAT 'ARU.EXMPL09.&DDNAME'
DDTYPE LOCPFCPY
  DSNPAT 'ARU.EXMPL09.&DDNAME'
DDTYPE LOCBFCPY ACTIVE YES
  DSNPAT 'ARU.EXMPL09.&DDNAME'
DDTYPE REMPFCPY ACTIVE YES
  DSNPAT 'ARU.EXMPL09.&DDNAME'
DDTYPE REMBFCPY ACTIVE YES
  DSNPAT 'ARU.EXMPL09.&DDNAME'
/*
```

Figure 38 shows the SYSPRINT output for example 9.

Figure 38 SYSPRINT for example 9 (part 1 of 5)

```
                               *****  B M C   R E O R G   P L U S   F O R   D B 2   V10R1.00  *****
                                             (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
               REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
BMC50001I UTILITY EXECUTION STARTING    1/27/2011   15:43:35 ...
BMC50002I UTILITY ID = 'EXMPL09'.  DB2 SUBSYSTEM ID = 'DHV'.  OPTION MODULE = 'ARU$OPTS'.
BMC50024I CONNECTED TO SUBSYSTEM 'DHV2'
BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
BMC50471I REGION=0M,BELOW 16M=8852K,ABOVE 16M=1409960K,IEFUSI=NO,CPUS=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

BMC50471I REORG PLUS FOR DB2--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.03.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
```

Figure 38 SYSPRINT for example 9 (part 2 of 5)

```
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
```

Figure 38 SYSPRINT for example 9 (part 3 of 5)

```
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
```

Figure 38 SYSPRINT for example 9 (part 4 of 5)

```
BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DHV
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'

BMC50102I REORG TABLESPACE ARUDB006.TS006
BMC50102I COPY YES
BMC50102I ORDER YES
BMC50102I SORTDATA
BMC50102I ON FAILURE UNLOAD TERMINATE UTILITY RETCODE 31
BMC50102I DDTYPE UNLOAD ACTIVE NO
BMC50102I DDTYPE WORK
BMC50102I   DSNPAT 'ARU.EXMPL09.&DDNAME'
BMC50102I DDTYPE LOCPFCPY
BMC50102I   DSNPAT 'ARU.EXMPL09.&DDNAME'
BMC50102I DDTYPE LOCBFCPY ACTIVE YES
BMC50102I   DSNPAT 'ARU.EXMPL09.&DDNAME'
BMC50102I DDTYPE REMPFCPY ACTIVE YES
BMC50102I   DSNPAT 'ARU.EXMPL09.&DDNAME'
BMC50102I DDTYPE REMBFCPY ACTIVE YES
BMC50102I   DSNPAT 'ARU.EXMPL09.&DDNAME'

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:06

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 1: SAMPLING STATISTICS: PART=0,TP=766,SP=312,SR=6982,AVGR=110,SD=66,SE=0,AVGF=2237,SD=65,SE=62,EP=54
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.ARUDB006.TS006.I0001.A001'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 17095  AVG SORTWK ROW LENGTH = 110  AVG UNLOAD ROW LENGTH = 110
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0061 = 1851  AVG SORTWK ROW LENGTH = 172  AVG UNLOAD ROW LENGTH = 172
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0062 = 4235  AVG SORTWK ROW LENGTH = 98  AVG UNLOAD ROW LENGTH = 98
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0063 = 5852  AVG SORTWK ROW LENGTH = 100  AVG UNLOAD ROW LENGTH = 100
BMC50484I ESTIMATED CARDINALITY OF TABLE TS0064 = 5157  AVG SORTWK ROW LENGTH = 109  AVG UNLOAD ROW LENGTH = 109
BMC51264I UNLOAD WILL READ 764 DATA PAGES FROM SPACE 'ARUDB006.TS006'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGAEXM9  &STEPNAME   BMCREORG  &DB         ARUDB006  &TSIX       TS006     &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       154335    &SSID       DHV2      &UTIL       EXMPL09
BMC50425I &UTILPFX    EXMPL09   &UTILSFX              &DATE8      01272011  &GRPNM      DHV       &VCAT       DSNDHV
BMC50425I &TIME4      1543      &DATEJ      2011027   &JDATE      11027
```

Figure 38   SYSPRINT for example 9 (part 5 of 5)

```
BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                       UNIT OR                      KBYTES   KBYTES     ALOC      ALOC
BMC50447I DDNAME   DSNAME                                       DATACLAS MGMTCLAS STORCLAS      PRI      SEC      PRI       SEC

BMC50448I BMCCPY   ARU.EXMPL09.BMCCPY                           SYSALLDA                       3064      306       64         7  TRK
BMC50448I BMCCPZ   ARU.EXMPL09.BMCCPZ                           SYSALLDA                       3064      306       64         7  TRK
BMC50448I BMCRCY   ARU.EXMPL09.BMCRCY                           SYSALLDA                       3064      306       64         7  TRK
BMC50448I BMCRCZ   ARU.EXMPL09.BMCRCZ                           SYSALLDA                       3064      306       64         7  TRK
BMC50448I SYSUT101 ARU.EXMPL09.SYSUT101                         SYSALLDA                        104       11        2         1  TRK
BMC50448I SYSUT102 ARU.EXMPL09.SYSUT102                         SYSALLDA                        157       16        4         1  TRK
BMC50448I SYSUT103 ARU.EXMPL09.SYSUT103                         SYSALLDA                         96       10        2         1  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50474I BELOW 16M = 8448K, ABOVE 16M = 1400904K, CPUS = 3
BMC50479I TOTAL PAGES: 2106215, ALLOWED: 0; AVAILABLE PAGES: 234587, ALLOWED: 234587
BMC51302I MAX TASKS = 1, MAX PARTITIONS PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50411U REORG TASK NUMBER 1 ABNORMALLY TERMINATED WITH SYSTEM ABEND CODE = D37, REASON CODE = 00000004
BMC50376I 0 PAGES COPIED TO DATASET = 'ARU.EXMPL09.BMCCPY'
BMC50376I 0 PAGES COPIED TO DATASET = 'ARU.EXMPL09.BMCCPZ'
BMC50376I 0 PAGES COPIED TO DATASET = 'ARU.EXMPL09.BMCRCY'
BMC50376I 0 PAGES COPIED TO DATASET = 'ARU.EXMPL09.BMCRCZ'
BMC50313S RDJFCB FAILED FOR DDNAME = 'BMCCPY'
BMC50476I DDNAME = SYSREC, I/OS = 2, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT101, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT103, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT102, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50313S RDJFCB FAILED FOR DDNAME = 'BMCCPY'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT103', DSNAME = 'ARU.EXMPL09.SYSUT103'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT102', DSNAME = 'ARU.EXMPL09.SYSUT102'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT101', DSNAME = 'ARU.EXMPL09.SYSUT101'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'BMCRCZ  ', DSNAME = 'ARU.EXMPL09.BMCRCZ'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'BMCRCY  ', DSNAME = 'ARU.EXMPL09.BMCRCY'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'BMCCPZ  ', DSNAME = 'ARU.EXMPL09.BMCCPZ'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'BMCCPY  ', DSNAME = 'ARU.EXMPL09.BMCCPY'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC  ', DSNAME = 'ARU.EXMPL09.SYSREC'
BMC50271I ATTEMPTING TO START SPACE 'ARUDB006.TS006'...
BMC50272I SPACE 'ARUDB006.TS006' IS STARTED
BMC50271I ATTEMPTING TO START SPACE 'ARUDB006.TS00611'...
BMC50272I SPACE 'ARUDB006.TS00611' IS STARTED
BMC50271I ATTEMPTING TO START SPACE 'ARUDB006.TS00612'...
BMC50272I SPACE 'ARUDB006.TS00612' IS STARTED
BMC50271I ATTEMPTING TO START SPACE 'ARUDB006.TS00621'...
BMC50272I SPACE 'ARUDB006.TS00621' IS STARTED
BMC50271I ATTEMPTING TO START SPACE 'ARUDB006.TS00641'...
BMC50272I SPACE 'ARUDB006.TS00641' IS STARTED
BMC50019I UTILITY REQUESTED TO TERMINATE DUE TO FAILURE IN PHASE 'REORG'
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 31
```

Example 10: DSNUTILB reorganization and index that contains keys with random ordering

In this example, REORG PLUS invokes DSNUTILB to reorganize a segmented table space that has an associated index that contains keys with random ordering. This job is a two-phase table space reorganization. This job specifies a different options module on the EXEC statement that activates DSNUTILB when random key indexes are participating in the reorganization (IXRANDOM=YES).

Because this job invokes DSNUTILB, ACTIVE YES must be specified for all work files that the reorganization job requires. ACTIVE YES is required for at least the primary local copy data set (DDTYPE LOCPFCPY) when COPY YES is specified.

NOTE

All copy data sets for a DSNUTILB reorganization are dynamically allocated, even if you specify ACTIVE YES for only the primary local copy data set.

DSNU050I messages show the REORG command and the TEMPLATE statements that are created from the REORG PLUS command and installation option values.

Table 73 describes the key command options for this job.

Table 73 Key command options used in example 10 (part 1 of 2)

Command options used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
COPY YES INLINE YES	creates an inline DB2 image copy of the table space
COPYDDN BMCPY	■ specifies the ddname BMCPY for the data set that receives the full local image copy of the table space that you are reorganizing ■ overrides the default name of BMCCPY
RECOVERYDDN BMRPY	■ specifies a ddname of BMRPY for data set that receives the full remote image copy of the table space that you are reorganizing ■ overrides the default names of BMCRCY
ORDER YES	tells REORG PLUS to sort the rows ORDER YES is passed to DSNUTILB as SORTDATA YES, and DSNUTILB will sort the clustering keys
REDEFINE NO	tells REORG PLUS not to delete and redefine the VSAM data sets for the table space Because this job invokes DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility as REUSE.
SORTDEVT	specifies the device type for the sort work files that are allocated dynamically Because this job invokes DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility for processing.
SORTNUM	affects the allocation of sort work files Because this job invokes DSNUTILB, REORG PLUS passes this option to the DB2 REORG utility as the number of sort work files to allocate dynamically.

Table 73 Key command options used in example 10 (part 2 of 2)

Command options used in JCL	Description
DDTYPE LOCBFCPY ACTIVE YES DDTYPE REMPFCPY ACTIVE YES DDTYPE REMBFCPY ACTIVE YES	activates dynamic data set allocation for the specified data set types
IFALLOC	Because this job invokes DSNUTILB, REORG PLUS ignores this option.
DSNPAT	specifies a pattern of variables and text that REORG PLUS uses to create data set names for dynamic data set allocation, overriding the default patterns Because this job invokes DSNUTILB, REORG PLUS includes this pattern in the TEMPLATE control statement that it builds for the DB2 REORG utility.
UNIT MAXEXTSZ UNITCNT VOLCNT THRESHLD	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.
UPDATEDB2STATS YES	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.

Figure 39 shows the JCL for example 10.

Figure 39 JCL for example 10 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  SEGMENTED TABLESPACE WITH AN ASSOCIATED INDEX CONTAINING A       *
//*  RANDOM INDEX KEY. DSNUTILB INVOKED DUE TO RANDOM INDEX KEY.      *
//*  NOTE THE BMC OPTIONS TRANSLATED TO DSNUTILB OPTIONS.             *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//LARREORG EXEC PGM=ARUUMAIN,REGION=0M,COND=(7,LT),
// PARM='DEDR,ARUDB070.RG1,NEW,,MSGLEVEL(1),JG9$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//*
//SYSIN     DD *
REORG TABLESPACE ARUDB070.LARS$JBA
  SHRLEVEL NONE
  UNLOAD CONTINUE
  COPY YES
```

Figure 39 JCL for example 10 (part 2 of 2)

```
  INLINE YES
  COPYLVL FULL
  COPYDDN BMCPY
  RECOVERYDDN BMRPY
  PREFORMAT YES
  REDEFINE NO
  ORDER YES
  UPDATEDB2STATS YES
  SORTDEVT 3390
  SORTNUM 12
  SORTDATA
  DDTYPE UNLOAD UNIT (WORK) IFALLOC USE
  DSNPAT 'ARU.LARREORG.DEDR.ARUDB070.&DDNAME'
  MAXEXTSZ 100 UNITCNT (1,15) VOLCNT AUTO
  DDTYPE WORK UNIT (WORK) IFALLOC USE
  DSNPAT 'ARU.LARREORG.DEDR.ARUDB070.&DDNAME'
  DDTYPE LOCPFCPY IFALLOC USE  DSNPAT
  'ARU.LARREORG.DEDR.&DB.&TSIX.&DDNAME'
  UNIT (WORK) THRESHLD 10000
  DDTYPE LOCBFCPY ACTIVE YES IFALLOC USE  DSNPAT
  'ARU.LARREORG.DEDR.&DB.&TSIX.&DDNAME' UNIT (WORK)
  DDTYPE REMPFCPY ACTIVE YES UNIT (WORK) IFALLOC USE
  DSNPAT 'ARU.LARREORG.DEDR.&DB.&TSIX.&DDNAME'
  DDTYPE REMBFCPY ACTIVE YES UNIT (WORK) IFALLOC USE
  DSNPAT 'ARU.LARREORG.DEDR.&DB.&TSIX.&DDNAME'
/*
```

Figure 40 shows the SYSPRINT output for example 10.

Figure 40 SYSPRINT for example 10 (part 1 of 8)

```
                                *****  B M C   R E O R G   P L U S   F O R   D B 2    V10R1.00  *****
                                              (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
                REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
 BMC50001I UTILITY EXECUTION STARTING    1/27/2011   15:47:18 ...
 BMC50002I UTILITY ID = 'ARUDB070.RG1'.  DB2 SUBSYSTEM ID = 'DEDR'.  OPTION MODULE = 'JG9$OPTS'.
 BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
 BMC50471I REGION=0M,BELOW 16M=8848K,ABOVE 16M=1409760K,IEFUSI=NO,CPUS=3
 BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

 BMC50471I REORG PLUS FOR DB2--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I SOLUTION COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMCSORT ENGINE--V02.03.01
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMC STATS API--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT

 BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
 BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
 BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
 BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
 BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
 BMC50471I AUXREORG=DEFAULT                        IXRANDOM=YES                            SORTDEVT=(,SYSALLDA)
 BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
 BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
 BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
```

Figure 40 SYSPRINT for example 10 (part 2 of 8)

```
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO
```

Figure 40 SYSPRINT for example 10 (part 3 of 8)

```
BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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
```

Figure 40 SYSPRINT for example 10 (part 4 of 8)

```
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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DEDR
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'
BMC50471I ...EXCEPTIONS  ='ASU101.EXCEPTIONS2'

BMC50102I REORG TABLESPACE ARUDB070.LARS$JBA
BMC50102I   SHRLEVEL NONE
BMC50102I   UNLOAD CONTINUE
BMC50102I   COPY YES
BMC50102I   INLINE YES
BMC50102I   COPYLVL FULL
BMC50102I   COPYDDN BMCPY
BMC50102I   RECOVERYDDN BMRPY
BMC50102I   PREFORMAT YES
BMC50102I   REDEFINE NO
BMC50102I   ORDER YES
BMC50102I   UPDATEDB2STATS YES
BMC50102I   SORTDEVT 3390
BMC50102I   SORTNUM 12
BMC50102I   SORTDATA
BMC50102I   DDTYPE UNLOAD UNIT (WORK) IFALLOC USE
BMC50102I   DSNPAT 'ARU.LARREORG.DEDR.ARUDB070.&DDNAME'
BMC50102I   MAXEXTSZ 100 UNITCNT (1,15) VOLCNT AUTO
BMC50102I   DDTYPE WORK UNIT (WORK) IFALLOC USE
BMC50102I   DSNPAT 'ARU.LARREORG.DEDR.ARUDB070.&DDNAME'
BMC50102I   DDTYPE LOCPFCPY IFALLOC USE  DSNPAT
BMC50102I   'ARU.LARREORG.DEDR.&DB.&TSIX.&DDNAME'
BMC50102I   UNIT (WORK) THRESHLD 10000
BMC50102I   DDTYPE LOCBFCPY ACTIVE YES IFALLOC USE  DSNPAT
BMC50102I   'ARU.LARREORG.DEDR.&DB.&TSIX.&DDNAME' UNIT (WORK)
BMC50102I   DDTYPE REMPFCPY ACTIVE YES UNIT (WORK) IFALLOC USE
BMC50102I   DSNPAT 'ARU.LARREORG.DEDR.&DB.&TSIX.&DDNAME'
BMC50102I   DDTYPE REMBFCPY ACTIVE YES UNIT (WORK) IFALLOC USE
BMC50102I   DSNPAT 'ARU.LARREORG.DEDR.&DB.&TSIX.&DDNAME'

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:05
```

Figure 40 SYSPRINT for example 10 (part 5 of 8)

```
DSNU000I    027 15:47:28.39 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = ARUDB070.RG1
DSNU1044I   027 15:47:29.84 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    027 15:47:29.88 DSNUGUTC -  TEMPLATE BMC00001 DSN 'ARU.LARREORG.DEDR.&DB..&SN..BMCRCZ' UNIT WORK VOLCNT
 25 PCTPRIME 100
DSNU1035I   027 15:47:29.90 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I    027 15:47:29.90 DSNUGUTC -  TEMPLATE BMC00002 DSN 'ARU.LARREORG.DEDR.&DB..&SN..BMRPY' UNIT WORK VOLCNT
25 PCTPRIME 100
DSNU1035I   027 15:47:29.90 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I    027 15:47:29.90 DSNUGUTC -  TEMPLATE BMC00003 DSN 'ARU.LARREORG.DEDR.&DB..&SN..BMCCPZ' UNIT WORK VOLCNT
 25 PCTPRIME 100
DSNU1035I   027 15:47:29.90 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I    027 15:47:29.90 DSNUGUTC -  TEMPLATE BMC00004 DSN 'ARU.LARREORG.DEDR.&DB..&SN..BMCPY' UNIT WORK VOLCNT
25 PCTPRIME 100 LIMIT(9 MB, BMC00005)
DSNU1035I   027 15:47:29.90 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I    027 15:47:29.90 DSNUGUTC -  TEMPLATE BMC00005 DSN 'ARU.LARREORG.DEDR.&DB..&SN..BMCPY' UNIT SYSALLDA
VOLCNT 25 PCTPRIME 100
DSNU1035I   027 15:47:29.90 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I    027 15:47:29.90 DSNUGUTC -  TEMPLATE BMC00006 DSN 'ARU.LARREORG.DEDR.ARUDB070.SYSREC' UNIT WORK UNCNT 1
 DISP(NEW, DELETE, CATLG) PCTPRIME 100 MAXPRIME 100
DSNU1035I   027 15:47:29.90 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I    027 15:47:29.90 DSNUGUTC -  TEMPLATE BMC00007 DSN '&USERID..ARUDB070.SYSPUNCH' UNIT SYSALLDA VOLCNT 25
PCTPRIME 100
DSNU1035I   027 15:47:29.90 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I    027 15:47:29.90 DSNUGUTC -  REORG TABLESPACE ARUDB070.LARS$JBA REUSE COPYDDN(BMC00004, BMC00003)
RECOVERYDDN(BMC00002, BMC00001) UNLDDN(BMC00006) SHRLEVEL NONE UNLOAD CONTINUE STATISTICS TABLE ALL INDEX ALL
REPORT YES UPDATE ALL HISTORY ALL FORCEROLLUP YES PUNCHDDN BMC00007 SORTDEVT 3390 SORTNUM 12 PREFORMAT
DSNU1038I   027 15:47:32.36 DSNUGDYN - DATASET ALLOCATED.  TEMPLATE=BMC00004
                       DDNAME=SYS00003
                       DSN=ARU.LARREORG.DEDR.ARUDB070.LARS$JBA.BMCPY
DSNU1038I   027 15:47:32.49 DSNUGDYN - DATASET ALLOCATED.  TEMPLATE=BMC00003
                       DDNAME=SYS00004
                       DSN=ARU.LARREORG.DEDR.ARUDB070.LARS$JBA.BMCCPZ
DSNU1038I   027 15:47:32.54 DSNUGDYN - DATASET ALLOCATED.  TEMPLATE=BMC00002
                       DDNAME=SYS00005
                       DSN=ARU.LARREORG.DEDR.ARUDB070.LARS$JBA.BMRPY
DSNU1038I   027 15:47:32.69 DSNUGDYN - DATASET ALLOCATED.  TEMPLATE=BMC00001
                       DDNAME=SYS00006
                       DSN=ARU.LARREORG.DEDR.ARUDB070.LARS$JBA.BMCRCZ
DSNU1038I   027 15:47:33.04 DSNUGDYN - DATASET ALLOCATED.  TEMPLATE=BMC00006
                       DDNAME=SYS00007
                       DSN=ARU.LARREORG.DEDR.ARUDB070.SYSREC
DSNU3340I   027 15:47:33.24 DSNUGSRT - UTILITY PERFORMS DYNAMIC ALLOCATION OF SORT DISK SPACE
DSNU252I    027 15:47:35.06 DSNURULD - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS UNLOADED=2048 FOR TABLESPACE
ARUDB070.LARS$JBA
DSNU250I    027 15:47:35.07 DSNURULD - UNLOAD PHASE COMPLETE, ELAPSED TIME=00:00:01
DSNU3340I   027 15:47:42.20 DSNURPIB - UTILITY PERFORMS DYNAMIC ALLOCATION OF SORT DISK SPACE
DSNU3342I   027 15:47:42.51 DSNURPIB - NUMBER OF OPTIMAL SORT TASKS = 2, NUMBER OF ACTIVE SORT TASKS = 2
DSNU395I    027 15:47:42.51 DSNURPIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 6
DSNU400I    027 15:47:42.95 DSNURBID - COPY PROCESSED FOR TABLESPACE ARUDB070.LARS$JBA
                      NUMBER OF PAGES=315
                      AVERAGE PERCENT FREE SPACE PER PAGE = 25.45
                      PERCENT OF CHANGED PAGES =100.00
                      ELAPSED TIME=00:00:10
DSNU304I  *DEDR 027 15:47:43.74 DSNURWT - (RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=2048 FOR TABLE ARUDB070.LART001
DSNU302I    027 15:47:43.77 DSNURILD - (RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS PROCESSED=2048
DSNU300I    027 15:47:43.77 DSNURILD - (RE)LOAD PHASE COMPLETE, ELAPSED TIME=00:00:08
DSNU394I  *DEDR 027 15:47:44.57 DSNURBXA - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=2048 FOR INDEX ARUDB070.LARX001A
DSNU394I  *DEDR 027 15:47:44.60 DSNURBXA - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=2048 FOR INDEX ARUDB070.LARX001D
DSNU391I    027 15:47:44.74 DSNURPTB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 2
DSNU392I    027 15:47:44.74 DSNURPTB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:01
DSNU428I    027 15:47:45.64 DSNURORG - DB2 IMAGE COPY SUCCESSFUL FOR TABLESPACE ARUDB070.LARS$JBA
DSNU613I  *DEDR 027 15:47:46.32 DSNUSUTP - SYSTABLEPART CATALOG STATISTICS FOR ARUDB070.LARS$JBA PARTITION 0
                              CARD              = 2048
                              CARDF             = 2.048E+03
                              NEARINDREF        = 0
                              FARINDREF         = 0
                              PERCACTIVE        = 8
                              PERCDROP          = 0
                              PAGESAVE          = 0
                              SPACE             = 10080
                              SPACEF            = 1.008E+04
                              PQTY              = 2500
                              SQTY              = 1000
                              DSNUM             = 1
                              EXTENTS           = 1
```

Figure 40 SYSPRINT for example 10 (part 6 of 8)

```
DSNU610I  *DEDR 027 15:47:46.32 DSNUSUTP - SYSTABLEPART CATALOG UPDATE FOR ARUDB070.LARS$JBA SUCCESSFUL
DSNU614I  *DEDR 027 15:47:46.34 DSNUSUTB - SYSTABLES CATALOG STATISTICS FOR ARUDB070.LART001
                              CARD              = 2048
                              CARDF             = 2.048E+03
                              NPAGES            = 257
                              NPAGESF           = 2.57E+02
                              PCTPAGES          = 80
                              PCTROWCOMP        = 0
                              AVGROWLEN         = 449
                              SPACEF            = 1.28E+03
DSNU610I  *DEDR 027 15:47:46.35 DSNUSUTB - SYSTABLES CATALOG UPDATE FOR ARUDB070.LART001 SUCCESSFUL
DSNU615I  *DEDR 027 15:47:46.38 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C001_INTEGER
                              COLCARD           = 2048
                              COLCARDF          = 2.048E+03
                              HIGH2KEY          = X'008009C41E'
                              LOW2KEY           = X'0080000002'
DSNU615I  *DEDR 027 15:47:46.66 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C002_SMALLINT
                              COLCARD           = 1888
                              COLCARDF          = 1.888E+03
                              HIGH2KEY          = X'00A6E6'
                              LOW2KEY           = X'008008'
DSNU615I  *DEDR 027 15:47:46.66 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C003_REAL
                              COLCARD           = 1888
                              COLCARDF          = 1.888E+03
                              HIGH2KEY          = X'00C47FD500'
                              LOW2KEY           = X'00C21D0000'
DSNU615I  *DEDR 027 15:47:46.69 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C004_DOUBLE
                              COLCARD           = 2048
                              COLCARDF          = 2.048E+03
                              HIGH2KEY          = X'00C47FEF0000000000'
                              LOW2KEY           = X'00C21F000000000000'
DSNU615I  *DEDR 027 15:47:46.71 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C005_CHAR
                              COLCARD           = 2048
                              COLCARDF          = 2.048E+03
                              HIGH2KEY          = X'C1C4F9F6C1C1C1F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0'
                              LOW2KEY           = X'C1C1C1C3C1C1C1F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
                                                    F0F0F0F0F0F0F0F0F0F0F0F0F0F0'
DSNU615I  *DEDR 027 15:47:46.73 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C006_VARCHAR
                              COLCARD           = 1824
                              COLCARDF          = 1.824E+03
                              HIGH2KEY          = X'00F9F7F9C9D9D8D2D7E2F6D4C8F7'
                              LOW2KEY           = X'00C1C1E4C4C4F7F6F6C4F3C4D7E7F2C3D2D2F6F6C1C8C9E3E6E6E8D6F3C5
                                                    F6E2E2F6C1'
DSNU615I  *DEDR 027 15:47:46.75 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C007_DATE
                              COLCARD           = 1952
                              COLCARDF          = 1.952E+03
                              HIGH2KEY          = X'19981129'
                              LOW2KEY           = X'19100226'
DSNU615I  *DEDR 027 15:47:46.77 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C008_TIME
                              COLCARD           = 1888
                              COLCARDF          = 1.888E+03
                              HIGH2KEY          = X'235914'
                              LOW2KEY           = X'000215'
DSNU615I  *DEDR 027 15:47:46.78 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C010_DECIMAL
                              COLCARD           = 2048
                              COLCARDF          = 2.048E+03
                              HIGH2KEY          = X'00F0000000000327410000000000000000'
                              LOW2KEY           = X'00F0000000000000420000000000000000'
DSNU615I  *DEDR 027 15:47:46.79 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C011_LVARCHAR
                              COLCARD           = 2048
                              COLCARDF          = 2.048E+03
```

Figure 40 SYSPRINT for example 10 (part 7 of 8)

```
                              HIGH2KEY          = X'00F9F8E9F0C4C6E4E2D6D5C3E3D1E8F3F2F1D7E8E6E2F1F6E9C8C8D2E5C7
                                                    D1D2E5D6E7E4F0D6F9F0C6F6E4D4F4D6E3C8E7C3F4D3D8E3C3C5C6D1C7C8
                                                    F8C6C5C2D5F1C3F3E4C4F4D1E5C4E9D3F4C4D1C5F8F2C3D9C2F0E9C6E3E7
                                                    C9F3D6F0C2F5F3F5D5E9E9'
                              LOW2KEY           = X'00C1C1D8F7E9F8F6D5D2E3D1F4F4E4F0E8E9E3F5E8C8D3C1F7D4D8C9C2C4
                                                    F1C1E5D5F0C7F7F5D4D1C1D5C8D7D9E5E2C6C4F8F0F9F5D1F7D6C1E4D8C7
                                                    F9C3D1C9F0C4C4F3F4D8D5F6E8D5E6F9F7F7E2E4D1D8C2E2D5F4C8E8F6F5
                                                    F9F0E6E2D5C6D7C3F0C4D2'
DSNU610I  *DEDR 027 15:47:46.79 DSNUSUCO - SYSCOLUMNS CATALOG UPDATE FOR ARUDB070.LART001 SUCCESSFUL
DSNU612I  *DEDR 027 15:47:46.80 DSNUSUTS - SYSTABLESPACE CATALOG STATISTICS FOR ARUDB070.LARS$JBA
                              NACTIVE           = 2520
                              NACTIVEF          = 2.52E+03
DSNU610I  *DEDR 027 15:47:46.80 DSNUSUTS - SYSTABLESPACE CATALOG UPDATE FOR ARUDB070.LARS$JBA SUCCESSFUL
DSNU618I  *DEDR 027 15:47:47.01 DSNUSUIP - SYSINDEXPART CATALOG STATISTICS FOR ARUDB070.LARX001A PARTITION 0
                              CARD              = 2048
                              CARDF             = 2.048E+03
                              NEAROFFPOS        = 51
                              NEAROFFPOSF       = 5.1E+01
                              FAROFFPOS         = 0
                              FAROFFPOSF        = 0.0E0
                              LEAFDIST          = 20
                              LEAFNEAR          = 2
                              LEAFFAR           = 0
                              SPACE             = 4320
                              SPACEF            = 4.32E+03
                              DSNUM             = 1
                              EXTENTS           = 1
                              P_DEL_ENT         = 0
                              PQTY              = 1000
                              SQTY              = 500
DSNU610I  *DEDR 027 15:47:47.01 DSNUSUIP - SYSINDEXPART CATALOG UPDATE FOR ARUDB070.LARX001A SUCCESSFUL
DSNU618I  *DEDR 027 15:47:47.18 DSNUSUIP - SYSINDEXPART CATALOG STATISTICS FOR ARUDB070.LARX001D PARTITION 0
                              CARD              = 2048
                              CARDF             = 2.048E+03
                              NEAROFFPOS        = 1807
                              NEAROFFPOSF       = 1.807E+03
                              FAROFFPOS         = 240
                              FAROFFPOSF        = 2.4E+02
                              LEAFDIST          = 18
                              LEAFNEAR          = 2
                              LEAFFAR           = 0
                              SPACE             = 4320
                              SPACEF            = 4.32E+03
                              DSNUM             = 1
                              EXTENTS           = 1
                              P_DEL_ENT         = 0
                              PQTY              = 1000
                              SQTY              = 500
DSNU610I  *DEDR 027 15:47:47.18 DSNUSUIP - SYSINDEXPART CATALOG UPDATE FOR ARUDB070.LARX001D SUCCESSFUL
DSNU615I  *DEDR 027 15:47:47.20 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C009_TIMESTAMP
                              COLCARD           = 2048
                              COLCARDF          = 2.048E+03
                              HIGH2KEY          = X'44600702000000000000'
                              LOW2KEY           = X'19100202000000000000'
DSNU610I  *DEDR 027 15:47:47.20 DSNUSUCO - SYSCOLUMNS CATALOG UPDATE FOR ARUDB070.LARX001A SUCCESSFUL
DSNU617I  *DEDR 027 15:47:47.21 DSNUSUIX - SYSINDEXES CATALOG STATISTICS FOR ARUDB070.LARX001A
                              CLUSTERED         = Y
                              CLUSTERRATIO      = 100
                              CLUSTERRATIOF     = 1.0E+00
                              FIRSTKEYCARD      = 2048
                              FIRSTKEYCARDF     = 2.048E+03
                              FULLKEYCARD       = 2048
                              FULLKEYCARDF      = 2.048E+03
                              NLEAF             = 10
                              NLEVELS           = 2
                              SPACEF            = 4.32E+03
                              DATAREPEATFACTORF = 2.57E+02
DSNU610I  *DEDR 027 15:47:47.21 DSNUSUIX - SYSINDEXES CATALOG UPDATE FOR ARUDB070.LARX001A SUCCESSFUL
DSNU615I  *DEDR 027 15:47:47.24 DSNUSUCO - SYSCOLUMNS CATALOG STATISTICS FOR C009_TIMESTAMP
                              COLCARD           = 2048
                              COLCARDF          = 2.048E+03
                              HIGH2KEY          = X'44600702000000000000'
                              LOW2KEY           = X'19100202000000000000'
DSNU610I  *DEDR 027 15:47:47.24 DSNUSUCO - SYSCOLUMNS CATALOG UPDATE FOR ARUDB070.LARX001D SUCCESSFUL
DSNU617I  *DEDR 027 15:47:47.26 DSNUSUIX - SYSINDEXES CATALOG STATISTICS FOR ARUDB070.LARX001D
                              CLUSTERED         = N
```

Figure 40 SYSPRINT for example 10 (part 8 of 8)

```
                              CLUSTERRATIO      = 30
                              CLUSTERRATIOF     = 3.056640625E-01
                              FIRSTKEYCARD      = 2048
                              FIRSTKEYCARDF     = 2.048E+03
                              FULLKEYCARD       = 2048
                              FULLKEYCARDF      = 2.048E+03
                              NLEAF             = 11
                              NLEVELS           = 2
                              SPACEF            = 4.32E+03
                              DATAREPEATFACTORF = 1.972E+03
DSNU610I  *DEDR 027 15:47:47.26 DSNUSUIX - SYSINDEXES CATALOG UPDATE FOR ARUDB070.LARX001D SUCCESSFUL
DSNU616I  *DEDR 027 15:47:47.26 DSNUSUCD - SYSCOLDIST CATALOG STATISTICS FOR
                                       C009_TIMESTAMP
                              FREQUENCY           COLVALUE
                              ---------           --------
                              4.8828125E-04       X'44500702000000000000'
                              4.8828125E-04       X'40500302000000000000'
                              4.8828125E-04       X'41500202000000000000'
                              4.8828125E-04       X'42500102000000000000'
                              4.8828125E-04       X'37200402000000000000'
                              4.8828125E-04       X'36200502000000000000'
                              4.8828125E-04       X'32200102000000000000'
                              4.8828125E-04       X'31200202000000000000'
                              4.8828125E-04       X'35200602000000000000'
                              4.8828125E-04       X'30200302000000000000'
DSNU610I  *DEDR 027 15:47:47.28 DSNUSUCD - SYSCOLDIST CATALOG UPDATE FOR ARUDB070.LARX001D SUCCESSFUL
DSNU620I  *DEDR 027 15:47:47.29 DSNUSEOF - RUNSTATS CATALOG TIMESTAMP = 2011-01-27-15.47.42.082113
DSNU010I    027 15:47:47.76 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
BMC50004I DSNUTILB PHASE COMPLETE.  ELAPSED TIME = 00:00:20

BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Example 11: Partition-by-growth table space

In this example, REORG PLUS reorganizes a partition-by-growth table space. It is a SHRLEVEL NONE, two-phase, full table space reorganization. During the reorganization, REORG PLUS adds new partitions, indicated by message BMC50175I.

Message 50041I indicates the status of zIIP processing. For this example, the ZIIP installation option is ENABLED, but an XBM subsystem was not specified. REORG PLUS automatically located an available XBM subsystem to provide zIIP processing.

Table 74 describes the key command options for this job.

Table 74 Key command options used in example 11 (part 1 of 2)

Command options used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
UNLOAD CONTINUE	tells REORG PLUS to continue the reorganization with two-phase processing after the UNLOAD phase has unloaded the data
MAXNEWPARTS	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

Table 74 Key command options used in example 11 (part 2 of 2)

Command options used in JCL	Description
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.
COPYLVL PART	allocates an image copy data set for each partition that you are reorganizing
REDEFINE YES	tells REORG PLUS to delete and redefine the VSAM data sets containing the DB2 objects
ORDER YES	sorts the data rows
BMCSTATS YES UPDATEDB2STATS YES	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. Statistics are not included for the partitions that REORG PLUS adds during this job.

Figure 41 shows the JCL for example 11.

Figure 41 JCL for example 11 (part 1 of 2)

```
//        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='DEDR,RDAJRGD3.RG1,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//*
//SYSIN     DD *
REORG TABLESPACE RDAJRGD3.LARSZABA
 SHRLEVEL NONE
 UNLOAD CONTINUE
 MAXNEWPARTS 10
 COPY YES
 REGISTER ALL
 COPYLVL PART
 ORDER YES
 ANALYZE SAMPLE
 KEEPDICTIONARY NO
 PREFORMAT YES
```

Figure 41 JCL for example 11 (part 2 of 2)

```
 UPDATEDB2STATS YES
 BMCSTATS YES
 DDTYPE UNLOAD   UNIT (WORK)
 DSNPAT 'ARU.EXMPL11.&DB.&TSIX.&DDNAME'
 DDTYPE WORK     UNIT (WORK)
 DSNPAT 'ARU.EXMPL11.&DB.&TSIX.&DDNAME'
 DDTYPE LOCPFCPY UNIT (WORK)
 DSNPAT 'ARU.EXMPL11.&DB.&TSIX.&DDNAME'
/*
```

Figure 42 shows the SYSPRINT output for example 11.

Figure 42 SYSPRINT for example 11 (part 1 of 9)

```
                                *****  B M C   R E O R G   P L U S   F O R   D B 2    V10R1.00  *****
                                              (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
                REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
 BMC50001I UTILITY EXECUTION STARTING    1/27/2011   15:52:13 ...
 BMC50002I UTILITY ID = 'RDAJRGD3.RG1'.  DB2 SUBSYSTEM ID = 'DEDR'.  OPTION MODULE = 'ARU$OPTS'.
 BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
 BMC50471I REGION=0M,BELOW 16M=8852K,ABOVE 16M=1409864K,IEFUSI=NO,CPUS=3
 BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

 BMC50471I REORG PLUS FOR DB2--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I SOLUTION COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMCSORT ENGINE--V02.03.01
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMC STATS API--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT

 BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
 BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
 BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
 BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
 BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(OK,OK)
 BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
 BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
 BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
 BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
 BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
 BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
 BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
 BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
 BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
 BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
 BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
 BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
 BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
 BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
 BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
 BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
 BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
 BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
 BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
 BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
 BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
 BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
 BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
 BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
 BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
 BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
 BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
 BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
 BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
```

Figure 42 SYSPRINT for example 11 (part 2 of 9)

```
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
```

Figure 42 SYSPRINT for example 11 (part 3 of 9)

```
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DEDR
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)
```

Figure 42 SYSPRINT for example 11 (part 4 of 9)

```
BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'
BMC50471I ...EXCEPTIONS  ='ASU101.EXCEPTIONS2'

BMC50102I REORG TABLESPACE RDAJRGD3.LARSZABA
BMC50102I  SHRLEVEL NONE
BMC50102I  UNLOAD CONTINUE
BMC50102I  MAXNEWPARTS 10
BMC50102I  COPY YES
BMC50102I  REGISTER ALL
BMC50102I  COPYLVL PART
BMC50102I  ORDER YES
BMC50102I  ANALYZE SAMPLE
BMC50102I  KEEPDICTIONARY NO
BMC50102I  PREFORMAT YES
BMC50102I  UPDATEDB2STATS YES
BMC50102I  BMCSTATS YES
BMC50102I  DDTYPE UNLOAD   UNIT (WORK)
BMC50102I  DSNPAT 'ARU.EXMPL11.&DB.&TSIX.&DDNAME'
BMC50102I  DDTYPE WORK     UNIT (WORK)
BMC50102I  DSNPAT 'ARU.EXMPL11.&DB.&TSIX.&DDNAME'
BMC50102I  DDTYPE LOCPFCPY UNIT (WORK)
BMC50102I  DSNPAT 'ARU.EXMPL11.&DB.&TSIX.&DDNAME'

BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:04

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 1: SAMPLING STATISTICS: PART=1,TP=54308,SP=208,SR=4228,AVGR=52,SD=66,SE=0,AVGF=2032,SD=65,SE=138,EP=102
BMC51301I 1: SAMPLING STATISTICS: PART=1,TP=54308,SP=415,SR=8466,AVGR=52,SD=66,SE=0,AVGF=2040,SD=65,SE=98,EP=203
BMC51301I 1: SAMPLING STATISTICS: PART=1,TP=54308,SP=623,SR=12004,AVGR=52,SD=66,SE=0,AVGF=1926,SD=65,SE=79,EP=322
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A001'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 1045336  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 42
BMC50484I ESTIMATED CARDINALITY OF PART 0001 = 130986  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 42
BMC50484I ESTIMATED CARDINALITY OF PART 0002 = 130986  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 42
BMC50484I ESTIMATED CARDINALITY OF PART 0003 = 130986  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 42
BMC50484I ESTIMATED CARDINALITY OF PART 0004 = 130986  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 42
BMC50484I ESTIMATED CARDINALITY OF PART 0005 = 130986  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 42
BMC50484I ESTIMATED CARDINALITY OF PART 0006 = 130986  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 42
BMC50484I ESTIMATED CARDINALITY OF PART 0007 = 130986  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 42
BMC50484I ESTIMATED CARDINALITY OF PART 0008 = 128434  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 42
BMC51264I UNLOAD WILL READ 54275 DATA PAGES FROM SPACE 'RDAJRGD3.LARSZABA'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGEXM11  &STEPNAME   LARREORG  &DB         RDAJRGD3  &TSIX       LARSZABA  &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       155213    &SSID       DEDR      &UTIL       RDAJRGD3
BMC50425I &UTILPFX    RDAJRGD3  &UTILSFX    RG1       &DATE8      01272011  &GRPNM      DEDR      &VCAT       DEDRCAT
BMC50425I &TIME4      1552      &DATEJ      2011027   &JDATE      11027

BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                       UNIT OR                      KBYTES   KBYTES     ALOC      ALOC
BMC50447I DDNAME   DSNAME                                       DATACLAS MGMTCLAS STORCLAS      PRI      SEC      PRI       SEC

BMC50448I BMCCPY01 ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY01      *WORK                        1048572   104857    21846      2185  TRK
BMC50448I BMCCPY02 ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY02      *WORK                        1048572   104857    21846      2185  TRK
BMC50448I BMCCPY03 ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY03      *WORK                        1048572   104857    21846      2185  TRK
BMC50448I BMCCPY04 ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY04      *WORK                        1048572   104857    21846      2185  TRK
BMC50448I BMCCPY05 ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY05      *WORK                        1048572   104857    21846      2185  TRK
```

Figure 42 SYSPRINT for example 11 (part 5 of 9)

```
BMC50448I BMCCPY06 ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY06      *WORK                          1048572   104857    21846      2185  TRK
BMC50448I BMCCPY07 ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY07      *WORK                          1048572   104857    21846      2185  TRK
BMC50448I BMCCPY08 ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY08      *WORK                          1048572   104857    21846      2185  TRK
BMC50448I BMCCPY09 ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY09      *WORK                          1048572   104857    21846      2185  TRK
BMC50448I BMCCPY10 ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY10      *WORK                          1048572   104857    21846      2185  TRK
BMC50448I SYSREC01 ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC01      *WORK                             8187     1279      152        24  TRK
BMC50448I SYSREC02 ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC02      *WORK                             8187     1279      152        24  TRK
BMC50448I SYSREC03 ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC03      *WORK                             8187     1279      152        24  TRK
BMC50448I SYSREC04 ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC04      *WORK                             8187     1279      152        24  TRK
BMC50448I SYSREC05 ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC05      *WORK                             8187     1279      152        24  TRK
BMC50448I SYSREC06 ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC06      *WORK                             8187     1279      152        24  TRK
BMC50448I SYSREC07 ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC07      *WORK                             8187     1279      152        24  TRK
BMC50448I SYSREC08 ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC08      *WORK                             8028     1254      149        24  TRK
BMC50448I SYSREC09 ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC09      *WORK                                1        1        1         1  TRK
BMC50448I SYSREC10 ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC10      *WORK                                1        1        1         1  TRK
BMC50448I SYSUT1   ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSUT1        *WORK                            17362     1737      322        33  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50474I BELOW 16M = 8160K, ABOVE 16M = 1394200K, CPUS = 3
BMC50479I TOTAL PAGES: 2238323, ALLOWED: 0; AVAILABLE PAGES: 359913, ALLOWED: 359913
BMC51302I MAX TASKS = 1, MAX PARTITIONS PER TASK = 10, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 10
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 2068K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 1081344, I/O WAITS = 199 ,DDNAME = SYS00006
BMC50367I COMPRESSION DICTIONARY BUILT FOR TABLESPACE RDAJRGD3.LARSZABA PARTITION 1
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:09
BMC50481I 1: UNLOAD TASK COMPLETE.  ELAPSED TIME = 00:00:21
BMC50476I DDNAME = SYSREC01, I/OS = 28, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC02, I/OS = 28, I/O WAITS = 12, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC03, I/OS = 28, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC04, I/OS = 28, I/O WAITS = 6, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC05, I/OS = 28, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC06, I/OS = 28, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC07, I/OS = 28, I/O WAITS = 11, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC08, I/OS = 28, I/O WAITS = 9, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC09, I/OS = 8, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC10, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT1, I/OS = 56, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51271I UNLOAD STATISTICS:  130986 ROWS/KEYS UNLOADED FROM PARTITION 1
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 1
BMC51286I RELOADING OF DATASET 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A001' WILL REQUIRE 262142 PAGES
BMC51271I UNLOAD STATISTICS:  130986 ROWS/KEYS UNLOADED FROM PARTITION 2
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 2
BMC51286I RELOADING OF DATASET 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A002' WILL REQUIRE 262142 PAGES
BMC51271I UNLOAD STATISTICS:  130986 ROWS/KEYS UNLOADED FROM PARTITION 3
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 3
BMC51286I RELOADING OF DATASET 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A003' WILL REQUIRE 262142 PAGES
BMC51271I UNLOAD STATISTICS:  130986 ROWS/KEYS UNLOADED FROM PARTITION 4
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 4
BMC51286I RELOADING OF DATASET 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A004' WILL REQUIRE 262142 PAGES
BMC51271I UNLOAD STATISTICS:  130986 ROWS/KEYS UNLOADED FROM PARTITION 5
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 5
BMC51286I RELOADING OF DATASET 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A005' WILL REQUIRE 262142 PAGES
BMC51271I UNLOAD STATISTICS:  130986 ROWS/KEYS UNLOADED FROM PARTITION 6
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 6
BMC51286I RELOADING OF DATASET 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A006' WILL REQUIRE 262142 PAGES
BMC51271I UNLOAD STATISTICS:  130986 ROWS/KEYS UNLOADED FROM PARTITION 7
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 7
BMC51286I RELOADING OF DATASET 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A007' WILL REQUIRE 262142 PAGES
BMC51271I UNLOAD STATISTICS:  130986 ROWS/KEYS UNLOADED FROM PARTITION 8
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 8
BMC51286I RELOADING OF DATASET 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A008' WILL REQUIRE 262142 PAGES
BMC51271I UNLOAD STATISTICS:  33456 ROWS/KEYS UNLOADED FROM PARTITION 9
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 9
BMC51286I RELOADING OF DATASET 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A009' WILL REQUIRE 66968 PAGES
BMC51288I RELOADING OF DATASET 'DEDRCAT.DSNDBD.RDAJRGD3.LARX003A.I0001.A001' MAY REQUIRE 5386 PAGES
BMC51272I UNLOAD STATISTICS:  1081344 ROWS UNLOADED FROM SPACE 'RDAJRGD3.LARSZABA', 0 ROWS DISCARDED, 0 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR SPACE 'RDAJRGD3.LARSZABA'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I UNLOAD PHASE COMPLETE.  ELAPSED TIME = 00:00:41

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50175I PARTITION  2 ADDED TO PARTITION BY GROWTH OBJECT RDAJRGD3.LARSZABA
BMC50175I PARTITION  3 ADDED TO PARTITION BY GROWTH OBJECT RDAJRGD3.LARSZABA
BMC50175I PARTITION  4 ADDED TO PARTITION BY GROWTH OBJECT RDAJRGD3.LARSZABA
BMC50175I PARTITION  5 ADDED TO PARTITION BY GROWTH OBJECT RDAJRGD3.LARSZABA
BMC50175I PARTITION  6 ADDED TO PARTITION BY GROWTH OBJECT RDAJRGD3.LARSZABA
```

Figure 42 SYSPRINT for example 11 (part 6 of 9)

```
BMC50175I PARTITION  7 ADDED TO PARTITION BY GROWTH OBJECT RDAJRGD3.LARSZABA
BMC50175I PARTITION  8 ADDED TO PARTITION BY GROWTH OBJECT RDAJRGD3.LARSZABA
BMC50175I PARTITION  9 ADDED TO PARTITION BY GROWTH OBJECT RDAJRGD3.LARSZABA
BMC50474I BELOW 16M = 8128K, ABOVE 16M = 1393232K, CPUS = 3
BMC50479I TOTAL PAGES: 2224867, ALLOWED: 0; AVAILABLE PAGES: 346044, ALLOWED: 346044
BMC51303I MAX TASKS = 1, INDEXES PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A005'
BMC50476I DDNAME = SYSUT1, I/OS = 55, I/O WAITS = 13, RDB LOCK WAITS = 0
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:03
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:06  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARX003A.I0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 1081344, I/O WAITS = 7 ,DDNAME = SYS00013
BMC51276I BUILD  STATISTICS:  1081344 KEYS LOADED INTO INDEX 'RDAJRGD3.LARX003A'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A003'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A006'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A007'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A008'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A002'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A004'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A001'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A009'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A009'
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:00:15  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A009'
BMC50477I 0: PARTITION =    9, ROWS/KEYS = 33456, I/O WAITS = 165 ,DDNAME = SYS00037
BMC51274I RELOAD STATISTICS:  33456 ROWS/KEYS LOADED INTO PARTITION 9
BMC50476I DDNAME = BMCCPY09, I/OS = 671, I/O WAITS = 621, RDB LOCK WAITS = 0
BMC50376I 67009 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY09'
BMC50476I DDNAME = SYSREC09, I/OS = 6, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A005'
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:04:40  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A005'
BMC50477I 0: PARTITION =    5, ROWS/KEYS = 130986, I/O WAITS = 397 ,DDNAME = SYS00010
BMC51274I RELOAD STATISTICS:  130986 ROWS/KEYS LOADED INTO PARTITION 5
BMC50476I DDNAME = BMCCPY05, I/OS = 2623, I/O WAITS = 2537, RDB LOCK WAITS = 0
BMC50376I 262297 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY05'
BMC50476I DDNAME = SYSREC05, I/OS = 26, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A003'
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:04:40  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A003'
BMC50477I 0: PARTITION =    3, ROWS/KEYS = 130986, I/O WAITS = 442 ,DDNAME = SYS00016
BMC51274I RELOAD STATISTICS:  130986 ROWS/KEYS LOADED INTO PARTITION 3
BMC50476I DDNAME = BMCCPY03, I/OS = 2623, I/O WAITS = 2494, RDB LOCK WAITS = 0
BMC50376I 262297 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY03'
BMC50476I DDNAME = SYSREC03, I/OS = 26, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A006'
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:04:11  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A006'
BMC50477I 0: PARTITION =    6, ROWS/KEYS = 130986, I/O WAITS = 444 ,DDNAME = SYS00019
BMC51274I RELOAD STATISTICS:  130986 ROWS/KEYS LOADED INTO PARTITION 6
BMC50476I DDNAME = BMCCPY06, I/OS = 2623, I/O WAITS = 2375, RDB LOCK WAITS = 0
BMC50376I 262297 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY06'
BMC50476I DDNAME = SYSREC06, I/OS = 26, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A007'
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:04:39  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A007'
BMC50477I 0: PARTITION =    7, ROWS/KEYS = 130986, I/O WAITS = 689 ,DDNAME = SYS00022
BMC51274I RELOAD STATISTICS:  130986 ROWS/KEYS LOADED INTO PARTITION 7
BMC50476I DDNAME = BMCCPY07, I/OS = 2623, I/O WAITS = 2288, RDB LOCK WAITS = 0
BMC50376I 262297 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY07'
BMC50476I DDNAME = SYSREC07, I/OS = 26, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A008'
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:06:40  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A008'
BMC50477I 0: PARTITION =    8, ROWS/KEYS = 130986, I/O WAITS = 356 ,DDNAME = SYS00025
BMC51274I RELOAD STATISTICS:  130986 ROWS/KEYS LOADED INTO PARTITION 8
BMC50476I DDNAME = BMCCPY08, I/OS = 2623, I/O WAITS = 2501, RDB LOCK WAITS = 0
BMC50376I 262297 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY08'
BMC50476I DDNAME = SYSREC08, I/OS = 26, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A002'
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:07:40  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A002'
BMC50477I 0: PARTITION =    2, ROWS/KEYS = 130986, I/O WAITS = 367 ,DDNAME = SYS00028
BMC51274I RELOAD STATISTICS:  130986 ROWS/KEYS LOADED INTO PARTITION 2
BMC50476I DDNAME = BMCCPY02, I/OS = 2623, I/O WAITS = 2482, RDB LOCK WAITS = 0
BMC50376I 262297 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY02'
BMC50476I DDNAME = SYSREC02, I/OS = 26, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A004'
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:08:01  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A004'
BMC50477I 0: PARTITION =    4, ROWS/KEYS = 130986, I/O WAITS = 693 ,DDNAME = SYS00031
BMC51274I RELOAD STATISTICS:  130986 ROWS/KEYS LOADED INTO PARTITION 4
BMC50476I DDNAME = BMCCPY04, I/OS = 2623, I/O WAITS = 2255, RDB LOCK WAITS = 0
BMC50376I 262297 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY04'
```

Figure 42 SYSPRINT for example 11 (part 7 of 9)

```
 BMC50476I DDNAME = SYSREC04, I/OS = 26, I/O WAITS = 1, RDB LOCK WAITS = 0
 BMC50375I INLINE COPY STARTED FOR DATASET = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A001'
 BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:08:21  DSN = 'DEDRCAT.DSNDBD.RDAJRGD3.LARSZABA.I0001.A001'
 BMC50477I 0: PARTITION =    1, ROWS/KEYS = 130986, I/O WAITS = 468 ,DDNAME = SYS00034
 BMC51274I RELOAD STATISTICS:  130986 ROWS/KEYS LOADED INTO PARTITION 1
 BMC50476I DDNAME = BMCCPY01, I/OS = 2623, I/O WAITS = 2331, RDB LOCK WAITS = 0
 BMC50376I 262297 PAGES COPIED TO DATASET = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY01'
 BMC50476I DDNAME = SYSREC01, I/OS = 26, I/O WAITS = 1, RDB LOCK WAITS = 0
 BMC51275I RELOAD STATISTICS:  1081344 ROWS LOADED INTO SPACE 'RDAJRGD3.LARSZABA'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC10', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC10'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC01', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC01'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC04', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC04'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC02', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC02'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC08', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC08'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC07', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC07'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC06', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC06'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC03', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC03'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC05', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC05'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC09', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSREC09'
 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT1  ', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.SYSUT1'
 BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
 BMC50004I RELOAD PHASE COMPLETE.  ELAPSED TIME = 00:12:01

 BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'BMCCPY10', DSNAME = 'ARU.EXMPL11.RDAJRGD3.LARSZABA.BMCCPY10'
1BMCSTATS V10.1.0 REPORT FOR DEDR V910              TIME 2011-01-27-15.52.59.608616                                             1


   INDEX --------------- RDAJRGD3.LARX003A
     ASSOCIATIONS
       TABLESPACE ------ RDAJRGD3.LARSZABA
       TABLE ----------- RDAJRGD3.LART003
     HISTORY
       SAVESTATS--------------------Y  UPDATEDB2--------------------Y  UTILCODE----------------(NULL)  LOCATION------------------DEDR
       SAMPLING---------------------N
     ATTRIBUTES
       PIECESIZE--------------4194304  TYPE TS----------------------G  UNIQUERULE-------------------D  KEYLENGTH-------------------10
       SUBPAGES---------------------0  INDEXTYPE--------------------2  CLUSTERING-------------------Y  COLCOUNT---------------------1
       INDEXSPACE------------LARX003A  PGSIZE-----------------------4  EXTTYPE----------------(BLANK)  COMPRESS---------------------N
       COLNAME---------C006_TIMESTAMP
     STATISTICS
       FIRSTKEYCARD-----------1081344  FULLKEYCARD------------1081344  NACTIVE------------------10620  TBCARDF----------------1081344
       NUMNONLEAF------------------20  NLEVELS----------------------3  CLUSTERRATIO---------------100  NLEAF---------------------5380
       AVGKEYLEN-------------------10  REORGSPACE-----------------465  REORGSPACE_KB------------22320  REORGNLEVELS-----------------3
       HIGH2K-----X'4460062522000000'  HIGH2K (CHAR)---------.-......  LOW2K------X'1910010101000000'  LOW2K (CHAR)----------........
       DATAREPEAT-------------1081344
     ALLOCATION
       SPACE----------------------885  SPACE_KB-----------------42480  NUMDATASETS------------------1

   INDEXPART ----------- RDAJRGD3.LARX003A
     ASSOCIATIONS
       TABLESPACE ------ RDAJRGD3.LARSZABA
       TABLE ----------- RDAJRGD3.LART003
     HISTORY
       SAVESTATS--------------------Y  UPDATEDB2--------------------Y  UTILCODE----------------(NULL)  LOCATION----------------(NULL)
       SAMPLING---------------------N
     ATTRIBUTES
       PQTY------------------------59  SQTY------------------------59  PCTFREE----------------------0  FREEPAGE---------------------0
       ALLOCUNIT--------------------C  STORTYPE---------------------I  NLAVGKEYLEN------------------5  VCATNAME---------------DEDRCAT
       STORNAME--------------ARULARSG
     STATISTICS
       FIRSTKEYCARD-----------1081344  FULLKEYCARD------------1081344  CARD-------------------1081344  NACTIVE------------------10620
       CLUSTERRATIO---------------100  NUMNONLEAF------------------20  NLEVELS----------------------3  FAROFFPOS--------------------8
       NEAROFFPOS-------------1081335  LEAFDIST---------------------0  NLEAF---------------------5380  FREE------------------------69
       FULL----------------------5397  AVGKEYLEN-------------------10
       LEAFNEAR--------------------19  LEAFFAR----------------------0  PSEUDO_DEL_RIDS--------------0  REORGSPACE-----------------465
       REORGSPACE_KB------------22320  REORGNLEVELS-----------------3  PCTUSED--------------------100  PQTYROWS---------------2126379
       DATAREPEAT-------------1081344
     ALLOCATION
       IPREFIX----------------------I  SPACE----------------------885  SPACE_KB-----------------42480  NUMDATASETS------------------1
       EXTENTS----------------------2  VOLCOUNT---------------------1  DEVTYPE-------------------3390  VOLUME------------------TM0002

   COLUMN -------------- C006_TIMESTAMP
     ASSOCIATIONS
       TABLESPACE ------ RDAJRGD3.LARSZABA
       TABLE ----------- RDAJRGD3.LART003
```

Figure 42 SYSPRINT for example 11 (part 8 of 9)

```
     ATTRIBUTES
       COLNO------------------------6  COLTYPE---------------TIMESTMP  LENGTH----------------------10  NULLS------------------------N
     STATISTICS
       COLCARD----------------1081344  COLNULLS---------------------0
       COLAVG----------------------10  COLMIN----------------------10  COLMAX----------------------10
       HIGH2K-----X'4460062522000000'  HIGH2K (CHAR)---------.-......  LOW2K------X'1910010101000000'  LOW2K (CHAR)----------........

       MOST FREQUENT VALUES
1BMCSTATS V10.1.0 REPORT FOR DEDR V910              TIME 2011-01-27-15.52.59.608616                                            2

        FREQUENCY---------9.247751E-07     COLVAL-X'19100101190000000000'
                                           (CHAR)--------------..........
        FREQUENCY---------9.247751E-07     COLVAL-X'19100101180000000000'
                                           (CHAR)--------------..........
        FREQUENCY---------9.247751E-07     COLVAL-X'19100101170000000000'
                                           (CHAR)--------------..........
        FREQUENCY---------9.247751E-07     COLVAL-X'19100101160000000000'
                                           (CHAR)--------------..........
        FREQUENCY---------9.247751E-07     COLVAL-X'19100101150000000000'
                                           (CHAR)--------------..........
        FREQUENCY---------9.247751E-07     COLVAL-X'19100101140000000000'
                                           (CHAR)--------------..........
        FREQUENCY---------9.247751E-07     COLVAL-X'19100101130000000000'
                                           (CHAR)--------------..........
        FREQUENCY---------9.247751E-07     COLVAL-X'19100101120000000000'
                                           (CHAR)--------------..........
        FREQUENCY---------9.247751E-07     COLVAL-X'19100101110000000000'
                                           (CHAR)--------------..........
        FREQUENCY---------9.247751E-07     COLVAL-X'19100101100000000000'
                                           (CHAR)--------------..........
1BMCSTATS V10.1.0 REPORT FOR DEDR V910              TIME 2011-01-27-15.52.59.608616                                            1


 TABLESPACE ------------ RDAJRGD3.LARSZABA
     HISTORY
       SAVESTATS--------------------Y  UPDATEDB2--------------------Y  UTILCODE----------------(NULL)  LOCATION------------------DEDR
       SAMPLING---------------------N
     ATTRIBUTES
       PARTITIONS-------------------1  NTABLES----------------------1  PGSIZE-----------------------4  SEGSIZE----------------------4
       DSSIZE-----------------1048576
       MAXROWS----------------------1  TYPE-------------------------G  ENCODING---------------------E  MAXPARTITIONS---------------10
       LOB--------------------------N  XML--------------------------N  MEMBER_CLUSTER---------(BLANK)
     STATISTICS
       NACTIVE-----------------262143  REORGSPACE---------------21870  REORGSPACE_KB----------1049760  ROWMAXFOUND-----------------57
       ROWMINFOUND-----------------33  ROWAVG----------------------41  REORGSPACE_PARTS-------------2
     ALLOCATION
       SPACE--------------------21855  SPACE_KB---------------1049040

   TABLEPART ----------- RDAJRGD3.LARSZABA PART 001
     HISTORY
       SAVESTATS--------------------Y  UPDATEDB2--------------------Y  LOCATION----------------(NULL)  UTILCODE----------------(NULL)
       SAMPLING---------------------N
     ATTRIBUTES
       PQTY-----------------------100  SQTY-----------------------100  PCTFREE---------------------50  FREEPAGE---------------------1
       ALLOCUNIT--------------------C  STORTYPE---------------------I  COMPRESS---------------------Y  TRACKMOD---------------(BLANK)
       FORMAT---------------------RRF  VCATNAME---------------DEDRCAT  STORNAME--------------ARULARSG
     STATISTICS
       CARD--------------------130986  NACTIVE-----------------262143  NPAGES------------------130986  ROWAVG----------------------41
       ROWMAXFOUND-----------------57  ROWMINFOUND-----------------33  AVGNONCOMPROWLEN----------(-1)  DIRTY------------------------0
       FULL-------------------------0  FARINDREF--------------------0  NEARINDREF-------------------0  PERCACTIVE-------------------0
       PCTPAGES--------------------49  PERCDROP---------------------0  REORGSPACE---------------21855  REORGSPACE_KB----------1049040
       PCTUSED--------------------100  PQTYROWS------------------8986  NDICTIONARY-----------------16  PCTROWCOMP-----------------100
       PAGESAVE---------------------0
     ALLOCATION
       IPREFIX----------------------I  SPACE--------------------21855  SPACE_KB---------------1049040  NUMDATASETS------------------1
       EXTENTS---------------------18  VOLCOUNT---------------------4  DEVTYPE-------------------3390  VOLUME------------------TM0002

   TABLE --------------- RDAJRGD3.LART003
     ASSOCIATIONS
       TABLESPACE ------ RDAJRGD3.LARSZABA
     HISTORY
       SAVESTATS--------------------Y  UPDATEDB2--------------------Y  UTILCODE----------------(NULL)  LOCATION----------------(NULL)
```

Figure 42 SYSPRINT for example 11 (part 9 of 9)

```
     STATISTICS
       CARD--------------------130986  NPAGES------------------130986  SPACE_KB---------------1049040  AVGNONCOMPROWLEN----------(-1)
       ROWAVG----------------------41  ROWMAXFOUND-----------------57  ROWMINFOUND-----------------33  INDREF-----------------------0
       PCTPAGES-------------------100  PCTROWCOMP-----------------100
 BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
 BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Example 12: SHRLEVEL CHANGE with DDLIN and online repartitioning

In this example, REORG PLUS reorganizes a partitioned table space with three partitions. The table has one clustering index and one nonpartitioned secondary index. This example uses SHRLEVEL CHANGE to keep the objects in RW status during most of the reorganization.

In the LOGAPPLY phase, REORG PLUS determines the rate at which it is applying log records. If REORG PLUS estimates that it can finish applying the log records (LOGFINAL phase) by the time specified by the DEADLINE option, and without exceeding the maximum time specified by MAXRO, REORG PLUS moves from the LOGAPPLY phase into the LOGFINAL phase and completes the reorganization.

The DDLIN data set supplies the ALTER INDEX commands. REORG PLUS performs online partition rebalancing in the staging data sets, and performs the ALTER commands in the UTILTERM phase.

Message 50041I indicates the status of zIIP processing. For this example, an XBM subsystem ID was specified for SHRLEVEL CHANGE processing. REORG PLUS uses the same XBM subsystem ID for zIIP processing.

Table 75 describes the key command options, installation options, and DD statements for this job.

Table 75 Command options, installation options, and DD statements used in example 12 (part 1 of 3)

Command options and DD statements used in JCL; installation options	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
SHRLEVEL CHANGE	specifies the reorganization type REORG PLUS will keep the original data sets in their original status (which can be RW) as long as possible.
UNLOAD CONTINUE	tells REORG PLUS to continue the reorganization with two-phase processing after the UNLOAD phase has unloaded the data

Table 75 Command options, installation options, and DD statements used in example 12 (part 2 of 3)

Command options and DD statements used in JCL; installation options	Description
XBMID XBMA	specifies to use the XBM subsystem with the SSID of XBMA
COPY YES	creates full and incremental DB2 image copies of the table space SHRLEVEL CHANGE sets REGISTER ALL, so all copies are registered in SYSIBM.SYSCOPY.
ORDER YES	sorts the rows by table as well as by each table's clustering key If no clustering key exists, x'00's are used.
DEADLINE	continues the reorganization only if REORG PLUS estimates that it can finish LOGFINAL processing by the time and date specified in the timestamp The UTILTERM phase can continue past the deadline.
MAXRO 600	starts the LOGFINAL phase when REORG PLUS estimates that it can apply the remaining log records within 10 minutes (600 seconds) This estimate does not include the time needed to produce the incremental image copies and complete the UTILTERM phase processing.
LONGLOG CONTINUE (installation option)	tells REORG PLUS to continue the reorganization if the longlog condition still exists after the DELAY time expires
DELAY 300	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.
RIDMAPMEM 0 (installation option)	REORG PLUS automatically calculates the RIDMAPMEM value to be used to store the RID translation maps.
LOGMEM 0 (installation option)	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
LOGSPILL (1024, 1024)	determines the primary and secondary space allocation (in kilobytes) for the spill data sets that hold the log records Each number can be a nonzero positive integer.
SPILLDSNPAT	tells REORG PLUS to use a particular pattern of variables and text to create a prefix for the spill data set 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.

Table 75 Command options, installation options, and DD statements used in example 12 (part 3 of 3)

Command options and DD statements used in JCL; installation options	Description
SPILLUNIT	specifies the DASD unit where REORG PLUS can allocate spill data sets The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.
SPILLSTORCLAS	specifies the SMS storage class that REORG PLUS uses to allocate spill data sets The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.
ANALYZE	gathers information about the objects that you are reorganizing and provides estimated sizes for SYSREC, SYSUT1, SORTWK, and copy data sets
SIZEPCT (150,150)	tells REORG PLUS to allocate 150% of the data set size that it calculated for the UNLOAD, WORK, and SORTWORK files The increase allows room for the unknown number of updates that are being applied to the data sets.
DSNPAT	specifies a pattern of variables and text that REORG PLUS uses to create data set names for dynamic data set allocation, overriding the default pattern
DDLIN	DDLIN data set contains the SQL ALTER INDEX and ALTER TABLESPACE statements with the new limit key values to use to rebalance partitions

//BMCCPY
//BMCCPZ
//BMCRCY
//BMCRCZ
	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.

//BMCICY
//BMCICZ
//BMCIRY
//BMCIRZ
	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.

Figure 43 shows the JCL for example 12.

Figure 43 JCL for example 12 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  PARTITIONED TABLESPACE, CLUSTERING INDEX AND NONPARTITIONED      *
//*  SECONDARY INDEX.                                                 *
//*  REORG TABLESPACE WITH SHRLEVEL CHANGE OPTION.                    *
//*  ALTER PARTITIONING INDEX KEYS USING DDLIN OPTION.                *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//LARREORG EXEC PGM=ARUUMAIN,COND=(7,LT),
// PARM='DHV,LARCOPB.LARDBXAB,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//*
//BMCCPY    DD DSN=ARU.EXMPL12.BMCCPY,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCCPZ    DD DSN=ARU.EXMPL12.BMCCPZ,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCRCY    DD DSN=ARU.EXMPL12.BMCRCY,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCRCZ    DD DSN=ARU.EXMPL12.BMCRCZ,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCICY    DD DSN=ARU.EXMPL12.BMCICY,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCICZ    DD DSN=ARU.EXMPL12.BMCICZ,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCIRY    DD DSN=ARU.EXMPL12.BMCIRY,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//BMCIRZ    DD DSN=ARU.EXMPL12.BMCIRZ,
//  DISP=(,CATLG),
//  SPACE=(CYL,(50,20),RLSE),UNIT=WORK
//*
//SYSIN     DD *
REORG TABLESPACE LARDBXAB.LARSXABA
  SHRLEVEL CHANGE
  UNLOAD CONTINUE
  XBMID XBMA
  FASTSWITCH YES
```

Figure 43   JCL for example 12 (part 2 of 2)

```
  COPY YES
  ORDER YES
  DEADLINE 2011-01-24-18.30.00.00
  MAXRO 600
  DELAY 300
  LOGSPILL (1024,1024)
  SPILLDSNPAT 'ARU.LARWORK.DHV'
  SPILLUNIT NONE
  SPILLSTORCLAS COPYCLAS
  ANALYZE
  DDTYPE UNLOAD   UNIT (WORK) SIZEPCT (150,150)
  DSNPAT 'ARU.EXMPL12.&DB.&TSIX.&DDNAME'
  DDTYPE WORK     UNIT (WORK) SIZEPCT (150,150)
  DSNPAT 'ARU.EXMPL12.&DB.&TSIX.&DDNAME'
  DDTYPE LOCPFCPY ACTIVE NO
//DDLIN DD *
  ALTER INDEX LARDBXAB.LARX001A
       PART 1 VALUES ('2800-12-31-00.00.00.000000'),
       PART 2 VALUES ('3800-12-31-00.00.00.000000'),
       PART 3 VALUES ('4500-12-31-00.00.00.000000');
/*
```

Figure 44 shows the SYSPRINT output for example 12.

Figure 44   SYSPRINT for example 12 (part 1 of 8)

```
                                *****  B M C   R E O R G   P L U S   F O R   D B 2   V10R1.00  *****
                                              (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
                REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
 BMC50001I UTILITY EXECUTION STARTING    1/27/2011   16:10:38 ...
 BMC50002I UTILITY ID = 'LARCOPB.LARDBXAB'.  DB2 SUBSYSTEM ID = 'DHV'.  OPTION MODULE = 'ARU$OPTS'.
 BMC50024I CONNECTED TO SUBSYSTEM 'DHV2'
 BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
 BMC50471I REGION=0M,BELOW 16M=8828K,ABOVE 16M=1409344K,IEFUSI=NO,CPUS=3
 BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

 BMC50471I REORG PLUS FOR DB2--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I SOLUTION COMMON CODE--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMCSORT ENGINE--V02.03.01
 BMC50471I   NO MAINTENANCE TO REPORT
 BMC50471I BMC STATS API--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT

 BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
 BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
 BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
 BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
 BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
 BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
 BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
 BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
 BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
 BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
 BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
 BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
 BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
 BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
 BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
```

Figure 44 SYSPRINT for example 12 (part 2 of 8)

```
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
```

Figure 44 SYSPRINT for example 12 (part 3 of 8)

```
BMC50470I SIZEPCT  = (100,100)                          (100,100)                           (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                 (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                               (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                             (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))           ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                         (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                         (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                         (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                   0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                       ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                   5
BMC50470I GDGEMPTY = NO                                 NO                                  NO
BMC50470I GDGSCRAT = NO                                 NO                                  NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                            REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                  NO
BMC50470I IFALLOC  = USE                                USE                                 USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                  NO
BMC50470I SMSUNIT  = NO                                 NO                                  NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                           (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                 (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                               (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                             (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))           ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                         (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                         (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                         (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                   0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                       ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                   5
BMC50470I GDGEMPTY = NO                                 NO                                  NO
BMC50470I GDGSCRAT = NO                                 NO                                  NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME
```

Figure 44 SYSPRINT for example 12 (part 4 of 8)

```
BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                = 910
BMC50471I SUBSYSTEM DEFAULT      = DHV
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'

BMC50102I REORG TABLESPACE LARDBXAB.LARSXABA
BMC50102I   SHRLEVEL CHANGE
BMC50102I   UNLOAD CONTINUE
BMC50102I   XBMID XBMA
BMC50102I   FASTSWITCH YES
BMC50102I   COPY YES
BMC50102I   ORDER YES
BMC50102I   DEADLINE 2011-01-27-19.30.00.00
BMC50102I   MAXRO 600
BMC50102I   DELAY 300
BMC50102I   LOGSPILL (1024,1024)
BMC50102I   SPILLDSNPAT 'ARU.LARWORK.DHV'
BMC50102I   SPILLUNIT NONE
BMC50102I   SPILLSTORCLAS COPYCLAS
BMC50102I   ANALYZE
BMC50102I   DDTYPE UNLOAD   UNIT (WORK) SIZEPCT (150,150)
BMC50102I   DSNPAT 'ARU.EXMPL12.&DB.&TSIX.&DDNAME'
BMC50102I   DDTYPE WORK     UNIT (WORK) SIZEPCT (150,150)
BMC50102I   DSNPAT 'ARU.EXMPL12.&DB.&TSIX.&DDNAME'
BMC50102I   DDTYPE LOCPFCPY ACTIVE NO

BMC50109I 'FASTSWITCH=YES' OPTION SPECIFIED.  OPTION 'STAGEDSN=BMC' IGNORED
BMC51291I A DDLIN DATASET HAS BEEN FOUND AND CONTAINS THE FOLLOWING STATEMENTS:

BMC50102I   ALTER INDEX LARDBXAB.LARX001A
BMC50102I        PART 1 VALUES ('2800-12-31-00.00.00.000000'),
BMC50102I        PART 2 VALUES ('3800-12-31-00.00.00.000000'),
BMC50102I        PART 3 VALUES ('4500-12-31-00.00.00.000000');
BMC51232I ALTER STATEMENT 1 WILL BE PROCESSED
BMC51293I 1 ALTER STATEMENT(S) WILL BE PROCESSED FROM THE DDLIN FILE
BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:03

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 3: SAMPLING STATISTICS: PART=1,TP=7095,SP=113,SR=655,AVGR=439,SD=65,SE=1,AVGF=579,SD=65,SE=10,EP=4
BMC50482I 3: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.I0001.A001'
BMC51301I 2: SAMPLING STATISTICS: PART=2,TP=7013,SP=112,SR=661,AVGR=440,SD=65,SE=1,AVGF=590,SD=65,SE=7,EP=2
BMC50482I 2: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.I0001.A002'
BMC51301I 1: SAMPLING STATISTICS: PART=3,TP=7012,SP=109,SR=590,AVGR=441,SD=65,SE=0,AVGF=541,SD=65,SE=17,EP=11
```

Figure 44 SYSPRINT for example 12 (part 5 of 8)

```
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.I0001.A003'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 120356  AVG SORTWK ROW LENGTH = 440  AVG UNLOAD ROW LENGTH = 440
BMC50484I ESTIMATED CARDINALITY OF PART 0001 = 41068  AVG SORTWK ROW LENGTH = 439  AVG UNLOAD ROW LENGTH = 439
BMC50484I ESTIMATED CARDINALITY OF PART 0002 = 41364  AVG SORTWK ROW LENGTH = 440  AVG UNLOAD ROW LENGTH = 440
BMC50484I ESTIMATED CARDINALITY OF PART 0003 = 37924  AVG SORTWK ROW LENGTH = 441  AVG UNLOAD ROW LENGTH = 441
BMC51264I UNLOAD WILL READ 21114 DATA PAGES FROM SPACE 'LARDBXAB.LARSXABA'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:01

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGREXMC  &STEPNAME   LARREORG  &DB         LARDBXAB  &TSIX       LARSXABA  &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       161038    &SSID       DHV2      &UTIL       LARCOPB.
BMC50425I &UTILPFX    LARCOPB   &UTILSFX    LARDBXAB  &DATE8      01272011  &GRPNM      DHV       &VCAT       DSNDHV
BMC50425I &TIME4      1610      &DATEJ      2011027   &JDATE      11027

BMC50702I LOG PROCESSING STARTED AT 01/27/2011 16:10:58
BMC50773I 101: LOG SCAN TASK STARTED AT 01/27/2011 16:10:58
BMC50706I MONITOR TRACE CLASS(1) STARTED, TRACE NUMBER = 03
BMC50877I POINT OF CONSISTENCY ESTABLISHED AT RBA/LRSN = C73EB8395B90
BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                       UNIT OR                      KBYTES   KBYTES     ALOC      ALOC
BMC50447I DDNAME   DSNAME                                       DATACLAS MGMTCLAS STORCLAS      PRI      SEC      PRI       SEC

BMC50448I SYSREC01 ARU.EXMPL12.LARDBXAB.LARSXABA.SYSREC01      *WORK                          26952     6738      500       125  TRK
BMC50448I SYSREC02 ARU.EXMPL12.LARDBXAB.LARSXABA.SYSREC02      *WORK                          27207     6803      504       126  TRK
BMC50448I SYSREC03 ARU.EXMPL12.LARDBXAB.LARSXABA.SYSREC03      *WORK                          24999     6251      463       116  TRK
BMC50448I SYSUT1   ARU.EXMPL12.LARDBXAB.LARSXABA.SYSUT1        *WORK                           1941      195       36         4  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50474I BELOW 16M = 8340K, ABOVE 16M = 1397948K, CPUS = 3
BMC50479I TOTAL PAGES: 2333029, ALLOWED: 0; AVAILABLE PAGES: 441746, ALLOWED: 441746
BMC50719I 1 LOG APPLY TASKS, 4 INDEX APPLY TASKS, AVAILABLE MEMORY: 1242853K
BMC50887I RIDMAP: RIDMAPMEM CHANGED TO 2097152K
BMC50864I LOGMEM OF OK SPECIFIED, CHANGED TO 25600K
BMC51302I MAX TASKS = 1, MAX PARTITIONS PER TASK = 3, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 3
BMC50773I 107: LOG RECORD SORT TASK STARTED AT 01/27/2011 16:11:01
BMC50773I 108: LOG RECORD SPILL TASK STARTED AT 01/27/2011 16:11:01
BMC50773I 109: INLINE IMAGE COPY APPEND TASK STARTED AT 01/27/2011 16:11:01
BMC50847I 109: BUFFER ALLOCATIONS: CHANGED PAGE BUFFERS = 0, I/O BUFFERS = 200,LARGEST BLOCK SIZE = 4096
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1836K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 40592, I/O WAITS = 13 ,DDNAME = SYS00006
BMC50477I 1: PARTITION =    2, ROWS/KEYS = 40120, I/O WAITS = 14 ,DDNAME = SYS00007
BMC50477I 1: PARTITION =    3, ROWS/KEYS = 40120, I/O WAITS = 12 ,DDNAME = SYS00008
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:03
BMC50481I 1: UNLOAD TASK COMPLETE.  ELAPSED TIME = 00:00:06
BMC50476I DDNAME = SYSREC01, I/OS = 59, I/O WAITS = 45, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC02, I/OS = 65, I/O WAITS = 41, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSREC03, I/OS = 44, I/O WAITS = 21, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT1, I/OS = 5, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51271I UNLOAD STATISTICS:  42373 ROWS/KEYS UNLOADED FROM PARTITION 1
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 1
BMC51286I RELOADING OF DATASET 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.J0001.A001' WILL REQUIRE 7410 PAGES
BMC51290I RELOADING OF DATASET 'DSNDHV.DSNDBD.LARDBXAB.LARX001A.J0001.A001' WILL REQUIRE 268 PAGES (APPROX.)
BMC51271I UNLOAD STATISTICS:  47200 ROWS/KEYS UNLOADED FROM PARTITION 2
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 2
BMC51286I RELOADING OF DATASET 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.J0001.A002' WILL REQUIRE 8251 PAGES
BMC51290I RELOADING OF DATASET 'DSNDHV.DSNDBD.LARDBXAB.LARX001A.J0001.A002' WILL REQUIRE 298 PAGES (APPROX.)
BMC51271I UNLOAD STATISTICS:  31259 ROWS/KEYS UNLOADED FROM PARTITION 3
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 3
BMC51286I RELOADING OF DATASET 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.J0001.A003' WILL REQUIRE 5465 PAGES
BMC51290I RELOADING OF DATASET 'DSNDHV.DSNDBD.LARDBXAB.LARX001A.J0001.A003' WILL REQUIRE 196 PAGES (APPROX.)
BMC51288I RELOADING OF DATASET 'DSNDHV.DSNDBD.LARDBXAB.LARX001B.J0001.A001' MAY REQUIRE 621 PAGES
BMC51272I UNLOAD STATISTICS:  120832 ROWS UNLOADED FROM SPACE 'LARDBXAB.LARSXABA', 0 ROWS DISCARDED, 0 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR SPACE 'LARDBXAB.LARSXABA'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I UNLOAD PHASE COMPLETE.  ELAPSED TIME = 00:00:10

BMC50277I XBM STATISTICS: DSN='DSNDHV.DSNDBD.LARDBXAB.LARSXABA.I0001.A001   ', READS 7095, HITS 0, WRITES 86, CACHE 0
BMC50277I XBM STATISTICS: DSN='DSNDHV.DSNDBD.LARDBXAB.LARSXABA.I0001.A002   ', READS 7013, HITS 0, WRITES 92, CACHE 0
BMC50277I XBM STATISTICS: DSN='DSNDHV.DSNDBD.LARDBXAB.LARSXABA.I0001.A003   ', READS 7012, HITS 0, WRITES 80, CACHE 0
BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50474I BELOW 16M = 8280K, ABOVE 16M = 1395736K, CPUS = 3
BMC50479I TOTAL PAGES: 2334925, ALLOWED: 0; AVAILABLE PAGES: 442572, ALLOWED: 442572
BMC51303I MAX TASKS = 1, INDEXES PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
```

Figure 44 SYSPRINT for example 12 (part 6 of 8)

```
BMC50476I DDNAME = SYSUT1, I/OS = 4, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:01
BMC50375I INLINE COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.J0001.A002'
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:01  DSN = 'DSNDHV.DSNDBD.LARDBXAB.LARX001B.J0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 120832, I/O WAITS = 6 ,DDNAME = SYS00019
BMC50773I 106: INDEX APPLY TASK STARTED AT 01/27/2011 16:11:13
BMC51276I BUILD  STATISTICS:  120832 KEYS LOADED INTO INDEX 'LARDBXAB.LARX001B'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.J0001.A001'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.J0001.A003'
BMC50482I 0: BUILD COMPLETE.  ELAPSED TIME = 00:00:10  DSN = 'DSNDHV.DSNDBD.LARDBXAB.LARX001A.J0001.A003'
BMC50477I 0: PARTITION =    3, ROWS/KEYS = 31259, I/O WAITS = 5 ,DDNAME = SYS00069
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:00:16  DSN = 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.J0001.A003'
BMC50477I 0: PARTITION =    3, ROWS/KEYS = 31259, I/O WAITS = 27 ,DDNAME = SYS00058
BMC51274I RELOAD STATISTICS:  31259 ROWS/KEYS LOADED INTO PARTITION 3
BMC50773I 103: INDEX APPLY TASK STARTED AT 01/27/2011 16:11:48
BMC50476I DDNAME = SYSREC03, I/OS = 42, I/O WAITS = 4, RDB LOCK WAITS = 0
BMC50482I 0: BUILD COMPLETE.  ELAPSED TIME = 00:00:42  DSN = 'DSNDHV.DSNDBD.LARDBXAB.LARX001A.J0001.A002'
BMC50477I 0: PARTITION =    2, ROWS/KEYS = 47200, I/O WAITS = 5 ,DDNAME = SYS00030
BMC50773I 104: INDEX APPLY TASK STARTED AT 01/27/2011 16:11:55
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:00:46  DSN = 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.J0001.A002'
BMC50477I 0: PARTITION =    2, ROWS/KEYS = 47200, I/O WAITS = 33 ,DDNAME = SYS00021
BMC51274I RELOAD STATISTICS:  47200 ROWS/KEYS LOADED INTO PARTITION 2
BMC50476I DDNAME = SYSREC02, I/OS = 64, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50482I 0: BUILD COMPLETE.  ELAPSED TIME = 00:00:34  DSN = 'DSNDHV.DSNDBD.LARDBXAB.LARX001A.J0001.A001'
BMC50477I 0: PARTITION =    1, ROWS/KEYS = 42373, I/O WAITS = 5 ,DDNAME = SYS00047
BMC50773I 105: INDEX APPLY TASK STARTED AT 01/27/2011 16:11:56
BMC50482I 0: RELOAD COMPLETE.  ELAPSED TIME = 00:00:38  DSN = 'DSNDHV.DSNDBD.LARDBXAB.LARSXABA.J0001.A001'
BMC50477I 0: PARTITION =    1, ROWS/KEYS = 42373, I/O WAITS = 27 ,DDNAME = SYS00038
BMC51274I RELOAD STATISTICS:  42373 ROWS/KEYS LOADED INTO PARTITION 1
BMC50476I DDNAME = SYSREC01, I/OS = 57, I/O WAITS = 8, RDB LOCK WAITS = 0
BMC51276I BUILD  STATISTICS:  120832 KEYS LOADED INTO INDEX 'LARDBXAB.LARX001A'
BMC51275I RELOAD STATISTICS:  120832 ROWS LOADED INTO SPACE 'LARDBXAB.LARSXABA'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC01', DSNAME = 'ARU.EXMPL12.LARDBXAB.LARSXABA.SYSREC01'
BMC50773I 102: LOG APPLY TASK STARTED AT 01/27/2011 16:11:58
BMC50829I LOGAPPLY PHASE STARTING AT 01/27/2011 16:11:58
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC02', DSNAME = 'ARU.EXMPL12.LARDBXAB.LARSXABA.SYSREC02'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSREC03', DSNAME = 'ARU.EXMPL12.LARDBXAB.LARSXABA.SYSREC03'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT1  ', DSNAME = 'ARU.EXMPL12.LARDBXAB.LARSXABA.SYSUT1'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I RELOAD PHASE COMPLETE.  ELAPSED TIME = 00:00:49

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50877I POINT OF CONSISTENCY ESTABLISHED AT RBA/LRSN = C73EB8A0925B
BMC50778I LOG RECORD QUEUE SIZE AT OR BELOW THRESHOLD
BMC50004I LOGAPPLY PHASE COMPLETE.  ELAPSED TIME = 00:00:48
BMC50830I LOGFINAL PHASE STARTING AT 01/27/2011 16:12:47
BMC50709I MONITOR TRACE CLASS(1) TRACE NUMBER = 03 STOPPED
BMC50786I 101: LOG SCAN TASK ENDED AT 01/27/2011 16:12:47, ELAPSED TIME = 00:01:49
BMC50720I 102: 1996 NEW LOG APPLY BUFFERS, 40226 REUSED, 18105 INITIAL MAXIMUM, 0 WAITS, BUFFER SIZE=8328
BMC50741I 102: LOGAPPLY STATISTICS
BMC50742I 102: PROCESSED TRANSACTIONS
BMC50744I 102:    INSERTS:      24576 INSERTS(COMPENSATION):          0
BMC50745I 102:    UPDATES:      12800 UPDATES(COMPENSATION):          0
BMC50746I 102:    DELETES:       2560 DELETES(COMPENSATION):          0
BMC50743I 102: CONSOLIDATED TRANSACTIONS
BMC50744I 102:    INSERTS:      24576 INSERTS(COMPENSATION):          0
BMC50745I 102:    UPDATES:       6289 UPDATES(COMPENSATION):          0
BMC50746I 102:    DELETES:       2560 DELETES(COMPENSATION):          0
BMC50747I 102: PAGES UPDATED COUNTERS FOR LARDBXAB.LARSXABA
BMC50748I 102:     PART     TOTAL PAGES     DATA PAGES     SPACE MAPS     HEADER PAGES
BMC50749I 102:        1           7128           7125              2               1
BMC50749I 102:        2           7826           7823              2               1
BMC50749I 102:        3           5481           5479              1               1
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A003
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A003
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A003
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A002
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A002
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A002
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A002
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A001
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A001
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A001
BMC50737I 102: EXTEND SUCCESSFUL FOR DATASET DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A001
BMC50716I 102: Buffer Manager Statistics for the following datasets:
BMC50717I 102:  DATASET=DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A001
```

Figure 44 SYSPRINT for example 12 (part 7 of 8)

```
BMC50717I 102:  DATASET=DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A002
BMC50717I 102:  DATASET=DSNDHV.DSNDBC.LARDBXAB.LARSXABA.J0001.A003
BMC50718I 102:        Page Requests     405090 |            Page Hits     384653
BMC50718I 102:           Page Reads       7496 |          Page Writes        160
BMC50718I 102:   Write I/O Requests        165 |            New Pages      12941
BMC50716I 102:       Write I/O time    0:00:10 |        Read I/O time    0:00:36
BMC50786I 102: LOG APPLY TASK ENDED AT 01/27/2011 16:13:10, ELAPSED TIME = 00:01:11
BMC50786I 107: LOG RECORD SORT TASK ENDED AT 01/27/2011 16:13:10, ELAPSED TIME = 00:02:08
BMC50786I 109: INLINE IMAGE COPY APPEND TASK ENDED AT 01/27/2011 16:13:10, ELAPSED TIME = 00:02:08
BMC50794I 103: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXAB.LARX001A DSNUM 3
BMC50791I 103:   PAGE READS  :     21917  KEY INSERTS     :      6432
BMC50791I 103:   RID INSERTS :         0  RID DELETES (PS):       662
BMC50791I 103:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 103:   LEAF SPLITS :        66  PAGE FREES      :         0
BMC50791I 103:   NEW PAGES   :        66  BUFF PAGE READS :        66
BMC50716I 103: Buffer Manager Statistics for the following datasets:
BMC50717I 103:  DATASET=DSNDHV.DSNDBC.LARDBXAB.LARX001A.J0001.A003
BMC50718I 103:        Page Requests      28484 |            Page Hits      28237
BMC50718I 103:           Page Reads        198 |          Page Writes          2
BMC50718I 103:   Write I/O Requests          3 |            New Pages         49
BMC50716I 103:       Write I/O time    0:00:00 |        Read I/O time    0:00:01
BMC50786I 103: INDEX APPLY TASK ENDED AT 01/27/2011 16:13:10, ELAPSED TIME = 00:01:22
BMC50794I 104: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXAB.LARX001A DSNUM 2
BMC50791I 104:   PAGE READS  :     35201  KEY INSERTS     :      9600
BMC50791I 104:   RID INSERTS :         0  RID DELETES (PS):      1000
BMC50791I 104:   ROOT SPLITS :         1  NON-LEAF SPLITS :         0
BMC50791I 104:   LEAF SPLITS :       100  PAGE FREES      :         0
BMC50791I 104:   NEW PAGES   :       102  BUFF PAGE READS :       152
BMC50716I 104: Buffer Manager Statistics for the following datasets:
BMC50717I 104:  DATASET=DSNDHV.DSNDBC.LARDBXAB.LARX001A.J0001.A002
BMC50718I 104:        Page Requests      45008 |            Page Hits      44632
BMC50718I 104:           Page Reads        300 |          Page Writes          3
BMC50718I 104:   Write I/O Requests          4 |            New Pages         76
BMC50716I 104:       Write I/O time    0:00:00 |        Read I/O time    0:00:01
BMC50786I 104: INDEX APPLY TASK ENDED AT 01/27/2011 16:13:12, ELAPSED TIME = 00:01:16
BMC50794I 105: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXAB.LARX001A DSNUM 1
BMC50791I 105:   PAGE READS  :     30327  KEY INSERTS     :      8544
BMC50791I 105:   RID INSERTS :         0  RID DELETES (PS):       898
BMC50791I 105:   ROOT SPLITS :         1  NON-LEAF SPLITS :         0
BMC50791I 105:   LEAF SPLITS :        89  PAGE FREES      :         0
BMC50791I 105:   NEW PAGES   :        91  BUFF PAGE READS :       104
BMC50716I 105: Buffer Manager Statistics for the following datasets:
BMC50717I 105:  DATASET=DSNDHV.DSNDBC.LARDBXAB.LARX001A.J0001.A001
BMC50718I 105:        Page Requests      39056 |            Page Hits      38719
BMC50718I 105:           Page Reads        270 |          Page Writes          3
BMC50718I 105:   Write I/O Requests          4 |            New Pages         67
BMC50716I 105:       Write I/O time    0:00:00 |        Read I/O time    0:00:01
BMC50786I 105: INDEX APPLY TASK ENDED AT 01/27/2011 16:13:13, ELAPSED TIME = 00:01:16
BMC50794I 106: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXAB.LARX001B
BMC50791I 106:   PAGE READS  :    130387  KEY INSERTS     :         0
BMC50791I 106:   RID INSERTS :     26824  RID DELETES (PS):      5120
BMC50791I 106:   ROOT SPLITS :         0  NON-LEAF SPLITS :         1
BMC50791I 106:   LEAF SPLITS :        72  PAGE FREES      :         0
BMC50791I 106:   NEW PAGES   :        73  BUFF PAGE READS :       145
BMC50716I 106: Buffer Manager Statistics for the following datasets:
BMC50717I 106:  DATASET=DSNDHV.DSNDBC.LARDBXAB.LARX001B.J0001.A001
BMC50718I 106:        Page Requests     157773 |            Page Hits     157294
BMC50718I 106:           Page Reads        445 |          Page Writes          4
BMC50718I 106:   Write I/O Requests          5 |            New Pages         34
BMC50716I 106:       Write I/O time    0:00:00 |        Read I/O time    0:00:05
BMC50786I 106: INDEX APPLY TASK ENDED AT 01/27/2011 16:13:14, ELAPSED TIME = 00:02:01
BMC50867I LOG RECORD STORE SPILL REQUESTS = 0, HIGH SPILL PAGE = 0, SPILL DATASETS CREATED = 0
BMC50868I LOG RECORD STORE WRITE REQUESTS = 0, WRITE WAIT TIME =   0.000, READ REQUESTS = 0 READ WAIT TIME =   0.000
BMC50786I 108: LOG RECORD SPILL TASK ENDED AT 01/27/2011 16:13:14, ELAPSED TIME = 00:02:13
BMC50476I DDNAME = BMCCPY, I/OS = 416, I/O WAITS = 194, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCCPZ, I/OS = 416, I/O WAITS = 194, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCY, I/OS = 416, I/O WAITS = 194, RDB LOCK WAITS = 0
BMC50476I DDNAME = BMCRCZ, I/OS = 416, I/O WAITS = 194, RDB LOCK WAITS = 0
BMC50376I 41564 PAGES COPIED TO DATASET = 'ARU.EXMPL12.BMCCPY'
BMC50376I 41564 PAGES COPIED TO DATASET = 'ARU.EXMPL12.BMCCPZ'
BMC50376I 41564 PAGES COPIED TO DATASET = 'ARU.EXMPL12.BMCRCY'
BMC50376I 41564 PAGES COPIED TO DATASET = 'ARU.EXMPL12.BMCRCZ'
BMC50865I TOTAL RECORDS INSERTED INTO LOG RECORD STORE, DATA =42213, INDEX = 59392
BMC50866I TOTAL TABLESPACE STRINGS PROCESSED = 57, WHEN NOT FULL = 3, FROM SPILL = 0
BMC50866I TOTAL INDEXSPACE STRINGS PROCESSED = 51, WHEN NOT FULL = 36, FROM SPILL = 0
BMC50866I TOTAL INDEXSPACE REQUE STRINGS PROCESSED = 0, WHEN NOT FULL = 0, FROM SPILL = 0
```

Figure 44 SYSPRINT for example 12 (part 8 of 8)

```
BMC50859I LOG RECORD STORE STATISTICS:  MEMORY AVAILABLE = 25600K, MEMORY USED = 20480K
BMC50860I TOTAL WAIT TIME FOR LOG RECORD STORE MEMORY =   0.000 DATA =   0.000 INDEX =   0.000
BMC50880I RIDMAP: PART 0001: ROWS=40592, PAIRS=40592, STORAGE=388K, MEM WAITS=0, WAIT TIME=0
BMC50881I RIDMAP: PART 0001: TRANS=12680, ADDS=9019, DELS=1623, SPILL READS=0, WAIT TIME=0
BMC50880I RIDMAP: PART 0002: ROWS=40120, PAIRS=40120, STORAGE=384K, MEM WAITS=0, WAIT TIME=0
BMC50881I RIDMAP: PART 0002: TRANS=12750, ADDS=8944, DELS=1634, SPILL READS=0, WAIT TIME=0
BMC50880I RIDMAP: PART 0003: ROWS=40120, PAIRS=40120, STORAGE=384K, MEM WAITS=0, WAIT TIME=0
BMC50881I RIDMAP: PART 0003: TRANS=12599, ADDS=8890, DELS=1580, SPILL READS=0, WAIT TIME=0
BMC50880I RIDMAP: SUMMARY: ROWS=120832, PAIRS=120832, STORAGE=1156K, MEM WAITS=0, WAIT TIME=0
BMC50881I RIDMAP: SUMMARY: TRANS=38029, ADDS=26853, DELS=4837, SPILL READS=0, WAIT TIME=0
BMC50882I RIDMAP: SUMMARY: MAX DATASPACES=1, USED DATASPACES=1, RIDMAPMEM=0K, USED MEM=2044K
BMC50883I RIDMAP: SUMMARY: MAX PIPES=9, PIPES USED=4, PIPE BUFFER SIZE=262144, PIPE WAITS=0, WAIT TIME=0
BMC50868I RIDMAP WRITE REQUESTS = 0, WRITE WAIT TIME =   0.000, READ REQUESTS = 0 READ WAIT TIME =   0.000
BMC50703I LOG PROCESSING COMPLETED, RC = 0, AT 01/27/2011 16:13:15, ELAPSED TIME = 00:02:17
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I LOGFINAL PHASE COMPLETE.  ELAPSED TIME = 00:00:28

BMC50894I SWITCH PROCESS STARTING AT   1/27/2011  16:13:16
BMC50895I SWITCH PROCESS COMPLETE.  ELAPSED TIME =  00:00:00

BMC51297I ALL SPECIFIED LIMIT KEYS HAVE BEEN SUCCESSFULLY ALTERED
BMC50890I 1: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARSXABA.I0001.A001'...
BMC50890I 5: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARX001A.I0001.A002'...
BMC50890I 2: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARSXABA.I0001.A002'...
BMC50890I 4: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARX001A.I0001.A001'...
BMC50890I 3: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARSXABA.I0001.A003'...
BMC50890I 6: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARX001A.I0001.A003'...
BMC50890I 7: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARX001B.I0001.A001'...
BMC50891I 1: DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARSXABA.I0001.A001' IS DELETED
BMC50891I 5: DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARX001A.I0001.A002' IS DELETED
BMC50891I 3: DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARSXABA.I0001.A003' IS DELETED
BMC50891I 7: DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARX001B.I0001.A001' IS DELETED
BMC50891I 6: DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARX001A.I0001.A003' IS DELETED
BMC50891I 4: DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARX001A.I0001.A001' IS DELETED
BMC50891I 2: DATASET 'DSNDHV.DSNDBC.LARDBXAB.LARSXABA.I0001.A002' IS DELETED

BMC50501I DB2 OBJECT STATISTICS
BMC50515I TABLESPACE  LARDBXAB.LARSXABA  PARTS   = 3         TABLES  = 1     SEGSIZE = 0     DSSIZE =    0G    PAGESIZE = 4K
BMC50516I             PART  NACTIVE       CARD   EXTENTS   DBCARD PCOMP   KSAVED  PSAVED
BMC50517I                1     7410      42373        10        0     0        0       0
BMC50517I                2     8251      47200        11        0     0        0       0
BMC50517I                3     5465      31259         7        0     0        0       0
BMC50518I   TABLE     LARDBXAB.LART001
BMC50519I             ROWAVG    NPAGES       CARD   PCTPAGES
BMC50520I                441     20116     120832         95
BMC50525I     INDEX   LARDBXAB.LARX001A
BMC50526I             FIRST KEY COLUMN = C009_TIMESTAMP
BMC50527I             CLUSTER = Y   UNIQUE = U   COMPRESS = N   PAGESIZE = 4K   KEYLEN =   10   COLCOUNT = 1
BMC50528I             PART   NACTIVE       CARD   EXTENTS   LEVELS
BMC50529I                1       268      42373         1        2
BMC50529I                2       297      47200         1        2
BMC50529I                3       198      31259         1        2
BMC50525I     INDEX   LARDBXAB.LARX001B
BMC50526I             FIRST KEY COLUMN = C001_INTEGER
BMC50527I             CLUSTER = N   UNIQUE = D   COMPRESS = N   PAGESIZE = 4K   KEYLEN =    5   COLCOUNT = 1
BMC50528I             PART   NACTIVE       CARD   EXTENTS   LEVELS
BMC50529I                0       439     120832         2        3
BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Example 13: SHRLEVEL CHANGE with LONGLOG and DELAY

In this example, REORG PLUS performs a SHRLEVEL CHANGE reorganization on a partitioned table space with three partitions. The table has one clustering index and one nonpartitioned secondary index. During the LOGAPPLY phase, if REORG PLUS determines that a longlog condition still exists after the time specified by DELAY has expired, REORG PLUS terminates. Otherwise, the reorganization completes normally.

This example uses the default AUTO for the ICTYPE option. Because there is an individual full copy data set on DASD for each partition, REORG PLUS changes ICTYPE to UPDATE and updates the full copy data sets rather than creating incremental copy data sets.

Message 50041I indicates the status of zIIP processing. For this example, an XBM subsystem ID was specified for SHRLEVEL CHANGE processing. REORG PLUS uses the same XBM subsystem ID for zIIP processing.

Table 76 describes the key command options for this job.

Table 76 Command options used in example 13 (part 1 of 2)

Command options used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
SHRLEVEL CHANGE	specifies the reorganization type REORG PLUS keeps the original data sets in their original status (which can be RW) through most of the reorganization.
XBMID XBMA	tells REORG PLUS to use XBM subsystem with the SSID of XBMA
FASTSWITCH YES	tells REORG PLUS to bypass the VSAM rename process and point the DB2 catalog to the staging data sets
COPY YES	creates a full and incremental DB2 image of the table space SHRLEVEL CHANGE sets REGISTER ALL, so all copies are registered in SYSIBM.SYSCOPY.
ORDER YES	sorts the data rows
MAXRO DEFER	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.

Table 76 Command options used in example 13 (part 2 of 2)

Command options used in JCL	Description
LONGLOG TERM	tells REORG PLUS to end the reorganization if the longlog condition still exists after the DELAY time expires
DELAY 300	tells REORG PLUS to wait for 5 minutes (300 seconds) and test again if it detects a longlog condition If the longlog condition still exists after the test, REORG PLUS terminates the reorganization.
RIDMAPMEM	specifies the amount of data space memory (in kilobytes) that REORG PLUS can use to store the RID translation maps
LOGMEM	specifies the amount of data space memory (in kilobytes) needed to hold the internal structures that REORG PLUS uses to store the log records
LOGSPILL	LOGSPILL (1024, 1024) determines the primary and secondary space allocation (in kilobytes) for the spill data sets that hold the log records Each number can be a nonzero positive integer.
SPILLDSNPAT	tells REORG PLUS to use a particular pattern of variables and text to create a prefix for the spill data set names The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.
SPILLUNIT	specifies the DASD unit where REORG PLUS can allocate spill data sets The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.
SPILLSTORCLAS	specifies the SMS storage class that REORG PLUS uses to allocate spill data sets The spill data sets hold the log records and RID maps when they overflow memory. REORG PLUS dynamically allocates spill data sets as they are needed. The spill data sets are VSAM objects.
ANALYZE	gathers information about the objects that you are reorganizing and provides estimated sizes for SYSREC, SYSUT1, SORTWK, and copy data sets
SIZEPCT (150,150)	tells REORG PLUS to allocate 150% of the data set size that it calculated for the UNLOAD, WORK, and SORTWORK files The increase allows room for the unknown number of updates that are being applied to the data sets.
DSNPAT	specifies a pattern of variables and text that REORG PLUS uses to create data set names for dynamic data set allocation, overriding the default pattern

Figure 45 shows the JCL for example 13.

Figure 45 JCL for example 13

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//* PARTITIONED TABLESPACE, CLUSTERING INDEX AND NONPARTITIONED       *
//* SECONDARY INDEX.                                                  *
//* REORG TABLESPACE WITH SHRLEVEL CHANGE OPTION.                     *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//LARREORG EXEC PGM=ARUUMAIN,COND=(7,LT),
// PARM='DHV,LARCOPB.LARDBXBB,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//*
//SYSIN     DD *
REORG TABLESPACE LARDBXBB.LARSXBBA
  SHRLEVEL CHANGE
  XBMID XBMA
  FASTSWITCH YES
  COPY YES
  ORDER YES
  MAXRO DEFER
  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'
  DDTYPE LOCPFCPY UNIT (WORK) SIZEPCT (150,150)
  DSNPAT 'ARU.EXMPL13.&DB.&TSIX.&DDNAME'
/*
```

Figure 46 shows the SYSPRINT output for example 13.

Figure 46 SYSPRINT for example 13 (part 1 of 9)

```
                                *****  B M C   R E O R G   P L U S   F O R   D B 2    V10R1.00  *****
                                                 (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
                   REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
 BMC50001I UTILITY EXECUTION STARTING   1/27/2011   17:22:02 ...
```

Figure 46 SYSPRINT for example 13 (part 2 of 9)

```
BMC50002I UTILITY ID = 'LARCOPB.LARDBXBB'.  DB2 SUBSYSTEM ID = 'DHV'.  OPTION MODULE = 'ARU$OPTS'.
BMC50024I CONNECTED TO SUBSYSTEM 'DHV2'
BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
BMC50471I REGION=0M,BELOW 16M=8856K,ABOVE 16M=1406220K,IEFUSI=NO,CPUS=3
BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

BMC50471I REORG PLUS FOR DB2--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.03.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(0K,0K)
BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
```

Figure 46 SYSPRINT for example 13 (part 3 of 9)

```
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
```

Figure 46 SYSPRINT for example 13 (part 4 of 9)

```
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DHV
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'

BMC50102I REORG TABLESPACE LARDBXBB.LARSXBBA
BMC50102I   SHRLEVEL CHANGE
BMC50102I   XBMID XBMA
BMC50102I   FASTSWITCH YES
BMC50102I   COPY YES
BMC50102I   ORDER YES
```

Figure 46   SYSPRINT for example 13 (part 5 of 9)

```
BMC50102I   MAXRO DEFER
BMC50102I   LONGLOG TERM
BMC50102I   DELAY 300
BMC50102I   RIDMAPMEM 20480
BMC50102I   LOGMEM 30720
BMC50102I   LOGSPILL (1024,1024)
BMC50102I   SPILLDSNPAT 'ARU.LARWORK.EX13'
BMC50102I   SPILLUNIT NONE
BMC50102I   SPILLSTORCLAS COPYCLAS
BMC50102I   ANALYZE
BMC50102I   DDTYPE UNLOAD   UNIT (WORK) SIZEPCT (150,150)
BMC50102I   DSNPAT 'ARU.EXMPL13.&DB.&TSIX.&DDNAME'
BMC50102I   DDTYPE WORK     UNIT (WORK) SIZEPCT (150,150)
BMC50102I   DSNPAT 'ARU.EXMPL13.&DB.&TSIX.&DDNAME'
BMC50102I   DDTYPE LOCPFCPY UNIT (WORK) SIZEPCT (150,150)
BMC50102I   DSNPAT 'ARU.EXMPL13.&DB.&TSIX.&DDNAME'

BMC50109I 'FASTSWITCH=YES' OPTION SPECIFIED.  OPTION 'STAGEDSN=BMC' IGNORED
BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:02

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 3: SAMPLING STATISTICS: PART=1,TP=7095,SP=112,SR=655,AVGR=441,SD=65,SE=0,AVGF=584,SD=65,SE=9,EP=3
BMC50482I 3: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.I0001.A001'
BMC51301I 2: SAMPLING STATISTICS: PART=2,TP=7013,SP=113,SR=648,AVGR=441,SD=65,SE=1,AVGF=573,SD=65,SE=11,EP=5
BMC50482I 2: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.I0001.A002'
BMC51301I 1: SAMPLING STATISTICS: PART=3,TP=7012,SP=111,SR=643,AVGR=441,SD=65,SE=0,AVGF=579,SD=65,SE=10,EP=4
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.I0001.A003'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 122183  AVG SORTWK ROW LENGTH = 441  AVG UNLOAD ROW LENGTH = 441
BMC50484I ESTIMATED CARDINALITY OF PART 0001 = 41423  AVG SORTWK ROW LENGTH = 441  AVG UNLOAD ROW LENGTH = 441
BMC50484I ESTIMATED CARDINALITY OF PART 0002 = 40173  AVG SORTWK ROW LENGTH = 441  AVG UNLOAD ROW LENGTH = 441
BMC50484I ESTIMATED CARDINALITY OF PART 0003 = 40587  AVG SORTWK ROW LENGTH = 441  AVG UNLOAD ROW LENGTH = 441
BMC51264I UNLOAD WILL READ 21114 DATA PAGES FROM SPACE 'LARDBXBB.LARSXBBA'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGREXMD  &STEPNAME   LARREORG  &DB         LARDBXBB  &TSIX       LARSXBBA  &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       172202    &SSID       DHV2      &UTIL       LARCOPB.
BMC50425I &UTILPFX    LARCOPB   &UTILSFX    LARDBXBB  &DATE8      01272011  &GRPNM      DHV       &VCAT       DSNDHV
BMC50425I &TIME4      1722      &DATEJ      2011027   &JDATE      11027

BMC50702I LOG PROCESSING STARTED AT 01/27/2011 17:22:10
BMC50773I 101: LOG SCAN TASK STARTED AT 01/27/2011 17:22:10
BMC50706I MONITOR TRACE CLASS(1) STARTED, TRACE NUMBER = 03
BMC50877I POINT OF CONSISTENCY ESTABLISHED AT RBA/LRSN = C73EC82436AB
BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                       UNIT OR                      KBYTES   KBYTES     ALOC      ALOC
BMC50447I DDNAME   DSNAME                                       DATACLAS MGMTCLAS STORCLAS      PRI      SEC      PRI       SEC

BMC50448I BMCCPY01 ARU.EXMPL13.LARDBXBB.LARSXBBA.BMCCPY01      *WORK                         42570     4257       887        89  TRK
BMC50448I BMCCPY02 ARU.EXMPL13.LARDBXBB.LARSXBBA.BMCCPY02      *WORK                         42078     4208       877        88  TRK
BMC50448I BMCCPY03 ARU.EXMPL13.LARDBXBB.LARSXBBA.BMCCPY03      *WORK                         42072     4206       877        88  TRK
BMC50448I SYSUT1   ARU.EXMPL13.LARDBXBB.LARSXBBA.SYSUT1        *WORK                          1971      198        37         4  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50474I BELOW 16M = 8372K, ABOVE 16M = 1396764K, CPUS = 3
BMC50479I TOTAL PAGES: 2324855, ALLOWED: 0; AVAILABLE PAGES: 420012, ALLOWED: 420012
BMC50719I 3 LOG APPLY TASKS, 4 INDEX APPLY TASKS, AVAILABLE MEMORY: 1234083K
BMC51302I MAX TASKS = 3, MAX PARTITIONS PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50773I 109: LOG RECORD SORT TASK STARTED AT 01/27/2011 17:22:13
BMC50773I 110: LOG RECORD SPILL TASK STARTED AT 01/27/2011 17:22:13
BMC50773I 111: INLINE IMAGE COPY APPEND TASK STARTED AT 01/27/2011 17:22:13
BMC50847I 111: BUFFER ALLOCATIONS: CHANGED PAGE BUFFERS = 0, I/O BUFFERS = 600,LARGEST BLOCK SIZE = 4096
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1284K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 3: BMCSORT STARTED, 256K BELOW 16M, 1284K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1296K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 2: PARTITION =    2, ROWS/KEYS = 40120, I/O WAITS = 14 ,DDNAME = SYS00008
BMC51271I UNLOAD STATISTICS:  40120 ROWS/KEYS UNLOADED FROM PARTITION 2
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 2
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:01
BMC50477I 3: PARTITION =    3, ROWS/KEYS = 40120, I/O WAITS = 11 ,DDNAME = SYS00009
BMC51271I UNLOAD STATISTICS:  40120 ROWS/KEYS UNLOADED FROM PARTITION 3
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 3
BMC50481I 3: SORT COMPLETE.  ELAPSED TIME = 00:00:01
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 40592, I/O WAITS = 12 ,DDNAME = SYS00010
```

Figure 46 SYSPRINT for example 13 (part 6 of 9)

```
BMC51271I UNLOAD STATISTICS:  40592 ROWS/KEYS UNLOADED FROM PARTITION 1
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 1
BMC51272I UNLOAD STATISTICS:  120832 ROWS UNLOADED FROM SPACE 'LARDBXBB.LARSXBBA', 0 ROWS DISCARDED, 0 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR SPACE 'LARDBXBB.LARSXBBA'
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:01
BMC50375I INLINE COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.J0001.A001'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.J0001.A002'
BMC50375I INLINE COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.J0001.A003'
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DSNDHV.DSNDBD.LARDBXBB.LARX001A.J0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 40592, I/O WAITS = 5 ,DDNAME = SYS00061
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:06  DSN = 'DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.J0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 40592, I/O WAITS = 40 ,DDNAME = SYS00032
BMC51274I RELOAD STATISTICS:  40592 ROWS/KEYS LOADED INTO PARTITION 1
BMC50773I 105: INDEX APPLY TASK STARTED AT 01/27/2011 17:22:22
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DSNDHV.DSNDBD.LARDBXBB.LARX001A.J0001.A002'
BMC50477I 2: PARTITION =    2, ROWS/KEYS = 40120, I/O WAITS = 5 ,DDNAME = SYS00064
BMC50773I 102: LOG APPLY TASK STARTED AT 01/27/2011 17:22:22
BMC50481I 1: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:09
BMC50482I 2: RELOAD COMPLETE.  ELAPSED TIME = 00:00:08  DSN = 'DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.J0001.A002'
BMC50477I 2: PARTITION =    2, ROWS/KEYS = 40120, I/O WAITS = 41 ,DDNAME = SYS00034
BMC51274I RELOAD STATISTICS:  40120 ROWS/KEYS LOADED INTO PARTITION 2
BMC50482I 3: BUILD COMPLETE.  ELAPSED TIME = 00:00:06  DSN = 'DSNDHV.DSNDBD.LARDBXBB.LARX001A.J0001.A003'
BMC50477I 3: PARTITION =    3, ROWS/KEYS = 40120, I/O WAITS = 5 ,DDNAME = SYS00066
BMC50481I 2: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:09
BMC50482I 3: RELOAD COMPLETE.  ELAPSED TIME = 00:00:08  DSN = 'DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.J0001.A003'
BMC50477I 3: PARTITION =    3, ROWS/KEYS = 40120, I/O WAITS = 41 ,DDNAME = SYS00038
BMC51274I RELOAD STATISTICS:  40120 ROWS/KEYS LOADED INTO PARTITION 3
BMC50773I 106: INDEX APPLY TASK STARTED AT 01/27/2011 17:22:23
BMC50773I 103: LOG APPLY TASK STARTED AT 01/27/2011 17:22:23
BMC50773I 107: INDEX APPLY TASK STARTED AT 01/27/2011 17:22:23
BMC50773I 104: LOG APPLY TASK STARTED AT 01/27/2011 17:22:23
BMC50481I 3: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:10
BMC50277I XBM STATISTICS: DSN='DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.I0001.A001   ', READS 7095, HITS 0, WRITES 174, CACHE 0
BMC50277I XBM STATISTICS: DSN='DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.I0001.A002   ', READS 7013, HITS 0, WRITES 176, CACHE 0
BMC50277I XBM STATISTICS: DSN='DSNDHV.DSNDBD.LARDBXBB.LARSXBBA.I0001.A003   ', READS 7012, HITS 0, WRITES 168, CACHE 0
BMC50476I DDNAME = SYSUT1, I/OS = 5, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51276I BUILD  STATISTICS:  120832 KEYS LOADED INTO INDEX 'LARDBXBB.LARX001A'
BMC51275I RELOAD STATISTICS:  120832 ROWS LOADED INTO SPACE 'LARDBXBB.LARSXBBA'
BMC50474I BELOW 16M = 8176K, ABOVE 16M = 1376116K, CPUS = 3
BMC50479I TOTAL PAGES: 2314173, ALLOWED: 0; AVAILABLE PAGES: 415266, ALLOWED: 415266
BMC51303I MAX TASKS = 1, INDEXES PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 1
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50476I DDNAME = SYSUT1, I/OS = 4, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:00
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DSNDHV.DSNDBD.LARDBXBB.LARX001B.J0001.A001'
BMC50477I 1: PARTITION =    0, ROWS/KEYS = 120832, I/O WAITS = 7 ,DDNAME = SYS00077
BMC50773I 108: INDEX APPLY TASK STARTED AT 01/27/2011 17:22:28
BMC50829I LOGAPPLY PHASE STARTING AT 01/27/2011 17:22:28
BMC51276I BUILD  STATISTICS:  120832 KEYS LOADED INTO INDEX 'LARDBXBB.LARX001B'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT1  ', DSNAME = 'ARU.EXMPL13.LARDBXBB.LARSXBBA.SYSUT1'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I REORG PHASE COMPLETE.  ELAPSED TIME = 00:00:17

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50795I XBM COMMAND RECEIVED AT 01/27/2011 17:23:06 FROM RDAXBM
BMC50796I COMMAND : MAXRO 999
BMC50796I RESPONSE: BMC50801  MAXRO VALUE SET TO 999
BMC50877I POINT OF CONSISTENCY ESTABLISHED AT RBA/LRSN = C73EC86DAD95
BMC50778I LOG RECORD QUEUE SIZE AT OR BELOW THRESHOLD
BMC50004I LOGAPPLY PHASE COMPLETE.  ELAPSED TIME = 00:01:00
BMC50830I LOGFINAL PHASE STARTING AT 01/27/2011 17:23:28
BMC50709I MONITOR TRACE CLASS(1) TRACE NUMBER = 03 STOPPED
BMC50786I 101: LOG SCAN TASK ENDED AT 01/27/2011 17:23:28, ELAPSED TIME = 00:01:18
BMC50720I 104: 522 NEW LOG APPLY BUFFERS, 13481 REUSED, 5442 INITIAL MAXIMUM, 0 WAITS, BUFFER SIZE=8328
BMC50741I 104: LOGAPPLY STATISTICS
BMC50742I 104: PROCESSED TRANSACTIONS
BMC50744I 104:    INSERTS:       8160 INSERTS(COMPENSATION):          0
BMC50745I 104:    UPDATES:       4250 UPDATES(COMPENSATION):          0
BMC50746I 104:    DELETES:        850 DELETES(COMPENSATION):          0
BMC50743I 104: CONSOLIDATED TRANSACTIONS
BMC50744I 104:    INSERTS:       8160 INSERTS(COMPENSATION):          0
BMC50745I 104:    UPDATES:       3757 UPDATES(COMPENSATION):          0
BMC50746I 104:    DELETES:        850 DELETES(COMPENSATION):          0
BMC50747I 104: PAGES UPDATED COUNTERS FOR LARDBXBB.LARSXBBA
BMC50748I 104:     PART     TOTAL PAGES     DATA PAGES     SPACE MAPS     HEADER PAGES
BMC50749I 104:        3           6793           6790              2               1
```

Figure 46 SYSPRINT for example 13 (part 7 of 9)

```
BMC50716I 104: Buffer Manager Statistics for the following datasets:
BMC50717I 104:  DATASET=DSNDHV.DSNDBC.LARDBXBB.LARSXBBA.J0001.A003
BMC50718I 104:        Page Requests     170607 |            Page Hits     163814
BMC50718I 104:           Page Reads       2675 |          Page Writes         54
BMC50718I 104:   Write I/O Requests         55 |            New Pages       4118
BMC50716I 104:       Write I/O time    0:00:01 |        Read I/O time    0:00:15
BMC50786I 104: LOG APPLY TASK ENDED AT 01/27/2011 17:23:30, ELAPSED TIME = 00:01:07
BMC50720I 102: 611 NEW LOG APPLY BUFFERS, 13583 REUSED, 5442 INITIAL MAXIMUM, 0 WAITS, BUFFER SIZE=8328
BMC50741I 102: LOGAPPLY STATISTICS
BMC50742I 102: PROCESSED TRANSACTIONS
BMC50744I 102:    INSERTS:       8256 INSERTS(COMPENSATION):          0
BMC50745I 102:    UPDATES:       4300 UPDATES(COMPENSATION):          0
BMC50746I 102:    DELETES:        860 DELETES(COMPENSATION):          0
BMC50743I 102: CONSOLIDATED TRANSACTIONS
BMC50744I 102:    INSERTS:       8256 INSERTS(COMPENSATION):          0
BMC50745I 102:    UPDATES:       3963 UPDATES(COMPENSATION):          0
BMC50746I 102:    DELETES:        860 DELETES(COMPENSATION):          0
BMC50747I 102: PAGES UPDATED COUNTERS FOR LARDBXBB.LARSXBBA
BMC50748I 102:     PART     TOTAL PAGES     DATA PAGES     SPACE MAPS     HEADER PAGES
BMC50749I 102:        1           6843           6840               2               1
BMC50720I 103: 620 NEW LOG APPLY BUFFERS, 13436 REUSED, 5442 INITIAL MAXIMUM, 0 WAITS, BUFFER SIZE=8328
BMC50741I 103: LOGAPPLY STATISTICS
BMC50742I 103: PROCESSED TRANSACTIONS
BMC50744I 103:    INSERTS:       8160 INSERTS(COMPENSATION):          0
BMC50745I 103:    UPDATES:       4250 UPDATES(COMPENSATION):          0
BMC50746I 103:    DELETES:        850 DELETES(COMPENSATION):          0
BMC50743I 103: CONSOLIDATED TRANSACTIONS
BMC50744I 103:    INSERTS:       8160 INSERTS(COMPENSATION):          0
BMC50745I 103:    UPDATES:       3737 UPDATES(COMPENSATION):          0
BMC50746I 103:    DELETES:        850 DELETES(COMPENSATION):          0
BMC50747I 103: PAGES UPDATED COUNTERS FOR LARDBXBB.LARSXBBA
BMC50748I 103:     PART     TOTAL PAGES     DATA PAGES     SPACE MAPS     HEADER PAGES
BMC50749I 103:        2           6788           6785               2               1
BMC50716I 102: Buffer Manager Statistics for the following datasets:
BMC50717I 102:  DATASET=DSNDHV.DSNDBC.LARDBXBB.LARSXBBA.J0001.A001
BMC50718I 102:        Page Requests     176085 |            Page Hits     169242
BMC50718I 102:           Page Reads       2697 |          Page Writes         54
BMC50718I 102:   Write I/O Requests         55 |            New Pages       4146
BMC50716I 102:       Write I/O time    0:00:02 |        Read I/O time    0:00:16
BMC50786I 102: LOG APPLY TASK ENDED AT 01/27/2011 17:23:37, ELAPSED TIME = 00:01:14
BMC50716I 103: Buffer Manager Statistics for the following datasets:
BMC50717I 103:  DATASET=DSNDHV.DSNDBC.LARDBXBB.LARSXBBA.J0001.A002
BMC50718I 103:        Page Requests     170111 |            Page Hits     163323
BMC50718I 103:           Page Reads       2612 |          Page Writes         54
BMC50718I 103:   Write I/O Requests         55 |            New Pages       4176
BMC50716I 103:       Write I/O time    0:00:02 |        Read I/O time    0:00:16
BMC50786I 103: LOG APPLY TASK ENDED AT 01/27/2011 17:23:38, ELAPSED TIME = 00:01:15
BMC50786I 111: INLINE IMAGE COPY APPEND TASK ENDED AT 01/27/2011 17:23:38, ELAPSED TIME = 00:01:25
BMC50786I 109: LOG RECORD SORT TASK ENDED AT 01/27/2011 17:23:38, ELAPSED TIME = 00:01:25
BMC50794I 105: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXBB.LARX001A DSNUM 1
BMC50791I 105:   PAGE READS  :     28115  KEY INSERTS     :      8256
BMC50791I 105:   RID INSERTS :         0  RID DELETES (PS):       860
BMC50791I 105:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 105:   LEAF SPLITS :        85  PAGE FREES      :         0
BMC50791I 105:   NEW PAGES   :        85  BUFF PAGE READS :        85
BMC50716I 105: Buffer Manager Statistics for the following datasets:
BMC50717I 105:  DATASET=DSNDHV.DSNDBC.LARDBXBB.LARX001A.J0001.A001
BMC50718I 105:        Page Requests      36544 |            Page Hits      36223
BMC50718I 105:           Page Reads        259 |          Page Writes          3
BMC50718I 105:   Write I/O Requests          3 |            New Pages         62
BMC50716I 105:       Write I/O time    0:00:00 |        Read I/O time    0:00:01
BMC50786I 105: INDEX APPLY TASK ENDED AT 01/27/2011 17:23:39, ELAPSED TIME = 00:01:17
BMC50794I 107: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXBB.LARX001A DSNUM 3
BMC50791I 107:   PAGE READS  :     27789  KEY INSERTS     :      8160
BMC50794I 106: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXBB.LARX001A DSNUM 2
BMC50791I 107:   RID INSERTS :         0  RID DELETES (PS):       850
BMC50791I 107:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 106:   PAGE READS  :     27789  KEY INSERTS     :      8160
BMC50791I 107:   LEAF SPLITS :        84  PAGE FREES      :         0
BMC50791I 107:   NEW PAGES   :        84  BUFF PAGE READS :        84
BMC50791I 106:   RID INSERTS :         0  RID DELETES (PS):       850
BMC50791I 106:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 106:   LEAF SPLITS :        84  PAGE FREES      :         0
BMC50791I 106:   NEW PAGES   :        84  BUFF PAGE READS :        84
BMC50716I 107: Buffer Manager Statistics for the following datasets:
BMC50717I 107:  DATASET=DSNDHV.DSNDBC.LARDBXBB.LARX001A.J0001.A003
```

Figure 46 SYSPRINT for example 13 (part 8 of 9)

```
BMC50718I 107:        Page Requests      36120 |            Page Hits      35803
BMC50718I 107:           Page Reads        255 |          Page Writes          3
BMC50718I 107:   Write I/O Requests          4 |            New Pages         62
BMC50716I 107:       Write I/O time    0:00:00 |        Read I/O time    0:00:02
BMC50786I 107: INDEX APPLY TASK ENDED AT 01/27/2011 17:23:40, ELAPSED TIME = 00:01:17
BMC50716I 106: Buffer Manager Statistics for the following datasets:
BMC50717I 106:  DATASET=DSNDHV.DSNDBC.LARDBXBB.LARX001A.J0001.A002
BMC50718I 106:        Page Requests      36120 |            Page Hits      35803
BMC50718I 106:           Page Reads        255 |          Page Writes          3
BMC50718I 106:   Write I/O Requests          4 |            New Pages         62
BMC50716I 106:       Write I/O time    0:00:00 |        Read I/O time    0:00:02
BMC50786I 106: INDEX APPLY TASK ENDED AT 01/27/2011 17:23:40, ELAPSED TIME = 00:01:17
BMC50794I 108: INDEX MAINTENANCE STATISTICS FOR INDEX LARDBXBB.LARX001B
BMC50791I 108:   PAGE READS  :    130677  KEY INSERTS     :         0
BMC50791I 108:   RID INSERTS :     26854  RID DELETES (PS):      5120
BMC50791I 108:   ROOT SPLITS :         0  NON-LEAF SPLITS :         1
BMC50791I 108:   LEAF SPLITS :       119  PAGE FREES      :         0
BMC50791I 108:   NEW PAGES   :       120  BUFF PAGE READS :       239
BMC50716I 108: Buffer Manager Statistics for the following datasets:
BMC50717I 108:  DATASET=DSNDHV.DSNDBC.LARDBXBB.LARX001B.J0001.A001
BMC50718I 108:        Page Requests     158219 |            Page Hits     157698
BMC50718I 108:           Page Reads        440 |          Page Writes          5
BMC50718I 108:   Write I/O Requests          6 |            New Pages         81
BMC50716I 108:       Write I/O time    0:00:00 |        Read I/O time    0:00:03
BMC50786I 108: INDEX APPLY TASK ENDED AT 01/27/2011 17:23:42, ELAPSED TIME = 00:01:13
BMC50867I LOG RECORD STORE SPILL REQUESTS = 0, HIGH SPILL PAGE = 0, SPILL DATASETS CREATED = 0
BMC50868I LOG RECORD STORE WRITE REQUESTS = 0, WRITE WAIT TIME =   0.000, READ REQUESTS = 0 READ WAIT TIME =   0.000
BMC50786I 110: LOG RECORD SPILL TASK ENDED AT 01/27/2011 17:23:42, ELAPSED TIME = 00:01:28
BMC50476I DDNAME = BMCCPY01, I/OS = 140, I/O WAITS = 36, RDB LOCK WAITS = 0
BMC50376I 13942 PAGES COPIED TO DATASET = 'ARU.EXMPL13.LARDBXBB.LARSXBBA.BMCCPY01'
BMC50476I DDNAME = BMCCPY02, I/OS = 139, I/O WAITS = 34, RDB LOCK WAITS = 0
BMC50376I 13802 PAGES COPIED TO DATASET = 'ARU.EXMPL13.LARDBXBB.LARSXBBA.BMCCPY02'
BMC50476I DDNAME = BMCCPY03, I/OS = 139, I/O WAITS = 51, RDB LOCK WAITS = 0
BMC50376I 13807 PAGES COPIED TO DATASET = 'ARU.EXMPL13.LARDBXBB.LARSXBBA.BMCCPY03'
BMC50865I TOTAL RECORDS INSERTED INTO LOG RECORD STORE, DATA =42222, INDEX = 59392
BMC50866I TOTAL TABLESPACE STRINGS PROCESSED = 66, WHEN NOT FULL = 19, FROM SPILL = 0
BMC50866I TOTAL INDEXSPACE STRINGS PROCESSED = 54, WHEN NOT FULL = 42, FROM SPILL = 0
BMC50866I TOTAL INDEXSPACE REQUE STRINGS PROCESSED = 0, WHEN NOT FULL = 0, FROM SPILL = 0
BMC50859I LOG RECORD STORE STATISTICS:  MEMORY AVAILABLE = 30720K, MEMORY USED = 15360K
BMC50860I TOTAL WAIT TIME FOR LOG RECORD STORE MEMORY =   0.000 DATA =   0.000 INDEX =   0.000
BMC50880I RIDMAP: PART 0001: ROWS=40592, PAIRS=40592, STORAGE=388K, MEM WAITS=0, WAIT TIME=0
BMC50881I RIDMAP: PART 0001: TRANS=15059, ADDS=9024, DELS=1628, SPILL READS=0, WAIT TIME=0
BMC50880I RIDMAP: PART 0002: ROWS=40120, PAIRS=40120, STORAGE=384K, MEM WAITS=0, WAIT TIME=0
BMC50881I RIDMAP: PART 0002: TRANS=14642, ADDS=8946, DELS=1636, SPILL READS=0, WAIT TIME=0
BMC50880I RIDMAP: PART 0003: ROWS=40120, PAIRS=40120, STORAGE=384K, MEM WAITS=0, WAIT TIME=0
BMC50881I RIDMAP: PART 0003: TRANS=14480, ADDS=8892, DELS=1582, SPILL READS=0, WAIT TIME=0
BMC50880I RIDMAP: SUMMARY: ROWS=120832, PAIRS=120832, STORAGE=1156K, MEM WAITS=0, WAIT TIME=0
BMC50881I RIDMAP: SUMMARY: TRANS=44181, ADDS=26862, DELS=4846, SPILL READS=0, WAIT TIME=0
BMC50882I RIDMAP: SUMMARY: MAX DATASPACES=1, USED DATASPACES=1, RIDMAPMEM=20480K, USED MEM=2044K
BMC50883I RIDMAP: SUMMARY: MAX PIPES=12, PIPES USED=4, PIPE BUFFER SIZE=262144, PIPE WAITS=0, WAIT TIME=0
BMC50868I RIDMAP WRITE REQUESTS = 0, WRITE WAIT TIME =   0.000, READ REQUESTS = 0 READ WAIT TIME =   0.000
BMC50703I LOG PROCESSING COMPLETED, RC = 0, AT 01/27/2011 17:23:42, ELAPSED TIME = 00:01:31
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I LOGFINAL PHASE COMPLETE.  ELAPSED TIME = 00:00:14


BMC50894I SWITCH PROCESS STARTING AT   1/27/2011  17:23:43
BMC50895I SWITCH PROCESS COMPLETE.  ELAPSED TIME =  00:00:00

BMC50890I 2: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARSXBBA.I0001.A002'...
BMC50890I 1: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARSXBBA.I0001.A001'...
BMC50890I 3: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARSXBBA.I0001.A003'...
BMC50890I 5: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARX001A.I0001.A002'...
BMC50890I 6: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARX001A.I0001.A003'...
BMC50890I 4: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARX001A.I0001.A001'...
BMC50890I 7: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARX001B.I0001.A001'...
BMC50891I 1: DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARSXBBA.I0001.A001' IS DELETED
BMC50891I 2: DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARSXBBA.I0001.A002' IS DELETED
BMC50891I 6: DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARX001A.I0001.A003' IS DELETED
BMC50891I 4: DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARX001A.I0001.A001' IS DELETED
BMC50891I 7: DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARX001B.I0001.A001' IS DELETED
BMC50891I 3: DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARSXBBA.I0001.A003' IS DELETED
BMC50891I 5: DATASET 'DSNDHV.DSNDBC.LARDBXBB.LARX001A.I0001.A002' IS DELETED
```

Figure 46 SYSPRINT for example 13 (part 9 of 9)

```
BMC50501I DB2 OBJECT STATISTICS
BMC50515I TABLESPACE  LARDBXBB.LARSXBBA  PARTS   = 3         TABLES  = 1     SEGSIZE = 0     DSSIZE =    0G    PAGESIZE = 4K
BMC50516I             PART  NACTIVE       CARD   EXTENTS   DBCARD PCOMP   KSAVED  PSAVED
BMC50517I                1     7098      40592         1        0     0        0       0
BMC50517I                2     7013      40120         1        0     0        0       0
BMC50517I                3     7013      40120         1        0     0        0       0
BMC50518I   TABLE     LARDBXBB.LART001
BMC50519I             ROWAVG    NPAGES       CARD   PCTPAGES
BMC50520I                441     20115     120832         95
BMC50525I     INDEX   LARDBXBB.LARX001A
BMC50526I             FIRST KEY COLUMN = C009_TIMESTAMP
BMC50527I             CLUSTER = Y   UNIQUE = U   COMPRESS = N   PAGESIZE = 4K   KEYLEN =   10   COLCOUNT = 1
BMC50528I             PART   NACTIVE       CARD   EXTENTS   LEVELS
BMC50529I                1       257      40592         1        2
BMC50529I                2       253      40120         1        2
BMC50529I                3       253      40120         1        2
BMC50525I     INDEX   LARDBXBB.LARX001B
BMC50526I             FIRST KEY COLUMN = C001_INTEGER
BMC50527I             CLUSTER = N   UNIQUE = D   COMPRESS = N   PAGESIZE = 4K   KEYLEN =    5   COLCOUNT = 1
BMC50528I             PART   NACTIVE       CARD   EXTENTS   LEVELS
BMC50529I                0       439     120832         1        3
BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY FAILED OR PARTIALLY COMPLETE
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Example 14: SHRLEVEL CHANGE with table-controlled partitioning and partition rebalancing

In this example, REORG PLUS performs a SHRLEVEL CHANGE reorganization of a partitioned table space that uses table-controlled partitioning and has data partitioned secondary (DPSI) and nonpartitioned secondary (NPSI) indexes defined.

The REBALANCE command option tells REORG PLUS to define new partition boundaries and evenly redistribute rows across the reorganized partitions. The JCL also includes a DD statement for a DDLOUT data set. This DD statement tells REORG PLUS to write the ALTER statements that REORG PLUS used for rebalancing to this data set. Figure 49 on page 520 shows the contents of the DDLOUT data set for this example.

Message 50041I indicates the status of zIIP processing. For this example, an XBM subsystem ID was specified for SHRLEVEL CHANGE processing. REORG PLUS uses the same XBM subsystem ID for zIIP processing.

Table 77 describes the key command options for this job.

Table 77 Key command options used in example 14 (part 1 of 2)

Command options used in JCL	Description
REORG TABLESPACE	specifies that the table space named in the statement is to be reorganized
ORDER YES	sorts the data rows

Table 77 Key command options used in example 14 (part 2 of 2)

Command options used in JCL	Description
REBALANCE	defines new partition boundaries and evenly redistributes rows across the reorganized partitions
COPY YES	creates a DB2 image copy of the table space Because the INLINECP installation option is YES, REORG PLUS would normally create an inline image copy. However, in this example, REORG PLUS changes INLINECP=YES to INLINECP=NO because the ICTYPE is INCREMENTAL.
ICTYPE INCREMENTAL	tells REORG PLUS to create an incremental image copy instead of updating the full image copy
ANALYZE SAMPLE	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.
PREFORMAT YES	REORG PLUS preformats the unused pages of the data set, overriding the default After REORG PLUS reorganizes the data and indexes, it writes pages that have been initialized with zeros, up to the high-allocated relative byte address (RBA) of the table space and index spaces that were reorganized.
BMCSTATS YES UPDATEDB2STATS YES	tells REORG PLUS to update statistics in the DASD MANAGER PLUS database statistics tables and the DB2 catalog 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.
SIZEPCT (150,150)	tells REORG PLUS to allocate 150% of the data set size that it calculated for the local primary full copy data set
DSNPAT	specifies a pattern of variables and text that REORG PLUS uses to create data set names for dynamic data set allocation, overriding the default pattern

Figure 47 shows the JCL for example 14.

Figure 47 JCL for example 14 (part 1 of 2)

```
//        JOB
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  PARTITIONED TABLESPACE THAT USES TABLE-BASED PARTITIONING.       *
//*  DATA-PARTITIONED SECONDARY INDEX (DPSI) AND NON-PARTITIONED      *
//*  SECONDARY (NPSI) INDEXES DEFINED.                                *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 47 JCL for example 14 (part 2 of 2)

```
//LARREORG EXEC PGM=ARUUMAIN,REGION=0M,COND=(7,LT),
// PARM='DHV,ARUEXP14.RG1,NEW,,MSGLEVEL(1),ARU$OPTS'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//UTPRINT   DD  SYSOUT=*
//DDLOUT    DD  SYSOUT=*
//ASUSRPRT  DD  SYSOUT=*
//*
//SYSIN     DD *
REORG TABLESPACE ARUEXP14.LARS$XBA
 SHRLEVEL CHANGE
 XBMID XBMA
 ORDER YES
 REBALANCE
 COPY YES
 ICTYPE INCREMENTAL
 LOGTHRESHLD 1000
 DELAY 300
 LOGSPILL (20480,10240)
 SPILLDSNPAT 'ARU.LARWORK.DHV'
 SPILLUNIT NONE
 SPILLSTORCLAS COPYCLAS
 ANALYZE SAMPLE
 PREFORMAT YES
 UPDATEDB2STATS YES
 BMCSTATS YES
 FASTSWITCH YES
 DDTYPE WORK     UNIT (WORK) SIZEPCT (150,150)
 DSNPAT 'ARU.EXMPL14.&DB.&TSIX.&DDNAME'
 DDTYPE LOCPFCPY UNIT (WORK) SIZEPCT (150,150)
 DSNPAT 'ARU.EXMPL14.&DB.&TSIX.&DDNAME'
 DDTYPE LOCPICPY UNIT (WORK)
 DSNPAT 'ARU.EXMPL14.&DB.&TSIX.&DDNAME'
/*
```

Figure 48 shows the SYSPRINT output for example 14.

Figure 48 SYSPRINT for example 14 (part 1 of 10)

```
                                 *****  B M C   R E O R G   P L U S   F O R   D B 2   V10R1.00  *****
                                                (C) COPYRIGHT 1988 - 2011 BMC SOFTWARE, INC.
                   REORG PLUS TECHNOLOGY IS PROTECTED BY ONE OR MORE OF THE FOLLOWING U.S. PATENTS: 5,222,235 AND 7,809,762
 BMC50001I UTILITY EXECUTION STARTING    1/27/2011   17:32:56 ...
 BMC50002I UTILITY ID = 'ARUEXP14.RG1'.  DB2 SUBSYSTEM ID = 'DHV'.  OPTION MODULE = 'ARU$OPTS'.
 BMC50024I CONNECTED TO SUBSYSTEM 'DHV2'
 BMC50471I z/OS 1.10.0,PID=HBB7750,DFSMS FOR Z/OS=1.10.0,DB2=9.1.0
 BMC50471I REGION=0M,BELOW 16M=8852K,ABOVE 16M=1406152K,IEFUSI=NO,CPUS=3
 BMC50471I MEMLIMIT=17592186040320M,AVAILABLE=17592186040320M,MEMLIMIT SET BY:REGION=0

 BMC50471I REORG PLUS FOR DB2--V10.01.00
 BMC50471I   NO MAINTENANCE TO REPORT
```

Figure 48 SYSPRINT for example 14 (part 2 of 10)

```
BMC50471I DB2 UTILITIES COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I SOLUTION COMMON CODE--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMCSORT ENGINE--V02.03.01
BMC50471I   NO MAINTENANCE TO REPORT
BMC50471I BMC STATS API--V10.01.00
BMC50471I   NO MAINTENANCE TO REPORT

BMC50471I ACFORTSS=YES                            INDREFLM=10                             SDUMP=YES
BMC50471I ALTRFAIL=RCVRPEND                       INLINECP=YES                            SHORTMEMORY=CONTINUE
BMC50471I ANALMAX=1000%                           INLOB=YES                               SIXSNAP=NO
BMC50471I ARC=NO                                  IXINCLCOL=YES                           SMAX=0
BMC50471I ARCHDDN=SYSARC                          IXONEX=NO                               SMCORE=(OK,OK)
BMC50471I AUXREORG=DEFAULT                        IXRANDOM=NO                             SORTDEVT=(,SYSALLDA)
BMC50471I AVAILPAGEPCT=100                        KEEPDICTIONARY=NO                       SORTNUM=32
BMC50471I BILDMAX=300%                            LEAFDSLM=200                            SPILDSNP=&UID
BMC50471I BMCHIST=YES                             LOB=YES                                 SPILSCLS=NONE
BMC50471I CBUFFS=30                               LOCKROW=YES                             SPILUNIT=WORK
BMC50471I CLONE=YES                               LOGFINAL=NONE                           SQLDELAY=3
BMC50471I CONDEXEC=NO                             LOGMEM=0                                SQLRETRY=100
BMC50471I COPYDDN=(BMCCPY,BMCCPZ)                 LOGSPIL=(20000,10000)                   STAGEDSN=BMC
BMC50471I COPYLVL=PART                            LOGTHRSH=0                              STOP@CMT=YES
BMC50471I COPYMAX=1000%                           LONGLOG=CONTINUE                        STOPDELAY=1
BMC50471I COPYSUBSET=NO                           LONGNAMETRUNC=MIDDLE                    STOPRETRY=300
BMC50471I CPYRFAIL=TERM                           MAXNEWPARTS=2                           TAPEDISP=DELETE
BMC50471I DATACAP=NO                              MAXRO=300                               TASKMAX=1000%
BMC50471I DDLDDN=DDLIN                            MAXSORTMEMORY=0                         TEMPRALDATA=YES
BMC50471I DEADLINE=NONE                           MAXTAPE=3                               TERMEXIT=(NONE,REXX)
BMC50471I DELAY=1200                              MGEXTENT=CONTINUE                       TIMEOUT=TERM
BMC50471I DELFILES=YES                            MINSORTMEMORY=0                         TOTALPAGEPCT=0
BMC50471I DESCCDE=(3,7)                           MSGLEVEL=1                              TSPREC=YES
BMC50471I DRAINTYP=ALL                            OFFPOSLM=10                             TSSAMPLEPCT=100
BMC50471I DRNDELAY=1                              OPNDB2ID=YES                            TSTZ=YES
BMC50471I DRNRETRY=255                            ORIGDISP=DELETE                         UBUFFS=20
BMC50471I DRNWAIT=NONE                            PENDDDL=YES                             UNLDDN=SYSREC
BMC50471I DSNUEXIT=(NONE,ASM)                     PREFORMAT=NO                            UNLDMAX=300%
BMC50471I DSNUTILB=YES                            RCVICDDN=(BMCIRY,BMCIRZ)                UNLOAD=RELOAD
BMC50471I DSPLOCKS=DRNFAIL                        RCVYDDN=(BMCRCY,BMCRCZ)                 UTSMEM=YES
BMC50471I DSRSEXIT=(NONE,REXX)                    REDEFINE=YES                            UXSTATE=SUP
BMC50471I EXCLDUMP=(X37,X22,X06)                  RENMMAX=30                              WBUFFS=(20,10)
BMC50471I FASTSWITCH=NO                           RIDMDSSZ=2097152                        WORKDDN=SYSUT1
BMC50471I FILECHK=WARN                            RIDMMAXD=1                              WORKUNIT=SYSALLDA
BMC50471I HASHAX=YES                              RMAPMEM=0                               XBMID=
BMC50471I ICDDN=(BMCICY,BMCICZ)                   RORGMAX=300%                            XML=YES
BMC50471I ICTYPE=AUTO                             ROUTCDE=(11,1)                          ZIIP=ENABLED
BMC50471I IDCACHE=10000                           SCPYMAX=8

BMC50471I PLAN=ARUQA

BMC50470I DDTYPE   = UNLOAD                             WORK                                 SORTWORK
BMC50470I ACTIVE   = YES                                YES                                  NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  ANY
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
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 DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        N/A
BMC50470I EXPDT    = N/A                                N/A                                  N/A
BMC50470I RETPD    = N/A                                N/A                                  N/A
BMC50470I GDGLIMIT = N/A                                N/A                                  N/A
BMC50470I GDGEMPTY = N/A                                N/A                                  N/A
BMC50470I GDGSCRAT = N/A                                N/A                                  N/A

BMC50470I DDTYPE   = ARCHIVE                            LOCPFCPY                             LOCPICPY
BMC50470I ACTIVE   = NO                                 YES                                  YES
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
```

Figure 48 SYSPRINT for example 14 (part 3 of 10)

```
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (5,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = LOCBFCPY                           LOCBICPY                             REMPFCPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A                                  N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = REMPICPY                           REMBFCPY                             REMBICPY
BMC50470I ACTIVE   = NO                                 NO                                   NO
BMC50470I IFALLOC  = USE                                USE                                  USE
BMC50470I ALLOC    = N/A                                N/A
BMC50470I SMS      = NO                                 NO                                   NO
BMC50470I SMSUNIT  = NO                                 NO                                   NO
BMC50470I SIZEPCT  = (100,100)                          (100,100)                            (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)                (SYSALLDA,SYSALLDA)                  (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)                              (0,0)                                (0,0)
BMC50470I VOLCNT   = (25,25)                            (25,25)                              (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))          ((30000,TRK),(30000,TRK))            ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)                        (NONE,NONE)                          (NONE,NONE)
BMC50470I THRESHLD = 0                                  0                                    0
BMC50470I MAXEXTSZ = ((0,K),(0,K))                      ((0,K),(0,K))                        ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 5                                  5                                    5
BMC50470I GDGEMPTY = NO                                 NO                                   NO
BMC50470I GDGSCRAT = NO                                 NO                                   NO

BMC50470I DDTYPE   = SYSPUNCH
BMC50470I ACTIVE   = YES
BMC50470I IFALLOC  = USE
BMC50470I ALLOC    = N/A
BMC50470I SMS      = NO
BMC50470I SMSUNIT  = NO
BMC50470I SIZEPCT  = (100,100)
BMC50470I UNIT     = (SYSALLDA,SYSALLDA)
BMC50470I UNITCNT  = (0,0)
BMC50470I VOLCNT   = (25,25)
BMC50470I AVGVOLSP = ((30000,TRK),(30000,TRK))
BMC50470I DATACLAS = (NONE,NONE)
BMC50470I MGMTCLAS = (NONE,NONE)
BMC50470I STORCLAS = (NONE,NONE)
```

Figure 48 SYSPRINT for example 14 (part 4 of 10)

```
BMC50470I THRESHLD = 0
BMC50470I MAXEXTSZ = ((0,K),(0,K))
BMC50470I EXPDT    =
BMC50470I RETPD    =
BMC50470I GDGLIMIT = 0
BMC50470I GDGEMPTY = NO
BMC50470I 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 LOCPICPY   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 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 REMBICPY   DSNPAT=&UID.&DDNAME.&TSIX..F&PART..T&TIME

BMC50471I DB2 DSNHDECP MODULE SETTINGS:
BMC50471I VERSION                 = 910
BMC50471I SUBSYSTEM DEFAULT       = DHV
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)

BMC50028I DB2 MODE = NFM
BMC50471I BMCUTIL ='BMCUTIL.CMN_BMCUTIL'
BMC50471I BMCSYNC ='BMCUTIL.CMN_BMCSYNC'
BMC50471I BMCHIST ='BMCUTIL.CMN_BMCHIST'
BMC50471I BMCDICT ='BMCUTIL.CMN_BMCDICT'
BMC50471I BMCXCOPY='BMCUTIL.CMN_BMCXCOPY'
BMC50471I DASD MANAGER PLUS TABLES:
BMC50471I ...TABLESPACE  ='ATS101.RS_TABLESPACE'
BMC50471I ...TABLEPART   ='ATS101.RS_TABLEPART'
BMC50471I ...TABLES      ='ATS101.RS_TABLES'
BMC50471I ...TSPART_DIS  ='ATS101.RS_TSPART_DIST'
BMC50471I ...INDEXES     ='ATS101.RS_INDEXES'
BMC50471I ...INDEXPART   ='ATS101.RS_INDEXPART'
BMC50471I ...IXPART_DIS  ='ATS101.RS_IXPART_DIST'
BMC50471I ...COLUMNS     ='ATS101.RS_COLUMNS'
BMC50471I ...COLSTATS    ='ATS101.RS_COLSTATS'
BMC50471I ...COLDIST     ='ATS101.RS_COLDIST'
BMC50471I ...STOGROUP    ='ATS101.RS_STOGROUP'

BMC50102I REORG TABLESPACE ARUEXP14.LARS$XBA
BMC50102I  SHRLEVEL CHANGE
BMC50102I  XBMID XBMA
BMC50102I  ORDER YES
BMC50102I  REBALANCE
BMC50102I  COPY YES
BMC50102I  ICTYPE INCREMENTAL
BMC50102I  LOGTHRESHLD 1000
BMC50102I  DELAY 300
BMC50102I  LOGSPILL (20480,10240)
BMC50102I  SPILLDSNPAT 'ARU.LARWORK.DHV'
BMC50102I  SPILLUNIT NONE
BMC50102I  SPILLSTORCLAS COPYCLAS
BMC50102I  ANALYZE SAMPLE
```

Figure 48 SYSPRINT for example 14 (part 5 of 10)

```
BMC50102I  PREFORMAT YES
BMC50102I  UPDATEDB2STATS YES
BMC50102I  BMCSTATS YES
BMC50102I  FASTSWITCH YES
BMC50102I  DDTYPE WORK     UNIT (WORK) SIZEPCT (150,150)
BMC50102I  DSNPAT 'ARU.EXMPL14.&DB.&TSIX.&DDNAME'
BMC50102I  DDTYPE LOCPFCPY UNIT (WORK) SIZEPCT (150,150)
BMC50102I  DSNPAT 'ARU.EXMPL14.&DB.&TSIX.&DDNAME'
BMC50102I  DDTYPE LOCPICPY UNIT (WORK)
BMC50102I  DSNPAT 'ARU.EXMPL14.&DB.&TSIX.&DDNAME'

BMC50138I 'COPY YES INLINE YES' CHANGED TO 'COPY YES INLINE NO' DUE TO ICTYPE INCREMENTAL
BMC50109I 'FASTSWITCH=YES' OPTION SPECIFIED.  OPTION 'STAGEDSN=BMC' IGNORED
BMC50004I UTILINIT PHASE COMPLETE.  ELAPSED TIME = 00:00:03

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC51301I 3: SAMPLING STATISTICS: PART=1,TP=8655,SP=116,SR=8372,AVGR=52,SD=66,SE=0,AVGF=7217,SD=65,SE=127,EP=4
BMC51301I 3: SAMPLING STATISTICS: PART=1,TP=8655,SP=233,SR=16531,AVGR=52,SD=66,SE=0,AVGF=7094,SD=65,SE=109,EP=12
BMC51301I 3: SAMPLING STATISTICS: PART=1,TP=8655,SP=350,SR=24820,AVGR=52,SD=66,SE=0,AVGF=7091,SD=65,SE=91,EP=19
BMC51301I 2: SAMPLING STATISTICS: PART=2,TP=4753,SP=108,SR=7417,AVGR=52,SD=66,SE=0,AVGF=6867,SD=65,SE=200,EP=9
BMC51301I 2: SAMPLING STATISTICS: PART=2,TP=4753,SP=216,SR=15037,AVGR=52,SD=66,SE=0,AVGF=6961,SD=65,SE=130,EP=15
BMC50482I 3: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.I0001.A001'
BMC51301I 2: SAMPLING STATISTICS: PART=2,TP=4753,SP=324,SR=22688,AVGR=52,SD=66,SE=0,AVGF=7002,SD=65,SE=103,EP=21
BMC50482I 2: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.I0001.A002'
BMC51301I 1: SAMPLING STATISTICS: PART=3,TP=1777,SP=102,SR=7284,AVGR=52,SD=66,SE=0,AVGF=7141,SD=65,SE=178,EP=6
BMC51301I 1: SAMPLING STATISTICS: PART=3,TP=1777,SP=202,SR=14572,AVGR=52,SD=66,SE=0,AVGF=7213,SD=65,SE=111,EP=9
BMC51301I 1: SAMPLING STATISTICS: PART=3,TP=1777,SP=303,SR=21378,AVGR=52,SD=66,SE=0,AVGF=7055,SD=65,SE=108,EP=20
BMC50482I 1: SAMPLE COMPLETE.  ELAPSED TIME = 00:00:00  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.I0001.A003'
BMC51265I ESTIMATED CARDINALITY OF SPACE = 1071475  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 52
BMC50484I ESTIMATED CARDINALITY OF PART 0001 = 357158  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 52
BMC50484I ESTIMATED CARDINALITY OF PART 0002 = 357158  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 52
BMC50484I ESTIMATED CARDINALITY OF PART 0003 = 357159  AVG SORTWK ROW LENGTH = 52  AVG UNLOAD ROW LENGTH = 52
BMC51264I UNLOAD WILL READ 15179 DATA PAGES FROM SPACE 'ARUEXP14.LARS$XBA'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I ANALYZE PHASE COMPLETE.  ELAPSED TIME = 00:00:00

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50425I &JOBNAME    JRGEX14A  &STEPNAME   LARREORG  &DB         ARUEXP14  &TSIX       LARS$XBA  &RTYPE      TS
BMC50425I &UID        RDAJRG4   &DATE       012711    &TIME       173256    &SSID       DHV2      &UTIL       ARUEXP14
BMC50425I &UTILPFX    ARUEXP14  &UTILSFX    RG1       &DATE8      01272011  &GRPNM      DHV       &VCAT       DSNDHV
BMC50425I &TIME4      1732      &DATEJ      2011027   &JDATE      11027

BMC50702I LOG PROCESSING STARTED AT 01/27/2011 17:33:08
BMC50773I 101: LOG SCAN TASK STARTED AT 01/27/2011 17:33:08
BMC50706I MONITOR TRACE CLASS(1) STARTED, TRACE NUMBER = 03
BMC50877I POINT OF CONSISTENCY ESTABLISHED AT RBA/LRSN = C73ECA98FB33
BMC50445I REORG PLUS DYNAMIC FILE ALLOCATION REPORT

BMC50446I                                                       UNIT OR                      KBYTES   KBYTES     ALOC      ALOC
BMC50447I DDNAME   DSNAME                                       DATACLAS MGMTCLAS STORCLAS      PRI      SEC      PRI       SEC

BMC50448I BMCCPY01 ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCCPY01      *WORK                          30371     3036      633        64  TRK
BMC50448I BMCCPY02 ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCCPY02      *WORK                          30371     3036      633        64  TRK
BMC50448I BMCCPY03 ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCCPY03      *WORK                          30371     3036      633        64  TRK
BMC50448I BMCICY01 ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCICY01      *WORK                           1013     1013       22        22  TRK
BMC50448I BMCICY02 ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCICY02      *WORK                           1013     1013       22        22  TRK
BMC50448I BMCICY03 ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCICY03      *WORK                           1013     1013       22        22  TRK
BMC50448I SYSUT101 ARU.EXMPL14.ARUEXP14.LARS$XBA.SYSUT101      *WORK                          36102     3611      669        67  TRK
BMC50448I SYSUT102 ARU.EXMPL14.ARUEXP14.LARS$XBA.SYSUT102      *WORK                          36102     3611      669        67  TRK
BMC50448I SYSUT103 ARU.EXMPL14.ARUEXP14.LARS$XBA.SYSUT103      *WORK                          31394     3140      582        59  TRK
BMC50448I SYSUT104 ARU.EXMPL14.ARUEXP14.LARS$XBA.SYSUT104      *WORK                          25115     2513      466        47  TRK
BMC50448I SYSUT105 ARU.EXMPL14.ARUEXP14.LARS$XBA.SYSUT105      *WORK                          15698     1571      291        30  TRK

BMC50394I UNABLE TO LOCATE SORT WORK DATASETS, DDNAME = 'SORTWKNN'
BMC50474I BELOW 16M = 8124K, ABOVE 16M = 1391100K, CPUS = 3
BMC50479I TOTAL PAGES: 2333105, ALLOWED: 0; AVAILABLE PAGES: 443478, ALLOWED: 443478
BMC50719I 1 LOG APPLY TASKS, 5 INDEX APPLY TASKS, AVAILABLE MEMORY: 1242292K
BMC50887I RIDMAP: RIDMAPMEM CHANGED TO 2097152K
BMC51302I MAX TASKS = 1, MAX PARTITIONS PER TASK = 3, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 3
BMC50864I LOGMEM OF 0K SPECIFIED, CHANGED TO 29184K
BMC50773I 108: LOG RECORD SORT TASK STARTED AT 01/27/2011 17:33:14
BMC50773I 109: LOG RECORD SPILL TASK STARTED AT 01/27/2011 17:33:14
BMC50370I STARTING 1 IMAGE COPY TASKS
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 2132K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 616704, I/O WAITS = 1 ,DDNAME = SYS00008
BMC50367I COMPRESSION DICTIONARY BUILT FOR TABLESPACE ARUEXP14.LARS$XBA PARTITION 1
```

Figure 48 SYSPRINT for example 14 (part 6 of 10)

```
BMC51271I UNLOAD STATISTICS:  616704 ROWS/KEYS UNLOADED FROM PARTITION 1
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 1
BMC50477I 1: PARTITION =    2, ROWS/KEYS = 337920, I/O WAITS = 1 ,DDNAME = SYS00009
BMC50367I COMPRESSION DICTIONARY BUILT FOR TABLESPACE ARUEXP14.LARS$XBA PARTITION 2
BMC51271I UNLOAD STATISTICS:  337920 ROWS/KEYS UNLOADED FROM PARTITION 2
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 2
BMC50477I 1: PARTITION =    3, ROWS/KEYS = 126720, I/O WAITS = 2 ,DDNAME = SYS00010
BMC50367I COMPRESSION DICTIONARY BUILT FOR TABLESPACE ARUEXP14.LARS$XBA PARTITION 3
BMC51271I UNLOAD STATISTICS:  126720 ROWS/KEYS UNLOADED FROM PARTITION 3
BMC51281I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR PARTITION 3
BMC51272I UNLOAD STATISTICS:  1081344 ROWS UNLOADED FROM SPACE 'ARUEXP14.LARS$XBA', 0 ROWS DISCARDED, 0 ROWS UPDATED
BMC51282I UNLOAD STATISTICS:  X'000000000000' IS THE HIGHEST LOGRBA FOR SPACE 'ARUEXP14.LARS$XBA'
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:05
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.J0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 360448, I/O WAITS = 10 ,DDNAME = SYS00015
BMC51274I RELOAD STATISTICS:  360448 ROWS/KEYS LOADED INTO PARTITION 1
BMC50375I FULL IMAGE COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.J0001.A001'
BMC50476I DDNAME = BMCCPY01, I/OS = 43, I/O WAITS = 41, RDB LOCK WAITS = 0
BMC50376I 5090 PAGES COPIED TO DATASET = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCCPY01'
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.J0001.A002'
BMC50477I 1: PARTITION =    2, ROWS/KEYS = 360448, I/O WAITS = 9 ,DDNAME = SYS00021
BMC51274I RELOAD STATISTICS:  360448 ROWS/KEYS LOADED INTO PARTITION 2
BMC50375I FULL IMAGE COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.J0001.A002'
BMC50476I DDNAME = BMCCPY02, I/OS = 43, I/O WAITS = 42, RDB LOCK WAITS = 0
BMC50376I 5074 PAGES COPIED TO DATASET = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCCPY02'
BMC50482I 1: RELOAD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.J0001.A003'
BMC50477I 1: PARTITION =    3, ROWS/KEYS = 360448, I/O WAITS = 10 ,DDNAME = SYS00027
BMC51274I RELOAD STATISTICS:  360448 ROWS/KEYS LOADED INTO PARTITION 3
BMC50375I FULL IMAGE COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.J0001.A003'
BMC50481I 1: REORG TASK COMPLETE.  ELAPSED TIME = 00:00:16
BMC50277I XBM STATISTICS: DSN='DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.I0001.A001   ', READS 8655, HITS 0, WRITES 0, CACHE 0
BMC50277I XBM STATISTICS: DSN='DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.I0001.A002   ', READS 4753, HITS 0, WRITES 0, CACHE 0
BMC50476I DDNAME = BMCCPY03, I/OS = 43, I/O WAITS = 39, RDB LOCK WAITS = 0
BMC50376I 5091 PAGES COPIED TO DATASET = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCCPY03'
BMC50773I 102: LOG APPLY TASK STARTED AT 01/27/2011 17:33:32
BMC50277I XBM STATISTICS: DSN='DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.I0001.A003   ', READS 1777, HITS 0, WRITES 0, CACHE 0
BMC50476I DDNAME = SYSUT104, I/OS = 53, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT101, I/OS = 76, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT105, I/OS = 34, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT102, I/OS = 76, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT103, I/OS = 66, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC51275I RELOAD STATISTICS:  1081344 ROWS LOADED INTO SPACE 'ARUEXP14.LARS$XBA'
BMC50474I BELOW 16M = 8060K, ABOVE 16M = 1379688K, CPUS = 3
BMC50479I TOTAL PAGES: 2333475, ALLOWED: 0; AVAILABLE PAGES: 436916, ALLOWED: 436916
BMC51303I MAX TASKS = 5, INDEXES PER TASK = 1, SORTWKS PER TASK = 32, MAX OPEN PARTITIONS PER TASK = 5
BMC50486I 2: BMCSORT STARTED, 256K BELOW 16M, 1396K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 1: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 4: BMCSORT STARTED, 256K BELOW 16M, 1396K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 3: BMCSORT STARTED, 256K BELOW 16M, 1280K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50486I 5: BMCSORT STARTED, 256K BELOW 16M, 1336K TOTAL MEMORY, 0 PAGES HYPERSPACE
BMC50476I DDNAME = SYSUT104, I/OS = 51, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT105, I/OS = 32, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50481I 1: SORT COMPLETE.  ELAPSED TIME = 00:00:03
BMC50481I 3: SORT COMPLETE.  ELAPSED TIME = 00:00:02
BMC50476I DDNAME = SYSUT101, I/OS = 74, I/O WAITS = 4, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT102, I/OS = 74, I/O WAITS = 2, RDB LOCK WAITS = 0
BMC50476I DDNAME = SYSUT103, I/OS = 64, I/O WAITS = 6, RDB LOCK WAITS = 0
BMC50481I 2: SORT COMPLETE.  ELAPSED TIME = 00:00:04
BMC50481I 4: SORT COMPLETE.  ELAPSED TIME = 00:00:04
BMC50481I 5: SORT COMPLETE.  ELAPSED TIME = 00:00:03
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:07  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003A.J0001.A001'
BMC50477I 1: PARTITION =    1, ROWS/KEYS = 360448, I/O WAITS = 13 ,DDNAME = SYS00052
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:07  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003C.J0001.A001'
BMC50477I 2: PARTITION =    1, ROWS/KEYS = 360448, I/O WAITS = 20 ,DDNAME = SYS00062
BMC50482I 4: BUILD COMPLETE.  ELAPSED TIME = 00:00:08  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003B.J0001.A001'
BMC50477I 4: PARTITION =    0, ROWS/KEYS = 1081344, I/O WAITS = 52 ,DDNAME = SYS00060
BMC50482I 3: BUILD COMPLETE.  ELAPSED TIME = 00:00:09  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003E.J0001.A001'
BMC50477I 3: PARTITION =    1, ROWS/KEYS = 360448, I/O WAITS = 7 ,DDNAME = SYS00053
BMC50773I 103: INDEX APPLY TASK STARTED AT 01/27/2011 17:33:46
BMC51276I BUILD  STATISTICS:  1081344 KEYS LOADED INTO INDEX 'ARUEXP14.LARX003B'
BMC50482I 5: BUILD COMPLETE.  ELAPSED TIME = 00:00:08  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003D.J0001.A001'
BMC50477I 5: PARTITION =    0, ROWS/KEYS = 1081344, I/O WAITS = 37 ,DDNAME = SYS00061
BMC51276I BUILD  STATISTICS:  1081344 KEYS LOADED INTO INDEX 'ARUEXP14.LARX003D'
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:05  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003A.J0001.A002'
BMC50477I 1: PARTITION =    2, ROWS/KEYS = 360448, I/O WAITS = 14 ,DDNAME = SYS00071
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:04  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003C.J0001.A002'
```

Figure 48 SYSPRINT for example 14 (part 7 of 10)

```
BMC50477I 2: PARTITION =    2, ROWS/KEYS = 360448, I/O WAITS = 20 ,DDNAME = SYS00079
BMC50773I 104: INDEX APPLY TASK STARTED AT 01/27/2011 17:33:49
BMC50482I 3: BUILD COMPLETE.  ELAPSED TIME = 00:00:04  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003E.J0001.A002'
BMC50477I 3: PARTITION =    2, ROWS/KEYS = 360448, I/O WAITS = 8 ,DDNAME = SYS00080
BMC50482I 2: BUILD COMPLETE.  ELAPSED TIME = 00:00:02  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003C.J0001.A003'
BMC50477I 2: PARTITION =    3, ROWS/KEYS = 360448, I/O WAITS = 15 ,DDNAME = SYS00095
BMC50482I 1: BUILD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003A.J0001.A003'
BMC50477I 1: PARTITION =    3, ROWS/KEYS = 360448, I/O WAITS = 14 ,DDNAME = SYS00091
BMC50773I 105: INDEX APPLY TASK STARTED AT 01/27/2011 17:33:52
BMC50773I 106: INDEX APPLY TASK STARTED AT 01/27/2011 17:33:52
BMC51276I BUILD  STATISTICS:  1081344 KEYS LOADED INTO INDEX 'ARUEXP14.LARX003C'
BMC51276I BUILD  STATISTICS:  1081344 KEYS LOADED INTO INDEX 'ARUEXP14.LARX003A'
BMC50482I 3: BUILD COMPLETE.  ELAPSED TIME = 00:00:03  DSN = 'DSNDHV.DSNDBD.ARUEXP14.LARX003E.J0001.A003'
BMC50477I 3: PARTITION =    3, ROWS/KEYS = 360448, I/O WAITS = 7 ,DDNAME = SYS00098
BMC50773I 107: INDEX APPLY TASK STARTED AT 01/27/2011 17:33:53
BMC50829I LOGAPPLY PHASE STARTING AT 01/27/2011 17:33:53
BMC51276I BUILD  STATISTICS:  1081344 KEYS LOADED INTO INDEX 'ARUEXP14.LARX003E'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT103', DSNAME = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.SYSUT103'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT102', DSNAME = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.SYSUT102'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT101', DSNAME = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.SYSUT101'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT105', DSNAME = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.SYSUT105'
BMC50318I DATASET SUCCESSFULLY DELETED, DDNAME = 'SYSUT104', DSNAME = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.SYSUT104'
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I REORG PHASE COMPLETE.  ELAPSED TIME = 00:00:44

BMC50041I 0: ZIIP  ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50877I POINT OF CONSISTENCY ESTABLISHED AT RBA/LRSN = C73ECAC79CCE
BMC50778I LOG RECORD QUEUE SIZE AT OR BELOW THRESHOLD
BMC50004I LOGAPPLY PHASE COMPLETE.  ELAPSED TIME = 00:00:06
BMC50830I LOGFINAL PHASE STARTING AT 01/27/2011 17:33:59
BMC50709I MONITOR TRACE CLASS(1) TRACE NUMBER = 03 STOPPED
BMC50786I 101: LOG SCAN TASK ENDED AT 01/27/2011 17:33:59, ELAPSED TIME = 00:00:51
BMC50720I 102: 0 NEW LOG APPLY BUFFERS, 0 REUSED, 282488 INITIAL MAXIMUM, 0 WAITS, BUFFER SIZE=534
BMC50741I 102: LOGAPPLY STATISTICS
BMC50742I 102: PROCESSED TRANSACTIONS
BMC50744I 102:    INSERTS:          0 INSERTS(COMPENSATION):          0
BMC50745I 102:    UPDATES:          0 UPDATES(COMPENSATION):          0
BMC50746I 102:    DELETES:          0 DELETES(COMPENSATION):          0
BMC50743I 102: CONSOLIDATED TRANSACTIONS
BMC50744I 102:    INSERTS:          0 INSERTS(COMPENSATION):          0
BMC50745I 102:    UPDATES:          0 UPDATES(COMPENSATION):          0
BMC50746I 102:    DELETES:          0 DELETES(COMPENSATION):          0
BMC50747I 102: PAGES UPDATED COUNTERS FOR ARUEXP14.LARS$XBA
BMC50748I 102:     PART     TOTAL PAGES     DATA PAGES     SPACE MAPS     HEADER PAGES
BMC50749I 102:        1              1              0               0               1
BMC50749I 102:        2              1              0               0               1
BMC50749I 102:        3              1              0               0               1
BMC50716I 102: Buffer Manager Statistics for the following datasets:
BMC50717I 102:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARS$XBA.J0001.A001
BMC50717I 102:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARS$XBA.J0001.A002
BMC50717I 102:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARS$XBA.J0001.A003
BMC50718I 102:        Page Requests          6 |            Page Hits          0
BMC50718I 102:           Page Reads          6 |          Page Writes          1
BMC50718I 102:   Write I/O Requests          3 |            New Pages          0
BMC50716I 102:       Write I/O time    0:00:00 |        Read I/O time    0:00:00
BMC50786I 102: LOG APPLY TASK ENDED AT 01/27/2011 17:34:02, ELAPSED TIME = 00:00:30
BMC50786I 108: LOG RECORD SORT TASK ENDED AT 01/27/2011 17:34:02, ELAPSED TIME = 00:00:47
BMC50375I INCREMENTAL COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.J0001.A001'
BMC50375I INCREMENTAL COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.J0001.A003'
BMC50476I DDNAME = BMCICY01, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 1 PAGES COPIED TO DATASET = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCICY01'
BMC50375I INCREMENTAL COPY STARTED FOR DATASET = 'DSNDHV.DSNDBD.ARUEXP14.LARS$XBA.J0001.A002'
BMC50476I DDNAME = BMCICY03, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 1 PAGES COPIED TO DATASET = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCICY03'
BMC50476I DDNAME = BMCICY02, I/OS = 1, I/O WAITS = 1, RDB LOCK WAITS = 0
BMC50376I 1 PAGES COPIED TO DATASET = 'ARU.EXMPL14.ARUEXP14.LARS$XBA.BMCICY02'
BMC50794I 105: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003E DSNUM 2
BMC50791I 105:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 105:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 105:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 105:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 105:   NEW PAGES   :         0  BUFF PAGE READS :         0
BMC50794I 105: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003C DSNUM 3
BMC50791I 105:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 105:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 105:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
```

Figure 48 SYSPRINT for example 14 (part 8 of 10)

```
BMC50791I 105:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 105:   NEW PAGES   :         0  BUFF PAGE READS :         0
BMC50716I 105: Buffer Manager Statistics for the following datasets:
BMC50717I 105:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003C.J0001.A003
BMC50717I 105:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003E.J0001.A002
BMC50718I 105:        Page Requests         10 |            Page Hits          2
BMC50718I 105:           Page Reads          8 |          Page Writes          1
BMC50718I 105:   Write I/O Requests          2 |            New Pages          0
BMC50716I 105:       Write I/O time    0:00:00 |        Read I/O time    0:00:00
BMC50786I 105: INDEX APPLY TASK ENDED AT 01/27/2011 17:34:02, ELAPSED TIME = 00:00:10
BMC50794I 106: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003D
BMC50791I 106:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 106:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 106:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 106:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 106:   NEW PAGES   :         0  BUFF PAGE READS :         0
BMC50794I 106: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003A DSNUM 3
BMC50791I 106:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 106:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 106:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 106:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 106:   NEW PAGES   :         0  BUFF PAGE READS :         0
BMC50716I 106: Buffer Manager Statistics for the following datasets:
BMC50717I 106:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003A.J0001.A003
BMC50717I 106:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003D.J0001.A001
BMC50718I 106:        Page Requests         10 |            Page Hits          2
BMC50718I 106:           Page Reads          8 |          Page Writes          1
BMC50718I 106:   Write I/O Requests          2 |            New Pages          0
BMC50716I 106:       Write I/O time    0:00:00 |        Read I/O time    0:00:00
BMC50786I 106: INDEX APPLY TASK ENDED AT 01/27/2011 17:34:02, ELAPSED TIME = 00:00:10
BMC50794I 107: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003B
BMC50791I 107:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 107:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 107:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 107:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 107:   NEW PAGES   :         0  BUFF PAGE READS :         0
BMC50794I 107: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003E DSNUM 3
BMC50791I 107:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 107:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 107:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 107:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 107:   NEW PAGES   :         0  BUFF PAGE READS :         0
BMC50716I 107: Buffer Manager Statistics for the following datasets:
BMC50717I 107:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003E.J0001.A003
BMC50717I 107:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003B.J0001.A001
BMC50718I 107:        Page Requests         10 |            Page Hits          2
BMC50718I 107:           Page Reads          8 |          Page Writes          1
BMC50718I 107:   Write I/O Requests          2 |            New Pages          0
BMC50716I 107:       Write I/O time    0:00:00 |        Read I/O time    0:00:00
BMC50786I 107: INDEX APPLY TASK ENDED AT 01/27/2011 17:34:03, ELAPSED TIME = 00:00:10
BMC50794I 104: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003A DSNUM 2
BMC50791I 104:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 104:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 104:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 104:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 104:   NEW PAGES   :         0  BUFF PAGE READS :         0
BMC50794I 104: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003C DSNUM 2
BMC50791I 104:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 104:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 104:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 104:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 104:   NEW PAGES   :         0  BUFF PAGE READS :         0
BMC50716I 104: Buffer Manager Statistics for the following datasets:
BMC50717I 104:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003A.J0001.A002
BMC50717I 104:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003C.J0001.A002
BMC50718I 104:        Page Requests         10 |            Page Hits          2
BMC50718I 104:           Page Reads          8 |          Page Writes          1
BMC50718I 104:   Write I/O Requests          2 |            New Pages          0
BMC50716I 104:       Write I/O time    0:00:00 |        Read I/O time    0:00:00
BMC50786I 104: INDEX APPLY TASK ENDED AT 01/27/2011 17:34:04, ELAPSED TIME = 00:00:15
BMC50794I 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003A DSNUM 1
BMC50791I 103:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 103:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 103:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 103:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 103:   NEW PAGES   :         0  BUFF PAGE READS :         0
```

Figure 48 SYSPRINT for example 14 (part 9 of 10)

```
BMC50794I 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003C DSNUM 1
BMC50791I 103:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 103:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 103:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 103:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 103:   NEW PAGES   :         0  BUFF PAGE READS :         0
BMC50794I 103: INDEX MAINTENANCE STATISTICS FOR INDEX ARUEXP14.LARX003E DSNUM 1
BMC50791I 103:   PAGE READS  :         1  KEY INSERTS     :         0
BMC50791I 103:   RID INSERTS :         0  RID DELETES (PS):         0
BMC50791I 103:   ROOT SPLITS :         0  NON-LEAF SPLITS :         0
BMC50791I 103:   LEAF SPLITS :         0  PAGE FREES      :         0
BMC50791I 103:   NEW PAGES   :         0  BUFF PAGE READS :         0
BMC50716I 103: Buffer Manager Statistics for the following datasets:
BMC50717I 103:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003A.J0001.A001
BMC50717I 103:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003C.J0001.A001
BMC50717I 103:  DATASET=DSNDHV.DSNDBC.ARUEXP14.LARX003E.J0001.A001
BMC50718I 103:        Page Requests         15 |            Page Hits          3
BMC50718I 103:           Page Reads         12 |          Page Writes          1
BMC50718I 103:   Write I/O Requests          3 |            New Pages          0
BMC50716I 103:       Write I/O time    0:00:00 |        Read I/O time    0:00:00
BMC50786I 103: INDEX APPLY TASK ENDED AT 01/27/2011 17:34:06, ELAPSED TIME = 00:00:20
BMC50867I LOG RECORD STORE SPILL REQUESTS = 0, HIGH SPILL PAGE = 0, SPILL DATASETS CREATED = 0
BMC50868I LOG RECORD STORE WRITE REQUESTS = 0, WRITE WAIT TIME =   0.000, READ REQUESTS = 0 READ WAIT TIME =   0.000
BMC50786I 109: LOG RECORD SPILL TASK ENDED AT 01/27/2011 17:34:06, ELAPSED TIME = 00:00:51
BMC50865I TOTAL RECORDS INSERTED INTO LOG RECORD STORE, DATA =0, INDEX = 0
BMC50866I TOTAL TABLESPACE STRINGS PROCESSED = 0, WHEN NOT FULL = 0, FROM SPILL = 0
BMC50866I TOTAL INDEXSPACE STRINGS PROCESSED = 0, WHEN NOT FULL = 0, FROM SPILL = 0
BMC50866I TOTAL INDEXSPACE REQUE STRINGS PROCESSED = 0, WHEN NOT FULL = 0, FROM SPILL = 0
BMC50859I LOG RECORD STORE STATISTICS:  MEMORY AVAILABLE = 29184K, MEMORY USED = 5120K
BMC50860I TOTAL WAIT TIME FOR LOG RECORD STORE MEMORY =   0.000 DATA =   0.000 INDEX =   0.000
BMC50880I RIDMAP: PART 0001: ROWS=616704, PAIRS=616704, STORAGE=4700K, MEM WAITS=1, WAIT TIME=0
BMC50881I RIDMAP: PART 0001: TRANS=0, ADDS=0, DELS=0, SPILL READS=0, WAIT TIME=0
BMC50880I RIDMAP: PART 0002: ROWS=337920, PAIRS=337920, STORAGE=2592K, MEM WAITS=2, WAIT TIME=0
BMC50881I RIDMAP: PART 0002: TRANS=0, ADDS=0, DELS=0, SPILL READS=0, WAIT TIME=0
BMC50880I RIDMAP: PART 0003: ROWS=126720, PAIRS=126720, STORAGE=3356K, MEM WAITS=5, WAIT TIME=0
BMC50881I RIDMAP: PART 0003: TRANS=0, ADDS=0, DELS=0, SPILL READS=0, WAIT TIME=0
BMC50880I RIDMAP: SUMMARY: ROWS=1081344, PAIRS=1081344, STORAGE=10648K, MEM WAITS=8, WAIT TIME=0
BMC50881I RIDMAP: SUMMARY: TRANS=0, ADDS=0, DELS=0, SPILL READS=0, WAIT TIME=0
BMC50882I RIDMAP: SUMMARY: MAX DATASPACES=1, USED DATASPACES=1, RIDMAPMEM=0K, USED MEM=11772K
BMC50883I RIDMAP: SUMMARY: MAX PIPES=4, PIPES USED=2, PIPE BUFFER SIZE=262144, PIPE WAITS=0, WAIT TIME=0
BMC50868I RIDMAP WRITE REQUESTS = 0, WRITE WAIT TIME =   0.000, READ REQUESTS = 0 READ WAIT TIME =   0.000
BMC50703I LOG PROCESSING COMPLETED, RC = 0, AT 01/27/2011 17:34:06, ELAPSED TIME = 00:00:57
BMC50041I 0: ZIIP NOT ENABLED (0) USING XBM SUBSYSTEM XBMA
BMC50004I LOGFINAL PHASE COMPLETE.  ELAPSED TIME = 00:00:07

BMC50894I SWITCH PROCESS STARTING AT   1/27/2011  17:34:07
BMC50895I SWITCH PROCESS COMPLETE.  ELAPSED TIME =  00:00:00

BMC51297I ALL SPECIFIED LIMIT KEYS HAVE BEEN SUCCESSFULLY ALTERED
BMC50890I 1: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARS$XBA.I0001.A001'...
BMC50890I 11: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003E.I0001.A002'...
BMC50890I 7: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003C.I0001.A001'...
BMC50890I 6: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003A.I0001.A003'...
BMC50890I 4: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003A.I0001.A001'...
BMC50890I 9: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003C.I0001.A003'...
BMC50890I 12: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003E.I0001.A003'...
BMC50890I 2: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARS$XBA.I0001.A002'...
BMC50890I 8: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003C.I0001.A002'...
BMC50890I 5: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003A.I0001.A002'...
BMC50890I 10: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003E.I0001.A001'...
BMC50890I 3: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARS$XBA.I0001.A003'...
BMC50890I 14: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003D.I0001.A001'...
BMC50890I 13: ATTEMPTING TO DELETE DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003B.I0001.A001'...
BMC50891I 11: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003E.I0001.A002' IS DELETED
BMC50891I 1: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARS$XBA.I0001.A001' IS DELETED
BMC50891I 6: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003A.I0001.A003' IS DELETED
BMC50891I 4: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003A.I0001.A001' IS DELETED
BMC50891I 9: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003C.I0001.A003' IS DELETED
BMC50891I 7: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003C.I0001.A001' IS DELETED
BMC50891I 12: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003E.I0001.A003' IS DELETED
BMC50891I 8: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003C.I0001.A002' IS DELETED
BMC50891I 5: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003A.I0001.A002' IS DELETED
BMC50891I 2: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARS$XBA.I0001.A002' IS DELETED
BMC50891I 3: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARS$XBA.I0001.A003' IS DELETED
BMC50891I 10: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003E.I0001.A001' IS DELETED
BMC50891I 14: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003D.I0001.A001' IS DELETED
```

Figure 48 SYSPRINT for example 14 (part 10 of 10)

```
BMC50891I 13: DATASET 'DSNDHV.DSNDBC.ARUEXP14.LARX003B.I0001.A001' IS DELETED

BMC50290I DB2 REAL-TIME-STATISTICS -RESET STATS- FUNCTION FOR REORG UTILITY SUCCESSFUL FOR ALL OBJECTS
BMC50006I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Figure 49 shows the DDLOUT output for example 14.

Figure 49 DDLOUT for example 14

```
ALTER TABLE "ARUEXP14"."LART003" ALTER PARTITION 0001 ENDING AT ('2760-02-28-15.00.00.000000',1000);
ALTER TABLE "ARUEXP14"."LART003" ALTER PARTITION 0002 ENDING AT ('3610-04-28-07.00.00.000000',7400);
```

Chapter 6

Tuning REORG PLUS jobs

This chapter presents the following topics:

Tuning for performance

REORG PLUS provides several options that you can use to influence its performance. Some options affect the performance of the entire product; others influence processing only for specific functions of REORG PLUS. This section explains the effect that these options have on the performance of REORG PLUS.

Setting installation options for optimal performance

In general, you should use the values that were shipped with REORG PLUS for the installation options. These values usually enable optimal performance. However, for certain environments, BMC recommends that you make adjustments to these values. Table 78 describes these recommendations. For a complete description of each installation option, see Appendix A, “REORG PLUS installation options.”

For additional installation option changes for SHRLEVEL CHANGE, see page 523.

NOTE

If you modify these options after installation, you must resubmit the installation job for the modifications to take effect.

Table 78 Installation default option changes for performance (part 1 of 2)

Installation option	Recommended value	Comments
BMCHIST	NO	Set this value if you do not need to update the BMCHIST table.
COPYDDN	four-character prefix	To avoid errors when making partition-level copies for a large number of partitions, specify prefixes that are four characters or less.
KEEPDICTIONARY	YES	Consider this value to avoid building a new dictionary if one already exists.
REDEFINE	NO	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.

Table 78 Installation default option changes for performance (part 2 of 2)

Installation option	Recommended value	Comments
SMCORE	(0K,0K)	This value is the default. BMC recommends that you do not change this value.
STAGEDSN	DSN	For SHRLEVEL CHANGE and SHRLEVEL REFERENCE, specify this value to avoid redundant data set rules.

Additional installation options for SHRLEVEL CHANGE only

The recommendations in Table 78 on page 522 apply to all reorganizations. BMC recommends the additional changes in Table 79 to enhance performance of SHRLEVEL CHANGE jobs. You can change most of these options on the REORG command. For more information, see “Specifying command options for optimal performance” on page 524. For more information about using SHRLEVEL CHANGE jobs, see Chapter 7, “Online reorganization.”

Table 79 SHRLEVEL CHANGE installation default option changes for performance

Installation option	Value and description
LOGSPIL	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.
MAXRO	The MAXRO option specifies the maximum number of seconds that you want REORG PLUS to spend applying log records in the LOGFINAL phase. **Note:** The value for MAXRO applies only to the time that REORG PLUS spends applying log records in the LOGFINAL phase, not to the read-only time spent producing the incremental image copies and running the UTILTERM phase.
SIXSNAP	If your environment contains the appropriate hardware, specify AUTO to ensure that REORG PLUS uses the hardware if it is available and falls back to a regular copy if the hardware is not available. Specifying AUTO can improve performance. However, if the object that you are reorganizing has a large number of indexes or multiple data set indexes, SQLCODE –911 timeouts might occur because of the serialization of the Instant Snapshot requests, which are performed on a per-index basis. In this scenario, change SIXSNAP to NO.
SPILDSNP	Specify the data set name pattern to use when allocating the LOGSPIL data set. The default value is &&UID. To prevent duplicate names when the same user runs two jobs concurrently, change the default value.
SPILSCLS	Specify the SMS storage class, if any, from which you want to obtain the LOGSPIL data set volume.
SPILUNIT	Specify the device type or generic unit type to which you want the LOGSPIL data set allocated.
XBMID	Specify the EXTENDED BUFFER MANAGER (XBM) subsystem ID for the processor on which you are running.

Specifying command options for optimal performance

Table 80 lists REORG command options that correspond to installation options that are listed in Table 78 on page 522. BMC recommends that you specify the REORG command options that are listed in Table 80 if they are not already specified in the installation options.

If you are using SHRLEVEL CHANGE, also see Table 81 for additional command options.

NOTE

Some of these recommendations apply only to specific environments or conditions. Review the information in these tables to determine whether the recommendations meet your needs.

Table 80 REORG PLUS command options that correspond to installation options

Command option	Corresponding installation option
BMCHIST NO	BMCHIST=NO
FASTSWITCH YES	FASTSWITCH=YES
KEEPDICTIONARY YES	KEEPDICTIONARY=YES
REDEFINE NO	REDEFINE=NO
UNITCNT *n* where *n* is only the number of devices that you need	UNITCNT=*n* where *n* is only the number of devices that you need
VOLCNT AUTO	VOLCNT=AUTO

Additional command options for SHRLEVEL CHANGE

Table 81 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 81 SHRLEVEL CHANGE command options that correspond to installation options (part 1 of 2)

Command option	Corresponding installation option
DEADLINE	DEADLINE
DELAY	DELAY
LOGFINAL	LOGFINAL
LOGMEM	LOGMEM
LOGSPILL	LOGSPIL

Table 81 SHRLEVEL CHANGE command options that correspond to installation options (part 2 of 2)

Command option	Corresponding installation option
LOGTHRESHLD	LOGTHRSH
LONGLOG	LONGLOG
MAXRO	MAXRO
RIDMAPMEM	RMAPMEM
Not available	RIDMDSSZ
Not available	RIDMMAXD
SIXSNAP	SIXSNAP
SPILLDSNPAT	SPILDSNP
SPILLSTORCLAS	SPILSCLS
SPILLUNIT	SPILUNIT
XBMID	XBMID

Table 82 describes additional recommended command options. (No installation options correspond to these command options.)

Table 82 Additional REORG PLUS command syntax options

Command option	Conditions and comments
ON FAILURE *phase* TERMINATE UTILITY RETCODE *integer*	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.
ORDER NO	Specify this option if you do not need sorted data.

Additional performance information for installation and command options

The following sections provide additional information for some installation and command options. This information can help you make decisions about these options that can affect the performance of your reorganization jobs. Use Table 83 to find information about a specific option.

Table 83 Additional performance information about options (part 1 of 2)

Option	Description	See page
ANALYZE	provides information to help you determine data set sizes and the optimal number of tasks	526
AVAILPAGEPCT	controls the amount of system memory that REORG PLUS can allocate to BMCSORT for sort processing	536
CBUFFS	controls buffers for writing copy data sets	531

Table 83 Additional performance information about options (part 2 of 2)

Option	Description	See page
DDTYPE	controls dynamic allocation	535
MAXSORTS	controls number of sort processes	536
ORDER	controls whether REORG PLUS sorts the data	536
MAXSORTMEMORY	maximum amount of memory that REORG PLUS can allocate to each sort task	536
MINSORTMEMORY	minimum amount of memory that REORG PLUS should allocate to each sort task	536
SHORTMEMORY	controls the action that REORG PLUS takes when a memory shortage exists that affects sort processing	536
SMAX	controls number of sort processes	536
SMCORE	controls sort memory usage	536
UXSTATE	controls DB2 user exit processing	540
WBUFFS	controls buffers for reading and writing SYSUT1 data sets	531

ANALYZE command options

The ANALYZE options limit abends caused by inadequate size allocations for data sets. These options provide input to dynamic allocation processing or provide information that allows you to allocate space more accurately. REORG PLUS uses the results of the ANALYZE phase to verify work data set sizes and optimize tasks. For details about the ANALYZE command options, including restrictions on each value, see “ANALYZE” on page 185.

The SCAN, SAMPLE, HURBA, and BMCSTATS options determine how REORG PLUS gathers the information that it needs for analysis. Table 84 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 84 Time versus accuracy for the ANALYZE options

Option	Time required	Accuracy
SCAN	most	most
SAMPLE	more	less
HURBA	less	least
BMCSTATS[a]	least	most 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 Database Performance solution.

ANALYZE

For both table space and index space reorganizations, REORG PLUS decides whether to sample or scan for cardinality if you specify one of the following options:

- ANALYZE with no other ANALYZE options
- ANALYZE ONLY or ANALYZE PAUSE with no other ANALYZE options

For a table space reorganization, unless you specify ANALYZE BMCSTATS, REORG PLUS always samples the table space for the average row length.

ANALYZE SAMPLE and SCAN

Table 85 describes the actions that REORG PLUS takes based on the object that you are reorganizing and the ANALYZE option that you specify.

Table 85 REORG PLUS actions for ANALYZE option

Reorganization type	ANALYZE option	REORG PLUS actions
table space	SAMPLE	Samples the table space to ■ estimate the cardinality ■ determine the average row length **Note**: For an XML table space reorganization, REORG PLUS obtains index cardinality from the DB2 real-time statistics tables.
	SCAN	■ To determine the cardinality: — scans one index on each table to find the exact cardinality. — samples each table with no index to estimate the cardinality for that table. ■ Samples the table space to determine the average row length.
index	SAMPLE	Samples the index to estimate the cardinality.
	SCAN	Scans each leaf page of the index to determine the exact cardinality.

ANALYZE HURBA

If you specify ANALYZE HURBA for either a table space or an index reorganization, REORG PLUS does not read the table space or the index during the ANALYZE phase. Instead, it uses the high-used relative byte address (HURBA) and information about column lengths from the DB2 catalog to estimate the cardinality and average row length.

ANALYZE BMCSTATS

The BMCSTATS option is available only if you are using REORG PLUS as a component of the Database Performance for DB2 solution. Otherwise, REORG PLUS changes BMCSTATS to SAMPLE and continues the reorganization.

If you specify ANALYZE BMCSTATS for either a table space or an index reorganization, REORG PLUS uses the information that already exists in the DASD MANAGER PLUS statistics tables to perform the analysis. If the information in the DASD MANAGER PLUS statistics tables is current, using ANALYZE BMCSTATS is as accurate as using ANALYZE SCAN, and is the fastest method available because REORG PLUS does not have to gather the data.

If the DASD MANAGER PLUS statistics tables contain incomplete information for the object that you are reorganizing, REORG PLUS changes BMCSTATS to SAMPLE and continues the reorganization.

ANALYZE PAUSE and ONLY

The ANALYZE PAUSE and ANALYZE ONLY options provide estimates of the space needed for the following data sets:

- unload (SYSREC)
- work (SYSUT1)
- sort (SORTWK)
- full image copy (BMCCPY, BMCCPZ, BMCRCY, and BMCRCZ)
- incremental image copy (BMCICY, BMCICZ, BMCIRY, and BMCIRZ)

REORG PLUS writes the statistics to the SYSPRINT data set. You must specify PAUSE or ONLY to get this statistics report.

If you specify ANALYZE ONLY and use the information to allocate your data sets, you can improve performance by changing the REORG command option to ANALYZE HURBA when you rerun the job. Specifying ANALYZE HURBA bypasses the ANALYZE phase. For exceptions when specifying HURBA, see the restrictions listed in “ANALYZE HURBA” on page 527.

ANALYZE messages

When you specify ANALYZE PAUSE or ANALYZE ONLY, REORG PLUS issues the messages listed in this section. For more information about using the information from the ANALYZE phase for allocating your data sets, see “ANALYZE option for estimating data set allocation” on page 357.

If you do not want to specify ANALYZE PAUSE or ANALYZE ONLY, you must either have REORG PLUS dynamically allocate your data sets (see "Dynamic allocation options" on page 269), or calculate the data set sizes yourself.

Message overview

The ANALYZE messages provide information about data set size allocation in a table format:

- Messages BMC51260I through BMC51262I provide heading information.
- Multiple BMC51263I messages provide the estimated values.
- Messages BMC51264I through BMC51266I provide estimates that you can use to gauge the elapsed time of the reorganization.

Message descriptions

This section lists and describes the messages that REORG PLUS issues when you specify ANALYZE PAUSE or ANALYZE ONLY.

```
BMC51260I             REORG PLUS DASD REQUIREMENT ESTIMATES

BMC51261I                                 3380 CYLS     3390 CYLS

BMC51262I        DDNAME      KBYTES      PRI   SEC     PRI   SEC     INDEX

BMC51263I        ddname   k              p    s        p    s     creator.indexName
```

Explanation: After providing heading information in messages BMC51260I through BMC51262I, REORG PLUS issues a separate BMC51263I message for each data set to provide the following information:

- data set name
- number of kilobytes
- primary and secondary 3380 cylinder quantities
- primary and secondary 3390 cylinder quantities
- index name, where applicable

User Response: No action is required.

BMC51264I **UNLOAD WILL READ *n* DATA PAGES FROM SPACE '*databaseName.tableSpaceName*'**

Explanation: This message provides the count of data pages that REORG PLUS will read during the unload process.

User Response: Because the reorganization process is I/O bound, you can use this number, along with the cardinality, to gauge the elapsed time of the reorganization, based on the performance of previous runs of the utility.

BMC51265I

ESTIMATED CARDINALITY OF SPACE = *n*
ESTIMATED CARDINALITY OF SPACE = *n*
AVG SORTWK ROW LENGTH = *n*
AVG UNLOAD ROW LENGTH = *n*

Explanation: This message provides an estimate of the cardinality of the table space or index space. For table space reorganizations, the message also provides an estimate of the average row length for the SORTWK and SYSREC data sets.

User Response: You can compare these estimates with estimates provided by other messages and previous runs of the utility to gauge the elapsed time of the reorganization.

BMC51266I

CARDINALITY OF SPACE = *n*
CARDINALITY OF SPACE = *n* AVG SORTWK ROW LENGTH = *n*
AVG UNLOAD ROW LENGTH = *n*

Explanation: This message provides the exact cardinality of the table space or index space. For table space reorganizations, the message also provides an estimate of the average row length for the SORTWK and SYSREC data sets.

User Response: You can compare these values with estimates provided by other messages and previous runs of the utility to gauge the elapsed time of the reorganization.

BMC50484I

ESTIMATED CARDINALITY OF *objectType* = *n* AVG SORTWK ROW LENGTH = *n*
AVG UNLOAD ROW LENGTH = *n*

Explanation: This message provides either an estimate of or the exact cardinality of each table in a nonpartitioned table space, or each partition in a partitioned table space. The message also provides an estimate of the average row length for the SORTWK and SYSREC data sets.

User Response: Use this message in conjunction with message BMC51265I or BMC51266I.

BMC50485I

ESTIMATED CARDINALITY OF PART *partitionNumber* = *n*
ESTIMATED CARDINALITY OF TABLE *tableName* = *n*

Explanation: REORG PLUS issues this message for an index-only reorganization. This message provides either an estimate of or the exact cardinality of each partition in a partitioned index space.

User Response: Use this message in conjunction with message BMC51265I or BMC51266I.

Buffer installation options

You can use installation options to control the amount of buffer space that the data sets use during REORG PLUS processing. For each option, REORG PLUS multiplies the number that you specify by 32 KB to determine the amount of virtual storage needed for the buffer pool. REORG PLUS then allocates as many buffers as will fit into that space.

REORG PLUS attempts to allocate all buffers above the 16-megabyte (MB) line.

SYSREC data set

REORG PLUS does not use the SYSREC data set when you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE for a single-phase table space reorganization. References to single-phase reorganizations in this section are for SHRLEVEL NONE only.

The SYSREC data set contains the unloaded rows from the table space that you are reorganizing. For a two-phase reorganization, REORG PLUS writes this data set during the UNLOAD phase and reads it during the RELOAD phase. For a single-phase reorganization, multiple SYSREC*nn* data sets (one per partition) are optional. When you specify them, REORG PLUS writes the data sets during unload processing in the REORG phase for restart purposes only. For detailed specification and allocation guidelines for the SYSREC data set, see “SYSREC data sets” on page 349. For information about having REORG PLUS dynamically allocate the SYSREC data set, see “Dynamic allocation options” on page 269.

For a partitioned table space, you can improve I/O performance by specifying one SYSREC*nn* 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 SYSREC*nn* data set in your JCL and not having REORG PLUS dynamically allocate one. Not having a SYSREC data set eliminates writing to the data set. However, not having the SYSREC data set also means that the job might not be restartable.

You can still achieve a performance gain when you specify a SYSREC data set or use dynamic data set allocation for a single-phase reorganization because the table space information is written to the SYSREC data set for restart purposes only. REORG PLUS does not reload the table space from this data set, thus eliminating reading from the SYSREC data set.

REORG PLUS uses the UBUFFS installation option to determine the amount of buffer pool storage that is allocated for reading and writing the SYSREC data sets. The multiplier that REORG PLUS applies to the UBUFFS option value is 32 KB. For the UBUFFS option, BMC recommends a value of 20 (the value that is shipped with REORG PLUS). REORG PLUS determines the optimal block size of the SYSREC data set, depending on the device type containing the data set.

SYSUT1 data set

For a table space reorganization, REORG PLUS requires one or more SYSUT1 data sets. For a single-phase index reorganization, REORG PLUS does not use the SYSUT1 data set when you specify SHRLEVEL REFERENCE or SHRLEVEL CHANGE. References to single-phase reorganizations in this section are for SHRLEVEL NONE only.

The SYSUT1 data set contains the information that REORG PLUS needs to build all indexes (when you specify ORDER NO) or only the non-data-sorting indexes (when you specify ORDER YES). For a two-phase reorganization, REORG PLUS writes this data set during the UNLOAD phase and reads it during the RELOAD phase. For a single-phase table space reorganization, REORG PLUS both writes and reads the data set during the REORG phase. For a single-phase index reorganization, the SYSUT1 data set is optional. When you specify it, REORG PLUS writes this data set during the REORG phase for restart purposes only. For specification and allocation guidelines for the SYSUT1 data set, see “SYSUT1 data sets” on page 353. For information about having REORG PLUS dynamically allocate the SYSUT1 data set, see “Dynamic allocation options” on page 269.

When more than one non-data-sorting index exists, you can improve I/O performance by using multiple SYSUT1 data sets. Using multiple data sets allows I/O operations to overlap.

For a single-phase index reorganization, you can improve I/O performance by not specifying a SYSUT1 data set in your JCL and not having REORG PLUS dynamically allocate one. Not having a SYSUT1 data set eliminates writing to the data set. However, not having the SYSUT1 data set also means that the job might not be restartable.

You can still achieve a performance gain when you specify a SYSUT1 data set or use dynamic data set allocation for a single-phase reorganization because REORG PLUS writes the index information to SYSUT1 for restart purposes only. REORG PLUS does not reload the index from this data set, thus eliminating reading from the SYSUT1 data set.

A single SYSUT1 data set for all indexes requires a record length long enough to hold information for the longest key. If shorter keys exist, REORG PLUS pads them so that they are as long as the longest key. With multiple SYSUT1 data sets, REORG PLUS writes information for each index to its own SYSUT1 data set, and does not pad the keys.

REORG PLUS uses the WBUFFS installation option to determine the amount of buffer pool storage that is allocated for reading and writing the SYSUT1 data sets. The multiplier that REORG PLUS applies to the WBUFFS option value is 32 KB. The WBUFFS option has two subparameter values:

- The first subparameter specifies the value to use when you use a single SYSUT1 data set.

- The second subparameter specifies the value to use for each data set when you use multiple SYSUT1 data sets.

For this option, BMC recommends a value of 20 for a single SYSUT1 data set and 10 for multiple SYSUT1 data sets.

REORG PLUS determines the optimal block size of the SYSUT1 data set, depending on the device type containing the data set.

Copy data sets

The copy data sets (shipped with the default names of BMCCPY, BMCCPZ, BMCRCY and BMCRCZ) contain identical copies of the reorganized table space or partitions of the table space after it is reorganized. For a two-phase table space reorganization, REORG PLUS writes these data sets during the RELOAD phase. For a single-phase table space reorganization, REORG PLUS writes these data sets during the REORG phase.

If the copies are registered as the DB2 local and remote copies, they can provide input to a DB2 recovery utility, such as the BMC Software RECOVER PLUS product or the IBM DB2 RECOVER utility. For guidelines about specifying and allocating the copy data sets, see “Copy data sets” on page 326.

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 BMCCPY 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 full copy data sets in a DD statement, you can have REORG PLUS dynamically allocate the data sets.

For additional performance considerations for copy data sets, see “Copy options” on page 535.

Incremental copy data sets

The incremental copy data sets (shipped with the default names of BMCICY, BMCICZ, BMCIRY, and BMCIRZ) contain identical copies of the pages that have changed since the full copies were made of the reorganized table space or partitions of the table space. When performing a SHRLEVEL CHANGE reorganization, you might need one of these data sets for each full copy data set that you specify in your JCL, depending on the value of the ICTYPE option.

REORG PLUS writes the incremental data sets during the LOGFINAL phase. The copies are registered as the DB2 local and remote copies and can provide input to a DB2 recovery utility, such as the BMC Software RECOVER PLUS product or the IBM DB2 RECOVER utility. For guidelines about specifying and allocating the incremental copy data sets, see “Incremental copy data sets” on page 605.

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 608.

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 545.

Copy options

Consider the following recommendations when making copies during the reorganization job:

- Specify the INLINECP installation option or the COPY YES INLINE YES command option to make inline image copies during the reorganization, particularly if you are reorganizing a partitioned object.

  The difference between inline image copies and standard image copies is that REORG PLUS creates inline copies *as* it reloads the table space, rather than *after* it reloads the table space. This process can reduce the elapsed time of the reorganization. For details about how to create inline image copies, see "Copy options for REORG TABLESPACE" on page 241.

- Specify a value of PART for the COPYLVL installation or command option if you are reorganizing a partitioned object.

  COPYLVL=PART facilitates multitasking by providing a unique full copy data set for each partition that you are reorganizing and ensures that each copy data set resides on DASD.

Dynamic allocation options

Although dynamically allocating each data set requires some processing time, enabling dynamic data set allocation can improve performance in many cases. If you are reorganizing very small quantities of data, the extra processing time might be a nontrivial percentage of the elapsed time. If you are reorganizing large quantities of data, especially in a partitioned data set, using dynamic data set allocation might improve performance; dynamic data set allocation always uses the optimal number of data sets, which enables REORG PLUS to use DASD more efficiently.

In general, BMC recommends that you dynamically allocate the following data sets with the DDTYPE options:

- SYSREC
- SYSUT1
- full copy (BMCCPY, BMCCPZ, BMCRCY, and BMCRCZ)
- incremental copy (BMCICY, BMCICZ, BMCIRY, and BMCIRZ)

For more information, see "Dynamic data set allocation" on page 88 and "Dynamic allocation options" on page 269.

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.

ORDER command option

The ORDER command option controls whether the unloaded rows are sorted. In determining how to optimize the performance of REORG PLUS, you must balance your need for fast execution and data availability with your need to have data sorted when it is reorganized. The following discussion can help you determine how to specify the ORDER command option to meet your specific performance needs.

ORDER YES

When you specify ORDER YES, REORG PLUS sorts the unloaded rows during unload processing. No data-sorting index work information is required in the SYSUT1 data set, thus reducing I/O operation and DASD space for that data set. Instead, REORG PLUS takes the index information from the data rows during reload processing.

ORDER NO

When you specify ORDER NO, REORG PLUS does not sort the data, thus decreasing processing time. When your tables have clustering or partitioning indexes, this option removes indirection and deleted table rows, and restores the PCTFREE and FREEPAGE values. This process avoids the overhead of sorting the unloaded rows if your tables have clustering or partitioning indexes. REORG PLUS writes the data-sorting index work information to the SYSUT1 data sets.

Sort processing options

The BMC BMCSORT technology provides REORG PLUS with more control of the sort process than external sort routines provide. This added control helps prevent memory-related problems during the sort process. REORG PLUS allocates the amount of resources to each sort process based on the amount of work that REORG PLUS determines the sort process will perform. Depending on the table space that you are reorganizing and on its index characteristics, REORG PLUS invokes BMCSORT one or more times.

REORG PLUS does not require sort work data sets when the following conditions exist. 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 the following conditions exist 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 539

- value that you specify for REGION in either your JCL or system exits

- amount of memory that is available during optimization

- percentage of available or total 4-KB system pages that you specify with the AVAILPAGEPCT and TOTALPAGEPCT options

NOTE

The SHORTMEMORY installation or command option controls the action that REORG PLUS takes when a memory shortage exists. For more information about SHORTMEMORY, see page 181 or page 670.

- minimum and maximum amounts of memory per sort that you specify with the following options:

  — MINSORTMEMORY installation (page 661) or command option (page 183)

  — MAXSORTMEMORY installation (page 659) or command option (page 183)

- values that you specify for the multitasking installation options described on page 540

Total memory

The first parameter value of the SMCORE option tells REORG PLUS how much total memory, both above and below the 16-MB line, that you want BMCSORT to use during a single invocation. BMC strongly recommends that you specify a value of 0K. However, other valid values are 4096K through 65536K. You can also specify the SMCORE value in megabytes (0M or 4M through 64M).

The region size available for your reorganization job in conjunction with the value that you specify for total memory can constrain the number of sort processes that REORG PLUS starts. Because the region size must include space for buffers and other required structures, the entire region size is not available for sort processing. You can avoid this constraint by using values of 0K and 0K, allowing REORG PLUS to determine the optimal amount of total memory to use.

Additional considerations

The following additional considerations apply when you use the sort optimization options and also specify a value greater than 0 for the SMCORE total memory parameter:

- REORG PLUS uses the SMCORE value that you specify, regardless of other memory factors, including values specified for the MINSORTMEMORY installation (page 661) or command option (page 183) and MAXSORTMEMORY installation (page 659) or command option (page 183).

  That is, if you specify an SMCORE total memory value that is lower than the MINSORTMEMORY value or greater than the MAXSORTMEMORY value, REORG PLUS still uses your specified value.

- If you specify a value greater than 0 for the total memory parameter, REORG PLUS honors your specified total memory value.

Below-the-line memory

The second parameter value of the SMCORE option indicates how much memory BMCSORT should use below the 16-MB line during a single invocation. Unless otherwise noted, BMC recommends that you specify a value of 0K, which allows REORG PLUS to determine the optimal amount of below-the-line memory to use. In addition to 0K, valid values are 256K through 4096K. You can also specify this value in megabytes (either 0M or 1M through 4M). If you specify a value of 0 for the below-the-line memory parameter, REORG PLUS computes below-the-line memory.

When you specify a value of 0 for the below-the-line memory parameter, the amount of memory that BMCSORT uses is controlled by the value of the SORTNUM option (page 174 or page 673), as follows:

SORTNUM value	Minimum amount of memory used for all sorts
32 or less	256 KB
33-99	384 KB
100-255	1024 KB

When you specify a value greater than 0 for the below-the-line memory parameter, REORG PLUS honors the other sort optimization options (which apply to total memory only) and also uses your specified SMCORE value.

The following example illustrate the use of SMCORE and SORTNUM to control the amount of memory that REORG PLUS uses for sort processing:

You specify SMCORE=(0K,700K) and SORTNUM=32.

REORG PLUS uses all of the sort optimization installation and command options to calculate the total memory that is required for sort processing. REORG PLUS uses 700 KB of memory below the line.

Controlling the number of sort processes

REORG PLUS determines the optimal number of sort processes that can execute concurrently, depending on available resources. Under normal circumstances, you should allow REORG PLUS to control the number of sorts processed concurrently.

You can specify the maximum number of concurrent sort processes by using the SMAX installation option. BMC recommends that you use the default value of 0 for the SMAX installation option.

You can also specify the MAXSORTS option of the REORG command to set the maximum number of concurrent sort processes. For more information, see the description of “MAXSORTS” on page 175 and Table 89 on page 542, which shows MAXSORTS combined with the other sort options.

If the value of the MAXSORTS command option or SMAX installation option is nonzero, REORG PLUS starts at most one task per CPU. You can improve performance by starting more than one task per CPU. To do so, set SMAX=0 (if you specify MAXSORTS in the command, use 0 for the MAXSORTS value), and use the multitasking installation options (TASKMAX, RORGMAX, UNLDMAX, and BILDMAX). For information about using these options, see “Multitasking installation options.” For information about specifying the installation options, see Appendix A, “REORG PLUS installation options.”

UXSTATE installation option

By default, during processing REORG PLUS invokes EDITPROC DB2 user exits in supervisor state and program status word (PSW) key=7. You can specify UXSTATE=PROB to tell REORG PLUS to call EDITPROCs in problem state and PSW key=7. The requirements of the exits dictate the UXSTATE setting. Check with the exit author (or vendor) before changing the value of UXSTATE to PROB.

Enabling multitasking for performance

During execution, REORG PLUS determines the most effective arrangement of tasks when running in a multiprocessor environment. Although REORG PLUS runs very efficiently on single-processor computers, it performs most efficiently on large multiprocessor systems.

REORG PLUS considers the following factors when it determines the maximum number of tasks to start:

- available memory above and below the 16-MB line
- number of SYSREC and SYSUT1 data sets that you specify
- governing limitations that you specify through the multitasking options
- number and size of the preallocated sort work data sets
- number of DB2 objects on which REORG PLUS must operate for each phase or process

Multitasking installation options

Table 86 on page 541 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 86 Multitasking options

Multitasking option	Phase or process affected
TASKMAX	global
ANALMAX	ANALYZE phase
RORGMAX[a]	REORG phase
UNLDMAX[a]	UNLOAD phase
BILDMAX[a]	index build process
COPYMAX	image copy process
RENMMAX	rename and delete process for SHRLEVEL REFERENCE and SHRLEVEL CHANGE
SCPYMAX	nonpartitioned index copy process for SHRLEVEL REFERENCE and SHRLEVEL CHANGE partial reorganizations

[a] To enable this option, you must specify SMAX=0, and either omit the MAXSORTS command option or specify MAXSORTS 0.

The TASKMAX option sets the default for all of the other multitasking options in the installation options module. TASKMAX does not directly provide the number of tasks to use for any phase or process. Instead, TASKMAX provides a default for any option that refers to it. Table 87 lists the valid values for TASKMAX.

Table 87 Values for the global multitasking option

TASKMAX value	Meaning
0	REORG PLUS determines the number of tasks to start.
n	The maximum number of tasks that REORG PLUS can start is *n* (where *n* is a positive integer from 1 through 32767).
n%	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).

Table 88 lists the valid values for all of the other multitasking options.

Table 88 Values for the individual multitasking options

Option value	Meaning
0	REORG PLUS automatically determines the number of tasks to start.
*	REORG PLUS uses the TASKMAX value.
n	The maximum number of tasks that REORG PLUS can start is *n* (where *n* is a positive integer from 1 through 32767).
n%	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 example, assume that you have three CPUs and you specify TASKMAX=5, ANALMAX=*, and RORGMAX=*. REORG PLUS starts up to five tasks for the ANALYZE phase and five for the REORG phase. If you change the TASKMAX value to 500%, REORG PLUS can start as many as 15 tasks for each of those phases. The more tasks that REORG PLUS can start, the better performance it will achieve.

RORGMAX, UNLDMAX, and BILDMAX are enabled only when all of the following conditions exist:

- The value of the SMAX installation option is 0.
- You have not specified the MAXSORTS command option or its value is 0.

Table 89 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 543.

Table 89 Sort task options hierarchy

When SMAX value	And MAXSORTS value	REORG PLUS uses up to
16	5	5 (MAXSORTS overrides SMAX)
16	0	16 (SMAX value)
0	16	16 (MAXSORTS value)
0	0	RORGMAX, UNLDMAX, and BILDMAX values

REORG PLUS always uses the ANALMAX, COPYMAX, SCPYMAX, and RENMMAX multitasking options. For information about determining the values for ANALMAX, COPYMAX, and SCPYMAX, see “Multitasking I/O bound processes.” For a description of the installation options, see Appendix A, “REORG PLUS installation options.”

Multitasking I/O bound processes

The following multitasked processes are I/O bound:

- ANALYZE phase
- table space copy process
- 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 542, the following processes are also typically I/O bound:

- single-phase REORG phase
- two-phase UNLOAD phase
- the index build process

You can control the multitasking for these processes with the RORGMAX, UNLDMAX, and BILDMAX installation options, respectively. However, for certain architectural considerations, REORG PLUS starts a maximum of 16 tasks for the REORG phase, UNLOAD phase even if you specify a value greater than 16 on the RORGMAX, UNLDMAX, or BILDMAX option.

NOTE

For REORG PLUS to honor the RORGMAX, UNLDMAX, and BILDMAX options, you must specify SMAX=0 in the installation options module, and either not specify the MAXSORTS command option or specify MAXSORTS 0.

To achieve the most flexibility and highest level of multitasking, observe the following guidelines:

- Do not include any SORTWK*nn* DD statements in your JCL.
- Specify the following:

  —DDTYPE SORTWORK ACTIVE NO on your REORG command, or disable dynamic allocation for the SORTWK*nn* DDs in your options module

  —the SORTDEVT installation or command option

  —one SYSREC*nn* DD per partition

  —one SYSUT1*nn* DD per index

  —SORTNUM 32

For more information about setting the environment for BMCSORT, see "Sort processing options" on page 536.

Using multiple SYSREC data sets

When you are reorganizing a partitioned table space, you can improve performance by specifying one SYSREC*nn* 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 SYSREC*nn* data sets, see page 531.

Using multiple SYSUT1 data sets

Using multiple SYSUT1 data sets provides the following performance advantages:

- I/O processing to each SYSUT1 data set is overlapped with other I/O processing and with CPU processing.

- REORG PLUS writes any non-unique index information when the index process receives it and does not pass the information to BMCSORT, thus reducing the amount of data that is sorted. This process also reduces the amount of DASD space that is required for all index information.

- With a single SYSUT1 data set, REORG PLUS pads all keys to the length of the longest key being processed. Padded keys require more DASD space, and more I/O operations are required to process the index information. With multiple SYSUT1 data sets, less padding of keys is needed.

- REORG PLUS attempts to build indexes concurrently. REORG PLUS determines resource utilization in the same way as for the UNLOAD phase (see "Resource allocation in the UNLOAD phase" on page 558) 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 SYSUT1*nn*, where *nn* is a unique suffix for each DD statement. The suffix is not used to identify which index is assigned to a specific SYSUT1 data set.

Additional performance tuning recommendations

The following sections describe additional steps that you can take to improve the performance of your reorganization jobs.

Tuning I/O

The single most important factor affecting performance in REORG PLUS is I/O processing. During typical reorganization processing, REORG PLUS reads and writes large amounts of data. To maximize I/O performance, REORG PLUS handles all of its own buffering and performs I/O operations at the lowest level possible. Doing so allows REORG PLUS to read or write several blocks of data with each I/O operation and permits REORG PLUS to prefetch subsequent data.

To avoid I/O queueing, allocate REORG PLUS data sets on separate channels and drives. If you do not have sufficient channels available, use separate drives and control units.

Because REORG PLUS I/O processing is primarily sequential, DASD caching provides no benefit. Avoid DASD caching because the overhead might slightly increase I/O processing time.

For more information that can help you tune your I/O processing, see “Buffer installation options” on page 531 and “Sort processing options” on page 536.

Providing maximum virtual storage

Because each task requires virtual storage for processing, REORG PLUS balances the multiprocessing performed with the amount of virtual storage that is available. The primary use of virtual storage is for I/O buffers by either REORG PLUS itself or by BMCSORT.

REORG PLUS uses as much virtual storage as needed for each task. For the best performance, BMC recommends that you specify REGION=0M in the JOB or EXEC statement of the execution JCL. If your data center does not permit you to specify REGION=0M, specify the amount that allows the most virtual storage, both above and below the 16-MB line.

Using a region size that is less than optimal risks the following potential issues:

- running less efficiently, which could result in additional CPU and elapsed time
- encountering memory failures or jobs that fail when new versions implement changes that require additional memory

If you specify a value for REGION other than 0M, ensure that you have an appropriate value set for the MEMLIMIT parameter, either as your site's default SMF option or on your JOB statement or EXEC statement. BMC makes the following recommendations for the MEMLIMIT option:

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

Selectively redefining VSAM data sets

You can use the DSRSEXIT user exit to selectively redefine each object. By not performing the VSAM DELETE / DEFINE process for an object, you can significantly reduce elapsed and CPU time. For information about the DSRSEXIT user exit, see "Using DSRSEXIT to manage VSAM data set redefinition" on page 780.

Additional performance information for SHRLEVEL CHANGE options

If INLINECP is not already specified in the installation options, BMC recommends that you specify COPY YES INLINE YES. If you do not want to make inline copies, specify the ICTYPE AUTO option. When you specify ICTYPE AUTO, REORG PLUS determines whether it can update the full copy data sets or will need to create an incremental image copy.

Specifying AUTO requires that you provide incremental copy data sets if REORG PLUS determines that it needs to create an incremental image copy. BMC recommends that you also specify DDTYPE LOCPICPY ACTIVE YES on the command to enable REORG PLUS to dynamically allocate the incremental copy data sets when they are needed.

Performance tuning for specific scenarios (any SHRLEVEL)

In addition to the general recommendations for REORG PLUS, use the following guidelines based on the needs of your site. The guidelines describe steps that you can take to tune specific types of reorganization jobs to improve performance.

Multiple indexes

Unless you are reorganizing an object that has many indexes but not much data, allocate one SYSUT1 data set per index. If you are reorganizing an object that has many indexes but not much data (such as an SAP object with hundreds of indexes), BMC recommends that you allocate a single SYSUT1 data set instead of one per index.

Many tables and indexes

Specify REDEFINE NO if both of the following conditions exist:

- You are reorganizing an object that has many tables and indexes.
- The purpose of the reorganization is solely to reorganize the data, not to redefine the data sets.

High availability

Specify the appropriate SHRLEVEL option for the performance goal. Note the following considerations when running SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganizations:

- If you are running in a non-RACF® site and you establish authority at a node lower than the highest node, see Chapter 2, “Operational considerations,” for information about setting authorizations for the data sets that are involved in the renaming process.

- Specify a value of DSN for the STAGEDSN installation option to avoid redundant data set rules.

Performance tuning for specific scenarios (SHRLEVEL CHANGE only)

The following sections describe steps that you can take to tune specific types of SHRLEVEL CHANGE reorganization jobs to improve performance.

Reorganization must complete

In an environment in which the reorganization must complete and you have transactions that will retry after a timeout, consider specifying the following values for the installation options:

- DRNWAIT=UTIL
- DRNRETRY=10
- DRNDELAY=3
- DSPLOCKS=RETRY

Transactions and reorganization are of equal priority

In an environment in which you want REORG PLUS to work like any other transaction, consider specifying the following values for the installation options:

- DRNWAIT=SQL
- DRNRETRY=15
- DRNDELAY=3
- DSPLOCKS=RETRY

Interpreting performance-related messages

REORG PLUS issues performance-related messages if you specify MSGLEVEL(1) on your EXEC statement or in your installation options. For more information about specifying this utility parameter, see “Utility parameters on the EXEC statement” on page 317.

Use the information provided in these messages to monitor REORG PLUS performance and to fine-tune future runs. For explanations and user responses, see the *Utility Products for DB2 Messages Manual*.

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 (SYSUT1*nn*).

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 =** ***n*****K, ABOVE 16M =** ***n*****K, 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 545 and messages BMC50475I and BMC50479I.

BMC50476I **DDNAME =** ***ddname*****, I/OS =** ***n*****, I/O WAITS =** ***w*****, RDB LOCK WAITS =** ***r***

This message displays the following performance information about I/O operations to sequential data sets:

- ddname associated with the I/O operations
- number of I/Os (blocks) written to or read from the data set
- number of waits issued for I/O completion
- number of waits for serialization of the data set

A wait count that is greater than 10 percent of the block count might indicate degraded performance.

Try allowing REORG PLUS additional buffer space for the associated data set (see “Buffer installation options” on page 531) or using multiple data sets. A high number of serialization waits might indicate the need to decrease the number of concurrent tasks or use multiple data sets. Also, with a single-phase reorganization, you have the option of not using SYSREC (for a table space reorganization) or SYSUT1 (for an index reorganization) data sets.

BMC50477I ***taskNumber*: PARTITION = *partitionNumber*, ROWS/KEYS = *n*, I/O WAITS = *w*, DDNAME = *ddname***

This message displays the following performance information about I/O operations to VSAM data sets:

- processing task number
- table space partition number
- number of rows or keys in the partition
- number of waits issued for I/O completion
- ddname associated with the I/O operations

Use the ddname to find the actual number of I/Os that were issued to the associated data set. A wait count that is greater than 20 percent of the actual I/Os might indicate degraded performance. Try allowing REORG PLUS more virtual storage if REORG PLUS also issues message BMC50397I. Provide more sort work file space if REORG PLUS also issues message BMC50398I. Changing these values enables REORG PLUS to start more concurrent tasks.

BMC50479I **TOTAL PAGES: *availablePages*, ALLOWED: *allowedAvailablePages*; AVAILABLE PAGES: *freePages*, ALLOWED: *allowedFreePages***

This message displays information that REORG PLUS obtains from the operating system (not the region). The *allowedAvailablePages* value is the value that you specified for the TOTALPAGEPCT option, and *allowedFreePages* is the value that you specified for the AVAILPAGEPCT option.

REORG PLUS considers the percentage of available and free pages when determining the maximum amount of memory that is allowed for sort processing. If insufficient memory exists to perform a minimum number of optimal sort processes, the job terminates with a constrained memory message.

No action is required. However, if you experience frequent system memory shortages, if possible, schedule your REORG PLUS jobs during periods of lower system activity.

BMC50486I ***taskNumber*: BMCSORT STARTED, *n*K BELOW 16M, *n*K TOTAL MEMORY, *n* PAGES HYPERSPACE**

This message displays the maximum amount of memory and 4-KB pages of hyperspace that the utility allows for each sort task. No action is required.

BMC50720I ***taskNumber*: *n* NEW LOG APPLY BUFFERS, *n* REUSED, *n* INITIAL MAXIMUM, *n* WAITS, BUFFER SIZE=*bufferSize***

This message displays buffer information for the log apply process. The initial maximum value is the number of buffers that the utility determined that it needs for each log apply task. No action is required. However, the information displayed in this message can help you tune performance of the utility.

BMC51302I **MAX TASKS = *t*, MAX PARTITIONS PER TASK = *p*, SORTWKS PER TASK = *s*, MAX OPEN PARTITIONS PER TASK = *o***

This message displays the following results of task optimization for unload processing:

- maximum number of concurrent tasks
- maximum number of partitions that can be processed per task
- number of sort work files assigned to each task
- maximum number of open partitions per task

The maximum number of tasks that REORG PLUS runs depends on the available system resources, such as

- virtual storage
- number of physical CPUs
- number of unload index work and sort work files
- sort work file space

REORG PLUS might be able to run more tasks concurrently if you increase the amount of virtual storage and allocate more sort work files.

Based on the size of the partition and available sort work file space, REORG PLUS determines the number of partitions to process per task in order to allow the sort to run in the fastest manner. The actual number of partitions processed per task might be less than the maximum stated in the message.

BMC51303I **MAX TASKS = *t*, INDEXES PER TASK = *i*, SORTWKS PER TASK = *s*, MAX OPEN PARTITIONS PER TASK = *o***

This message displays the following results of index build task optimization:

- maximum number of concurrent tasks
- number of indexes per task
- number of sort work files assigned to each task
- maximum number of open index partitions per task

The maximum number of tasks depends on the available system resources, such as

- virtual storage
- number of physical CPUs
- number of unload index work and sort work files
- sort work file space

REORG PLUS might be able to run more tasks concurrently if you increase the amount of virtual storage and allocate more sort work files.

Based on the number of indexes, work data sets, and sort work file space, REORG PLUS determines the number of indexes to process per task in order to allow the sort to run in the fastest manner.

Tuning to improve memory use

To help you determine how to improve memory use, consider the information in the following sections.

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.

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.

Multitasking

Multitasking can improve memory use in your reorganization jobs. For information, see “Enabling multitasking for performance” on page 540.

Making copies

The number of copies that you make during a reorganization can affect the memory that your system uses. Balance your site's backup copy needs with your system's memory use requirements to determine the optimal number of copy data sets to allocate.

Memory constraints and data set allocation constraints are particularly critical when you are making copies while reorganizing a large number of partitions. Consider one of the following options in this case:

- Specify COPYLVL FULL to have REORG PLUS allocate a single copy data set for the table space.

- If you are reorganizing a subset of contiguous partitions, create a single copy by specifying COPYSUBSET=YES in the installation options module and COPYLVL FULL on your REORG command.

- If you need copies by partition, specify as few copies per partition as possible or consider specifying fewer partitions per job.

Tuning for availability

This section describes recommendations for the different availability requirements of your REORG PLUS jobs.

Specifying SHRLEVEL

Specify the appropriate SHRLEVEL option for your availability requirements. Use Table 90 to help determine the appropriate option.

Table 90 REORG PLUS availability options

Access to data needed during the reorganization process	Additional site or application requirements	SHRLEVEL option
no access	not applicable	SHRLEVEL NONE
read-only access	limited outage required	SHRLEVEL REFERENCE
read / write access	limited or no outage required	SHRLEVEL CHANGE

Granting data set authority

For the data sets that REORG PLUS uses as staging data sets, ensure that UPDATE and CONTROL authorization is established when all of the following conditions exist:

- You are using SHRLEVEL CHANGE or SHRLEVEL REFERENCE.
- You are running REORG PLUS in a non-RACF environment.
- You establish authority at a node lower than the highest node.

For more information, see "Data set authorization" on page 67.

REORG PLUS processing phases

This section describes the processing phases of REORG PLUS in detail. This information can be useful when tuning your reorganization jobs.

NOTE

For a DSNUTILB reorganization, REORG PLUS passes processing to DSNUTILB after the UTILINIT phase and regains control during the UTILTERM phase. Therefore, the information in this section does not apply to a DSNUTILB reorganization.

REORG PLUS exploits the technology provided by large-scale processors. During execution, REORG PLUS examines available resources and uses as much of these resources as possible to maximize performance.

REORG PLUS architecture

The architecture of REORG PLUS differs from that of the IBM DB2 REORG utility. The DB2 REORG utility performs reorganization processing in several phases (UNLOAD, RELOAD, SORT, and BUILD), which run serially. However, REORG PLUS combines these phases into either a two-phase architecture or a single-phase architecture.

Two-phase architecture

When you specify UNLOAD CONTINUE or UNLOAD PAUSE, REORG PLUS uses two phases: UNLOAD and RELOAD. When you specify UNLOAD RELOAD, REORG PLUS combines the UNLOAD and RELOAD phases into a single processing phase named REORG. All functions of the DB2 REORG utility are performed in these phases of REORG PLUS.

The UNLOAD installation option determines the phase processing that REORG PLUS uses at a global level. You can override this value for a particular reorganization job by specifying the UNLOAD command option. You can only specify UNLOAD PAUSE on the command option.

The two-phase architecture allows REORG PLUS to perform several tasks concurrently, reducing the elapsed time for a table space reorganization. For example, REORG PLUS sorts the index keys and builds the indexes while it reloads the table space rows.

Single-phase architecture

The REORG PLUS single-phase architecture builds on the advantages of the two-phase processing and allows for even greater reductions in CPU usage and elapsed time. Single-phase processing eliminates some read and write processes by combining the UNLOAD and RELOAD phases.

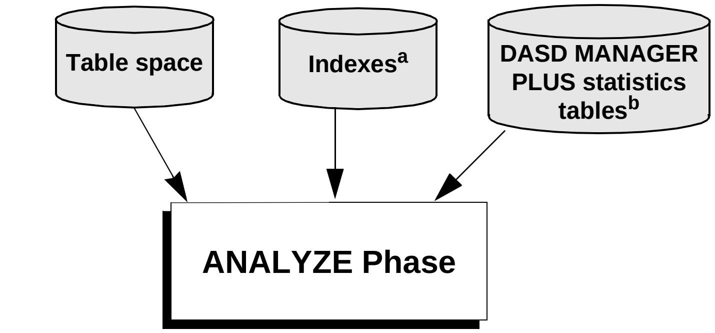
ANALYZE phase

The ANALYZE phase provides information about the number of rows (cardinality) and the average row size of the data to be reorganized. The following performance factors arise from the ANALYZE phase, and command options that you specify control these factors:

- amount of time that the phase requires to run
- accuracy of the information gathered during the phase

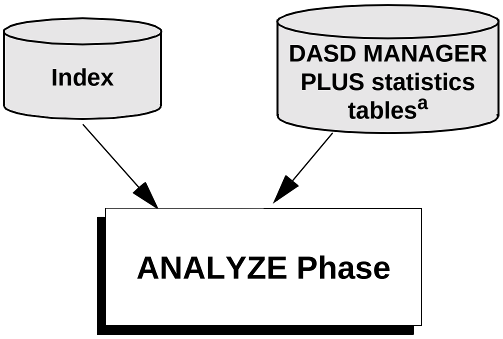
Figure 50 on page 557 and Figure 51 on page 557 show the objects that the ANALYZE phase might use (except ANALYZE HURBA, which uses no objects).

Figure 50 ANALYZE phase for a table space reorganization



[a] Only if you are using the SCAN option.
[b] Only if you are using REORG PLUS as part of the Database Performance solution.

Figure 51 ANALYZE phase for an index reorganization



[a] Only if you are using REORG PLUS as part of the Database Performance solution.

UNLOAD phase

During the UNLOAD phase, REORG PLUS prepares data to allow concurrent processing for reloading table spaces and indexes. The UNLOAD phase prepares the unloaded rows for reloading into the specified tables by the RELOAD phase.

Depending on the data and the characteristics of the tables, REORG PLUS performs one or more of the following tasks during the UNLOAD phase:

- reads the rows from the DB2 tables or indexes
- sorts the rows (except in the instances noted in “SORTWK data sets” on page 338)

- 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 573 in Chapter 7, "Online reorganization."

Resource allocation in the UNLOAD phase

REORG PLUS checks the amount of available virtual storage within the region and the number of processors in the CPU to determine how much data can be unloaded concurrently. If you specify a nonzero value for SMAX or MAXSORTS, REORG PLUS uses the *lowest* of the following values to determine the maximum number of tasks that can run concurrently:

- 16
- SMAX value specified in the installation options
- MAXSORTS value specified in the REORG command statement
- number of processors

If you specify 0 for SMAX and MAXSORTS, REORG PLUS uses the *lower* of the following values to determine the maximum number of tasks that can run concurrently:

- 16
- value of the multitasking options RORGMAX, UNLDMAX, or BILDMAX

In addition to using the various command and installation options, REORG PLUS considers the following factors when calculating the maximum number of tasks that can run concurrently:

- number of partitions in the table space and index space
- number of sort work data sets divided by two

After determining the maximum number of tasks, REORG PLUS calculates the number of sort work files per task by dividing the number of sort work files specified in the JCL by the number of tasks. REORG PLUS then calculates the amount of sort work space per task by multiplying the number of sort work files per task by the size of the sort work files.

Figure 52 and Figure 53 display the objects that REORG PLUS might use in the UNLOAD phase.

Figure 52 UNLOAD phase for a table space reorganization

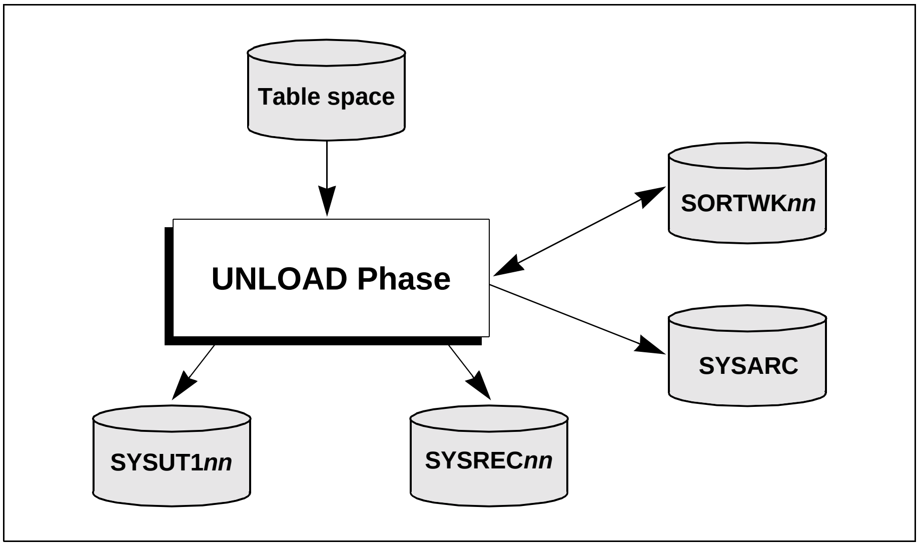
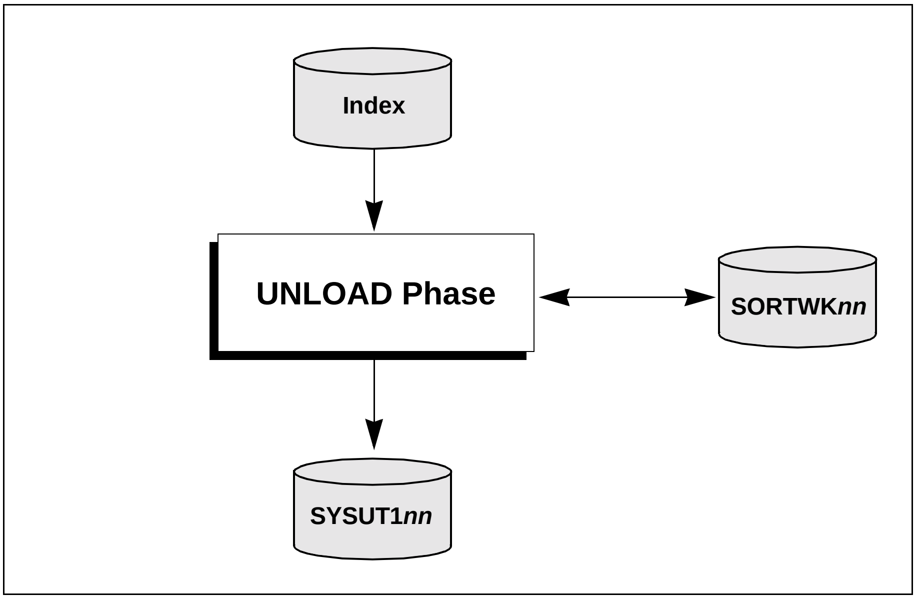


Figure 53 UNLOAD phase for an index reorganization



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 574 in Chapter 7, “Online reorganization.”

Resource allocation in the RELOAD phase

REORG PLUS checks the amount of available virtual storage within the region to determine how many indexes it can build concurrently. If you specify a nonzero value for SMAX or MAXSORTS, REORG PLUS uses the *lowest* of the following values to determine the maximum number of tasks that can run concurrently:

- SMAX value specified in the installation options
- MAXSORTS value specified in the REORG command statement
- number of processors

If you specify 0 for SMAX and MAXSORTS, REORG PLUS uses the value of the multitasking options RORGMAX, UNLDMAX, or BILDMAX to determine the maximum number of tasks that can run concurrently.

In addition to using the various command and installation options, REORG PLUS considers the following factors when calculating the maximum number of tasks that can run concurrently:

- number of indexes to be built
- number of SYSUT1 data sets
- number of sort work data sets divided by two

After REORG PLUS determines the maximum number of tasks that can run, it uses a recursive algorithm to balance the available virtual storage and sort work files. (Sort work files are not a consideration if you do not specify them in your JCL *and* you use the SORTNUM installation or command option.) As long as enough virtual storage space and sort work file space is available, processing continues. If either resource appears to be constrained, REORG PLUS adjusts downward the number of indexes built concurrently and repeats the checks.

If the maximum number of indexes to be built decreases to less than one, the reorganization terminates because of constrained resources. This balancing algorithm allows optimized reorganization when enough resources are available. It also allows processing to continue in a nonoptimized manner if only minimal resources are available.

Figure 54 on page 562 through Figure 57 on page 563 show the objects that REORG PLUS might use in the RELOAD phase.

Figure 54 RELOAD phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY table space reorganization

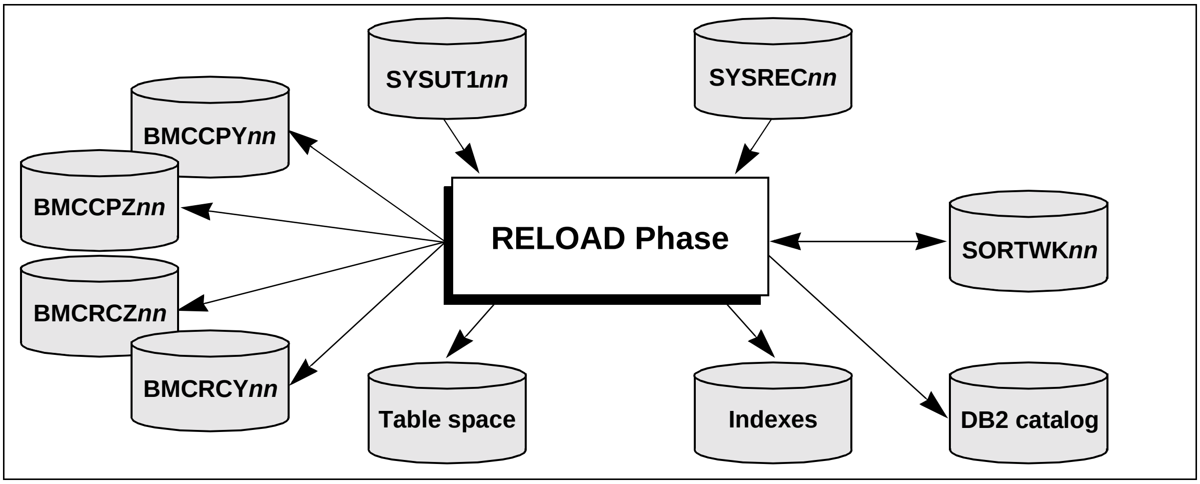
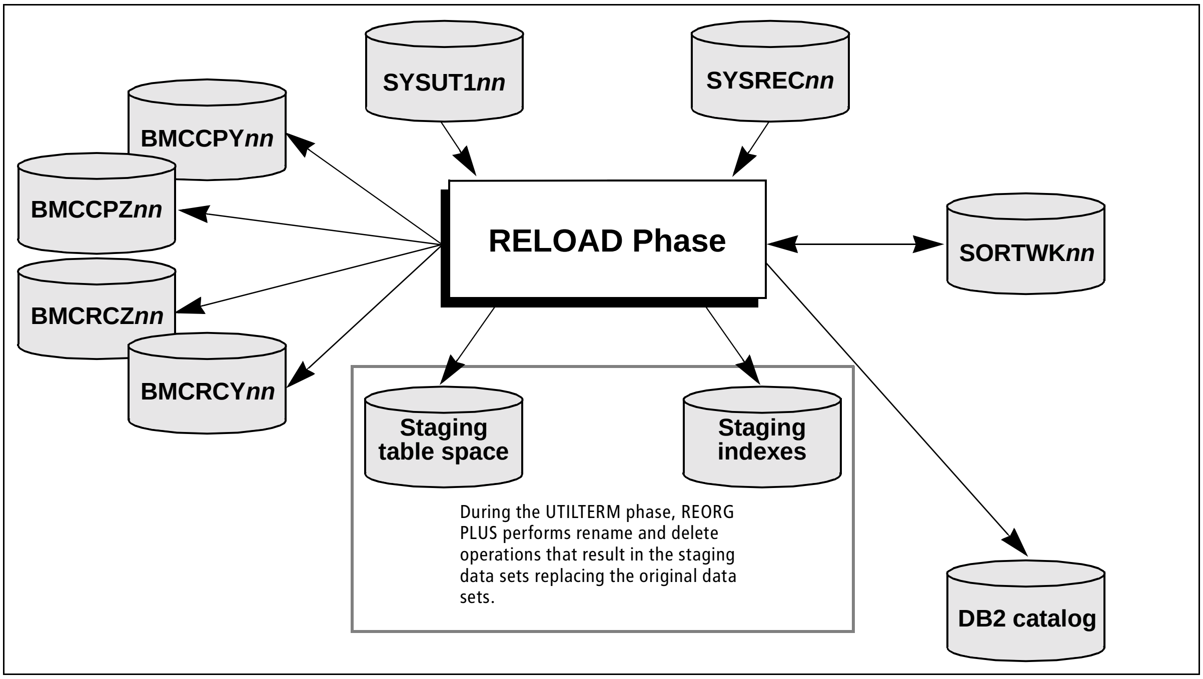


Figure 55 RELOAD phase: SHRLEVEL REFERENCE table space reorganization



During the UTILTERM phase, REORG PLUS performs rename and delete operations that result in the staging data sets replacing the original data sets.

Figure 56 RELOAD phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY index reorganization

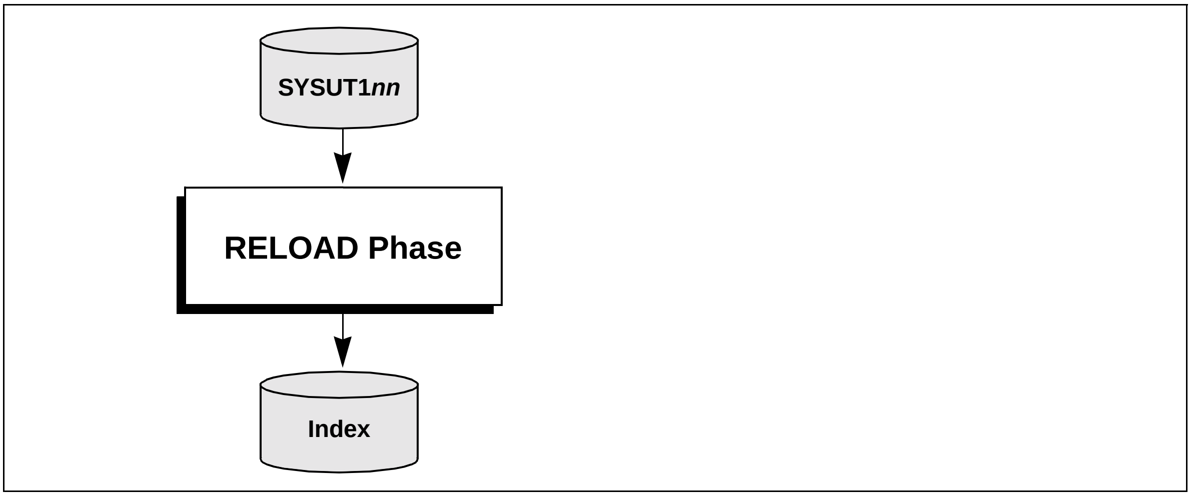
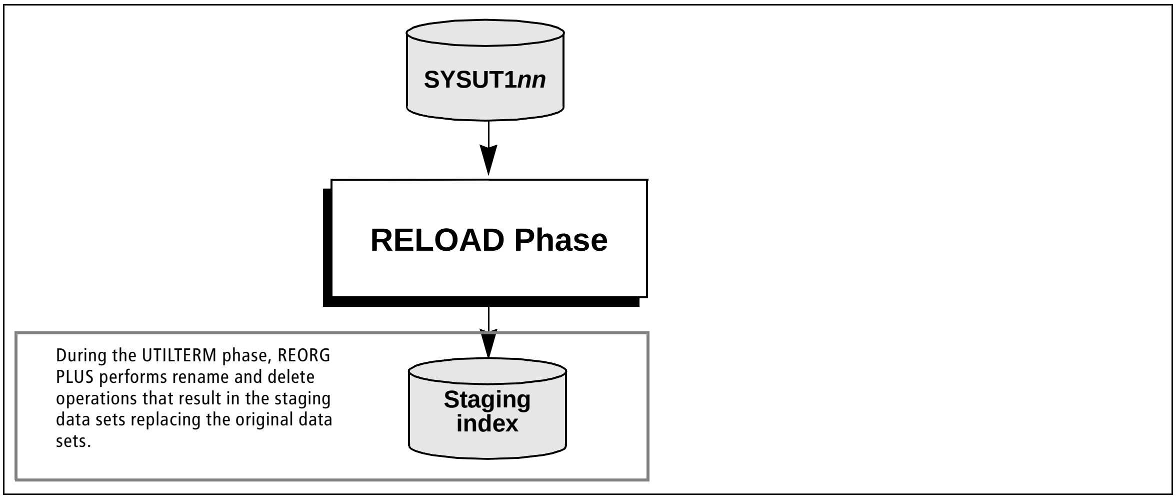


Figure 57 RELOAD phase: SHRLEVEL REFERENCE index reorganization



During the UTILTERM phase, REORG PLUS performs rename and delete operations that result in the staging data sets replacing the original data sets.

REORG phase for a single-phase reorganization

During the REORG phase, REORG PLUS performs almost all of the same processes as in the UNLOAD and RELOAD phases. Depending on your command specifications, REORG PLUS performs one or more of the following tasks:

- reads the rows from the DB2 tables or indexes
- sorts the rows (except in the instances noted in “SORTWK data sets” on page 338)
- writes the row images to the SYSREC data sets
- writes the index work records to the SYSUT1 data sets
- writes discarded rows from SELECT or DELETE processing to the SYSARC data set (if specified)
- builds or keeps the compression dictionary and compresses the rows
- if required, adds partitions to a partition-by-growth table space (SHRLEVEL NONE and SHRLEVEL REFERENCE only)
- redefines the VSAM data sets (including the staging data sets for SHRLEVEL REFERENCE or SHRLEVEL CHANGE) when the value of the REDEFINE command or installation option is YES
- checks for duplicate key values in a unique index
- concurrently loads multiple partitions
- concurrently loads the table space and builds data-sorting indexes
- concurrently sorts and builds non-data-sorting indexes
- concurrently builds multiple, non-data-sorting indexes
- collects statistics while loading table spaces and building indexes
- produces image copies either while reloading data or after reloading rows

If you have a large number of nonpartitioned indexes, consider using a two-phase reorganization to have REORG PLUS concurrently reload the table space and build the nonpartitioned indexes. Otherwise, a single-phase reorganization provides significant CPU and elapsed time savings over a two-phase reorganization because REORG PLUS does not have to read from the SYSREC data set (for a table space reorganization) or the SYSUT1 data set (for an index reorganization).

For a single-phase reorganization, the considerations in Table 91 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 91 Single-phase reorganization considerations for SYSREC or SYSUT1 data set

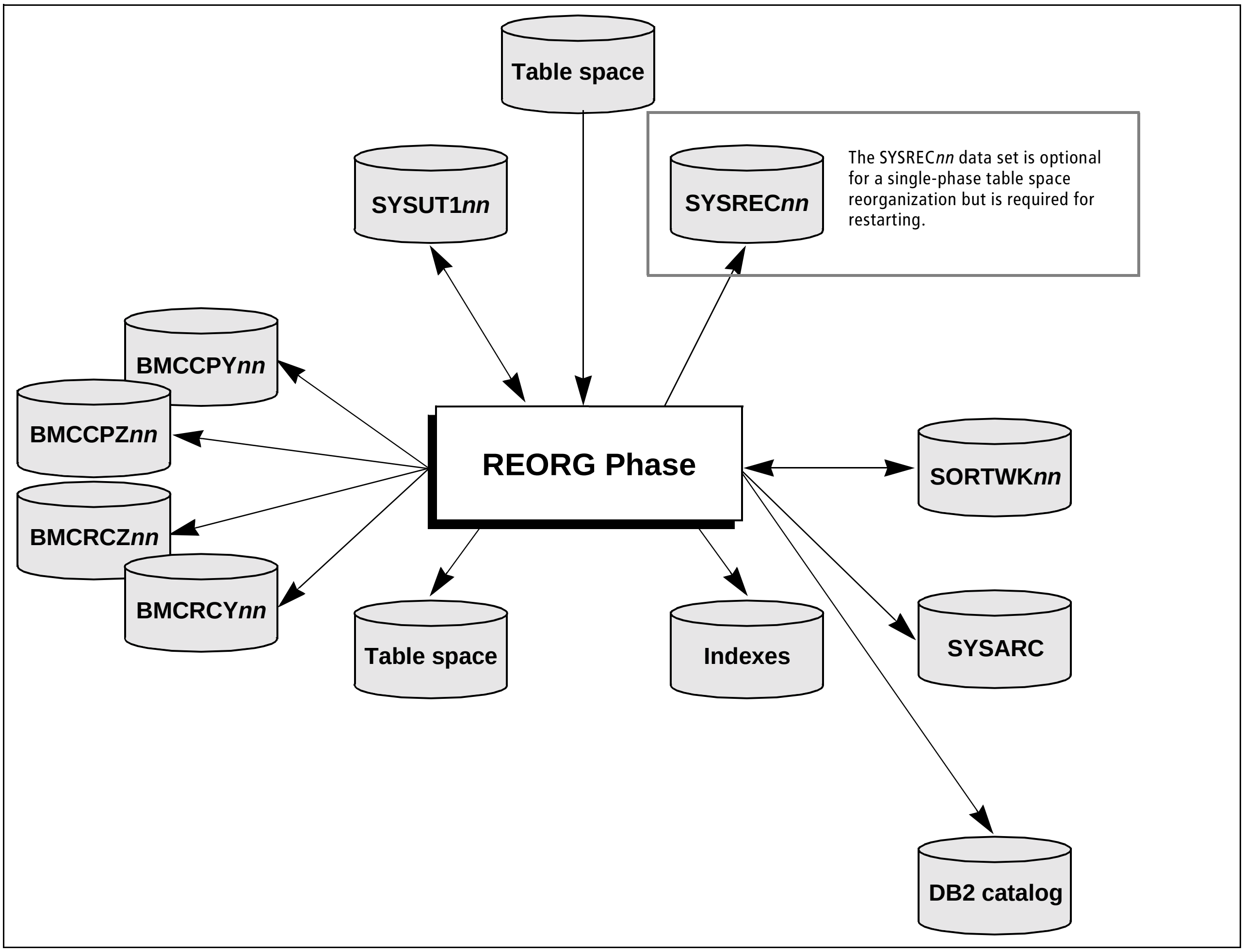
Specified SHRLEVEL	Type of reorganization		Effect on restartability[a]
	Table space	Index	
SHRLEVEL NONE (the default)	You can omit SYSREC data set for an additional performance gain.	You can omit SYSUT1 data set for an additional performance gain.	If you omit the data set, your job might not be restartable.
SHRLEVEL REFERENCE	REORG PLUS does not use SYSREC data set, even if you specify it.	REORG PLUS does not use SYSUT1 data set, even if you specify it.	The job is restartable due to the nondestructive nature of this type of reorganization.
SHRLEVEL CHANGE	REORG PLUS does not use SYSREC data set, even if you specify it.	REORG PLUS does not use SYSUT1 data set, even if you specify it.	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.

[a] For more information about restarting the job during a single-phase reorganization, see “Restarting REORG PLUS” on page 361.

Except for the option of omitting the SYSREC or SYSUT1 data set, all of the performance considerations for the REORG phase are the same as those in the UNLOAD phase (page 557) and the RELOAD phase (page 560).

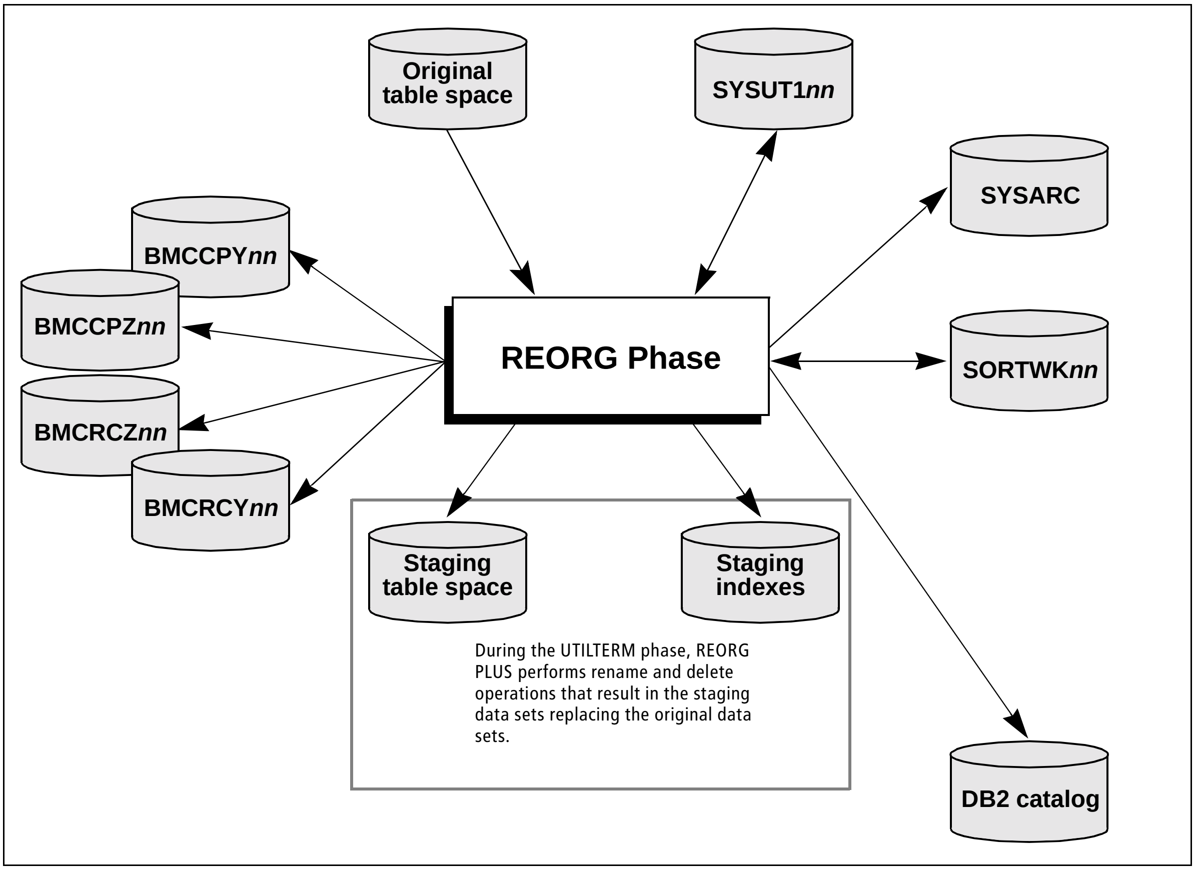
Figure 58 on page 566 through Figure 61 on page 568 show the objects that REORG PLUS uses in the REORG phase.

Figure 58 REORG phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY table space reorganization



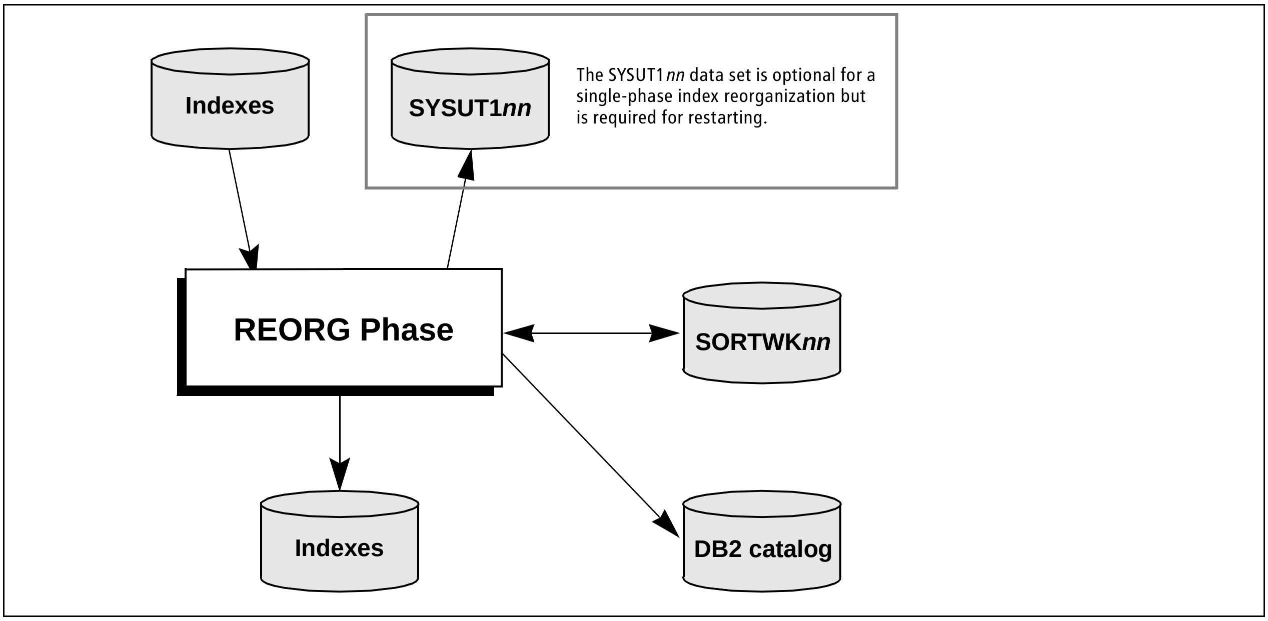
The SYSREC*nn* data set is optional for a single-phase table space reorganization but is required for restarting.

Figure 59 REORG phase: SHRLEVEL REFERENCE table space reorganization



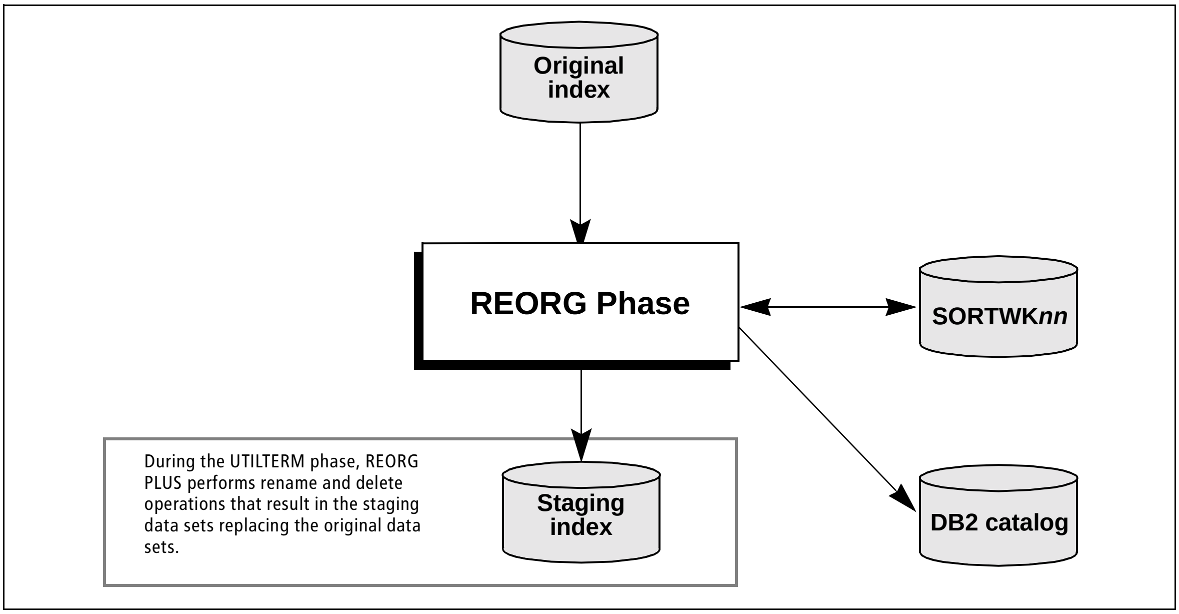
During the UTILTERM phase, REORG PLUS performs rename and delete operations that result in the staging data sets replacing the original data sets.

Figure 60 REORG phase: SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY index reorganization



The SYSUT1*nn* data set is optional for a single-phase index reorganization but is required for restarting.

Figure 61 REORG 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.

Chapter 7

Online reorganization

This chapter presents the following topics:

Introduction to online reorganizations

Performing an online reorganization by specifying SHRLEVEL CHANGE on the REORG command offers the following benefits:

- **allows full access** to DB2® data during most of the reorganization

- **delivers improved data availability** by greatly reducing the outage for the DB2 objects, thus helping to meet growing 24 x 7 requirements

- **operates in a nondestructive manner,** allowing you to make the objects available without having to recover in the event of a failure

- **optionally provides support for batch applications** that use the BMC Software APPLICATION RESTART CONTROL (AR/CTL) product

  Online reorganizations use the AR/CTL suspend-and-resume interface, which eliminates the outage for DB2 objects.

To use the SHRLEVEL CHANGE option, you must have installed the BMC Software EXTENDED BUFFER MANAGER (XBM) product or SNAPSHOT UPGRADE FEATURE (SUF) component of XBM.

This chapter discusses online reorganizations in depth, concentrating on the aspects of reorganization processing that are unique to SHRLEVEL CHANGE. Aspects that are the same as those for other types of reorganizations are covered in other chapters and appendixes.

Overview of SHRLEVEL CHANGE

To invoke an online reorganization, REORG PLUS provides a SHRLEVEL CHANGE keyword. The keyword allows read-write (RW) application access to DB2 table space and index space objects during the reorganization process.

When you specify the SHRLEVEL CHANGE keyword, REORG PLUS performs an online reorganization, which includes the following processing steps after the utility initializes:

1. (*optional*) analyzes the original DB2 table space and index space objects

2. externalizes changed pages for the original table space and index space objects and starts capturing log records

3. unloads table space or index space data from the original DB2 objects by using the XBM or SUF product

4. allocates staging objects that mirror the original table space or index space objects, or uses predefined staging objects

5. reloads data into the staging table space objects, and builds or updates the staging index space objects

6. creates a row ID (RID) translation map for a table space reorganization

7. applies log records to the staging objects to reflect all update activity since XBM was initialized

   REORG PLUS uses the RID translation map to reconcile the log records for a table space reorganization.

8. prevents any access to the original objects and switches the data sets, which causes the staging objects to replace the original objects

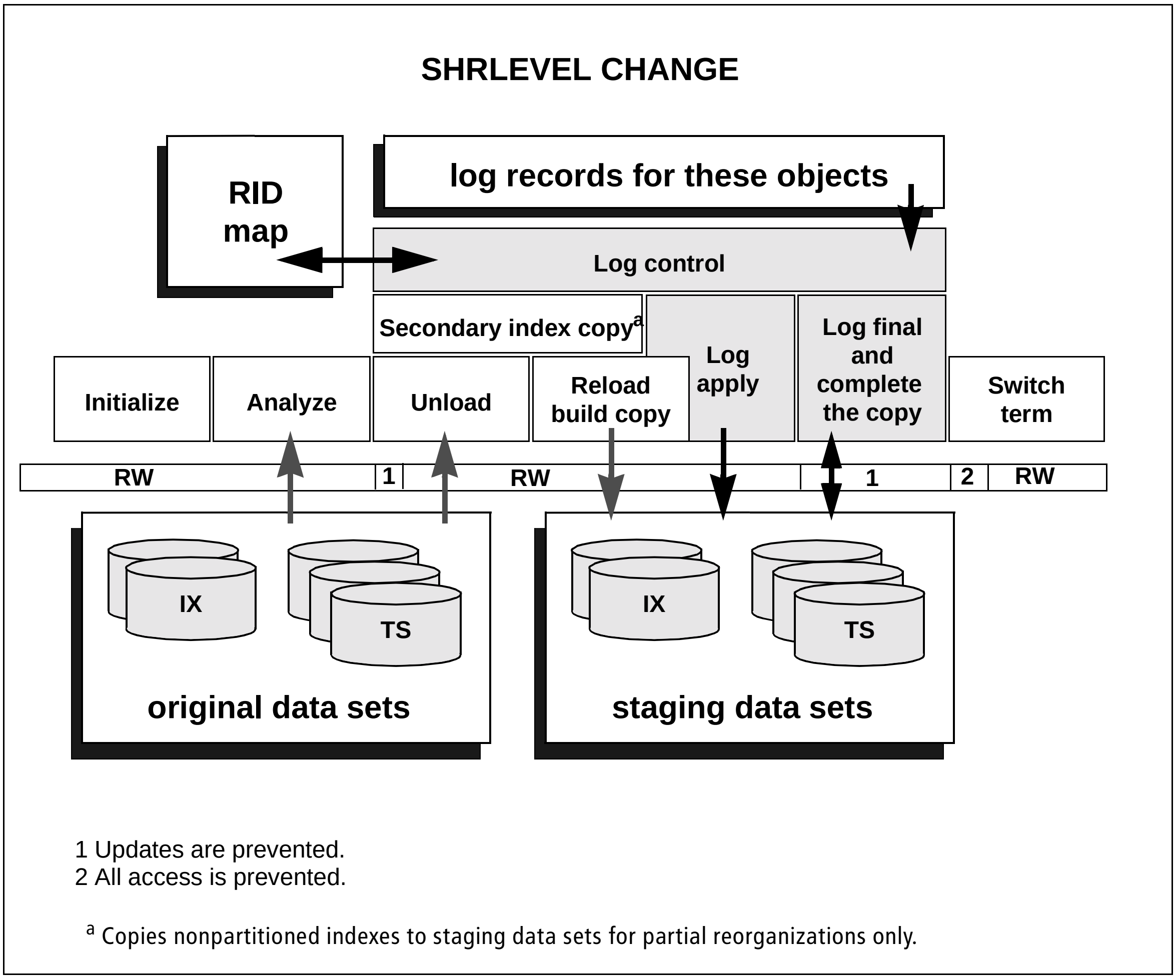
9. registers full and incremental copies, ensuring recoverability of objects

10. allows full access to the objects to resume

11. optionally updates the DB2 catalog and DASD MANAGER PLUS statistics tables

Figure 62 on page 572 illustrates the processing flow for a SHRLEVEL CHANGE reorganization. For details and information about the objects' status during the reorganization, see "How SHRLEVEL CHANGE works" on page 572.

Figure 62 Processing flow for SHRLEVEL CHANGE



[a] Copies nonpartitioned indexes to staging data sets for partial reorganizations only.

How SHRLEVEL CHANGE works

Processing for SHRLEVEL CHANGE is similar to SHRLEVEL REFERENCE. However, some additional tasks are performed in the UNLOAD and RELOAD phases, and two additional phases occur, LOGAPPLY and LOGFINAL. This section describes the key tasks that are performed during each execution phase. Figure 63 on page 577 through Figure 66 on page 582 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:

Topic	Reference
when REORG PLUS turns on Data Capture Changes	DATACAP installation option on page 636
spill data sets	"Allocation of spill data sets" on page 601
SHRLEVEL CHANGE options	"SHRLEVEL CHANGE options" on page 300
performance considerations for the UNLOAD phase	"UNLOAD phase" on page 557

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:

Reorganization type	REORG PLUS performs this action
full or partial table space	loads the table space data into the table space staging data sets
full table space	loads the index data into the index staging data sets
partial table space	updates the nonpartitioned indexes that REORG PLUS previously copied to the staging data sets
index-only	loads the index data into the index staging data sets

During this phase, REORG PLUS also sorts the indexes and collects statistics. For a table space reorganization, REORG PLUS makes a full image copy of the table space. However, REORG PLUS does not register the copy in the SYSIBM.SYSCOPY table until the UTILTERM phase. At the end of the RELOAD phase, the staging data sets contain a reorganized version of the original data sets as they appeared when XBM took the snapshot.

For information about performance considerations for the RELOAD phase, see "RELOAD phase" on page 560.

REORG phase

For SHRLEVEL CHANGE, REORG PLUS combines the operations of the UNLOAD and RELOAD phases of a two-phase reorganization into the REORG processing phase for a single-phase reorganization.

LOGAPPLY phase

When the RELOAD or REORG phase is complete, the LOGAPPLY phase begins. This phase applies the stored log records to the reorganized staging data sets. REORG PLUS continuously monitors parameters and events until it reaches one of the criteria that triggers the LOGFINAL phase, such as reaching the log threshold (as specified with the LOGTHRESHLD option). The criteria are based on the log apply control options that you set in the options module or with the REORG command. For more information, see "Control of the log apply process" on page 586.

REORG PLUS also provides statistics regarding the progress of the LOGAPPLY phase and the overall reorganization. This information is available through the XBM Utility Monitor interface or the MVS™ console. In addition to viewing the status of the job, you can dynamically change the settings of any of the log apply control options by using the XBM Utility Monitor interface or the MVS console. For more information, see "Using XBM to view and dynamically control the log apply process" on page 592.

Throughout this phase, REORG PLUS continues to monitor the DB2 log for records that pertain to the objects that you are reorganizing. If the value for ICTYPE is UPDATE, REORG PLUS updates the full copy data sets throughout the LOGAPPLY phase.

LOGFINAL phase

During the LOGFINAL phase, REORG PLUS controls access to the original table space and index space objects based on the value that you specify for the DRAIN command option or DRAINTYP installation option, as follows:

- prevents only updates if you specify WRITERS
- prevents all access if you specify ALL

Controlling access stops the arrival of new log records and allows REORG PLUS to finish applying the last of the existing log records to the staging data sets. If required and the table space is partition-by-growth, REORG PLUS adds partitions if the table space is partition-by-growth. Then, depending on the value of ICTYPE, REORG PLUS performs *one* of the following actions:

- creates incremental copy data sets to record changes made since REORG PLUS created the full copy data sets
- updates the full copy data sets

However, REORG PLUS does not register any copies in SYSIBM.SYSCOPY until the UTILTERM phase.

If Data Capture Changes was turned on in the UNLOAD phase, REORG PLUS turns it off in this phase. In some error conditions, REORG PLUS might not be able to turn off Data Capture Changes. For more information, see "Not completing before the UTILTERM phase" on page 610.

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 in the UTILTERM phase" on page 370.

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 process or the FASTSWITCH process, as shown in the following table. For more information about the rename and FASTSWITCH processes, see "Staging data sets" on page 98.

Process used	REORG PLUS action
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 in the UTILTERM phase" on page 370 for information about the actions to take next.

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" on page 124.

REORG PLUS registers the full image copy data sets and incremental copy data sets (if incremental copies were created) in the SYSIBM.SYSCOPY table.

After completing the rename or FASTSWITCH process, REORG PLUS allows all access to the objects to resume. REORG PLUS optionally deletes the original data sets or renames them in preparation for a subsequent reorganization. REORG PLUS updates the BMCHIST table and optionally updates

- real-time statistics
- statistics in the DB2 catalog
- DASD MANAGER PLUS statistics tables

When REORG PLUS completes this phase, the reorganization is complete.

Data sets by execution phase

Figure 63 on page 577 through Figure 66 on page 582 show the data sets that each execution phase uses during a SHRLEVEL CHANGE reorganization. For more information about the data sets, see "REORG PLUS data sets" on page 57 and "Staging data sets" on page 98.

Figure 63 Processing phases: SHRLEVEL CHANGE two-phase table space reorganization (part 1 of 2)

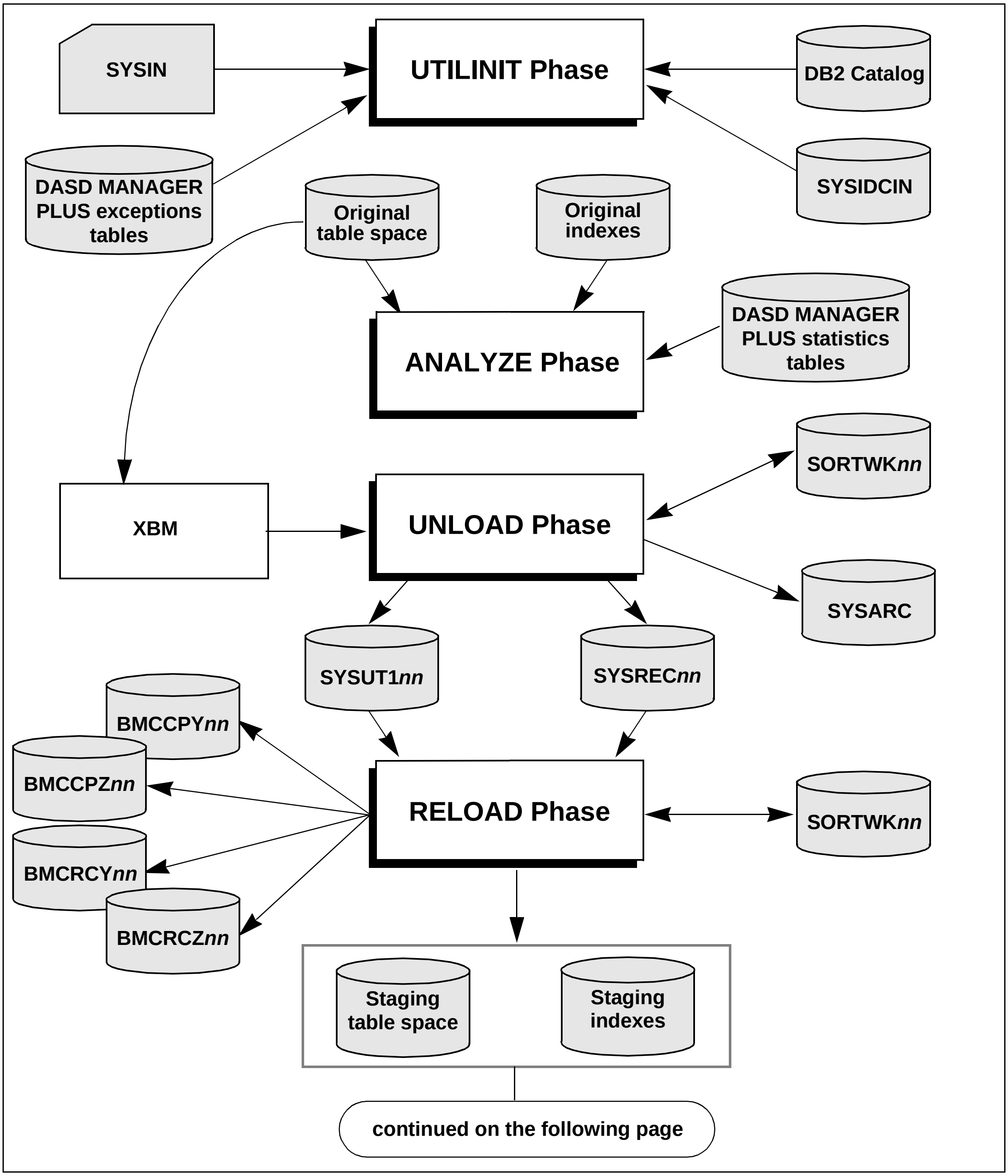
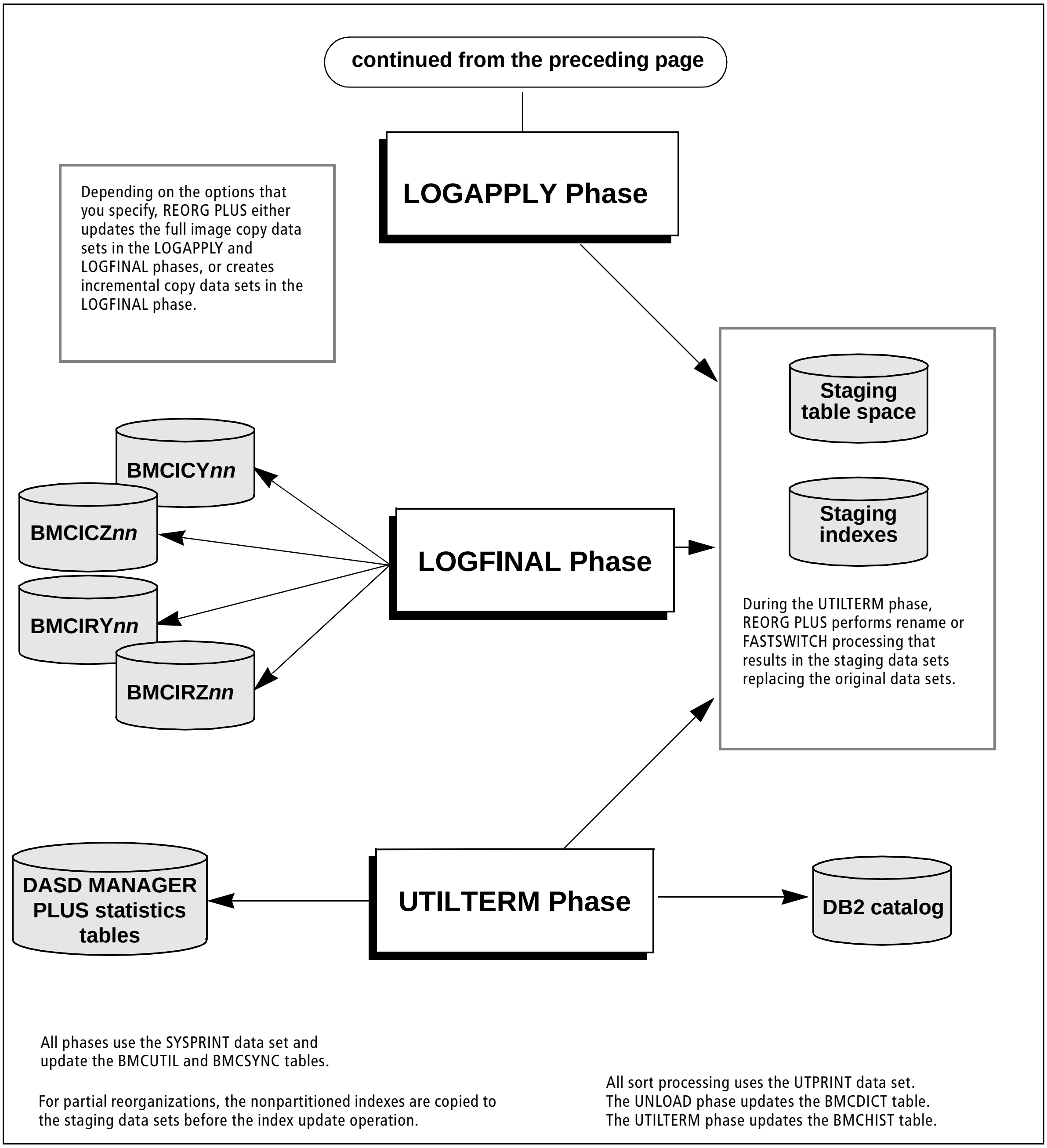


Figure 63 Processing phases: SHRLEVEL CHANGE two-phase table space reorganization (part 2 of 2)



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.

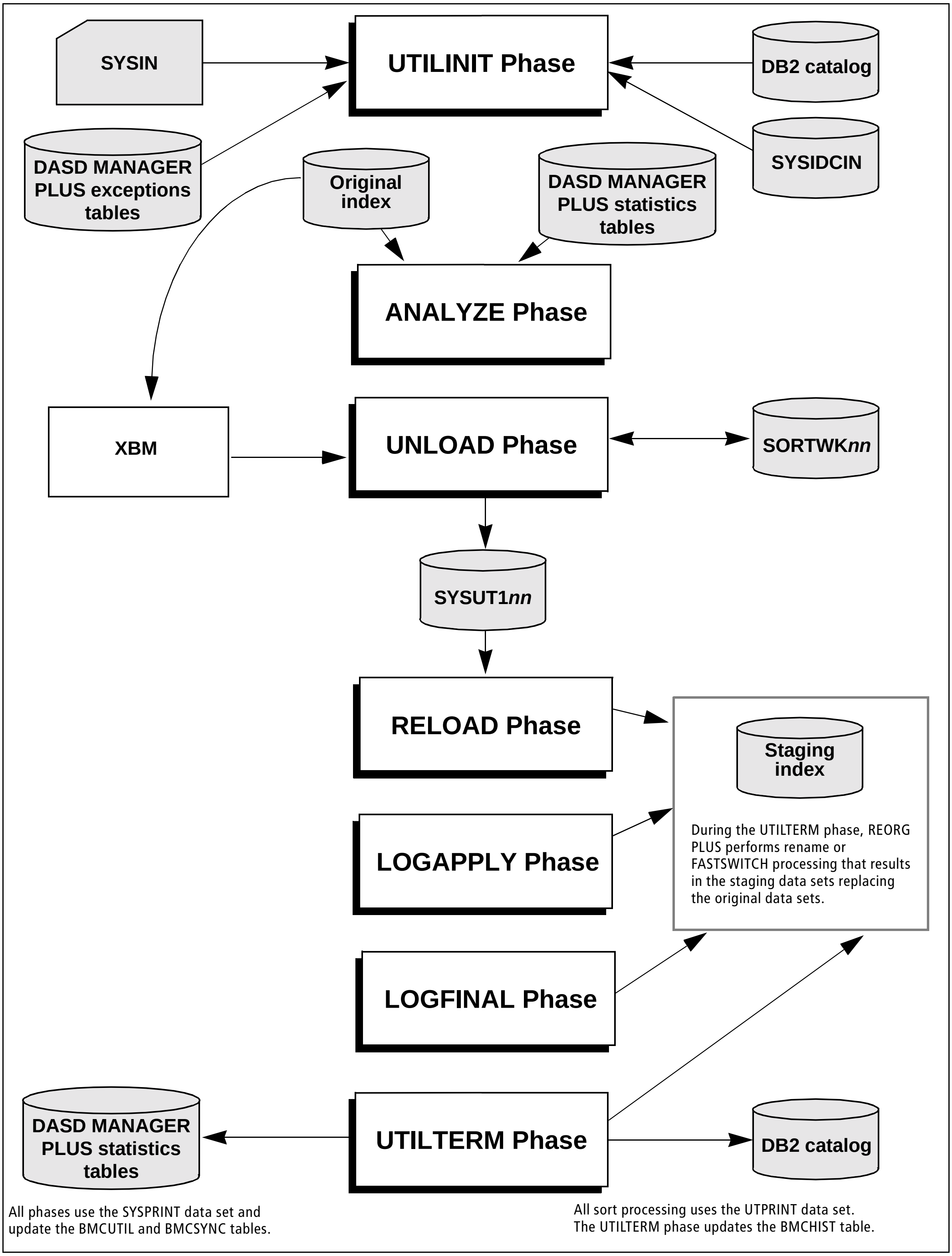
During the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

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

For partial reorganizations, the nonpartitioned indexes are copied to the staging data sets before the index update operation.

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

Figure 64 Processing phases: SHRLEVEL CHANGE two-phase index reorganization



During the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

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

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

Figure 65 Processing phases: SHRLEVEL CHANGE single-phase table space reorganization (part 1 of 2)

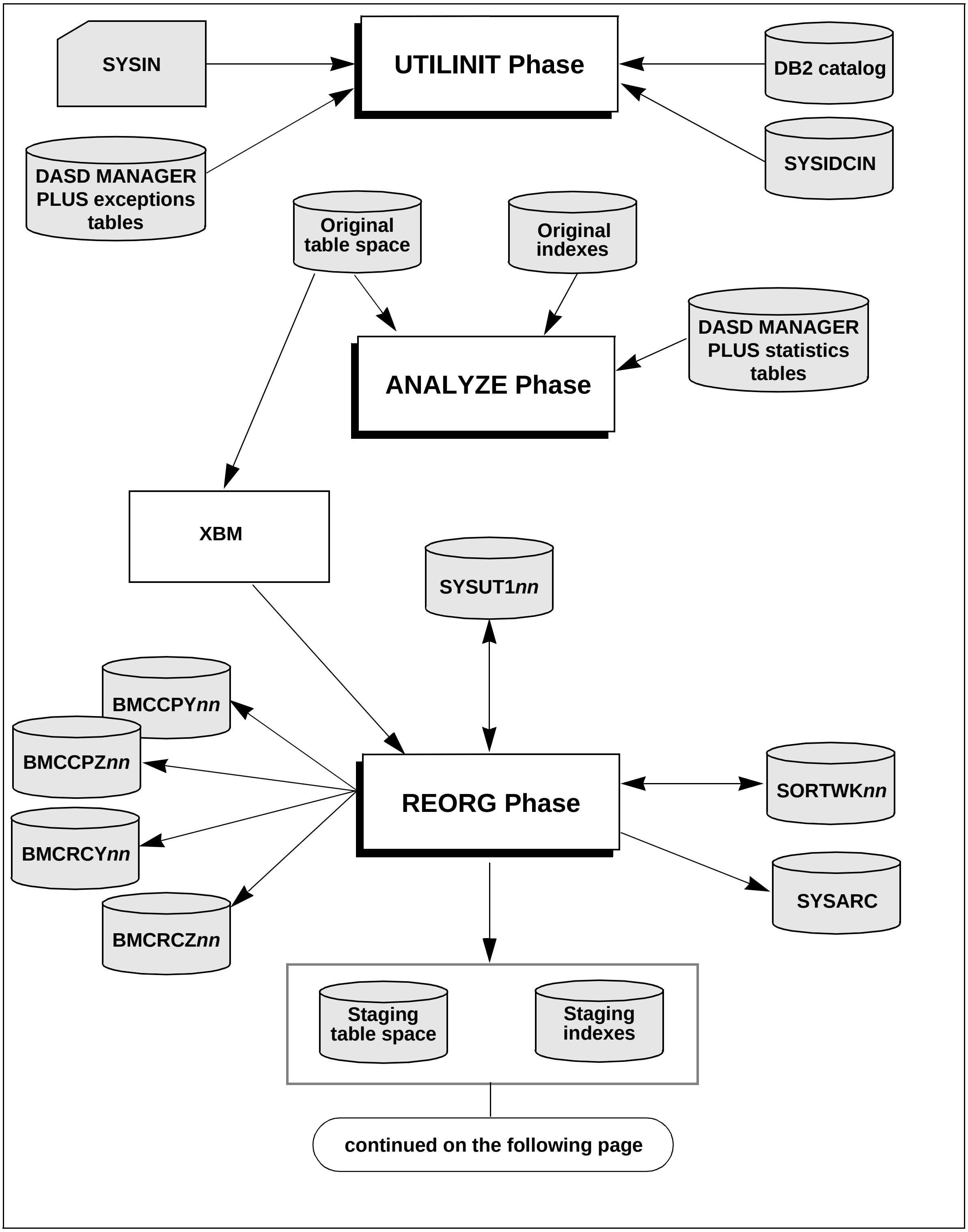
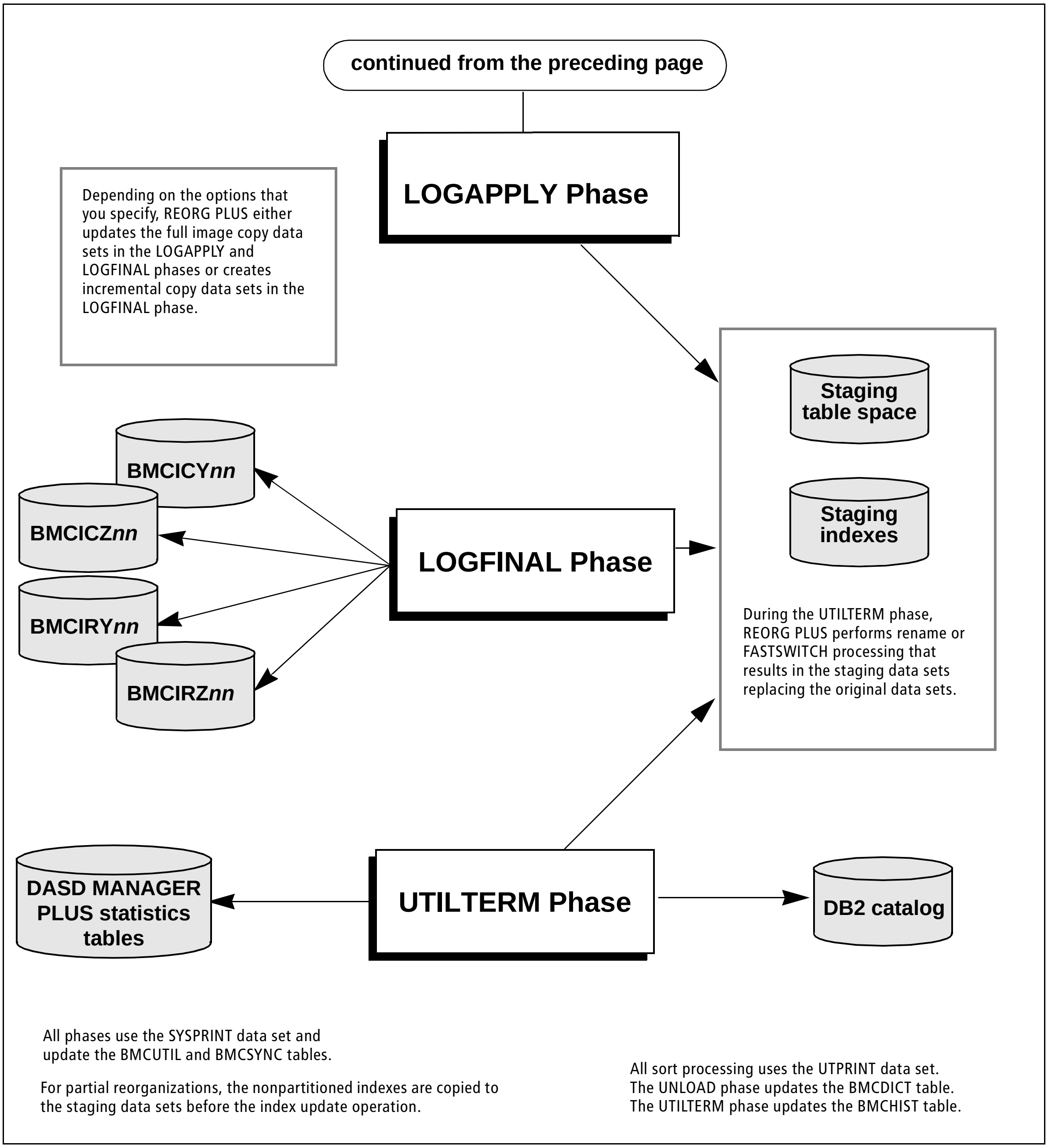


Figure 65 Processing phases: SHRLEVEL CHANGE single-phase table space reorganization (part 2 of 2)



Depending on the options that you specify, REORG PLUS either updates the full image copy data sets in the LOGAPPLY and LOGFINAL phases or creates incremental copy data sets in the LOGFINAL phase.

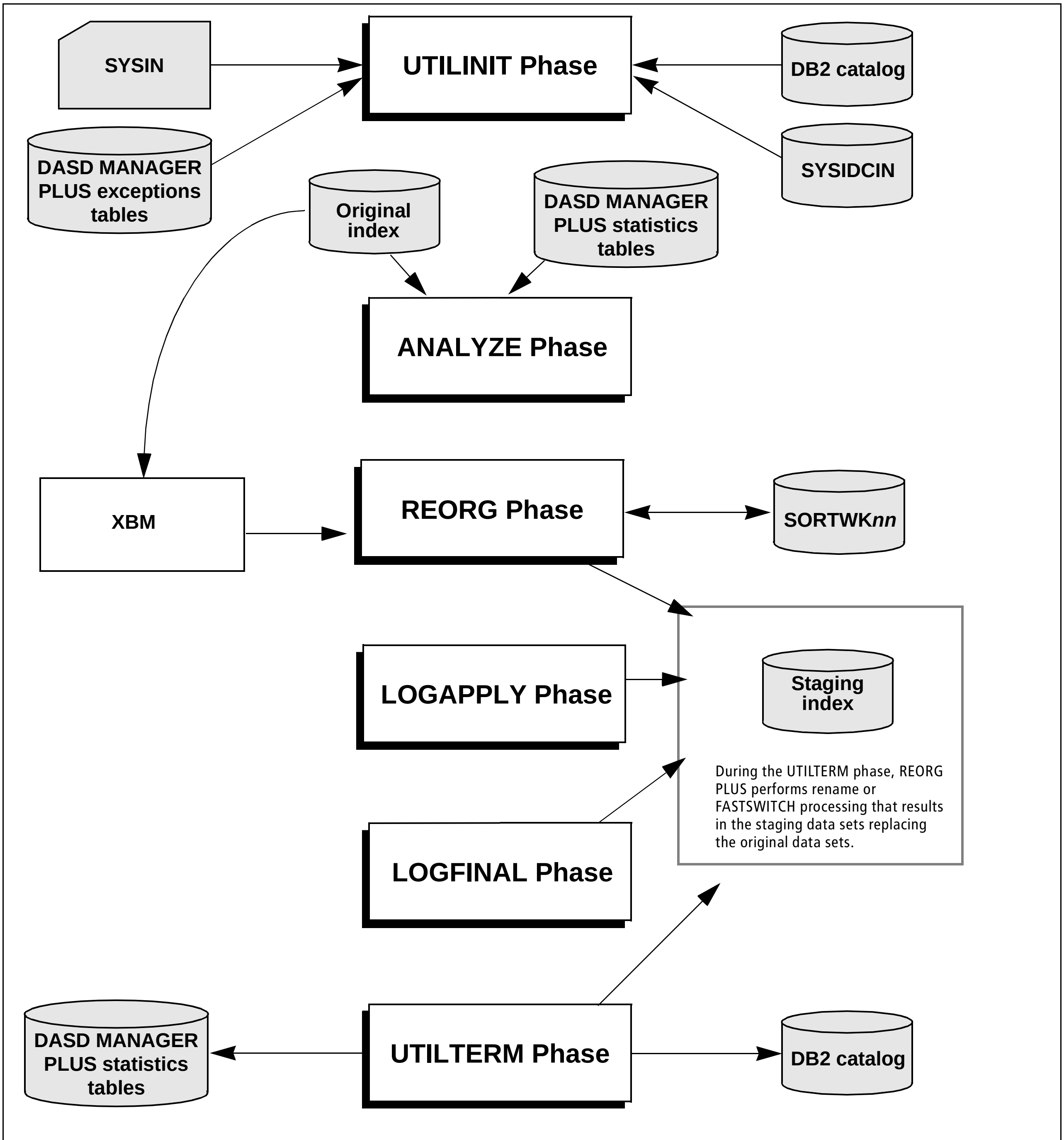
During the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

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

For partial reorganizations, the nonpartitioned indexes are copied to the staging data sets before the index update operation.

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

Figure 66 Processing phases: SHRLEVEL CHANGE single-phase index reorganization



During the UTILTERM phase, REORG PLUS performs rename or FASTSWITCH processing that results in the staging data sets replacing the original data sets.

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

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

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.

Processing differences

The SHRLEVEL CHANGE option of REORG PLUS allows DB2 table space and index space objects to be available for RW access by DB2 applications during most of the reorganization.

Allocating storage versus using DB2 resources

To be truly nondisruptive, a reorganization utility must do more than simply allow application access. If the reorganization utility competes with the application for database resources, disruption in the form of degraded performance can still occur. To avoid this situation, REORG PLUS does not use the DB2 buffer pool, buffer manager, data manager, or work databases to accomplish its tasks.

Instead, REORG PLUS allocates data sets, staging copies of the DB2 objects, and virtual storage as needed. This additional DASD and utilization of virtual storage space eliminates the use of DB2 resources to accomplish the reorganization. In addition, the RID map is not a DB2 object in REORG PLUS as it is in the IBM DB2 REORG utility. This factor prevents RID map access from interfering with DB2 processing. By eliminating contention for these resources between REORG PLUS and the application, the reorganization can proceed in a less disruptive manner.

Determining the longlog condition

REORG PLUS and DB2 REORG differ fundamentally in how they determine a longlog condition. DB2 REORG has an internal buffer that contains log records. When this buffer becomes full, DB2 REORG processes the records. DB2 REORG then repeats the process, making another pass at the next batch of log records. It uses an average of the last *n* passes to determine whether a longlog condition exists, or whether processing can be completed by the time that is set by the DEADLINE option or by the MAXRO value.

In contrast, REORG PLUS samples the arrival rate of the log records every 15 seconds to determine the rate at which the records are being applied. It uses a weighted average of the last 15 samples to determine whether a longlog condition exists, or whether processing can be completed by the time that is set by DEADLINE or by the MAXRO value.

By using more samples and a weighted average, REORG PLUS provides the following advantages over DB2 REORG:

- a greater level of accuracy in determining whether processing can be completed by the time that is set by DEADLINE or by the MAXRO value
- an increased probability that sudden changes in activity levels will not force unnecessary longlog conditions
- the ability to specify LOGTHRSHLD to define when the process will end

Updating nonpartitioned indexes during a partial reorganization

In a SHRLEVEL CHANGE reorganization with DB2 REORG, the object is partially unavailable during the BUILD2 phase, which updates logical partitions within the nonpartitioned index. In contrast, REORG PLUS keeps a copy of the nonpartitioned index current during log apply processing. Consequently, REORG PLUS does not need to update the index after renaming the data sets. For the table space partitions that you are reorganizing, this method reduces the time during which the partitions are unavailable.

Online partition rebalancing

When you use a DDLIN data set with ALTER statements, REORG PLUS performs online partition rebalancing and alters limit keys for you with no outage and without your space entering REORP status. Online partition rebalancing is performed on the staging data sets and the ALTER commands are performed in the UTILTERM phase. For more information, see “DDLIN data set” on page 124.

Alternatively, you can use the REBALANCE command option to rebalance partitions. By specifying REBALANCE, you can have REORG PLUS determine the limit keys for you. For more information, see “REBALANCE” on page 167.

Administrative differences

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, and 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

Control of the log apply process

You can control the log apply process with log control options:

- set defaults for the log apply control options in the installation options module
- override the default options by using the REORG command
- change the log apply control options while the reorganization is running (up to and including the LOGAPPLY phase)

  To change options while the reorganization is running, you can use the XBM Utility Monitor function or the MVS console. However, after beginning the LOGFINAL phase, REORG PLUS ignores any changes to the options. For more information, see "Using XBM to view and dynamically control the log apply process" on page 592.

Depending on the values that have been set for these options, you can allow the LOGAPPLY phase to continue indefinitely, start the LOGFINAL phase, or terminate the reorganization.

Log apply control options

Table 92 lists the options that REORG PLUS provides to allow you to control the log apply process during the reorganization. You do so by changing any one of the values for the options that REORG PLUS uses in the LOGAPPLY phase. In addition to the options that change execution, REORG PLUS provides the DISPLAY option to monitor the process.

Table 92 Log apply control options (part 1 of 2)

Option	Description
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 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.

Table 92 Log apply control options (part 2 of 2)

Option	Description
DELAY	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.
LOGFINAL	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.
LOGTHRESHLD	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.
LONGLOG	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. The following values are valid for LONGLOG: ■ TERM tells REORG PLUS to terminate the reorganization. ■ DRAIN tells REORG PLUS to start the LOGFINAL phase. ■ CONTINUE tells REORG PLUS to continue the LOGAPPLY phase.
MAXRO	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.
DISPLAY	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 "Display information about a job" on page 593.

Considerations for log apply control options

The following additional considerations apply when you use the log apply control options:

- If MAXRO is DEFER, the LOGAPPLY phase continues indefinitely, and REORG PLUS ignores all other options (except DEADLINE). You must change MAXRO to a value (and possibly set other options) to complete the reorganization.

- Because it uses a weighted average of samples, REORG PLUS does not detect a longlog condition during brief spurts of activity.

- If REORG PLUS detects a longlog condition but ends before reaching the DELAY time, REORG PLUS resets the DELAY time to its original value.

Criteria for entering the LOGFINAL phase

For REORG PLUS to stop the LOGAPPLY phase and enter the LOGFINAL phase, *both* of the following conditions must exist:

- REORG PLUS estimates that it can complete the LOGFINAL phase by the time that is specified in DEADLINE (if DEADLINE has a value other than NONE).

- MAXRO is set to a value other than DEFER.

In addition, at least *one* of the following conditions must exist:

- The time specified in the LOGFINAL option has been reached.

- LOGFINAL is NONE, a longlog condition does not exist, and either the MAXRO or the LOGTHRESHLD conditions exist.

- The longlog condition exists, the time that is specified in the DELAY option has been reached, and LONGLOG is set to DRAIN.

The reorganization cannot complete if the LOGFINAL phase does not start. Also, if too many changes are made to the data sets between the time that XBM is initialized and the time that REORG PLUS applies the incremental image copies, the new data sets might be relatively disorganized. However, the new data sets should be less disorganized than the originals.

Criteria for ending the reorganization

If necessary, you can also end the reorganization before it completes, based on how you set the log apply control options. The reorganization ends if either of the following conditions exists:

- REORG PLUS estimates that it cannot finish processing before the time that you specify in the DEADLINE option.

- A longlog condition exists, the time that you specify in the DELAY option is reached, and you set LONGLOG to TERM.

Hierarchy of options

Figure 67 illustrates how REORG PLUS uses the log apply control options to control SHRLEVEL CHANGE processing after the LOGAPPLY phase starts. The figure represents one 15-second sample interval. Processing repeats indefinitely, until either the LOGFINAL phase begins or the reorganization ends.

Figure 67 How options determine log apply processing (part 1 of 3)

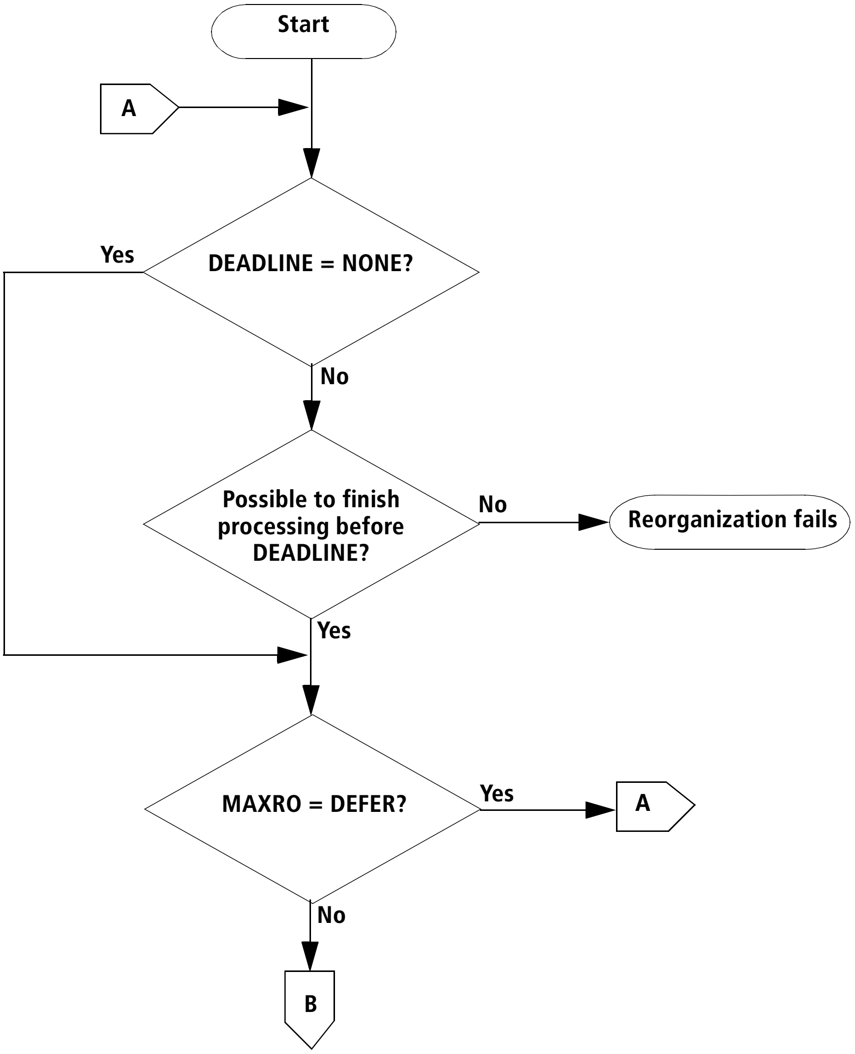


Figure 67 How options determine log apply processing (part 2 of 3)

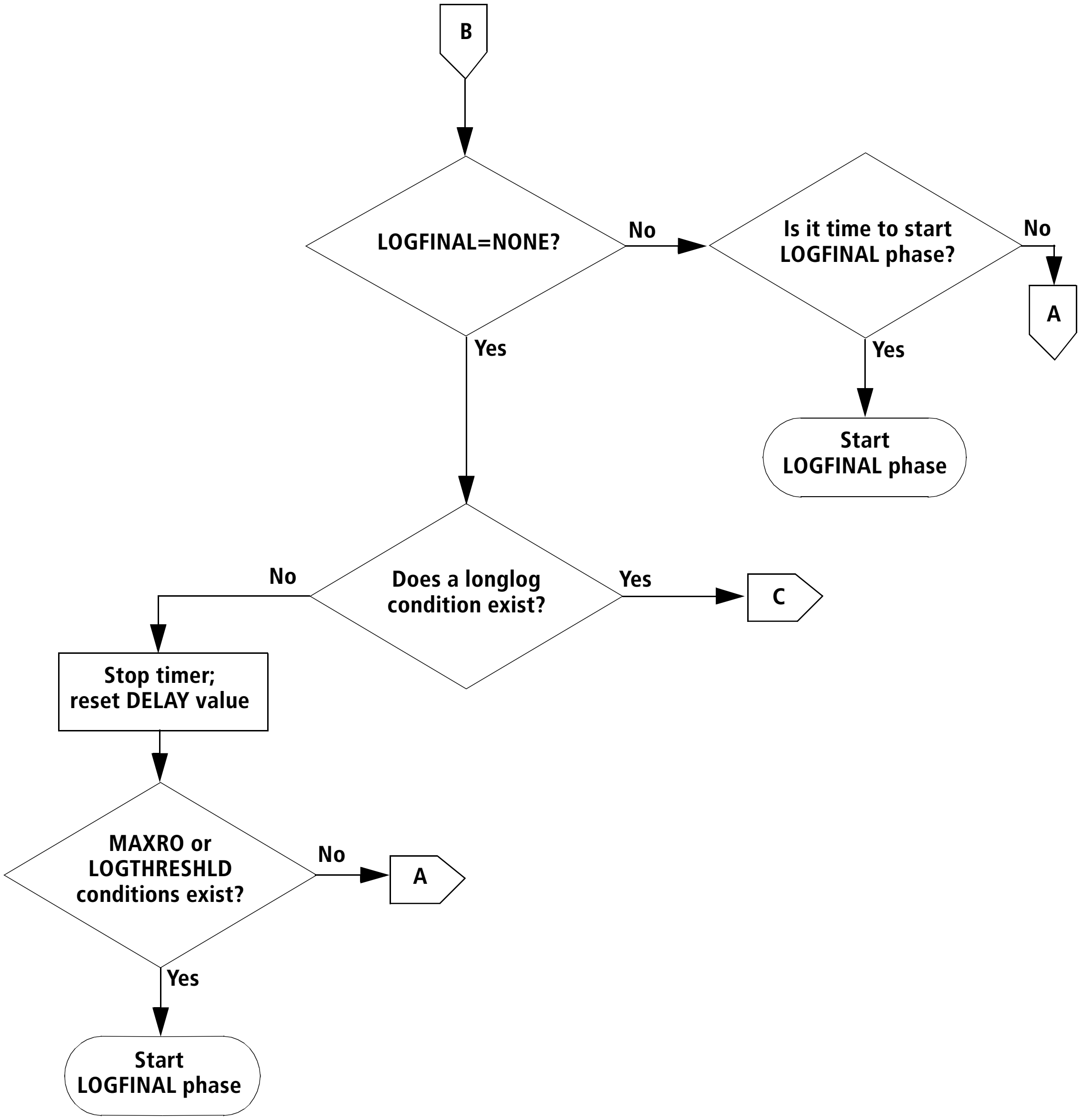
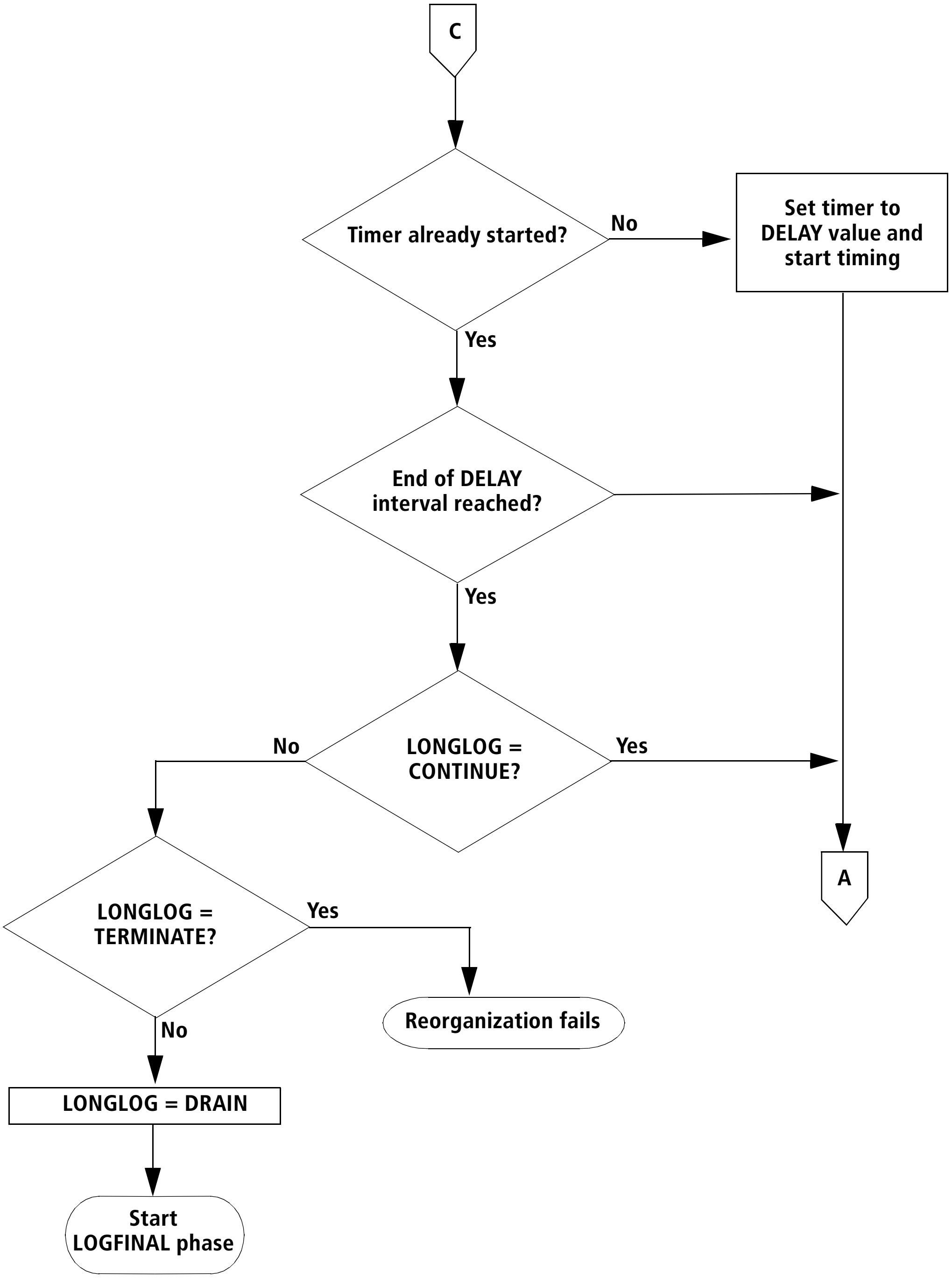

B
LOGFINAL=NONE?
No
Is it time to start LOGFINAL phase?
No
A
Yes
Yes
Start LOGFINAL phase
No
Does a longlog condition exist?
Yes
C
Stop timer; reset DELAY value
MAXRO or LOGTHRESHLD conditions exist?
No
A
Yes
Start LOGFINAL phase

Figure 67 How options determine log apply processing (part 3 of 3)



Using XBM to view and dynamically control the log apply process

With XBM or SUF, you can view information about the reorganization while it is running, and change the options that affect the LOGAPPLY phase. You can access REORG PLUS by using the XBM Utility Monitor function, or you can send a command from the MVS console through XBM to REORG PLUS.

Utility Monitor

The XBM ISPF interface provides a Utility Monitor. You can use this monitor to display statistics and information about the SHRLEVEL CHANGE reorganization jobs that are currently running. You can also use the Utility Monitor to change the log apply control options (as discussed in "Log apply control options" on page 586) for any of those jobs.

Monitor a job

You can use the XBM ISPF interface for a menu-driven approach to monitoring and changing a log apply reorganization. After you select the Utility Monitor function and a specific reorganization, the XBM REORG PLUS Online panel displays information about that reorganization job. The panel displays the job name, step name, start date, start time, and execution phase that is currently processing. The panel also displays the current values for the log apply control options, as well as information about the data sets that have been registered with XBM.

After REORG PLUS enters the LOGAPPLY phase, the REORG PLUS Online panel replaces the registered data set information with statistics about the log records for both the index and the table space. These statistics include the number of records that have been applied, the number that have been queued, the arrival rate, and the application rate.

You can also enter the REORG PLUS DISPLAY option from the XBM Send New command panel to get additional information about processing of the log records. For a description and syntax, see "Display information about a job" on page 593.

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.

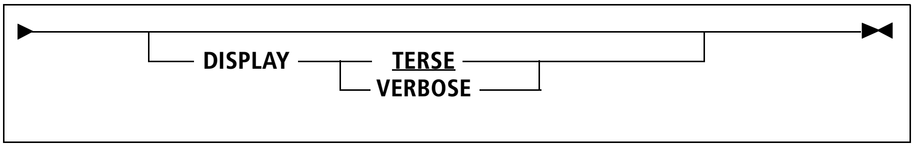
- Use the **Send New** command from the Command menu to send a new value for an option to REORG PLUS.

  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:



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.

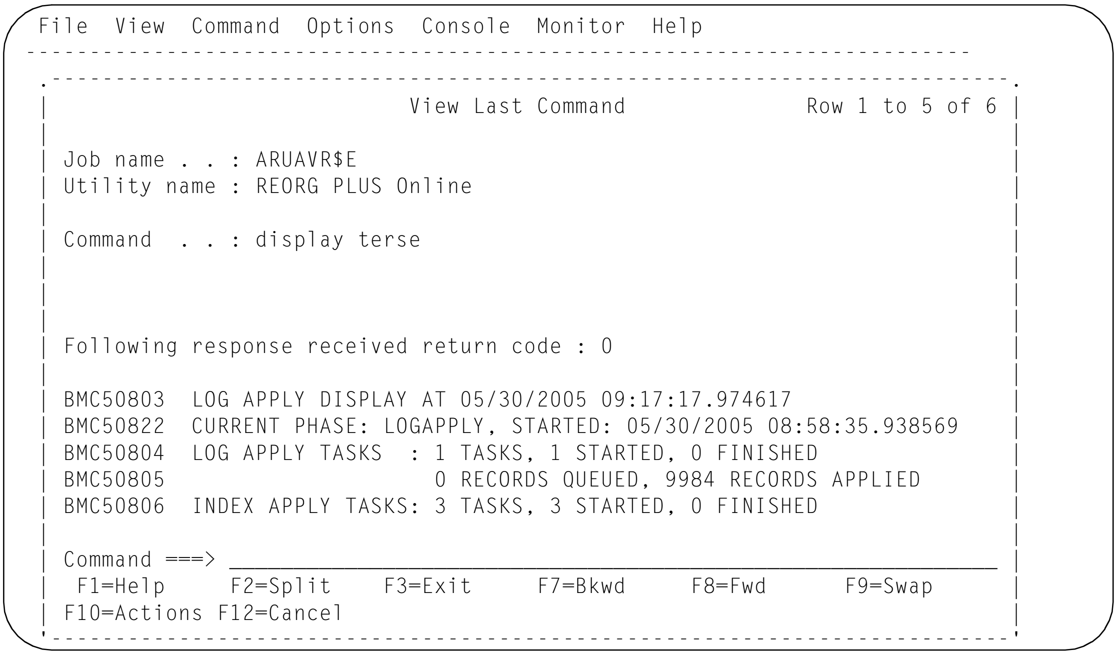
NOTE

Alternatively, you can enter the DISPLAY option with the XBM SEND command on the MVS console. For more information, see “MVS console” on page 597.

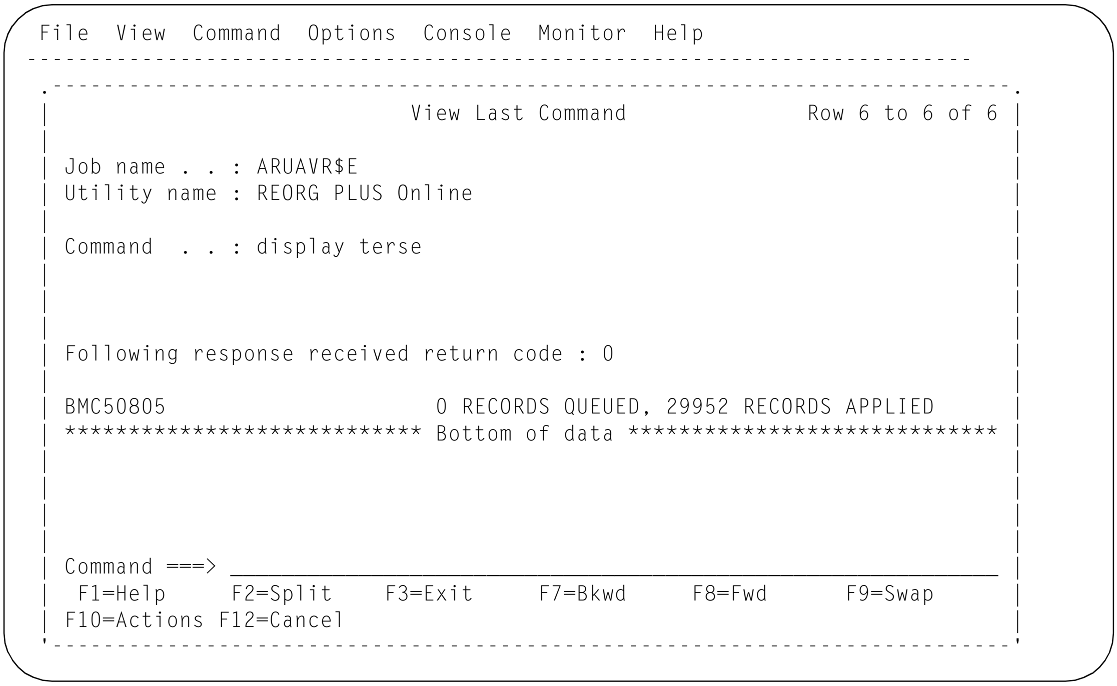
DISPLAY TERSE output

Figure 68 shows an example of the output from the DISPLAY TERSE command.

Figure 68 DISPLAY TERSE output



```
 File  View  Command  Options  Console  Monitor  Help
--------------------------------------------------------------------------
 .-----------------------------------------------------------------------------.
 |                            View Last Command            Row 1 to 5 of 6 |
 |                                                                            |
 | Job name . . : ARUAVR$E                                                    |
 | Utility name : REORG PLUS Online                                           |
 |                                                                            |
 | Command  . . : display terse                                               |
 |                                                                            |
 |                                                                            |
 |                                                                            |
 | Following response received return code : 0                                |
 |                                                                            |
 | BMC50803  LOG APPLY DISPLAY AT 05/30/2005 09:17:17.974617                  |
 | BMC50822  CURRENT PHASE: LOGAPPLY, STARTED: 05/30/2005 08:58:35.938569     |
 | BMC50804  LOG APPLY TASKS  : 1 TASKS, 1 STARTED, 0 FINISHED                |
 | BMC50805                     0 RECORDS QUEUED, 9984 RECORDS APPLIED        |
 | BMC50806  INDEX APPLY TASKS: 3 TASKS, 3 STARTED, 0 FINISHED                |
 |                                                                            |
 | Command ===> ____________________________________________________________ |
 |  F1=Help     F2=Split    F3=Exit     F7=Bkwd     F8=Fwd      F9=Swap      |
 | F10=Actions F12=Cancel                                                     |
 '----------------------------------------------------------------------------'
```

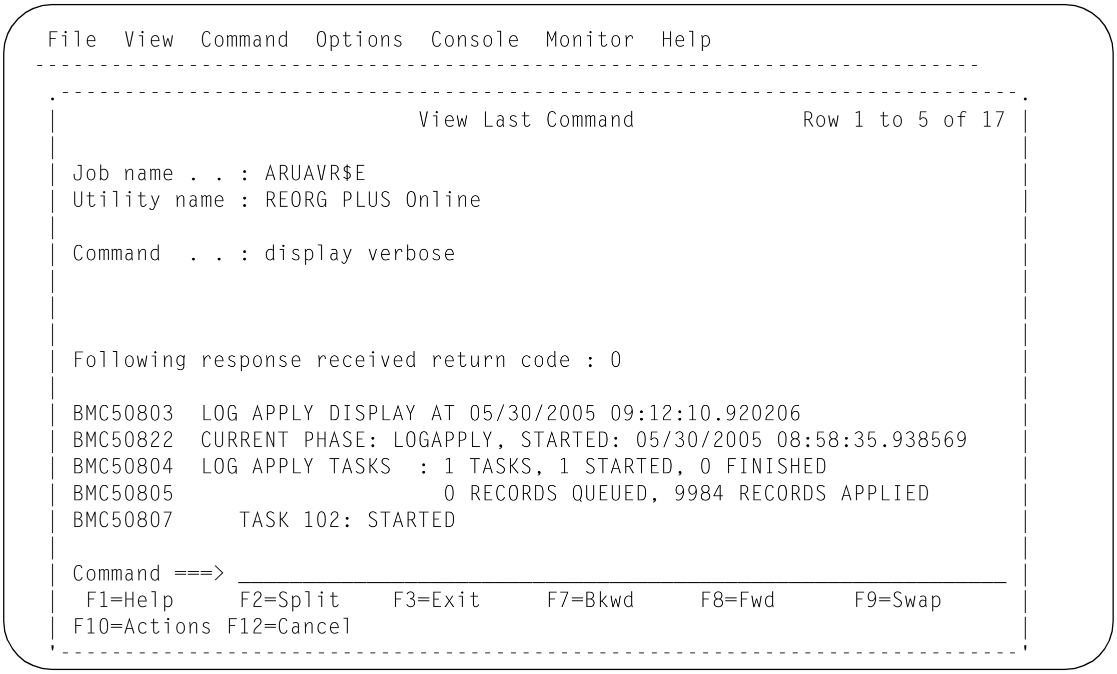


```
 File  View  Command  Options  Console  Monitor  Help
--------------------------------------------------------------------------
 .-----------------------------------------------------------------------------.
 |                            View Last Command            Row 6 to 6 of 6 |
 |                                                                            |
 | Job name . . : ARUAVR$E                                                    |
 | Utility name : REORG PLUS Online                                           |
 |                                                                            |
 | Command  . . : display terse                                               |
 |                                                                            |
 |                                                                            |
 |                                                                            |
 | Following response received return code : 0                                |
 |                                                                            |
 | BMC50805                     0 RECORDS QUEUED, 29952 RECORDS APPLIED       |
 | **************************** Bottom of data ***************************** |
 |                                                                            |
 |                                                                            |
 |                                                                            |
 |                                                                            |
 | Command ===> ____________________________________________________________ |
 |  F1=Help     F2=Split    F3=Exit     F7=Bkwd     F8=Fwd      F9=Swap      |
 | F10=Actions F12=Cancel                                                     |
 '----------------------------------------------------------------------------'
```

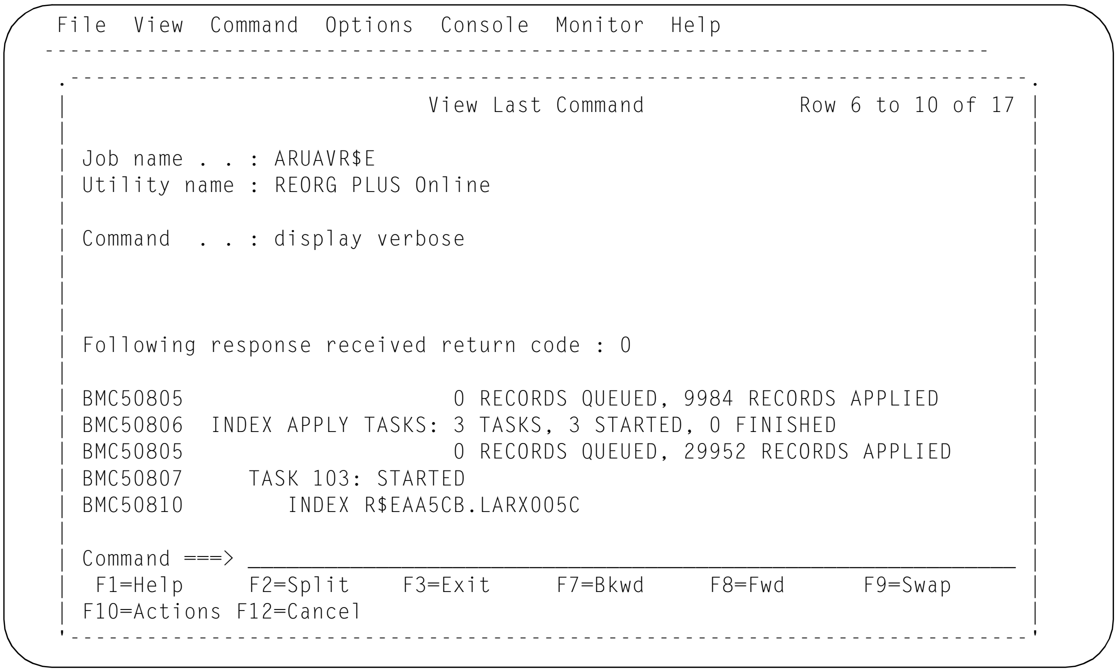
DISPLAY VERBOSE output

Figure 69 shows an example of the output from the DISPLAY VERBOSE command.

Figure 69 DISPLAY VERBOSE output (part 1 of 2)

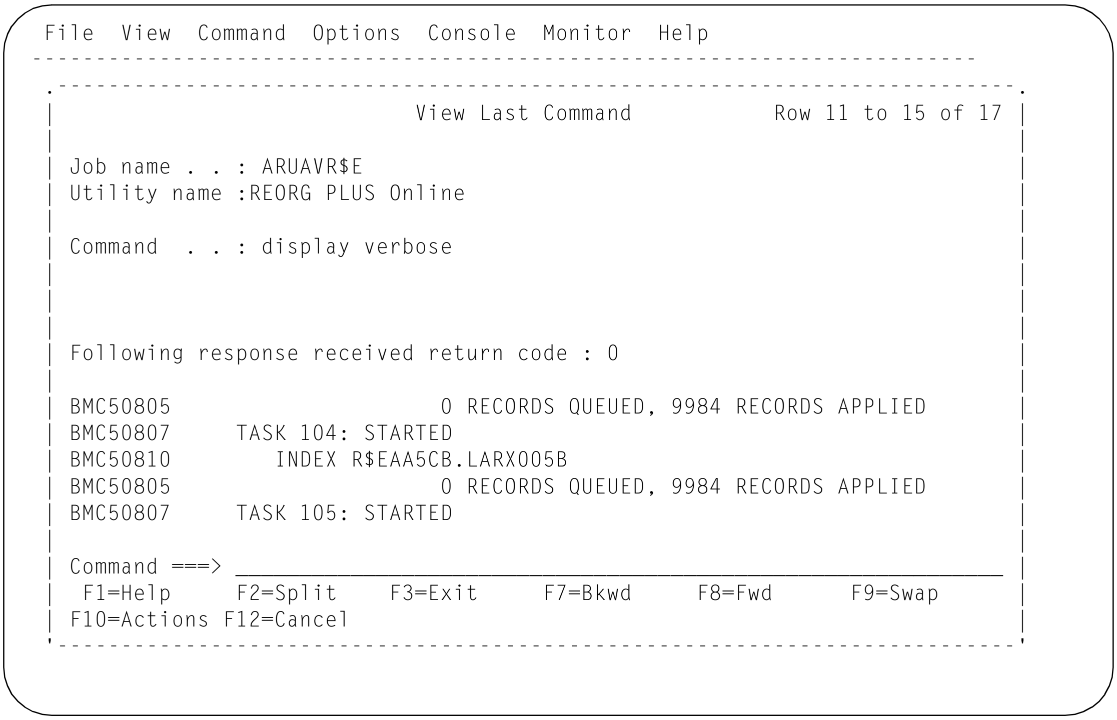


```
 File  View  Command  Options  Console  Monitor  Help
--------------------------------------------------------------------------
 .----------------------------------------------------------------------------.
 |                            View Last Command           Row 1 to 5 of 17 |
 |                                                                            |
 | Job name . . : ARUAVR$E                                                    |
 | Utility name : REORG PLUS Online                                           |
 |                                                                            |
 | Command   . . : display verbose                                            |
 |                                                                            |
 |                                                                            |
 |                                                                            |
 | 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                                             |
 |                                                                            |
 | Command ===> ____________________________________________________________ |
 |  F1=Help     F2=Split    F3=Exit     F7=Bkwd     F8=Fwd      F9=Swap       |
 | F10=Actions F12=Cancel                                                     |
 '----------------------------------------------------------------------------'
```

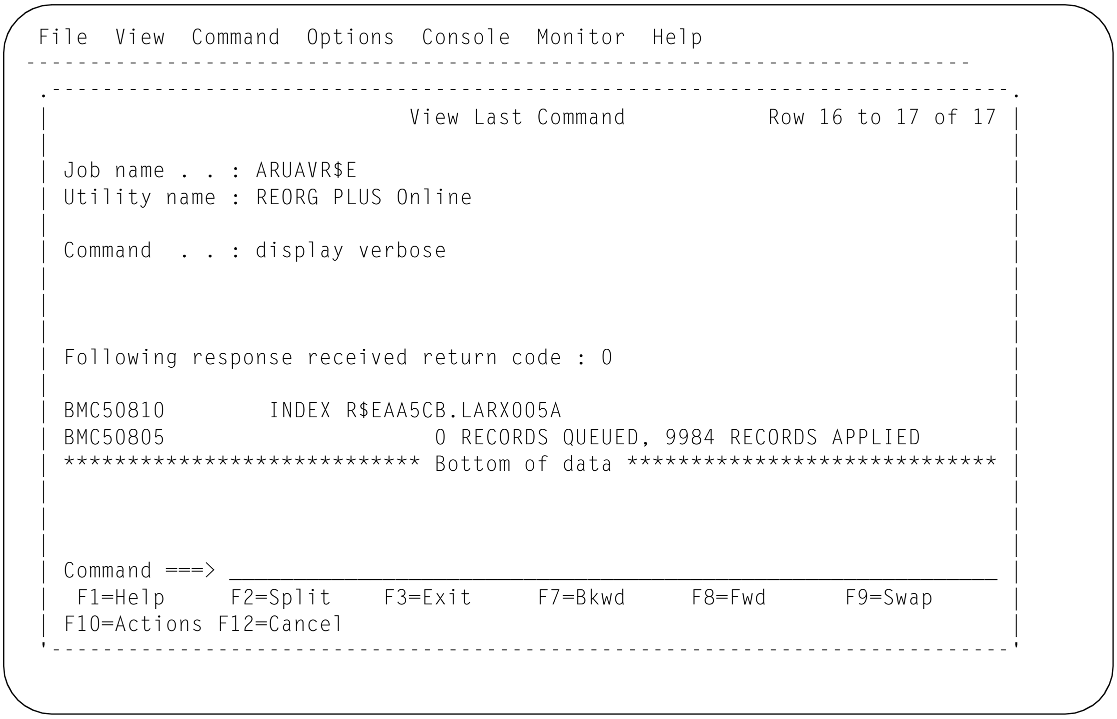


```
 File  View  Command  Options  Console  Monitor  Help
--------------------------------------------------------------------------
 .----------------------------------------------------------------------------.
 |                            View Last Command          Row 6 to 10 of 17 |
 |                                                                            |
 | Job name . . : ARUAVR$E                                                    |
 | Utility name : REORG PLUS Online                                           |
 |                                                                            |
 | Command   . . : display verbose                                            |
 |                                                                            |
 |                                                                            |
 |                                                                            |
 | Following response received return code : 0                               |
 |                                                                            |
 | BMC50805                     0 RECORDS QUEUED, 9984 RECORDS APPLIED        |
 | BMC50806  INDEX APPLY TASKS: 3 TASKS, 3 STARTED, 0 FINISHED                |
 | BMC50805                     0 RECORDS QUEUED, 29952 RECORDS APPLIED       |
 | BMC50807     TASK 103: STARTED                                             |
 | BMC50810        INDEX R$EAA5CB.LARX005C                                    |
 |                                                                            |
 | Command ===> ____________________________________________________________ |
 |  F1=Help     F2=Split    F3=Exit     F7=Bkwd     F8=Fwd      F9=Swap       |
 | F10=Actions F12=Cancel                                                     |
 '----------------------------------------------------------------------------'
```

Figure 69 DISPLAY VERBOSE output (part 2 of 2)



```
 File  View  Command  Options  Console  Monitor  Help
-------------------------------------------------------------------------
 .----------------------------------------------------------------------------.
 |                            View Last Command          Row 11 to 15 of 17 |
 |                                                                           |
 | Job name . . : ARUAVR$E                                                   |
 | Utility name :REORG PLUS Online                                           |
 |                                                                           |
 | Command  . . : display verbose                                            |
 |                                                                           |
 |                                                                           |
 |                                                                           |
 | Following response received return code : 0                              |
 |                                                                           |
 | BMC50805                    0 RECORDS QUEUED, 9984 RECORDS APPLIED        |
 | BMC50807     TASK 104: STARTED                                            |
 | BMC50810        INDEX R$EAA5CB.LARX005B                                   |
 | BMC50805                    0 RECORDS QUEUED, 9984 RECORDS APPLIED        |
 | BMC50807     TASK 105: STARTED                                            |
 |                                                                           |
 | Command ===> ____________________________________________________________ |
 |  F1=Help     F2=Split    F3=Exit     F7=Bkwd     F8=Fwd      F9=Swap     |
 | F10=Actions F12=Cancel                                                    |
 '----------------------------------------------------------------------------'
```



```
 File  View  Command  Options  Console  Monitor  Help
-------------------------------------------------------------------------
 .----------------------------------------------------------------------------.
 |                            View Last Command          Row 16 to 17 of 17 |
 |                                                                           |
 | Job name . . : ARUAVR$E                                                   |
 | Utility name : REORG PLUS Online                                          |
 |                                                                           |
 | Command  . . : display verbose                                            |
 |                                                                           |
 |                                                                           |
 |                                                                           |
 | Following response received return code : 0                              |
 |                                                                           |
 | BMC50810        INDEX R$EAA5CB.LARX005A                                   |
 | BMC50805                    0 RECORDS QUEUED, 9984 RECORDS APPLIED        |
 | **************************** Bottom of data ***************************** |
 |                                                                           |
 |                                                                           |
 |                                                                           |
 | Command ===> ____________________________________________________________ |
 |  F1=Help     F2=Split    F3=Exit     F7=Bkwd     F8=Fwd      F9=Swap     |
 | F10=Actions F12=Cancel                                                    |
 '----------------------------------------------------------------------------'
```

MVS console

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.

When REORG PLUS detects a longlog condition, it sends an action write-to-operator (WTO) notification. The ROUTCDE and DESCCDE installation options determine which console receives the WTO and how it is formatted. REORG PLUS deletes the WTO when the longlog condition ends or the LOGFINAL phase begins.

For information about the ROUTCDE and DESCCDE installation options, see Appendix A, “REORG PLUS installation options.” For information about the LONGLOG option, see “LONGLOG” on page 303.

Log apply control option scenarios

The following scenarios are typical scenarios that you might encounter when using the log apply control 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). You can use either the XBM Utility Monitor function or the MVS console to change the LONGLOG value.

Serialization and concurrency for SHRLEVEL CHANGE

DB2 applications might be using the affected objects during a SHRLEVEL CHANGE reorganization. Consequently, you cannot perform certain functions, and certain restrictions apply to the status of the objects before the reorganization.

Concurrency with other applications

During a reorganization, the SHRLEVEL CHANGE option of REORG PLUS allows DB2 SQL applications to continue to use the affected objects in RW mode. This option is not intended to allow concurrency with data definition language (DDL) or other DB2 or third-party utilities.

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 or index space that you are reorganizing

- any of these utilities against any table or index space that contains a part that you are reorganizing

- an SQL UPDATE statement for the clustering or partitioning key of a partitioned table space that you are reorganizing

- an SQL DELETE statement without a WHERE clause (mass DELETE)

Object status for SHRLEVEL CHANGE

For information about the initial statuses that REORG PLUS requires, see “Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE” on page 82. For a discussion of the statuses that REORG PLUS changes during execution, see Table 17 on page 86.

Operational considerations for online reorganizations

Some REORG PLUS options are not available when you perform a SHRLEVEL CHANGE reorganization. In addition to the considerations described in this section, additional considerations apply to online reorganizations. For more information, see the following sections:

For information about	See
status requirements	“Object status for SHRLEVEL REFERENCE UNLOADONLY, SHRLEVEL REFERENCE, and SHRLEVEL CHANGE” on page 82 Table 17 on page 86
partial reorganization considerations	“Partial reorganization” on page 114
single-phase reorganization considerations	“Single-phase reorganization” on page 135
staging data sets	“Staging data sets” on page 98
authorizations	“Additional authorizations for SHRLEVEL CHANGE” on page 66
performance considerations	“Additional performance information for SHRLEVEL CHANGE options” on page 546 “Performance tuning for specific scenarios (SHRLEVEL CHANGE only)” on page 547

Interacting with applications

When you use SHRLEVEL CHANGE, 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. For more information, see “WTOMSG” on page 231.

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

In addition, 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 (LOGSPIL, SPILUNIT, SPILSCLS, and SPILDSNP). These installation options are shipped with the following default values:

- LOGSPIL=20000, 10000
- SPILUNIT=WORK
- SPILSCLS=NONE
- SPILDSNP=&&UID

For information about these options, see Appendix A, “REORG PLUS installation options.”

You can also override the values through options on the REORG PLUS command (LOGSPILL, SPILLUNIT, SPILLSTORCLAS, and SPILLDSNPAT). For the syntax of these command options, see “SHRLEVEL CHANGE options” on page 300.

The LOGMEM command or installation option tells REORG PLUS how much data space memory (in kilobytes) that it can use for storing the log records. REORG PLUS stores only the log records for the DB2 rows that have been updated since XBM took the snapshot of the object. When this memory overflows, REORG PLUS begins using the spill data sets.

The RIDMAPMEM command or RMAPMEM installation option tells REORG PLUS how much data space memory (in kilobytes) it can use for storing the RID maps. REORG PLUS uses the RID maps to translate the RIDs of the original objects to the RIDs of the reorganized objects. When this memory overflows, REORG PLUS begins using the spill data sets.

The LOGSPILL command or LOGSPIL installation option tells REORG PLUS how much disk space to allocate for the log record spill data sets. You can set both primary and secondary allocations. You can use the SPILLUNIT command or SPILUNIT installation option to specify the DASD where you want REORG PLUS to allocate the spill data sets. Alternatively, you can use the SPILLSTORCLAS command or SPILSCLS installation option if you have SMS storage allocation.

Using the SPILLDSNPAT command or SPILDSNP installation option, you can direct REORG PLUS to use a particular pattern of variables and text to create the prefixes for the spill data set names.

The spill data sets are VSAM files. You might need to take this fact into consideration when you specify the SPILLUNIT, SPILLSTORCLAS, or SPILLDSNPAT command options or their equivalent installation options.

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.

Statistics considerations

Statistics that follow message BMC50501I in the REORG PLUS SYSPRINT do not reflect activity that occurs during the LOGAPPLY phase.

Support for APPLICATION RESTART CONTROL (AR/CTL)

For batch applications that are running the BMC AR/CTL product, an online reorganization can use the AR/CTL suspend-and-resume interface. The SHRLEVEL CHANGE reorganization must obtain temporary exclusive access to the object that you are reorganizing. AR/CTL suspends batch DB2 processing to allow exclusive access, and resumes batch processing when exclusive access is no longer needed.

Requirements

The suspend-and-resume interface between a SHRLEVEL CHANGE reorganization and AR/CTL has the following requirements:

- You must have a license for AR/CTL for DB2.

- The CPU authorization password for REORG PLUS must be available to the BMC Consolidated Subsystem (BCSS).

- In the MVS subsystem that you use to run the batch DB2 application, the BCSS must be active and the Application Enhancement Series non-IMS™ component of the BCSS must be initialized.

- The batch program must use AR/CTL checkpoint/restart services.

Interface processing

The suspend-and-resume interface with the AR/CTL product works as follows:

1. Batch job steps run under the control of AR/CTL.

2. When REORG PLUS is initialized, it tells AR/CTL which table spaces and indexes are affected by the reorganization.

3. When an application program attempts an SQL access to a table that would normally receive an SQLCODE –911 or –904 because of REORG PLUS, AR/CTL performs suspend processing for the application.

4. When REORG PLUS completes the function that would cause the SQLCODE –911 or –904, it signals AR/CTL to resume processing on the application unit of work (UOW).

The following REORG PLUS functions can cause AR/CTL to suspend an application:

- all processing during the LOGFINAL phase
- rename processing during the UTILTERM phase

Implementing the interface

To implement the suspend-and-resume interface with AR/CTL, perform the following steps:

1. Install the AR/CTL product as documented in the *APPLICATION RESTART CONTROL Installation Guide*.

2. Install REORG PLUS and specify ARC=YES in the installation options module.

3. Make the CPU authorization password for REORG PLUS available to the BCSS, using one of the following methods:

   - Include the library that contains the password module in the AESPAUTH DD statement concatenation in the BCSS startup procedure

   - Copy the password module to a library that is already in the AESPAUTH DD concatenation.

4. Set up the reorganization job step.

5. In each batch program that you want AR/CTL to control, implement AR/CTL checkpoint/restart services as described in the AR/CTL documentation.

6. Run the reorganization job.

   For more information about AR/CTL, see the *APPLICATION RESTART CONTROL User Guide*.

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.”

Copy data sets for SHRLEVEL CHANGE

During the RELOAD and REORG phases of a SHRLEVEL CHANGE reorganization, REORG PLUS makes full copies of the table space objects, just as it does for SHRLEVEL NONE and SHRLEVEL REFERENCE. However, these copies are not registered in SYSIBM.SYSCOPY until the UTILTERM phase.

COPY YES REGISTER ALL is required for a SHRLEVEL CHANGE table space reorganization and will be set regardless of your specifications. REORG PLUS does not make image copies for an index-only reorganization. For more information, see “Copy data sets” on page 326.

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

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 94 on page 608.

WARNING

REORG PLUS does not support stacking incremental copy data sets on tape. If you choose to stack incremental copy data sets with the corresponding full data sets, unpredictable results can occur with subsequent recoveries.

Overriding the default ddnames

To override the default incremental copy ddnames or ddname prefixes in your installation options module, use the ICDDN (page 252) and RECOVERYICDDN (page 254) command options. For performance and tuning information when using these options, see “Incremental copy data sets” on page 534 and “Improving performance” on page 608.

Registration

The ddname itself controls the registration information that is placed in the ICBACKUP column of SYSIBM.SYSCOPY. REORG PLUS forces REGISTER ALL for a SHRLEVEL CHANGE reorganization.

Table 93 describes the REORG PLUS incremental copy data sets that, when registered, correspond to the DB2 local and remote copies. It also states when they are required.

Table 93 Corresponding incremental image copy types (part 1 of 2)

Image copy type	Corresponding REORG PLUS incremental copy data set (when registered)	When the data set is required
local primary	BMCICY or *ddname1* specified in the ICDDN command or installation option	always required if you specify COPY YES, unless you are dynamically allocating your copy data sets
local backup	BMCICZ or *ddname2* specified in the ICDDN command or installation option	optional

Table 93 Corresponding incremental image copy types (part 2 of 2)

Image copy type	Corresponding REORG PLUS incremental copy data set (when registered)	When the data set is required
remote primary	BMCIRY or *ddname1* specified in the RECOVERYICDDN command option or the RCVIDDN installation option	required only if a BMCIRZ copy is made
remote backup	BMCIRZ or *ddname2* specified in the RECOVERYICDDN command option or the RCVIDDN installation option	optional

Allocating incremental copy data sets

You can use one of the methods described in "Methods for allocating copy and work data sets" on page 325 to allocate your copy data sets and determine the appropriate size for those data sets. REORG PLUS determines the optimal block size of the 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:

- Reorganization jobs that invoke DSNUTILB do not produce incremental copies.
- When dynamically allocating copy data sets for partition-by-growth table spaces, REORG PLUS uses the DSSIZE value to estimate the size.

Dynamic allocation

Based on the information in Table 94 on page 608, specify dynamic allocation options that result in the same number and type of incremental copy data sets as if you specified DD statements in your JCL. For more information, see "Dynamic data set allocation" on page 88.

Allocating copy data sets in your JCL

If you do not use dynamic allocation, you must allocate the incremental copy data sets in your JCL by specifying DD statements as described in Table 94 on page 608. The table lists the incremental copy DD statements that REORG PLUS requires based on the table space and type of reorganization.

Table 94 Incremental copy DD statements required for SHRLEVEL CHANGE when allocating data sets in your JCL

Table space being copied	Type of reorganization	Incremental copy DD statements required
nonpartitioned	any	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.
partitioned, including partition-by-growth	full (entire table space)	■ Specify one DD statement for each copy type that you want REORG PLUS to make. Do not use *nn* in the ddname. *or* ■ For each partition, specify one DD statement for each copy type that you want REORG PLUS to make. Use *nn* for all ddnames, where *nn* matches the partition number.[a] If necessary for partition-by-growth table spaces, specify additional DD statements as discussed in “Partition-by-growth table spaces” on page 331.
	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.[b] *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.)[a] If necessary for partition-by-growth table spaces, specify additional DD statements as discussed in “Partition-by-growth table spaces” on page 331.

[a] For more information, see “Naming partition-level copies Use the nn as part of the ddname only if you are reorganizing a partitioned table space and you want a separate data set for each partition.” on page 330.

[b] The value of the COPYSUBSET installation option must be YES.

Improving performance

When reorganizing a partitioned table space, REORG PLUS multitasks the incremental image copy process whenever possible. This multitasking decreases the elapsed time needed to run the reorganization. To facilitate multitasking, provide a unique incremental image copy data set for each partition that you are reorganizing, and ensure that each image copy data set resides on DASD.

When running a full reorganization of a partitioned table space, you can specify how the dynamically allocated incremental copy data sets are assigned. However, REORG PLUS does not allow dynamically allocated copy data sets to be stacked on tape. For more information, see "COPYLVL" on page 246.

You can use the SIZEPCT option (page 285) 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

If an error occurs that prevents the registration of the copy data sets for a SHRLEVEL CHANGE reorganization, the setting of the CPYRFAIL installation option determines the action that REORG PLUS takes:

- If CPYRFAIL=TERM, REORG PLUS backs out any renames that it completed and terminates the reorganization. In this case, the original data sets, with no modifications, are available for use by DB2. The condition of your objects is the same as if the reorganization had never run.
- If CPYRFAIL=COPYPEND is specified, REORG PLUS sets the copy pending (COPY) status for the objects and starts them with their original status. You will need to make an image copy of the objects in order to allow applications to have RW access again.

  If you are using a DDLIN data set that alters the object that you are reorganizing, REORG PLUS changes CPYRFAIL=COPYPEND to CPYRFAIL=TERM at the beginning of the job. If the copy registration fails, REORG PLUS backs out the renames, backs out the ALTER statement, 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 job" on page 366.

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 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 process or the FASTSWITCH process
- removes pending statuses

- alters limit keys
- registers the reorganization and image copies
- updates the primary and secondary quantities in the DB2 catalog

If REORG PLUS cannot complete all of these operations, you might need to perform recovery actions. However, REORG PLUS is designed to be restartable in this case to complete these operations successfully.

The following considerations apply when restarting in the UTILTERM phase:

- If a failure occurs during processing of limit-key ALTER statements and the value of the ALTRFAIL installation option is TERM, REORG PLUS backs out the reorganization.

- 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 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.

NOTE

If a failure occurs during processing of limit-key ALTER statements (either in a DDLIN data set or as a result of the REBALANCE option), the value of the ALTRFAIL installation option determines the action that REORG PLUS takes.

If you cannot restart your job, you can manually complete the reorganization or back out to the starting point. For more information, see “Not completing in the UTILTERM phase” on page 370.

Performance considerations

REORG PLUS uses virtual memory in the address space in which it is running and additional data spaces to hold the log records and the RID translation map. The amount of memory that you make available to REORG PLUS affects the performance of your job. The sections that follow provide information about the memory requirements for the following items:

- the RID translation map
- log records
- spill data sets
- log data

As with other REORG PLUS jobs, BMC recommends that you specify REGION=0M in the JOB or EXEC statement of the execution JCL whenever you run SHRLEVEL CHANGE. If your data center constrains region sizes or the use of REGION=0M, specify a region size that will allow the most virtual memory, both above and below the 16-megabyte line. Using a region size that is less than optimal risks the following potential issues:

- running less efficiently, which could result in additional CPU and elapsed time
- encountering memory failures or jobs that fail when new versions implement changes that require additional memory

NOTE

If you specify a value for REGION other than 0M, ensure that you have an appropriate value set for the MEMLIMIT parameter, either as your site’s default SMF option or on your JOB statement or EXEC statement.

BMC makes the following recommendations for the MEMLIMIT option:

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

Sizing memory for the RID translation map

REORG PLUS uses the RID translation map to translate the RIDs of the original objects to the RIDs of the reorganized objects. The RID map memory is the memory that holds the completed RID translation map. REORG PLUS allocates this memory in one or more data spaces. If REORG PLUS cannot build the entire RID map in the memory that you specified on the RIDMAPMEM command option (or RMAPMEM installation option), REORG PLUS spills the remaining portion of the RID map to DASD. During processing, as it needs space, REORG PLUS determines the amount of DASD that it needs for the spill data sets and allocates them for you.

Allowing REORG PLUS to calculate memory

BMC strongly recommends that you specify 0 as the value for RIDMAPMEM (and RMAPMEM) to allow REORG PLUS to compute the amount of memory for the RID map automatically. To determine the value, REORG PLUS multiplies the value of the installation option RIDMMAXD by the value of the installation option RIDMDSSZ. The result is the maximum amount of memory that REORG PLUS can allocate. However, REORG PLUS uses only as much memory as needed to hold the RID map.

RIDMMAXD specifies the maximum number of data spaces, and RIDMDSSZ specifies the maximum size for each data space. For more information about these options, see Appendix A, “REORG PLUS installation options.”

Specifying the amount of memory

Although BMC recommends that you specify 0 for RIDMAPMEM (and RMAPMEM), you can specify a nonzero value. REORG PLUS first ensures that the value you supplied is sufficient to satisfy the minimum storage requirements. If the value is not sufficient, REORG PLUS issues message BMC50887I and changes the value for RIDMAPMEM to the minimum required storage amount.

During execution, REORG PLUS allocates memory only as needed, up to the current value of RIDMAPMEM. If RIDMAPMEM is insufficient, REORG PLUS issues message BMC50885I and spills to the spill data set.

To calculate the RIDMAPMEM value, use the formula described in “Using formulas for the calculations.” 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 = (.0078 * r) + 2048
```

However, if you have fewer than 20 rows per page, consider using Formula B for a more exact computation:

```
B = (.0132 * number of pages participating in the
reorganization) + (.0071 * r) + 2048
```

REORG PLUS displays the total amount of memory (data space memory and DASD) that it used in the STORAGE field in message BMC50880I (SUMMARY version of the message).

```
BMC50880I RIDMAP: SUMMARY: ROWS=r, PAIRS=p, STORAGE=sK, MEM
WAITS=mw, WAIT TIME=wt
```

Regardless of the amount that you specify for RIDMAPMEM or RMAPMEM, REORG PLUS never allocates more memory than the value of RIDMMAXD multiplied by the value in RIDMDSSZ.

Sizing memory for log records

The total memory that REORG PLUS uses for log records consists of

- the data spaces that you allocate with the LOGMEM command or installation option

- the DASD that you allocate with the LOGSPILL command option or 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 616. For guidelines for using the LOGMEM and LOGSPILL options together, see “Calculating storage requirements for log data” on page 617.

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."

Specifying the amount of memory

Although BMC recommends specifying 0 for LOGMEM, you can specify a nonzero value for the LOGMEM option to control the amount of memory that is available for the log records. When that memory becomes full, REORG PLUS spills the log records to the spill data sets and issues the following message:

```
BMC50869I LOG RECORD: SPILLING TO DATASET 'dataSetName'
```

At the end of the LOGAPPLY phase, the following message displays the amount of time that REORG PLUS waited to store log records:

```
BMC50860I TOTAL WAIT TIME FOR LOG RECORD STORE MEMORY = seconds
DATA = seconds INDEX = seconds
```

To determine an appropriate amount of memory to specify, use the formulas in the next section to assist you. If the total wait time shown in message BMC50860I indicates that REORG PLUS spent a significant amount of time waiting, increase the value of LOGMEM to improve performance.

Using formulas for the calculations

This section contains formulas that can help you calculate the LOGMEM value. These are also the formulas that REORG PLUS uses when you specify 0 for LOGMEM. Use Table 95 to help determine each number in the formula.

Table 95 Calculating memory for LOGMEM (part 1 of 2)

Reorganization type	Value to use for *p*	Value to use for *c*	Value to use for *n*
full table space	number of partitions that you are reorganizing	number of partitioned indexes times the number of partitions that you are reorganizing This number is the same as the number of table space partitions that you are reorganizing.	number of nonpartitioned indexes that are defined on the table space

Table 95 Calculating memory for LOGMEM (part 2 of 2)

Reorganization type	Value to use for *p*	Value to use for *c*	Value to use for *n*
partial table space without nonpartitioned indexes	number of partitions that you are reorganizing	number of partitioned indexes times the number of partitions that you are reorganizing	0
partial table space with nonpartitioned indexes	total number of partitions in the table space	number of partitioned indexes times the number of partitions that you are reorganizing	number of nonpartitioned indexes that are defined on the table space
full or partial index (partitioned index)	number of index partitions that you are reorganizing	number of index partitions that you are reorganizing	0
index (nonpartitioned index)	total number of partitions in the table space	0	1

To determine the *minimum* amount of data space to specify for LOGMEM, use formula A:

$$A = ((p + c + n) * 128) + 2048$$

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:

$$B = 20480 + (p * 1024) + ((c + n) * 512)$$

Sizing the spill data sets

The more updates the application makes, the more space REORG PLUS needs for the spill data sets. When the amount of space that REORG PLUS needs to store the RID maps is greater than the space that you specified on the RIDMAPMEM command or RMAPMEM installation option, REORG PLUS uses the spill data sets. Therefore, it is important to have enough DASD to support spilling. When the amount of space that REORG PLUS needs to store the log records is greater than the space that you specified on the LOGMEM command or installation option, REORG PLUS uses a different set of spill data sets.

For log records, REORG PLUS allocates the spill data sets by using the extent size that you specify as primary on the LOGSPILL command or LOGSPIL installation option. After the spill data set becomes full, REORG PLUS allocates and uses additional secondary extents until it reaches the VSAM extent limit, or until the data set reaches a size of 4 gigabytes (GB). Then, REORG PLUS allocates another data set, and so on, for a maximum of 256 data sets. The following message displays the amount of memory that REORG PLUS actually used for the log record spill data sets:

```
BMC50867I LOG RECORD STORE SPILL REQUESTS = n, HIGH SPILL PAGE
= n, SPILL DATASETS CREATED = n
```

For optimal performance, BMC recommends allocating relatively large primary and secondary allocations to reduce the number of extents that are created. For more information, see "Calculating storage requirements for log data."

For spill data sets for RID maps, REORG PLUS calculates the primary and secondary extent sizes for you. To find the amount of storage that REORG PLUS used for the spill data sets, subtract the value displayed for *USED MEM* in message BMC50882I from the value displayed for *STORAGE* in message BMC50880I (the SUMMARY version of the message). For information about the options for spill data sets, see "Allocation of spill data sets" on page 601.

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 = number of updates * row length * 3
```

Use formula B if the table space has more than two nonpartitioned indexes:

```
B = number of updates * row length * ((sum of key lengths for
all indexes / row length) + 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

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.

Appendix

A

REORG PLUS installation options

This appendix presents the following topics:

Overview

The REORG PLUS product is installed by using the BMC Software Installation System. During this installation, the customization process generates a customized installation data set. This data set contains customized jobs that install REORG PLUS into your specific DB2® environment. Two of these jobs establish the default processing option values that REORG PLUS uses:

- $C30DOPT establishes the defaults for REORG PLUS processing options.
- $C32SOPT contains options for the BMC Software BMCSORT technology.

These jobs assemble options macros. The macros contain the REORG PLUS processing options and the values for those options that are shipped with REORG PLUS and BMCSORT. When the Installation System-generated customization job is submitted, it links the ARU$OPTS installation options module in the APF-authorized library that is designated by your site.

You can customize the installation of REORG PLUS by changing the values for the REORG PLUS installation options. However, if you change any of the values in $C30DOPT or $C32SOPT after REORG PLUS has been installed, you must rerun the jobs for these changes to take effect.

You can also create additional options modules that allow you to use different values of these options for different executions of REORG PLUS. For example, you might use the default installation options module for most jobs but create another options module with customized values for certain options for special situations. For information about specifying an options module at runtime, see Chapter 4, “Building and executing REORG PLUS jobs.” For more information about customizing your installation of REORG PLUS, see the *Utility Products for DB2 Installation Guide*.

To find a description of each options macro, use Table 96.

Table 96 REORG PLUS installation macros

Job	Macro name	Description	Reference
$C30DOPT	$ARUOPTS	basic options	page 620
	$ARUDYNA	options for dynamic data set allocation	page 689
C32SOPT	$AUPSMAC	options for BMCSORT	page 708

Basic REORG PLUS installation options

Table 97 shows the options contained in the $ARUOPTS macro in $C30DOPT. For each option, the table provides the value that ships with this version of REORG PLUS (or lowercase *none* for no 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 97 Basic REORG PLUS installation options (part 1 of 6)

Option	Shipped value	Brief description	Reference
ACFORTSS	YES	whether or not to use CA ACF2 or CA Top Secret for security	page 626
ALTRFAIL	RCVRPEND	which action to take when a failure occurs during limit-key ALTER processing	page 626
ANALMAX	*	maximum number analyze tasks to start	page 626
ARC	NO	whether to use the AR/CTL interface	page 627
ARCHDDN	SYSARC	default ddname for the archive data set	page 627
AUXREORG	DEFAULT	whether to enable reorganizing LOB table spaces when reorganizing the base table space (using DSNUTILB processing)	page 628
AVAILPAGEPCT	100	percentage of available 4-KB pages reported by the system that REORG PLUS can allocate to BMCSORT for sort processing	page 629
BILDMAX	300%	maximum number of tasks to start for the index build process	page 629
BMCHIST	YES	whether to update the BMCHIST table	page 630

Table 97 Basic REORG PLUS installation options (part 2 of 6)

Option	Shipped value	Brief description	Reference
CBUFFS	30	controls the amount of buffer pool storage for each copy data set	page 630
CLONE	YES	whether to enable support for clone objects or base objects that participate in a clone relationship (using DSNUTILB processing)	page 631
CONDEXEC	NO	whether REORG PLUS should consider performing a conditional reorganization	page 631
COPYDDN	(BMCCPY,BMCCPZ)	default ddname or prefix for the local primary and secondary copy data sets	page 632
COPYLVL	PART	whether to assign full or partition-level copy data sets for partitioned table spaces	page 633
COPYMAX	*	maximum number of tasks to start during the image copy process	page 634
COPYSUBSET	NO	whether to allow a single copy of a subset of partitions	page 634
CPYRFAIL	TERM	which action to take if the image copies cannot be registered in SYSIBM.SYSCOPY	page 635
DATACAP	NO	whether to turn on the Data Capture Changes flag	page 636
DDLDDN	DDLIN	default ddname or prefix of the DDLIN data set	page 636
DEADLINE	NONE	time by which the LOGFINAL phase should finish applying log records	page 636
DELAY	1200	number of seconds that are to elapse from the time REORG PLUS detects a longlog condition until it performs the action specified on the LONGLOG option	page 637
DELFILES	YES	whether to delete data sets on completion	page 637
DESCCDE	(3,7)	descriptor codes to control how WTO messages are displayed on a console	page 638
DRAINTYP	ALL	type of drain that REORG PLUS issues when it enters the LOGFINAL phase, after reaching the MAXRO threshold	page 639
DRNDELAY	1	minimum number of seconds that REORG PLUS waits before it retries to obtain the drain after a drain times out	page 639
DRNRETRY	255	maximum number of times that REORG PLUS attempts to obtain a drain before it terminates the job	page 639
DRNWAIT	NONE	drain timeout value to use	page 640
DSNUEXIT	(NONE,ASM)	name of the user exit for creating dynamically allocated data set names and the language it is written in	page 641

Table 97 Basic REORG PLUS installation options (part 3 of 6)

Option	Shipped value	Brief description	Reference
DSNUTILB	YES	whether to use DSNUTILB to process the reorganization job when a feature requires DSNUTILB	page 642
DSPLOCKS	DRNFAIL	whether REORG PLUS displays claims and locks if a drain attempt times out	page 642
DSRSEXIT	(NONE,REXX)	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	page 643
EXCLDUMP	(X37,X22,X06)	system codes to ignore during the dump process	page 643
FASTSWITCH	NO	whether to bypass the VSAM rename process and directly update the DB2 catalog to use the staging data set names	page 644
FILECHK	FAIL	which action to take when encountering a temporary data set	page 645
HASHAX	YES	whether to enable reorganizing table spaces that contain tables defined as ORGANIZE BY HASH (using DSNUTILB processing)	page 645
ICDDN	(BMCICY,BMCICZ)	ddnames or ddname prefixes for the local copy data sets that receive an incremental image copy	page 646
ICTYPE	AUTO	whether to update the full image copy data sets or create incremental image copy data sets	page 646
IDCACHE	10000	cache size of document ID values	page 648
INDREFLM	10	limit over which REORG PLUS is to reorganize the specified table space	page 649
INLINECP	YES	whether to create an inline image copy	page 649
INLOB	YES	whether to enable support for inline LOB data (using DSNUTILB processing)	page 651
IXINCLCOL	YES	whether to enable support for unique indexes that contain non-key columns (using DSNUTILB processing)	page 651
IXONEX	NO	whether to enable support for indexes that contain a key derived from an expression (using DSNUTILB processing)	page 651
IXRANDOM	NO	whether to enable support for indexes that contain keys with random ordering (using DSNUTILB processing)	page 652
KEEPDICTIONARY	NO	whether to keep an existing compression dictionary or build a new one	page 652
LEAFDSLM	200	limit over which REORG PLUS is to reorganize the index named in the reorganization	page 653
LOB	YES	whether to enable reorganizing LOB data when SHRLEVEL NONE or SHRLEVEL CHANGE is in effect (using DSNUTILB processing)	page 654

Table 97 Basic REORG PLUS installation options (part 4 of 6)

Option	Shipped value	Brief description	Reference
LOCKROW	YES	serialization method for BMCSYNC and BMCUTIL	page 655
LOGFINAL	NONE	time by which LOGFINAL is to start	page 655
LOGMEM	0	amount of memory for log records	page 656
LOGSPIL	(20000,10000)	space allocation for spill data sets that hold the log records	page 657
LOGTHRSH	0	number of remaining log records	page 657
LONGLOG	CONTINUE	which action to take when a long log condition occurs	page 657
LONGNAMETRUNC	MIDDLE (or M)	long name truncation method for messages	page 658
MAXNEWPARTS	2	maximum number of partitions that can be added to partition-by-growth table spaces	page 658
MAXRO	300	maximum number of seconds that REORG PLUS spends applying log records	page 659
MAXSORTMEMORY	0	maximum amount of memory that REORG PLUS can allocate to each sort task	page 659
MAXTAPE	3	maximum number of tape devices to allocate dynamically	page 660
MGEXTENT	CONTINUE	how to allocate extents when extending to a new data set	page 660
MINSORTMEMORY	0	minimum amount of memory that REORG PLUS should allocate to each sort task	page 661
MSGLEVEL	1	default for the message level execution parameter	page 661
OFFPOSLM	10	OFFPOS limit for table space reorganizations	page 661
OPNDB2ID	YES	whether to use the user's RACF® ID instead of the DB2 RACF ID	page 662
ORIGDISP	DELETE	disposition of original data set during staging data set renaming process	page 663
PENDDDL	YES	whether to enable reorganizing table spaces and indexes that contain pending DDL changes (using DSNUTILB processing)	page 663
PLAN	ARU1010	product plan name	page 664
PREFORMAT	NO	whether to preformat unused data set pages	page 664
RCVICDDN	(BMCIRY,BMCIRZ)	ddnames or ddname prefixes for the remote copy data sets that receive an incremental image copy	page 664
RCVYDDN	(BMCRCY,BMCRCZ)	ddnames or ddname prefixes for the remote copy data sets that receive a full image copy	page 665
REDEFINE	YES	whether to delete and redefine the VSAM data sets for the table space or index space as part of the reorganization	page 665

Table 97 Basic REORG PLUS installation options (part 5 of 6)

Option	Shipped value	Brief description	Reference
RENMMAX	30	maximum number of tasks to start during the staging data set rename and delete process	page 666
RIDMDSSZ	2097152	maximum data space size for RID maps	page 666
RIDMMAXD	1	number of data spaces for RID maps	page 667
RMAPMEM	0	amount of memory for RID maps	page 667
RORGMAX	300%	maximum number of tasks to start for the REORG phase	page 668
ROUTCDE	(11,1)	WTO console routing codes	page 668
SCPYMAX	8	maximum number of tasks to start during the nonpartitioned index copy process	page 669
SDUMP	(ALLPSA,CSA,RGN, SQA,LSQA,SUM,TRT, IO)	system dump parameters	page 669
SHORTMEMORY	CONTINUE	which action to take when a memory shortage exists	page 670
SIXSNAP	NO	whether to use the Instant Snapshot technology or SUF of XBM to create a copy of storage-group-defined nonpartitioned indexes	page 671
SMAX	0	maximum number of sort processes to invoke concurrently	page 672
SMCORE	(0K,0K)	maximum amount of sort memory	page 673
SORTDEVT	(,SYSALLDA)	sort device types	page 673
SORTNUM	32	number of sort work files per sort task for BMCSORT to allocate dynamically	page 673
SPILDSNP	&&UID	pattern for spill data set name prefix	page 674
SPILSCLS	NONE	SMS storage class for spill data sets	page 676
SPILUNIT	WORK	DASD unit for spill data sets	page 676
SQLDELAY	3	number of seconds between retry attempts after SQLCODE –911	page 676
SQLRETRY	100	number of retry attempts after SQLCODE -911	page 677
STAGEDSN	BMC	naming convention to use for staging data sets	page 677
STOP@CMT	YES	whether to add the 'AT (COMMIT)' parameter to all DB2 STOP commands that REORG PLUS issues	page 677
STOPDELAY	1	number of seconds between checking to determine whether DB2 has stopped the object	page 678
STOPRETRY	300	number of times to check to determine whether DB2 has stopped the object	page 678
TAPEDISP	DELETE	tape file disposition	page 678
TASKMAX	1000%	global maximum number of tasks to start	page 678

Table 97 Basic REORG PLUS installation options (part 6 of 6)

Option	Shipped value	Brief description	Reference
TEMPRALDATA	YES	whether to enable reorganizing temporal tables or indexes on those tables (using DSNUTILB processing)	page 679
TERMEXIT	(NONE,REXX)	name of user-written TERM exit and the programming language in which the exit is written	page 679
TIMEOUT	TERM	which action to take when a drain attempt times out	page 680
TOTALPAGEPCT	0	percentage of total 4-KB pages reported by the system that REORG PLUS can allocate to BMCSORT for sort processing	page 681
TSPREC	YES	whether to enable reorganizing table spaces that contain timestamp columns defined with a precision other than 6 (using DSNUTILB processing)	page 682
TSSAMPLEPCT	100	percentage of sampling to perform during statistics gathering	page 682
TSTZ	YES	whether to enable reorganizing table spaces that contain columns defined as TIMESTAMP WITH TIME ZONE (using DSNUTILB processing)	page 683
UBUFFS	20	controls the amount of buffer pool storage for unload data sets	page 683
UNLDDN	SYSREC	default ddname or ddname prefix for the output data set that contains the unloaded rows to be reorganized	page 683
UNLDMAX	300%	maximum number of unload tasks to start	page 684
UNLOAD	RELOAD	whether to use single- or two-phase processing	page 684
UTSMEM	YES	whether to enable reorganizing universal table spaces that are defined as MEMBER CLUSTER (using DSNUTILB processing)	page 685
UXSTATE	SUP	how to invoke DB2 user exits	page 685
WBUFFS	(20,10)	controls the amount of buffer pool storage for each work data set	page 685
WORKDDN	SYSUT1	default ddname or ddname prefix for the index work data set	page 686
WORKUNIT	SYSALLDA	temporary unit for work data sets	page 686
XBMID	none example value: XBMA	XBM subsystem that REORG PLUS accesses when it uses XBM or SUF	page 686
XML	YES	whether to enable reorganizing table spaces that contain XML columns that support XML versions (using DSNUTILB processing)	page 687
ZIIP	ENABLED	whether to enable zIIP processing	page 688

This section describes each of the options. For more information about setting the values of some of these options at runtime, see Chapter 3, “Syntax of the REORG command.” For information about the performance implications of these options, see Chapter 6, “Tuning REORG PLUS jobs.”

ACFORTSS=YES

The ACFORTSS option specifies whether REORG PLUS should look for the presence of the CA ACF2 or CA Top Secret security product from Computer Associates when verifying utility authorizations. If neither is present, the option has no effect.

If you specify YES, REORG PLUS uses the applicable CA product, provided that security is enabled at the correct version on the subsystem where REORG PLUS is running.

You must be using a version of your security product that enables external security calls for DB2. If you have one of these security products installed but the version does not support external security, specify NO for this option.

For more information about this option, see “Authorization verification mechanisms” on page 64.

ALTRFAIL=RCVRPEND

This option applies to the UTILTERM phase of a SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganization only.

The ALTRFAIL option specifies the action that REORG PLUS takes when a failure occurs during processing of limit-key ALTER statements. This option applies to ALTER statements that are processed either from a DDLIN data set or as a result of the REBALANCE option. You can specify one of the following values:

- TERM tells REORG PLUS to terminate and back out the ALTERs and renames so that the original data sets are restored. The condition of your objects is the same as if the reorganization had never run.

- RCVRPEND tells REORG PLUS to leave the newly reorganized data sets in use, but set the RECP (RECOVER pending) or RBDP (REBUILD pending) status and end with RC=12.

ANALMAX=*

The ANALMAX option indicates the maximum number of tasks to start for the ANALYZE phase. You can use any of the following values:

- 0 tells REORG PLUS to determine the number of tasks to start.

- An asterisk (*) tells REORG PLUS to use the TASKMAX value (page 678).

- *n* tells REORG PLUS to start a maximum of *n* tasks (where *n* is a positive integer from 1 through 32767).

- *n*% tells REORG PLUS that the maximum number of tasks that it can start is *n*% of the number of CPUs on the system (where *n* is a positive integer from 1 through 32768).

For information about using the multitasking options, see “Multitasking installation options” on page 540.

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

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
- rebalancing of a table space that is LARGE (either by definition or default) or defined with DSSIZE, and where the last partition key has been altered

If you change the default ddnames that BMC supplied, you must also change the names in your JCL. For information about specifying and using this data set, see “SYSARC data set” on page 342.

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.

REORG PLUS ignores this option when invoking DSNUTILB to reorganize a LOB table space.

Overriding this option

You can override the value for this option by using the ARCHDDN command option (page 172).

AUXREORG=DEFAULT

The AUXREORG option specifies whether to reorganize (by invoking DSNUTILB) associated LOB table spaces while reorganizing the base table space. If the base table space is partitioned, this option applies only to the LOB table spaces associated with the partitions that are being reorganized.

You can specify one of the following options:

- DEFAULT tells REORG PLUS to use the IBM DB2 REORG utility defaults to determine whether to invoke DSNUTILB for this reorganization when AUX is not specified on the REORG command.

  The DB2 REORG utility defaults to AUX YES only when certain conditions exist. If those conditions do not exist and you do not specify AUX on the REORG command, REORG PLUS does not invoke DSNUTILB. In this case, REORG PLUS reorganizes the base table, but no associated LOB table spaces. For information about when the DB2 REORG utility defaults to AUX YES, see the documentation for the DB2 REORG utility.

  If this option defaults to AUX YES, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS to reorganize only the base table space. No associated LOB table spaces are reorganized.

  The DB2 REORG utility requires that AUX YES be in effect when certain conditions exist. REORG PLUS terminates if AUX NO is in effect (either by default when AUXREORG=NO, or when you specify AUX NO on the REORG command) and either of the following conditions exists:

  — Participating partitions are in REORP status.

  — You are reorganizing a range-partitioned table space and you specify REBALANCE.

Overriding this option

You can override the value for this option by using the AUX command option (page 233).

AVAILPAGEPCT=100

The AVAILPAGEPCT option controls virtual storage above the 16-MB line that REORG PLUS allocates to BMCSORT for concurrent sort processing. AVAILPAGEPCT specifies the maximum percentage of available memory, in 4-KB pages in the system, that REORG PLUS can allocate. You can specify any integer from 0 through 100.

NOTE

REORG PLUS defines *available pages* as pages that have not been used. *Total pages* (which you can control with the TOTALPAGEPCT option), are pages that are underutilized and are available for use.

- A value of 0 tells REORG PLUS to ignore the number of available pages when allocating sort memory.

- A value of 1 through 100 tells REORG PLUS to use up to the specified percentage of available pages when allocating sort memory. For example, AVAILPAGEPCT 50 tells REORG PLUS to use no more than 50 percent of the available pages.

Additional considerations

The following additional information applies to the AVAILPAGEPCT option:

- Because available pages are rarely subject to system paging, changing this value will have a minimal effect, if any, on system performance.

- When you specify values greater than 0 for both AVAILPAGEPCT and TOTALPAGEPCT, REORG PLUS uses the lesser of the two calculated results as the maximum amount of memory for sort processing.

- If REORG PLUS is unable to perform an optimal sort due to an insufficient number of available pages that it is enabled to allocate, the SHORTMEMORY option (page 181 and page 670) controls the action that REORG PLUS takes.

- When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option

You can override the value for this option by using the AVAILPAGEPCT command option (page 179).

BILDMAX=300%

The BILDMAX option indicates the maximum number of tasks to start for the index build process. You can specify any of the following values:

- 0 tells REORG PLUS to determine the number of tasks to start.

- An asterisk (*) tells REORG PLUS to use the TASKMAX value (page 678).
- *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 540.

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 an use the TERMEXIT option and user exit to dynamically control processing of updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS at execution time. For more information about the TERMEXIT installation option, see page 679. For more information about the TERMEXIT user exit, see "Using TERMEXIT to control BMCHIST and statistics updates" on page 795.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option

You can override the value for this option by using the BMCHIST command option (page 193).

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 533.

CLONE=YES

The CLONE option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) the following types of objects:

- a clone object
- a base object that participates in a clone relationship when the instance number of the base table space is 2

NOTE

When a base object participates in a clone relationship and the instance number of the base table space is 1, REORG PLUS reorganizes that object natively.

You can specify one of the following options:

- YES tells REORG PLUS to use DSNUTILB processing to reorganize the object. To enable this feature, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS not to reorganize the object. If REORG PLUS encounters one of these conditions, REORG PLUS issues message BMC50920E and terminates.

You cannot use this option to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively. For more information about running a DSNUTILB reorganization, see “Reorganization jobs that invoke DSNUTILB” on page 70.

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 128. For more information about conditional execution using the DASD MANAGER PLUS tables, see “Using the DASD MANAGER PLUS exceptions table” on page 130.

Valid values are YES, NO, and BMC:

- NO tells REORG PLUS to ignore the limit installation options (OFFPOSLM, INDREFLM, and LEAFDSLM) and perform the reorganization unless you override this option with the CONDEXEC or limit command options.

- YES tells REORG PLUS to examine the values of the limit command options (if specified) and the limit installation options in conjunction with the values in the DB2 catalog to determine whether to perform the reorganization, and to issue a report with the information.

- BMC tells REORG PLUS to issue a report and reorganize the object of the current job if an exception for the object exists in the DASD MANAGER PLUS exceptions table. REORG PLUS issues one of the following return codes after examining the DASD MANAGER PLUS exceptions table:

  — 1, which indicates that no calculated value exceeded an exception value. REORG PLUS does not perform a reorganization.

  — 2, which indicates that a calculated value exceeded an exception value. REORG PLUS performs a reorganization. If REORG PLUS issues any return code greater than 2 during processing, that return code supersedes return code 2.

Restrictions

The following restrictions apply to CONDEXEC:

- CONDEXEC=BMC is available only if you are using REORG PLUS as a component of the Database Performance for DB2 solution.

- If CONDEXEC=BMC or CONDEXEC=YES is in effect, REORG PLUS terminates if you are reorganizing a LOB table space.

Additional considerations

The following additional considerations apply to CONDEXEC:

- When invoking DSNUTILB, REORG PLUS ignores CONDEXEC=BMC.

- You can obtain the report that recommends objects for reorganization without performing any reorganizations. To do so, specify REPORTONLY in the command.

Overriding this option

You can override the value for this option by using the CONDEXEC command option (page 210).

COPYDDN=(BMCCPY, BMCCPZ)

The COPYDDN option specifies default ddnames or ddname prefixes for the local copy data sets that receive an image copy or DSN1COPY-type copy of the table space or partitions that you are reorganizing. If you are registering the copies, the first name is the local primary, and the second name is the local backup.

When using this name as a prefix for a ddname for a partitioned table space, ensure that you allow sufficient bytes for the partition number to be added and still have a valid ddname (eight bytes or less). In addition, if you change the default ddnames that BMC supplied, you must also change the names in your JCL. For more information, see “Copy data sets” on page 326.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

Overriding this option

You can override the value for this option by using the COPYDDN command option (page 248).

COPYLVL=PART

The COPYLVL option specifies how REORG PLUS is to assign copy data sets when reorganizing partitioned data sets. You can specify one of the following values:

- COPYLVL=PART tells REORG PLUS to allocate individual full copy data sets and incremental copy data sets (if created) for each partition that you are reorganizing. If you also specify a tape device for the UNIT option, REORG PLUS allocates a tape unit for each partition.

  REORG PLUS appends the partition number to the ddname prefixes that you specify in the COPYDDN (page 632), ICDDN (page 646), RCVYDDN (page 665), and RCVICDDN (page 664) options. The values that you specify for these options plus the highest partition number must not exceed eight characters. For more information, see "Specifying ddname prefixes" on page 91.

  COPYLVL=PART is not valid when REORG PLUS invokes DSNUTILB (and you specify COPY YES). If this value is in effect at runtime, REORG PLUS converts it to COPYLVL FULL and DSNUTILB creates a full image copy.

- COPYLVL=FULL tells REORG PLUS to assign a single full copy data set to contain all of the partitions that you are reorganizing in either of the following cases:

  — You are reorganizing all partitions.

  — You are reorganizing a subset of contiguous partitions (specified on the PART command option) and you specify a value of YES for the COPYSUBSET installation option (page 634).

  If incremental copy data sets are created, REORG PLUS also allocates a single incremental copy data set to contain all of the partitions that you are reorganizing.

Additional considerations

The following considerations apply to the COPYLVL option:

- If you change ICTYPE in the options module to UPDATE, leave the COPYLVL installation option set to FULL, and do not specify either option on the command, REORG PLUS changes COPYLVL to PART to prevent the job from terminating.

- If you anticipate reorganizing a large number of partitions, consider specifying COPYLVL FULL. For more information, see "Large number of partitions" on page 105.

Overriding this option

You can override the value for this option by using the COPYLVL command option (page 246).

COPYMAX=*

The COPYMAX option indicates the maximum number of tasks to start during the image copy process. You can specify any of the following values:

- 0 to have REORG PLUS determine the number of tasks to start
- asterisk (*) to use the TASKMAX value (page 678)
- *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 540.

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.

NOTE

If you specify COPYSUBSET=YES, you might need to change the command syntax in existing REORG PLUS jobs to avoid syntax errors. For example, you might have a job that contains the following options on the REORG command:

```
SHRLEVEL CHANGE
PART 1:10
COPY YES COPYLVL FULL INLINE NO ICTYPE UPDATE
```

In versions earlier than 9.3, REORG PLUS changed COPYLVL FULL to COPYLVL PART in this case because ICTYPE UPDATE is not valid with INLINE NO when copying multiple partitions to a single copy data set. In version 9.3 or later, REORG PLUS still changes this to COPYLVL PART when COPYSUBSET is NO. However, if you specify COPYSUBSET=YES, REORG PLUS terminates with a syntax error.

BMC recommends that you set the INLINE command option or INLINECP installation option to YES to avoid this problem.

CPYRFAIL=TERM

This option applies to the UTILTERM phase of a SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganization only.

The CPYRFAIL option specifies the action that REORG PLUS takes if the image copies cannot be registered in SYSIBM.SYSCOPY:

- CPYRFAIL=TERM tells REORG PLUS to terminate and back out the renames, if necessary, so that the original data sets are restored. The condition of your objects is the same as if the reorganization had never run.
- CPYRFAIL=COPYPEND tells REORG PLUS to leave the newly reorganized data sets in use, but set the COPY (copy pending) status and end with RC=4.

  If you specify CPYRFAIL=COPYPEND and are using a DDLIN data set to alter limit key values, REORG PLUS changes the value to TERM for this job and continues terminating the job. 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.

- DATACAP=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.

- DATACAP=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 333.

Overriding this option

You can also specify the DDLIN data set name by using the DDLDDN command option (page 172).

DEADLINE=NONE

This option applies to SHRLEVEL CHANGE only.

The DEADLINE option specifies the time by which the LOGFINAL phase should finish applying log records. You can specify NONE or a time:

- NONE indicates no deadline.
- You can specify a time in *hh:mm:ss* format as the deadline, using a 24-hour clock. REORG PLUS replaces the time portion of the current system timestamp with the specified time to create a new timestamp. If the calculated timestamp time has already passed, REORG PLUS updates the date to the next day. Then, it uses this new timestamp to determine when the LOGFINAL phase should stop applying log records.

Additional considerations

The following considerations apply to the DEADLINE option:

- If REORG PLUS calculates timestamps for LOGFINAL and DEADLINE, the DEADLINE timestamp must be later than the LOGFINAL timestamp.
- You can use the XBM Utility Monitor function or the MVS™ operator console to dynamically change the DEADLINE values while the reorganization is in progress.
- The DEADLINE option interacts with other options to control the log apply process. For information, see "Control of the log apply process" on page 586.

Overriding this option

You can override the value for this option by using the DEADLINE command option (page 301).

DELAY=1200

This option applies to SHRLEVEL CHANGE only.

The DELAY option specifies the number of seconds, as a positive integer (0 or greater), that are to elapse from the time REORG PLUS detects a longlog condition until the time it performs the action specified on the LONGLOG option. If the longlog condition no longer exists at the end of the time period specified by DELAY, the DELAY value is reset.

During the window provided by the DELAY option, you can use the XBM Utility Monitor or the MVS operator console to terminate the reorganization or change the criteria that determines when LOGFINAL processing begins.

This option interacts with other options to control the log apply process. For information, see "Control of the log apply process" on page 586.

Overriding this option

You can override the value for this option by using the DELAY command option (page 304).

DELFILES=YES

The DELFILES option tells REORG PLUS whether to delete your work files after the reorganization.

- If you specify DELFILES=YES, the response from REORG PLUS depends on whether the reorganization completes successfully:

  — If a non-DSNUTILB reorganization completes successfully, DELFILES=YES tells REORG PLUS to delete all physical sequential data sets whose ddnames match the SYSREC, SYSUT1, and SORTWK ddnames or ddname prefixes.

- — For successful DSNUTILB reorganizations, REORG PLUS deletes the following data sets:
  - the SYSREC and SYSUT1 data sets that were used during the reorganization
  - all SYSREC, SYSUT1, and SORTWK data sets allocated in your JCL

- — If a non-DSNUTILB reorganization completes unsuccessfully, DELFILES=YES tells REORG PLUS to delete any unregistered full and incremental copy data sets (as well as the SYSREC, SYSUT1, and SORTWK data sets) when either of the following conditions exists:
  - You specify ON FAILURE *phase* TERMINATE UTILITY, where *phase* is UNLOAD, RELOAD, LOGAPPLY, or UTILTERM, and REORG PLUS terminates in the phase that you specified.
  - A drain timeout occurs and you specified TIMEOUT TERM as a command or installation option.

  In contrast, REORG PLUS does *not* delete the data sets if

  - You did not specify ON FAILURE or TIMEOUT TERM.
  - REORG PLUS ends with a STOP status.

- — If a DSNUTILB reorganization completes unsuccessfully, REORG PLUS does not perform DELETEFILES processing.

- DELFILES=NO tells REORG PLUS not to delete any files after either a successful or unsuccessful reorganization.

If you are running the reorganization job in a worklist environment, REORG PLUS ignores the value that you set in the installation option and processes the job as if you had specified DELFILES=NO. REORG PLUS does this so that the data sets will exist for subsequent executions in the job. If you want to delete the data sets, you must specify DELETEFILES YES on the REORG command in the worklist.

Overriding this option

You can override the value for this option by using the DELETEFILES command option (page 200).

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 (page 314).

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.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the RETRY_DELAY command option (page 225).

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.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option

You can override the value for this option by using the RETRY command option (page 225).

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)

DSNUTILB reorganizations

When invoking DSNUTILB, REORG PLUS converts the DRNWAIT option based on the DSNZPARM settings, and passes this option to the IBM DB2 REORG utility as DRNWAIT=*integer*, where *integer* is defined as follows:

- For DRNWAIT=UTIL, *integer* equals the product of SPRTMTOUT multiplied by SPRMUTO.

- For DRNWAIT=NONE, *integer* equals 1 (one second).

- For DRNWAIT=SQL, *integer* equals SPRTMTOU1T.

Overriding this option

You can override the value for this option by using the DRAIN_WAIT command option (page 224).

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 (page 289) or SPILLDSNPAT (page 305) command option.

The possible values for the language are

- ASM
- COBOL2
- LE_COBOL
- C
- LE_C

When invoking DSNUTILB, REORG PLUS ignores this option.

For more information about the DSNUEXIT user exit, see “Using DSNUEXIT to construct data set name patterns” on page 735.

Overriding this option

You can override the value for this option by using the DSNUEXIT command option (page 221).

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 70.

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 70.

Overriding this option

You can override the value for this option by using the DSNUTILB command option (page 217).

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.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the DSPLOCKS command option (page 226).

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 "Using DSRSEXIT to manage VSAM data set redefinition" on page 780.

Overriding this option
You can override the value for this option by using the DSRSEXIT command option (page 220).

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.

You can specify one of the following values:

- NO tells REORG PLUS to rename the staging data sets to the original data set names. This value can help to prevent potential contention issues between the DB2 catalog and directory.

- YES tells REORG PLUS to bypass the VSAM rename process and directly update the DB2 catalog to use the staging data set names.

  Consider the following information before using the FASTSWITCH process:

  — 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.

  — When reorganizing clone objects, REORG PLUS changes FASTSWITCH=YES to FASTSWITCH=NO.

- 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 the data set naming methods, see "Staging data sets and the rename process" on page 100. For more information about FASTSWITCH processing, see "Staging data sets and the FASTSWITCH process" on page 102.

Overriding this option
You can override the value for this option by using the FASTSWITCH command option (page 226).

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 94.

- 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.

When invoking DSNUTILB, REORG PLUS ignores this option.

HASHAX=YES

The HASHAX option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) a table space that contains a table that is defined as ORGANIZE BY HASH:

- YES tells REORG PLUS to reorganize this table space by using DSNUTILB processing. To enable this feature, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS not to reorganize this table space. REORG PLUS terminates.

NOTE

REORG PLUS natively reorganizes indexes on tables that are defined as ORGANIZE BY HASH.

You cannot use this option to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively. For more information about running a DSNUTILB job, see “Reorganization jobs that invoke DSNUTILB” on page 70.

ICDDN=(BMCICY, BMCICZ)

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 Software supplied, you must also change the names in your JCL. For more information, see “Incremental copy data sets” on page 605.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option

You can override the value for this option by using the ICDDN command option (page 252).

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 the 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 (page 244).

IDCACHE=10000

The IDCACHE option allows you to specify the size of the cache that REORG PLUS uses when populating document ID columns. Specify the size as the number of values to reserve. REORG PLUS reserves this cache in SYSIBM.SYSSEQUENCES and reserves at least one cache for each unload and log apply task that requires it. The valid values for this option are 1 through 2147483647.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option

You can override the value for this option by using the IDCACHE command option (page 232).

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.

```
(NEARINDREF + FARINDREF) * 100 / CARDF=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 (page 213). 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 128.

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 exist for a particular reorganization, REORG PLUS behaves as if you specified NO, regardless of the value specified 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.

  If all of the following conditions exist with INLINECP=NO, REORG PLUS terminates:

  — You specify ICTYPE UPDATE.

  — You attempt to create a single image copy for multiple partitions, as in the following scenarios:

    - You allocate a single copy data set in your JCL but specify multiple partitions.
    - You specify COPYLVL FULL and COPYSUBSET=YES.

Overriding this option

You can override the value for this option by using the INLINE command option (page 242).

INLOB=YES

The INLOB option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) a table space that contains an inline LOB column:

- YES tells REORG PLUS to reorganize the table space by using DSNUTILB processing. To enable this feature, DSNUTILB YES must also be in effect.

  NOTE

  If the table space that you are reorganizing contains an index on an inline LOB column, you must also specify IXONEX=YES.

- NO tells REORG PLUS not to reorganize the table space. REORG PLUS terminates.

For more information about running a DSNUTILB job, see “Reorganization jobs that invoke DSNUTILB” on page 70.

IXINCLCOL=YES

The IXINCLCOL option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) an index that contains non-key columns or a table space that contains this type of index:

- YES tells REORG PLUS to use DSNUTILB processing to reorganize this type of index or a table space that contains this type of index. To enable this feature, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS not to reorganize this type of index or a table space that contains this type of index. If REORG PLUS encounters this type of index, REORG PLUS terminates.

For more information about running a DSNUTILB job, see “Reorganization jobs that invoke DSNUTILB” on page 70.

IXONEX=NO

The IXONEX option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) indexes that contains a key derived from an expression, including an index on an inline LOB column, or table spaces that contain this type of index.

NOTE

REORG PLUS natively reorganizes indexes defined on table spaces that contain inline LOB data when the index does not contain a key derived from an expression on an inline LOB column.

You can specify one of the following options:

- YES tells REORG PLUS to use DSNUTILB processing to reorganize this type of index or a table space that contains this type of index. To enable this feature, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS not to reorganize this type of index or a table space that contains this type of index. If REORG PLUS encounters this type of index, REORG PLUS issues message BMC50920E and terminates.

For more information about running a DSNUTILB reorganization, see "Reorganization jobs that invoke DSNUTILB" on page 70.

IXRANDOM=NO

The IXRANDOM option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) indexes that contains a key with random ordering or table spaces that contain this type of index:

- YES tells REORG PLUS to use DSNUTILB processing to reorganize this type of index or a table space that contains this type of index. To enable this feature, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS not to reorganize this type of index or a table space that contains this type of index. If REORG PLUS encounters this type of index, REORG PLUS issues message BMC50920E and terminates.

For more information about running a DSNUTILB reorganization, see "Reorganization jobs that invoke DSNUTILB" on page 70.

KEEPDICTIONARY=NO

This option applies to a table space reorganization only.

The KEEPDICTIONARY option tells REORG PLUS whether to keep the existing compression dictionary. For more information about how REORG PLUS handles compression dictionaries, see "Table space compression" on page 135.

- 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 (page 191).

LEAFDSLM=200

This option applies to index-only reorganizations.

If the value of the CONDEXEC command or installation option is YES, REORG PLUS issues a report recommending which objects in a table space should be reorganized. REORG PLUS then uses the value in the LEAFDSLM option to conditionally perform the reorganization. Valid values are any positive integer (0 or greater) or NONE.

For the partitions that you are reorganizing for the specified index, REORG PLUS compares the specified LEAFDSLM value to the value of LEAFDIST in SYSIBM.SYSINDEXPART. If any LEAFDIST value exceeds the LEAFDSLM value, REORG PLUS reorganizes the index.

If the value of the CONDEXEC command or installation option is YES and you specify NONE in the LEAFDSLM installation option, REORG PLUS performs a conditional reorganization based on the values of the other limit options but not the value of LEAFDSLM.

Overriding this option

You can override the value for this option by using the LEAFDISTLIMIT command option (page 214). 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 128.

TIP

You can obtain the report that recommends objects for reorganization *without* performing any reorganizations. To do so, specify REPORTONLY in the command.

LOB=YES

The LOB option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) LOB table spaces. This option applies only when SHRLEVEL NONE or SHRLEVEL CHANGE is in effect.

NOTE

If SHRLEVEL REFERENCE is in effect, REORG PLUS reorganizes LOB table spaces and updates associated auxiliary indexes without invoking DSNUTILB. REORG PLUS ignores the LOB option in this case. For more information, see "LOB data" on page 110.

You can specify one of the following options:

- YES tells REORG PLUS to invoke DSNUTILB to reorganize LOB table spaces. SHRLEVEL NONE or SHRLEVEL CHANGE must be in effect, and the DSNUTILB option must be YES.

- NO tells REORG PLUS to not reorganize LOB table spaces when SHRLEVEL NONE or SHRLEVEL CHANGE is in effect.

You cannot use this option to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively. For more information about running a DSNUTILB reorganization, see "Reorganization jobs that invoke DSNUTILB" on page 70.

LOCKROW=YES

The LOCKROW option tells REORG PLUS which serialization method to use when updating the BMCSYNC and BMCUTIL tables:

- YES tells REORG PLUS to use MVS enqueues instead of the SQL LOCK TABLE statements for serialization. Using LOCKROW=YES should prevent most SQL –911 return codes that occur when multiple BMC products concurrently update the BMCSYNC and BMCUTIL tables.

- NO tells REORG PLUS to use the SQL LOCK TABLE statements for serialization when updating the BMCSYNC and BMCUTIL tables.

Additional considerations

The following considerations apply to the LOCKROW option:

- Using LOCKROW=YES requires row-level locking. You must define the BMCSYNC and BMCUTIL tables with LOCKSIZE ROW to achieve row-level locking.

- The following BMC products must use the same LOCKROW value within a subsystem: CHECK PLUS, COPY PLUS, LOADPLUS, RECOVER PLUS, REORG PLUS, and UNLOAD PLUS. Regardless of the value assigned for LOCKROW, COPY PLUS and RECOVER PLUS always behave as if LOCKROW=YES.

- In a future version, REORG PLUS will no longer require or support the LOCKROW installation option.

LOGFINAL=NONE

This option applies to SHRLEVEL CHANGE only.

The LOGFINAL option specifies when you want REORG PLUS to start the LOGFINAL phase. Through this option, you can control when updates are prevented to the table space or index space that you are reorganizing. You can specify NONE or a time:

- NONE indicates no set time, and the other log apply options will control the process. For information, see “Control of the log apply process” on page 586.

- You can specify a time in *hh:mm:ss* format as the value, using a 24-hour clock. REORG PLUS replaces the time portion of the current system timestamp with the time specified to create a new timestamp. If the calculated timestamp time has already passed, REORG PLUS updates the date to the next day. Then, it uses this new timestamp to determine when it will begin the LOGFINAL phase.

Additional considerations

The following considerations apply to the LOGFINAL option:

- REORG PLUS does not check the LOGFINAL value until the LOGAPPLY phase begins. At that point, if the current time is greater than the time specified on the LOGFINAL option, REORG PLUS immediately begins the LOGFINAL phase.
- You can use the XBM Utility Monitor function or the MVS operator console to dynamically change LOGFINAL values while the reorganization is in progress.
- When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option

You can override the value for this option by using the LOGFINAL command option (page 312).

LOGMEM=0

This option applies to SHRLEVEL CHANGE only.

The LOGMEM option specifies the amount of data space memory (in kilobytes) needed to hold the internal structures that REORG PLUS uses to store the log records (data and index components). The number must be 0 or a positive integer.

BMC strongly recommends that you use the value 0. A value of 0 tells REORG PLUS to automatically compute the amount of data space memory that it needs. To do so, REORG PLUS uses the formula described in “Sizing memory for log records” on page 614.

If you specify a nonzero value for LOGMEM, during execution REORG PLUS first ensures that your specified value is sufficient to satisfy the minimum memory requirements. If the value is not sufficient, REORG PLUS issues message BMC50864I and changes the value for LOGMEM to the minimum required memory amount.

If you specify a nonzero positive integer, REORG PLUS allocates up to that amount of data space memory.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option

You can override the value for this option by using the LOGMEM command option (page 311).

LOGSPIL=(20000,10000)

This option applies to SHRLEVEL CHANGE only.

The LOGSPIL option specifies the primary and secondary space allocation (in kilobytes) for the spill data sets that hold the log records. The number must be a nonzero positive integer. REORG PLUS uses these data sets when the memory allocated with LOGMEM becomes full.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the LOGSPILL command option (page 312).

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" on page 586.

Overriding this option
You can override the value for this option by using the LOGTHRESHLD command option (page 300).

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:

- LONGLOG=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.

- LONGLOG=TERM tells REORG PLUS to terminate the reorganization.

- LONGLOG=DRAIN tells REORG PLUS to begin the LOGFINAL phase, prevent updates to the objects that you are reorganizing, and apply the remaining log records.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 586.

Overriding this option
You can override the value for this option by using the LONGLOG command option (page 303).

LONGNAMETRUNC=MIDDLE

LONGNAMETRUNC tells REORG PLUS where to truncate names that are longer than the area that is available in a REORG PLUS report-style message. Table 98 describes the valid values and the symbol that appears in the message based on that value.

Table 98 LONGNAMETRUNC option values

Value	Description	Symbol
MIDDLE (or M)	truncates outward from the middle of the name	><
BEGINNING (or B)	truncates from the beginning of the name	<<
END (or E)	truncates from the end of the name	>>

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the LONGNAMETRUNC command option (page 166).

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” on page 326 and “SYSREC data sets” on page 349.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the MAXNEWPARTS command option (page 164).

MAXRO=300

This option applies to SHRLEVEL CHANGE only.

The MAXRO option specifies the maximum number of seconds that REORG PLUS spends applying log records in the LOGFINAL phase. This estimate does not include the time spent producing incremental image copies and running the UTILTERM phase. Specify the value as a positive integer (0 or greater). When the estimated number of seconds required to apply the remaining log records is less than this value, REORG PLUS ends the LOGAPPLY phase and begins the LOGFINAL phase.

Alternatively, you can specify MAXRO=DEFER, which tells REORG PLUS to continue applying log records indefinitely. The LOGFINAL phase will not begin unless one of the following conditions occurs:

- You specify a value for the DEADLINE option and the value is reached.
- You change DEFER to a number of seconds using the XBM Utility Monitor or the MVS operator console.

This option interacts with other options to control the log apply process. For information, see “Control of the log apply process” on page 586.

Overriding this option
You can override the value for this option by using the MAXRO command option (page 300).

MAXSORTMEMORY=0

The MAXSORTMEMORY option specifies the maximum amount of memory, in kilobytes, that REORG PLUS can allocate to each sort task. The shipped value is 0, which tells REORG PLUS to automatically compute the maximum amount of memory that is needed to perform each sort task. BMCSORT might increase the amount of memory that REORG PLUS allocates if necessary, to successfully complete the sort (if additional memory is available).

In addition to 0, you can specify any number of kilobytes between the value that you specify for the MINSORTMEMORY installation or command option and 2097152 (see page 183 or page 661).

NOTE

BMC recommends that you use a value of 0.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the MAXSORTMEMORY command option (page 183).

MAXTAPE=3

The MAXTAPE option specifies the maximum number of tape devices to dynamically allocate at one time. The value must be greater than 0.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the MAXTAPE command option (page 269).

MGEXTENT=CONTINUE

The MGEXTENT option specifies how to allocate the extents of a DB2 object when creating a new data set during extend processing. Specify one of the following options:

- CONTINUE tells REORG PLUS to allocate the extents as follows:

  — Allocate the primary extent with the greater of the original primary quantity or the last secondary quantity of the previous data set.

  — Allocate the first secondary extent with the last secondary quantity of the previous data set.

- RESET tells REORG PLUS to allocate the primary and first secondary extents by using the original values from the DB2 object allocation.

With the exception of this option, REORG PLUS allocates secondary extents in the same way that DB2 does. REORG PLUS uses a sliding-scale calculation for secondary extents, if applicable, that is similar to the method that DB2 uses. For more information, see the DB2 documentation that describes the sliding-scale algorithm.

MINSORTMEMORY=0

The MINSORTMEMORY option specifies the minimum amount of memory, in kilobytes, that REORG PLUS should allocate to each sort task. The shipped and BMC-recommended value is 0, which tell REORG PLUS to automatically compute the minimum amount of memory that is needed to optimally perform each sort task. In addition to 0, you can specify any number of kilobytes between 1024 and the value that you specify for the MAXSORTMEMORY installation or command option (page 183 and page 659).

For information about how this option interacts with the SMCORE installation option, see SMCORE on page 673.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the MINSORTMEMORY command option (page 183).

MSGLEVEL=1

The MSGLEVEL option controls which messages are returned to the user in the SYSPRINT data set. MSGLEVEL=0 returns minimal messages. MSGLEVEL=1 returns additional messages to help you diagnose problems and fine-tune performance.

Overriding this option
You can override the value for this option by using the MSGLEVEL parameter on the EXEC statement (page 321).

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.

```
(NEAROFFPOSF + FAROFFPOSF) * 100 / CARDF=integer
```

If any calculated integer value exceeds the OFFPOSLM value, REORG PLUS reorganizes the object.

If the value of the CONDEXEC command or installation option is YES and you specify NONE in the OFFPOSLM installation option, REORG PLUS performs a conditional reorganization based on the values of the other limit options but not the value of OFFPOSLM.

Overriding this option

You can override the value for this option by using the OFFPOSLIMIT command option (page 211). 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 128.

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.

- OPNDB2ID=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.
- OPNDB2ID=NO tells REORG PLUS to use the RACF ID of the user running REORG PLUS. If you specify NO, the user must have the appropriate RACF authority.

For any security system other than RACF, specify OPNDB2ID=NO to have REORG PLUS use the security authorization ID of the user who is running REORG PLUS.

For more information, see "Required authorization" on page 64.

TIP

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

ORIGDISP=DELETE

The ORIGDISP option specifies the action that REORG PLUS should take after it has successfully renamed the staging data sets and completed a SHRLEVEL REFERENCE or SHRLEVEL CHANGE reorganization:

- ORIGDISP=DELETE tells REORG PLUS to delete the original data sets.
- ORIGDISP=RENAME tells REORG PLUS to rename the original data sets to the staging data set names.

  ORIGDISP=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 either of the following conditions exists:

  — You specify YES for the FASTSWITCH option.

  — You specify ZPARM for the FASTSWITCH installation option, and the DB2 ZPARM value is YES.

  In these cases, 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 98.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the ORIGINALDISP command option (page 202).

PENDDDL=YES

The PENDDDL option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces or indexes that contain pending DDL changes. If the IBM DB2 REORG utility would not materialize the pending changes, REORG PLUS ignores this option and reorganizes the object natively. For information about the conditions under which the DB2 REORG utility would not materialize the pending changes, see the documentation for the DB2 REORG utilities.

You can specify one of the following options:

- YES tells REORG PLUS to invoke DSNUTILB when the following conditions exist:

  — REORG PLUS encounters a table space or index that contains pending DDL changes.

  — The DB2 REORG utility would materialize the changes.

  To enable this feature, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS not to invoke DSNUTILB. REORG PLUS terminates.

For more information about running a DSNUTILB job, see “Reorganization jobs that invoke DSNUTILB” on page 70.

PLAN=ARU1010

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.

- PREFORMAT=NO tells REORG PLUS not to preformat unused pages in a data set.
- PREFORMAT=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.

When invoking DSNUTILB, REORG PLUS passes PREFORMAT=YES to the IBM DB2 REORG utility as PREFORMAT and ignores PREFORMAT=NO.

Overriding this option
You can override the value for this option by using the PREFORMAT command option (page 208).

RCVICDDN=(BMCIRY, BMCIRZ)

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" on page 605.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option

You can override the value for this option by using the RECOVERYICDDN command option (page 254).

RCVYDDN=(BMCRCY, BMCRCZ)

The RCVYDDN option specifies default ddnames or ddname prefixes for remote copy data sets. These are the data sets that receive an image copy or a DSN1COPY-type copy of the table space or partitions that you are reorganizing. If you are registering the copies, the first name is the remote primary copy, and the second name is the remote backup copy.

When using this name as a prefix, ensure that you allow sufficient bytes for the partition number to be added and still have a valid ddname (eight bytes or less). In addition, if you change the default ddnames, you must change the names in your JCL. For more information, see "Copy data sets" on page 326.

When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

Overriding this option

You can override the value for this option by using the RECOVERYDDN command option (page 250).

REDEFINE=YES

The REDEFINE option tells REORG PLUS whether to delete and redefine the VSAM data sets for the table space or index space as part of the reorganization. REORG PLUS can delete and redefine both user-defined (VCAT-defined) data sets and data sets defined in DB2 storage groups (storage-group-defined).

- For SHRLEVEL NONE or SHRLEVEL REFERENCE UNLOADONLY, REDEFINE=YES tells REORG PLUS to delete and redefine the VSAM data sets for the space before reloading it. For SHRLEVEL REFERENCE or SHRLEVEL CHANGE, REORG PLUS deletes and defines the staging data sets.

- REDEFINE=NO tells REORG PLUS not to delete and redefine the existing VSAM data sets for the table space or indexes. Instead, REORG PLUS issues message BMC50391I, reuses the existing data sets, and resets the high-used relative byte address (HURBA).

Additional considerations

The following considerations apply to the REDEFINE option:

- If you are reorganizing a large number of partitions, consider specifying REDEFINE NO. This value minimizes the time that REORG PLUS requires to delete and redefine the existing VSAM data sets for the table space or indexes.
- When invoking DSNUTILB, REORG PLUS passes REDEFINE=NO to the IBM DB2 REORG utility as REUSE and ignores REDEFINE=YES.

Overriding this option

You can override the value for this option by using the REDEFINE command option (page 195). The command section also contains information about using the REDEFINE option with the DSRSEXIT user exit, the SYSIDCIN data set, and the UNLOAD command.

RENMMAX=30

The RENMMAX option indicates the maximum number of tasks to start during the rename and delete process for SHRLEVEL REFERENCE and SHRLEVEL CHANGE. You can specify any of the following values:

- 0 to have REORG PLUS determine the number of tasks to start
- asterisk (*) to use the TASKMAX value (page 678)
- *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 540.

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 613. However, if you specify a value greater than the result of multiplying the RIDMMAXD value by the RIDMDSSZ value, REORG PLUS reduces the value that you specify to the product of those two installation options.

- During execution, REORG PLUS allocates memory only as needed, up to the current value of RMAPMEM. If RMAPMEM is insufficient, REORG PLUS issues message BMC50885I and spills to the spill data set.

Overriding this option

You can override the value for this option by using the RIDMAPMEM command option (page 310).

RORGMAX=300%

The RORGMAX option indicates the maximum number of tasks to start for the REORG phase. You can specify any of the following values:

- 0 to have REORG PLUS determine the number of tasks to start
- asterisk (*) to use the TASKMAX value (page 678)
- *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 540.

ROUTCDE=(11,1)

The ROUTCDE option specifies the routing codes to route write-to-operator (WTO) messages to the designated console. The first subparameter is the code for informational WTOs. The second subparameter is the code for WTOs that require an action. The values (11,1) route informational WTOs to programmer information (11) and WTOs requiring an action to the master console (1).

For a complete listing of valid values, see the appropriate IBM reference manual.

SCPYMAX=8

The SCPYMAX option indicates the maximum number of tasks to start during the nonpartitioned index copy process for a SHRLEVEL REFERENCE or SHRLEVEL CHANGE partial reorganization. You can specify any of the following values:

- 0 to have REORG PLUS determine the number of tasks to start
- asterisk (*) to use the TASKMAX value (page 678)
- *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 540.

SDUMP=(ALLPSA,CSA,RGN,SQA,LSQA,SUM,TRT,IO)

The SDUMP option tells REORG PLUS to generate a system dump, using the information listed in the option, if the job abnormally terminates. The values listed provide diagnostic information to BMC Customer Support. Although you can specify any values that IBM allows for the SDATA parameter on the SDUMPX macro, BMC recommends that you do not change the values that were shipped with the product. For a complete list and description of the values that you can specify for this option, see the IBM SDUMPX macro description.

REORG PLUS uses the system-defined dump data set to hold the data. In cases where multiple abends occur, REORG PLUS generates the dump for only the first abend. The SDUMP option allows you to generate a system dump, regardless of your access to storage keys 0 through 7.

You can also specify SDUMP=NO (without parentheses) to tell REORG PLUS not to generate a system dump.

NOTE

REORG PLUS displays SDUMP=YES in message BMC50471I in the SYSPRINT list of options when SDUMP contains one or more values.

To limit the conditions under which REORG PLUS generates the system dump, you can exclude selected abend codes by using the EXCLDUMP option (page 643).

SHORTMEMORY=CONTINUE

The SHORTMEMORY option controls the action that REORG PLUS takes when one of the following memory shortages exist during sort processing:

- The system contains insufficient available pages of memory for REORG PLUS to perform an optimal sort

- The region contains insufficient memory for REORG PLUS to perform a minimum number of tasks, given the amount of memory required to perform an optimal sort. However, the region *does* contain at least 1024 KB of memory, or the minimum amount of memory specified by the MINSORTMEMORY option. If the region contain at least 1024 KB of memory *and* the amount of memory specified by the MINSORTMEMORY option is available, REORG PLUS uses the greater amount of memory.

You can specify one of the following values for the SHORTMEMORY option:

- CONTINUE, the shipped value, indicates that, when a memory shortage exists, REORG PLUS should issue message BMC50364I and continue sort processing.

- FAIL indicates that REORG PLUS should fail when a memory shortage exists.

  Be aware when specifying FAIL that sufficient memory might exist to sort during the UNLOAD phase of a two-phase reorganization or the REORG phase of a single-phase reorganization. However, because of other system conditions, insufficient available pages might exist during the index build process, which occurs during the RELOAD process (two-phase reorganization) or following the REORG process (single-phase reorganization).

Table 99 on page 671 shows the relationship between SHORTMEMORY values and the following conditions:

- memory data obtained from the system
- memory in the region
- value specified for the SMCORE installation option (page 673)
- value specified for the MINSORTMEMORY installation or command option (page 661 and page 183)

Table 99 Action REORG PLUS takes when memory resources are constrained

Memory shortage		SHORTMEMORY value	
Location	**Condition**	**CONTINUE**	**FAIL**
memory in the system	insufficient to run one optimal sort task based on the amount of data to be sorted	REORG PLUS runs one task with 1024 KB of memory or the amount of memory that you specified with MINSORTMEMORY, whichever is greater.	REORG PLUS fails.
	insufficient as specified by MINSORTMEMORY		
virtual memory in the region	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	REORG PLUS runs one task with the available memory.	
	insufficient as specified by MINSORTMEMORY or less than 1024 KB, whichever is greater	REORG PLUS fails.	
	insufficient as specified by the first parameter of SMCORE		

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option

You can override the value for this option by using the SHORTMEMORY command option (page 181).

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.

- SIXSNAP=NO tells REORG PLUS to use the software-based copy method to copy each nonpartitioned index.

- SIXSNAP=AUTO tells REORG PLUS to use Instant Snapshot.

  If the Instant Snapshot copy fails, REORG PLUS copies the index using the software copy method.

- SIXSNAP=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 be different from the order in your list, depending on the hardware vendor. For more information, see "Considerations for SIXSNAP use" on page 117.

For detailed information about the SIXSNAP function, see "Instant Snapshot with nonpartitioned indexes" on page 116. For a list of the supported devices, see the *EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.*

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the SIXSNAP command option (page 228).

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 540.

For information about the hierarchy of the SMAX, MAXSORTS, and multitasking options, see Table 89 on page 542. For more information, see "Controlling the number of sort processes" on page 539.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the MAXSORTS command option (page 175).

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 537.

SORTDEVT=(,SYSALLDA)

The SORTDEVT option specifies the device type for the sort work files that are allocated dynamically.

The first parameter of this option is the device type to use for non-DSNUTILB jobs. This parameter overrides the first parameter of the BMCSORT DYNALOC installation option. If the value of the third parameter in the BMCSORT DYNALOC installation option is OFF, specifying a value for this parameter turns BMCSORT dynamic allocation on.

The second parameter of this option is the device type to use for DSNUTILB jobs. When invoking DSNUTILB, REORG PLUS passes this option to the IBM DB2 REORG utility for processing.

Overriding this option
You can override the values for both parameters of this option by using the SORTDEVT command option (page 173).

SORTNUM=32

The SORTNUM option affects the allocation of sort work files in the following cases. The shipped value is 32, and you can specify any integer value from 0 through 255.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS passes this value to the IBM DB2 REORG utility as the number of sort work files to allocate dynamically. For this type of reorganization, the value must be 2 or greater.

All other reorganization jobs

This value is in effect when BMCSORT is allocating your sort work files dynamically. Table 100 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 100 SORTNUM values

Value	Description	Additional considerations
0	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.)	For more information about how this parameter affects dynamic allocation, see "Dynamically allocating SORTWK data sets" on page 339. For more information about the parameter itself, see "DYNALOC installation option" on page 708.
1–32	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.	*Preallocated sort work files* include sort work files that are allocated in your JCL and any sort work files that REORG PLUS dynamically allocates.
33–255	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.	

If the value of the third parameter in the BMCSORT DYNALOC installation option is OFF, specifying a value greater than 0 for the SORTNUM option turns BMCSORT dynamic allocation on and BMCSORT allocates sort work files as needed. For information about when BMCSORT allocates your sort work files dynamically, see the "SORTWK data sets" on page 338.

Overriding this option

You can override the value for this option with the SORTNUM command option (page 174).

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 Table 101 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 101 Symbolic variables for the SPILDSNP installation option

Symbolic variable	Definition	Length of result
&DATE	current date (in the form *MMDDYY*)	6 bytes
&DATEJ	current Julian date (in the form *YYYYDDD*)	7 bytes
&DB	database containing the space for this data set allocation	8 bytes maximum
&GRPNM	DB2 data-sharing group name In a non-data-sharing environment, GRPNM contains the DB2 SSID.	4 bytes
&JDATE	current Julian date (in the form *YYDDD*)	5 bytes
&JOBNAME	JOB name in the JCL	8 bytes maximum
&RTYPE	REORG type (TS or IX)	2 bytes maximum
&SSID	DB2 subsystem ID	4 bytes
&STEPNAME	STEP name used in the JCL REORG PLUS ignores PROC names.	8 bytes maximum
&TIME	current time (in the form *HHMMSS*)	6 bytes
&TIME4	current time (in the form *HHMM*)	4 bytes
&TSIX	table space or index space specified in your REORG command	8 bytes maximum
&USERID or &UID	job user ID You must have a security package to use the job user ID variable.	8 bytes maximum
&UTIL	BMC utility ID	8 bytes maximum REORG PLUS truncates longer utility IDs to eight characters.
&UTILPFX	BMC utility ID prefix	8 bytes maximum
&UTILSFX	BMC utility ID suffix	8 bytes maximum
&VCAT	VCATNAME specified in the DB2 catalog for the table space that you are reorganizing; or, if the table space is partitioned, the VCAT name from the first partition that you are reorganizing	8 bytes

For more information and guidelines for specifying data set name patterns, see page 305. User exits are discussed in Appendix C, "REORG PLUS user exits."

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the SPILLDSNPAT command option (page 305).

SPILSCLS=NONE

This option applies to SHRLEVEL CHANGE only.

The SPILSCLS option specifies the SMS storage class that REORG PLUS uses to allocate spill data sets. You can specify a valid SMS storage class name not exceeding eight characters, or NONE. The spill data sets are VSAM objects.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the SPILLSTORCLAS command option (page 305).

SPILUNIT=WORK

This option applies to SHRLEVEL CHANGE only.

The SPILUNIT option specifies the DASD unit to which REORG PLUS can allocate spill data sets. The unit name must be a valid DASD unit name not exceeding eight characters, or NONE. The spill data sets are VSAM objects.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the SPILLUNIT command option (page 304).

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 **BMC**DBC and **BMC**DBD for the cluster and data nodes.

STAGEDSN=DSN tells REORG PLUS to allocate staging data sets with a second node of **DSN**DBC and **DSN**DBD and a fifth node of I or J (whichever letter is not in the original data set name).

For more information about staging data sets and their naming conventions, see “Staging data sets” on page 98.

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 exist:

- 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 540.

TEMPRALDATA=YES

TEMPRALDATA tells REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces or indexes for the following types of temporal objects:

- system-period temporal tables
- history tables that are associated with system-period temporal tables
- indexes that are defined with BUSINESS_TIME WITHOUT OVERLAPS
- business-period temporal tables that have an index that is defined with BUSINESS_TIME WITHOUT OVERLAPS

For all other types of temporal objects, REORG PLUS natively reorganizes the table space or index.

You can specify one of the following options:

- YES tells REORG PLUS to reorganize this table space or index by using DSNUTILB processing. To enable this feature, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS not to reorganize this table space or index. REORG PLUS terminates.

You cannot use this option to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively. For more information about running a DSNUTILB job, see “Reorganization jobs that invoke DSNUTILB” on page 70.

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. For details about the TERMEXIT user exit, see “Using TERMEXIT to control BMCHIST and statistics updates” on page 795.

NOTE

You cannot use a user exit supplied by the TERMEXIT option to override BMCSTATS NO or UPDATEDB2STATS NO to YES.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the TERMEXIT command option (page 223).

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 Table 102.

Table 102 Return code hierarchy for the TIMEOUT installation option (part 1 of 2)

TIMEOUT installation option with TERM	ON FAILURE with RETCODE specified	Return code is returned from
TERM	yes	ON FAILURE: return code that you specified with RETCODE
TERM	no	TIMEOUT: return code 8

Table 102 Return code hierarchy for the TIMEOUT installation option (part 2 of 2)

TIMEOUT installation option with TERM	ON FAILURE with RETCODE specified	Return code is returned from
TERM, *rc*	does not matter	TIMEOUT: return code that you specified with TERM
no value specified	no	REORG: return code 8

Overriding this option

You can override the value for this option by specifying the TIMEOUT command option (page 218).

TOTALPAGEPCT=0

The TOTALPAGEPCT option controls memory above the 16-MB line that REORG PLUS allocates to BMCSORT for concurrent sort processing. TOTALPAGEPCT specifies the maximum percentage of total 4-KB pages, as obtained from the system, that REORG PLUS can allocate. You can specify any integer from 0 through 100.

NOTE

REORG PLUS defines *total pages* as pages that are underutilized and are available for use. *Available pages* (which you can control with the AVAILPAGEPCT option), are pages that have not been used.

A value of 0 tells REORG PLUS to ignore the number of total pages when allocating sort memory.

A value of 1 through 100 tells REORG PLUS to use up to the specified percentage of total pages when allocating sort memory. For example, TOTALPAGEPCT 50 tells REORG PLUS to use no more than 50 percent of the total pages.

Additional considerations

The following additional information applies to the TOTALPAGEPCT option:

- When you specify values greater than 0 for both TOTALPAGEPCT and AVAILPAGEPCT, REORG PLUS uses the lesser of the two calculated results as the maximum amount of memory for sort processing.
- If REORG PLUS is unable to start any tasks because of restraints on sort memory caused by a low number of total or available pages, the SHORTMEMORY installation option (page 670) or command option controls the action that REORG PLUS takes.
- When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by specifying the TOTALPAGEPCT command option (page 180).

TSPREC=YES

The TSPREC option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces that contain a timestamp column that is defined with a precision other than 6:

- YES tells REORG PLUS to reorganize this table space by using DSNUTILB processing. To enable this feature, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS not to reorganize this table space. REORG PLUS terminates.

For more information about running a DSNUTILB job, see “Reorganization jobs that invoke DSNUTILB” on page 70.

TSSAMPLEPCT=100

This option applies to a table space reorganization only.

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

- 1 through 50 tells REORG PLUS to sample the specified percentage of the table space pages.

- 100 tells REORG PLUS to read all table space pages instead of sampling.

NOTE

Values 51 through 99 are not valid.

Restrictions
REORG PLUS ignores the TSSAMPLEPCT option for either of the following types of reorganizations:

- a DSNUTILB reorganization
- an index reorganization

Overriding this option
You can override the value for this option by using the TSSAMPLEPCT command option (page 240).

TSTZ=YES

The TSTZ option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces that contain a timestamp column that is defined as TIMESTAMP WITH TIME ZONE:

- YES tells REORG PLUS to reorganize this table space by using DSNUTILB processing. To enable this feature, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS not to reorganize this table space. REORG PLUS terminates.

For more information about running a DSNUTILB job, see "Reorganization jobs that invoke DSNUTILB" on page 70.

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 531.

UNLDDN=SYSREC

The UNLDDN option specifies the default ddname or ddname prefix for the output data set that contains the unloaded rows to be reorganized.

When using this name as a prefix for a ddname for a partitioned table space, ensure that you allow sufficient bytes for the partition number to be added and still have a valid ddname (eight bytes or less). In addition, if you change the default ddname that BMC supplies, you must also change the name in your JCL. For information about specifying and using this data set, see "SYSREC data sets" on page 349.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS uses this option only to pass a value for the &DDNAME variable for the data set name pattern. REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

Overriding this option

You can override the value for this option by using the UNLDDN command option (page 169).

UNLDMAX=300%

The UNLDMAX option indicates the maximum number of tasks to start per CPU for the UNLOAD phase. You can specify any of the following values:

- 0 to have REORG PLUS determine the number of tasks to start
- asterisk (*) to use the TASKMAX value (page 678)
- *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 540.

UNLOAD=RELOAD

The UNLOAD option tells REORG PLUS whether to use single-phase or two-phase processing. You can specify either RELOAD or CONTINUE. For more information about these options, see the UNLOAD command option on page 189.

UNLOAD=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 135.

UNLOAD=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

When you are reorganizing a LOB table space and SHRLEVEL REFERENCE is in effect, REORG PLUS changes UNLOAD CONTINUE to UNLOAD RELOAD.

Overriding this option

You can override the value for this option by using the UNLOAD command option (page 189). An additional value is available with the command option.

UTSMEM=YES

The UTSMEM option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) universal table spaces that are defined with MEMBER CLUSTER.

NOTE

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

You can specify one of the following options:

- YES tells REORG PLUS to reorganize this table space by using DSNUTILB processing. To enable this feature, DSNUTILB YES must also be in effect.
- NO tells REORG PLUS not to reorganize this table space. REORG PLUS terminates.

You cannot use this option to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively. For more information about running a DSNUTILB job, see “Reorganization jobs that invoke DSNUTILB” on page 70.

UXSTATE=SUP

The UXSTATE option tells REORG PLUS how to invoke DB2 user exits:

- UXSTATE=SUP specifies that REORG PLUS should call EDITPROCs in supervisor state (and PSW key=7).
- UXSTATE=PROB tells REORG PLUS to call EDITPROCs in problem state and PSW key=7.

The requirements of the exits dictate the UXSTATE setting. Check with the exit author (or vendor) before changing the value of this option.

WBUFFS=(20,10)

The WBUFFS option specifies a multiple of 32 KB to use to define the amount of buffer pool storage that is allocated for each work (SYSUT1) data set. REORG PLUS uses the first number if you specify only one work data set. REORG PLUS uses the second number for each work data set if you specify multiple work data sets. For more information, see “SYSUT1 data set” on page 532.

WORKDDN=SYSUT1

The WORKDDN option specifies the default ddname or ddname prefix for the index work data set that contains the unloaded index keys and serves as input for BMCSORT.

When using this name as a prefix, allow sufficient bytes for the number of indexes to be added and still have a valid ddname (eight bytes or less). In addition, if you change the default ddname that BMC supplies, you must also change the name in your JCL. For information about specifying and using this data set, see “SYSUT1 data sets” on page 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 WORKDDN command option (page 170).

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=

This XBMID option specifies the XBM subsystem (SSID) that REORG PLUS accesses when it uses XBM or SUF. If you do not specify the SSID as an installation option, you must specify it as a command option in either of the following cases:

- You are going to use a feature that uses the snapshot-processing functions of XBM or SUF.

- You want to use a specific XBM subsystem for zIIP processing.

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.

For more information about using XBM or SUF with REORG PLUS, see “XBM and SUF considerations” on page 140. For more information about XBM and SUF, see the *EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide*.

Snapshot processing

SHRLEVEL CHANGE uses XBM to create a snapshot of the data sets to be reorganized. You can use the XBM Utility Monitor function or MVS console support to dynamically override certain SHRLEVEL CHANGE command options while the reorganization is running. For more information, see "Using XBM to view and dynamically control the log apply process" on page 592.

Both SHRLEVEL CHANGE and SHRLEVEL REFERENCE use the Instant Snapshot technology of XBM or SUF to copy nonpartitioned indexes if you specify YES for the SIXSNAP command or installation option. For more information about the SIXSNAP command option, see "SIXSNAP" on page 228.

zIIP processing

If you specify an XBM subsystem and ZIIP ENABLED is in effect, REORG PLUS attempts to use that subsystem to enable zIIP processing. If that subsystem is not available or not at the correct maintenance level, zIIP processing is not enabled.

If you do not specify an XBM subsystem (either here or with the XBMID command option), REORG PLUS searches for an XBM subsystem at the appropriate maintenance level to enable zIIP processing.

Restriction

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option

You can override the value for this option by using the XBMID command option (page 230).

XML=YES

The XML option tells REORG PLUS whether to reorganize (by invoking DSNUTILB) table spaces that contain XML columns that support XML versions, or associated XML table spaces.

NOTE

REORG PLUS natively reorganizes table spaces that contain XML columns that are not in a format that supports XML versions.

You can specify one of the following options:

- YES tells REORG PLUS to reorganize this table space by using DSNUTILB processing. To enable this feature, DSNUTILB YES must also be in effect.

- NO tells REORG PLUS not to reorganize this table space. REORG PLUS terminates.

You cannot use this option to request that REORG PLUS invoke DSNUTILB for a feature that REORG PLUS supports natively. For more information about running a DSNUTILB job, see "Reorganization jobs that invoke DSNUTILB" on page 70.

ZIIP=ENABLED

The ZIIP option tells REORG PLUS whether to attempt to use IBM System z® Integrated Information Processors (zIIPs). REORG PLUS can use enclave service request blocks (SRBs) to enable zIIP processing automatically while running jobs. Using zIIP processing can reduce the overall CPU time for REORG PLUS jobs.

You can specify one of the following values:

- ENABLED tells REORG PLUS to attempt to offload eligible processing to an available zIIP. If the zIIP is busy or not available, normal processing continues on a general-purpose processor.

- DISABLED tells REORG PLUS to not attempt to use zIIP processing.

To enable and use zIIP processing with REORG PLUS, you must

- have an installed authorized version of XBM or SUF

- start and maintain an XBM subsystem in your environment

NOTE

You can specify a particular XBM subsystem to use by specifying a value for the XBMID installation or command option. For more information, see "XBMID" on page 230 or page 686.

- 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 (page 229).

Dynamic allocation installation options

The $ARUDYNA macros produce the installation options for dynamic allocation (one macro for each data set type). Table 103 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 lowercase *none* for no value), a brief description, and a reference to more details. If an option ships with no value, the table shows a recommended value or example value.

Your $C30DOPT member must contain exactly 13 $ARUDYNA macros. Each macro must specify a different DDTYPE.

Table 103 REORG PLUS dynamic allocation installation options (part 1 of 2)

Option	Shipped value	Brief description	Reference
DDTYPE	Each macro has a different default value. For example, the default value for local primary copy data sets is LOCPFCPY.	data set type to which the remaining dynamic allocation options apply	page 690
ACTIVE	Each DDTYPE has a different default value. For example, the default value for DDTYPE UNLOAD is YES.	whether to dynamically allocate this type	page 691
ALLOC	ANY	method to use when dynamically allocating sort work files This installation option is valid only for the SORTWORK work file type.	page 692
AVGVOLSP	((30000,TRK),(30000,TRK))	average space available for data on volumes that are used for dynamic allocation	page 693
DATACLAS	(NONE,NONE)	SMS data class to use	page 694
DSNPAT	Each DDTYPE has a different default value. For example, the default value for DDTYPE UNLOAD is &&UID.&&JOBNAME.&&TSIX.&&DDNAME	data set name pattern	page 694
EXPDT	none example value: 2011087	expiration date for copy and archive data sets	page 698
GDGEMPTY	NO	whether to uncatalog all data sets when the limit is reached	page 699
GDGLIMIT	5	number of GDG generations to keep	page 699
GDGSCRATCH	NO	whether to delete uncataloged data sets	page 699
IFALLOC	USE	whether to use DDs in the JCL if they are coded	page 700

Table 103 REORG PLUS dynamic allocation installation options (part 2 of 2)

Option	Shipped value	Brief description	Reference
MAXEXTSZ	((0,K),(0,K))	maximum extent size	page 700
MGMTCLAS	(NONE,NONE)	SMS management class to use	page 701
RETPD	none example value: 30	number of days to retain copy and archive data sets	page 702
SIZEPCT	■ (5,100) for DDTYPE LOCPICPY ■ (100,100) for all other DDTYPEs	percentage of the REORG PLUS calculated space to use for allocation	page 702
SMS	NO	whether to perform SMS allocations	page 703
SMSUNIT	NO	whether to pass the unit value to SMS	page 703
STORCLAS	(NONE,NONE)	SMS storage class to use	page 704
THRESHLD	0	whether to use secondary units, classes, number and size of volumes, and extent size	page 704
UNIT	(SYSALLDA,SYSALLDA)	unit names used for dynamic allocation	page 705
UNITCNT	(0,0)	number of devices to dynamically allocate	page 706
VOLCNT	(25,25)	largest number of volumes to process	page 707

This section describes each of the dynamic data set allocation options shown in Table 103. You can also specify most of these options with the DDTYPE command option, enabling you to override the defaults established at installation. For more information, see “Dynamic allocation options” on page 269.

DDTYPE=UNLOAD

The DDTYPE option specifies the data set type for which you are establishing dynamic allocation options. Table 104 lists the valid values for the DDTYPE option. Each instance of the $ARUDYNA macro must contain a different value for this option.

Table 104 Valid DDTYPE values (part 1 of 2)

DDTYPE value	Corresponding default ddname
UNLOAD	SYSREC
WORK	SYSUT1
SORTWORK	SORTWK
ARCHIVE	SYSARC
SYSPUNCH	SYSPUNCH

Table 104 Valid DDTYPE values (part 2 of 2)

DDTYPE value	Corresponding default ddname
LOCPFCPY	BMCCPY
LOCPICPY	BMCICY
LOCBFCPY	BMCCPZ
LOCBICPY	BMCICZ
REMPFCPY	BMCRCY
REMPICPY	BMCIRY
REMBFCPY	BMCRCZ
REMBICPY	BMCIRZ

Overriding this option

You can override each instance of the DDTYPE installation option by using the DDTYPE command option (page 270).

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 105 ACTIVE option default values by DDTYPE

DDTYPE	ACTIVE value
UNLOAD	YES
WORK	YES
SORTWORK	NO
ARCHIVE	NO
SYSPUNCH	YES
LOCPFCPY	YES
LOCPICPY	YES
LOCBFCPY	NO
LOCBICPY	NO
REMPFCPY	NO
REMPICPY	NO
REMBFCPY	NO
REMBICPY	NO

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 (page 272).

ALLOC=ANY

The ALLOC option tells REORG PLUS what method to use when dynamically allocating your sort work files. This option is valid only for sort work files. The following values are valid:

- OPTIMIZED tells REORG PLUS to allocate as much DASD as necessary for the best performance results.

- MINIMAL tells REORG PLUS to allocate the smallest amount of DASD necessary to process the job.

- ANY (the default) tells REORG PLUS to first attempt to allocate as much DASD as necessary for the best performance results. If it cannot obtain an optimal allocation, REORG PLUS reduces DASD allocation and continues processing.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the ALLOC command option (page 275).

AVGVOLSP=((30000,TRK),(30000,TRK))

The AVGVOLSP option enables you to specify the average amount of space that is available on each device. Use this option to reflect the average space available on volumes that are eligible to contain the dynamically allocated data set.

Specify one or two integer values, and include the unit of measure as follows:

- K for kilobytes
- TRK for tracks (the default)
- CYL for cylinders

REORG PLUS uses AVGVOLSP only when you specify a value of AUTO for the corresponding first or second parameter of the VOLCNT installation or command option (see page 707 or page 282).

REORG PLUS uses the second parameter (*integer2*) when the value for the THRESHLD installation option or command is exceeded (see page 704 or page 286). This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

Restrictions
The following restrictions apply to the AVGVOLSP option:

- The AVGVOLSP option is not valid for DDTYPE SORTWORK.
- When invoking DSNUTILB, REORG PLUS ignores this option.

Additional considerations
Note the following additional information about the AVGVOLSP option:

- You should not use AVGVOLSP to specify the maximum space on all devices or volumes unless the volumes to be used are empty.

- If you specify a value for AVGVOLSP that is too small, REORG PLUS computes a value for VOLCNT that is too large.

- If you specify a value for AVGVOLSP that is too large, REORG PLUS computes a value for VOLCNT that is too small.

Overriding this option
You can override the value for this option by using the AVGVOLSP command option (page 284).

DATACLAS=(NONE,NONE)

The DATACLAS option specifies the primary and secondary SMS data classes that REORG PLUS uses for an SMS allocation. The two class names must be valid SMS data class names, not exceeding eight characters each, or NONE.

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the secondary data class in that template. For information about when REORG PLUS builds a secondary template, see page 704.

Overriding this option
You can override the value for this option by using the DATACLAS command option (page 299).

DSNPAT=&&UID.&&JOBNAME.&&TSIX.&&DDNAME

The DSNPAT option specifies the pattern that REORG PLUS uses to generate data set names during dynamic data set allocation.

For sort work data sets only, you can also specify the value NONE. Do not enclose NONE in single quotes with the DSNPAT *installation* option (even though the quotes are required in the DSNPAT *command* option).

Each DDTYPE has its own default value for the DSNPAT option:

Table 106 DSNPAT option default values by DDTYPE (part 1 of 2)

DDTYPE	DSNPAT value
UNLOAD	&&UID.&&JOBNAME.&&TSIX.&&DDNAME
WORK	&&UID.&&JOBNAME.&&TSIX.&&DDNAME
SORTWORK	NONE
ARCHIVE	&&UID.&&UTILPFX.&&DDNAME
SYSPUNCH	&&UID.&&UTILPFX.&&DDNAME
LOCPFCPY	&&UID.&&DDNAME.&&TSIX..F&&PART..T&&TIME
LOCPICPY	&&UID.&&DDNAME.&&TSIX..I&&PART..T&&TIME

Table 106 DSNPAT option default values by DDTYPE (part 2 of 2)

DDTYPE	DSNPAT value
LOCBFCPY	&&UID.&&DDNAME.&&TSIX..F&&PART..T&&TIME
LOCBICPY	&&UID.&&DDNAME.&&TSIX..I&&PART..T&&TIME
REMPFCPY	&&UID.&&DDNAME.&&TSIX..F&&PART..T&&TIME
REMPICPY	&&UID.&&DDNAME.&&TSIX..I&&PART..T&&TIME
REMBFCPY	&&UID.&&DDNAME.&&TSIX..F&&PART..T&&TIME
REMBICPY	&&UID.&&DDNAME.&&TSIX..I&&PART..T&&TIME

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 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.

You can use text or any of the symbolic variables in Table 107 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. The maximum total length allowed for a data set 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 TEMPLATE control statement or translatable (as shown in Table 107) 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.

NOTE

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 surround NONE with single quotation marks. The DSNPAT command option is described on page 289.

Table 107 Symbolic variables for the DSNPAT installation option (part 1 of 3)

Symbolic variable	Definition	Length of result[a]	DSNUTILB reorganization
&DATE	current date (in the form MMDDYY)	6 bytes	variable passed
&DATEJ	current Julian date (in the form YYYYDDD)	7 bytes	variable translated to IBM’s &JDATE

Table 107 Symbolic variables for the DSNPAT installation option (part 2 of 3)

Symbolic variable	Definition	Length of result[a]	DSNUTILB reorganization
&DB	database containing the space being used for this data set allocation	8 bytes maximum	variable passed
&DDNAME	ddname being used for this data set allocation	8 bytes maximum	value passed
&GRPNM	DB2 data-sharing group name In a nondata-sharing environment, GRPNM contains the DB2 SSID.	4 bytes	value passed
&JDATE	current Julian date (in the form YYDDD)	5 bytes	job terminated
&JOBNAME	JOB name used in the JCL	8 bytes maximum	variable passed
&PART	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.	3 bytes for table spaces with 999 partitions or less 4 bytes for table spaces with 1000 through 4096 partitions	variable passed

&PART5	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.&DB.&TSIX..P&PART5'`
	REORG PLUS generates the following 5-character partition number for partition 4096:
	`ABC.DSN1.DA.DBNAME.TSNAME.P04096`
		5 bytes	variable passed

&RTYPE	REORG type (TS or IX)	2 bytes maximum	job terminated
&SSID	DB2 subsystem ID	4 bytes	variable passed

Table 107 Symbolic variables for the DSNPAT installation option (part 3 of 3)

Symbolic variable	Definition	Length of result[a]	DSNUTILB reorganization
&STEPNAME	STEP name used in the JCL REORG PLUS ignores PROC names.	8 bytes maximum	variable passed
&TIME	current time (in the form HHMMSS)	6 bytes	variable passed
&TIME4	current time (in the form HHMM)	4 bytes	variable passed
&TSIX	table space or index space specified in your REORG command	8 bytes maximum	variable translated to IBM's &SN
&USERID or &UID	job user ID You must have a security package to use this variable.	8 bytes maximum	variable passed
&UTIL[b]	BMC utility ID REORG PLUS truncates longer utility IDs to eight characters.	8 bytes maximum	variable translated to IBM's &UTILID
&UTILPFX[b]	BMC utility ID prefix	8 bytes maximum	value passed
&UTILSFX[b]	BMC utility ID suffix	8 bytes maximum	value passed
&VCAT	VCATNAME specified in the DB2 catalog for the table space that you are reorganizing; or, if the table space is partitioned, the VCAT name from the first partition that you are reorganizing	8 bytes	job terminated

[a] REORG PLUS removes any trailing blanks in the result.

[b] Utility IDs that include special characters might cause REORG PLUS to generate invalid data set names. For more information, see page 292.

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" on page 89.

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 page 289.

EXPDT=

The EXPDT option specifies the expiration date for the SYSARC, SYSPUNCH, or copy data set that you are dynamically allocating. The date must be either blank or in the format *yyyyddd*:

- *yyyy* is the 4-digit year (1900 through 2155).
- *ddd* is the 3-digit Julian day (000 through 366).

A blank value means that REORG PLUS does not use an expiration date.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For more information about TEMPLATE control statements, see the documentation for the DB2 REORG utility.

When you specify the EXPDT installation option, it takes precedence over the RETPD installation option.

REORG PLUS ignores this option if you specify it for any other data set that you are dynamically allocating.

Overriding this option

You can override the EXPDT installation option by using the EXPDT or RETPD command option. For information about the commands, see “EXPDT” on page 296 and “RETPD” on page 296.

GDGEMPTY=NO

The GDGEMPTY option specifies whether to uncatalog all data sets when the GDGLIMIT is reached:

- NO indicates that the system uncatalog only the oldest GDG data set when the GDGLIMIT is reached.
- YES indicates that the system uncatalog *all* preexisting generations of this data set when the limit is reached.

REORG PLUS honors this option only when creating the GDG base for the SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating. REORG PLUS ignores this option if you specify it for any other data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS ignores this option.

GDGLIMIT=5

The GDGLIMIT option specifies the number of generations to keep for the SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating as a GDG data set. The number must be an integer in the range 1 through 255.

REORG PLUS honors this option only when creating the GDG base. REORG PLUS ignores this option if you specify it for any other data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

Overriding this option

You can override the value for this option by using the GDGLIMIT command option (page 297).

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.

Specifying IFALLOC=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.

Specifying IFALLOC=FREE tells REORG PLUS to free the data sets that you allocated in your JCL and use only dynamically allocated data sets.

When invoking DSNUTILB, REORG PLUS ignores this option.

Overriding this option
You can override the value for this option by using the IFALLOC command option (page 273).

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 MAXEXTSZ by using one of the following values:

- 0 if you do not want to set a limit for space allocation
- an integer for the units specified with the UNIT installation or command option (see page 705or page 280), with the unit of measure as follows:
  - —K for kilobytes (the default)
  - —TRK for tracks
  - —CYL for cylinders

REORG PLUS uses the second parameter when the value specified for the THRESHLD installation or command option is exceeded (see page 704 or page 286). This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

Regardless of the amount of space that REORG PLUS determines that it needs, REORG PLUS will not allocate more than your specified MAXEXTSZ limit for either the primary or the secondary quantity:

- If the amount of required space that REORG PLUS calculates is greater than the MAXEXTSZ limit for the primary quantity, REORG PLUS uses the secondary extents to hold the remainder of the required primary space.

- If the amount of required space that REORG PLUS calculates cannot be accommodated because of MAXEXTSZ restrictions, the job might terminate with an out-of-space condition on the data set.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes the first parameter of this option as the MAXPRIME value in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. For DSNUTILB, the unit of measure is always cylinders.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second parameter of this option as the MAXPRIME value in that template. For information about when REORG PLUS builds a secondary template, see page 704.

Additional considerations

Note the following additional information about the MAXEXTSZ option:

- The MAXEXTSZ option is not valid for DDTYPE SORTWORK.
- REORG PLUS ignores MAXEXTSZ when you specify SMS YES.
- REORG PLUS checks the value of MAXEXTSZ after applying SIZEPCT to the allocation amount.

Overriding this option

You can override the value for this option by using the MAXEXTSZ command option (page 278).

MGMTCLAS=(NONE,NONE)

The MGMTCLAS option specifies the primary and secondary SMS management classes that REORG PLUS uses for an SMS allocation. The two class names must be valid SMS management class names, not exceeding eight characters each, or NONE.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the secondary management class in that template. For information about when REORG PLUS builds a secondary template, see page 704.

Overriding this option
You can override the value for this option by using the MGMTCLAS command option (page 299).

RETPD=

Use RETPD to specify the retention period (in days) for the SYSARC, SYSPUNCH, or image copy data set that you are dynamically allocating. The number of days must be either blank or in the range 0 through 9999. A blank value means that REORG PLUS does not use a retention period.

REORG PLUS ignores this option if you specify it for any other data set that you are dynamically allocating.

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

Overriding this option
You can override the value for this option by using the RETPD (page 296) or EXPDT command option (page 296). 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 DDTYPE LOCPICPY, 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 discard data set and uses only ARCROWS to determine the discard data set size.

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS includes the first parameter of this option as the PCTPRIME value in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility. If you specify a value greater than 100, REORG PLUS converts it to 100.

Overriding this option
You can override the value for this option by using the SIZEPCT command option (page 285).

SMS=NO

The SMS option tells REORG PLUS whether to pass the SMS classes in the SMS allocation parameter list 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 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 during dynamic allocation.

  When you specify SMS=NO, REORG PLUS always passes the UNIT value during dynamic allocation.

Overriding this option
You can override the value for this option by using the SMS command option (page 276).

SMSUNIT=NO

The SMSUNIT option tells REORG PLUS whether to pass the UNIT value in the SMS allocation parameter list to SMS during dynamic allocation. REORG PLUS does not modify any other parameters based on this option. If the value of the SMS option is NO, REORG PLUS ignores the SMSUNIT option.

- NO tells REORG PLUS not to pass the value for the UNIT option.
- YES tells REORG PLUS to pass the value for the UNIT option.

  When invoking DSNUTILB, REORG PLUS includes the UNIT option values in the TEMPLATE control statements that it builds for the IBM DB2 REORG utility.

Overriding this option

You can override the value for this option by using the SMSUNIT command option (page 277).

STORCLAS=(NONE,NONE)

The STORCLAS option specifies the primary and secondary SMS storage classes that REORG PLUS uses for an SMS allocation. The two class names must be valid SMS storage class names, not exceeding eight characters each, or NONE.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes this information in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the secondary storage class in that template. For information about when REORG PLUS builds a secondary template, see page 704.

Overriding this option

You can override the value for this option by using the STORCLAS command option (page 299).

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)
  — VOLCNT (volume count)
  — AVGVOLSP (average volume space)
  — MAXEXTSZ (extent size)

- 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)
  - — if SMSUNIT is YES
    - UNIT (unit name)
    - UNITCNT (unit count)

If you specify 0 or if the threshold is not exceeded, REORG PLUS uses the primary values for these options.

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS handles this option differently, depending on the type of data set that you are allocating:

- For copy data sets, REORG PLUS translates this option to a LIMIT value in the TEMPLATE control statement that REORG PLUS builds for the IBM DB2 REORG utility. REORG PLUS also builds the secondary TEMPLATE control statement to which the DB2 REORG utility will switch when the LIMIT is exceeded.

  REORG PLUS assumes that the value that you supply for THRESHLD is in kilobytes. REORG PLUS translates that value to the appropriate value and unit of measure for the LIMIT keyword. Note the following additional information about this value:

  - — REORG PLUS rounds down to the nearest whole value.
  - — If you specify a value that would cause REORG PLUS to translate to a value less than 1 cylinder, REORG PLUS builds the template with a LIMIT value of 1 CYL.

- For all other data sets, REORG PLUS ignores this option.

Additional consideration

If you use THRESHLD to send larger data sets to tape, consider setting the MAXTAPE option to limit the number of tapes that you use. For examples of using the THRESHLD option with other dynamic allocation options, see page 286.

Overriding this option

You can override the value for this option by using the THRESHLD command option (page 286).

UNIT=(SYSALLDA,SYSALLDA)

For non-SMS-managed data sets, the UNIT option specifies the primary and secondary unit names that REORG PLUS uses for dynamic data set allocation. These unit names cannot exceed eight characters each.

For SMS-managed data sets when you specify SMS YES and SMSUNIT YES, this option supplies the unit names that REORG PLUS passes in the SMS allocation parameter list. DSNUTILB reorganization jobs function differently, as described in the DSNUTILB reorganization jobs section.

The THRESHLD command or installation option controls which of the specified unit names REORG PLUS selects. For more information about the THRESHLD installation option, see page 704.

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second UNIT parameter in that template. For information about when REORG PLUS builds a secondary template, see page 704.

Overriding this option
You can override the value for this option by using the UNIT command option (page 280).

UNITCNT=(0,0)

The UNITCNT option enables you to specify the number of devices to allocate when dynamically allocating data sets. Valid values are 0 through 59. A value of 0 tells REORG PLUS to use the system default.

If you specify a second value, REORG PLUS uses that value when the value for THRESHLD (page 704) is exceeded. This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

DSNUTILB reorganization jobs
When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second UNITCNT parameter in that template. For information about when REORG PLUS builds a secondary template, see page 704.

Additional considerations
The following additional considerations apply to the UNITCNT option:

- REORG PLUS ignores this option for SORTWORK files.
- To avoid performance problems, specify only the number of devices that you need.

Overriding this option
You can override the value for this option by using the UNITCNT command option (page 280).

VOLCNT=(25,25)

The VOLCNT option enables you to specify the maximum number of volumes to use during dynamic data set allocation. Valid values are

- 0, which tells REORG PLUS to not specify a volume count for dynamic allocation.
- Integer values 1 through 255 to specify the number of volumes.
- AUTO, which tells REORG PLUS to compute the volume count based on the amount of data, adjusted for the estimated space required.

  REORG PLUS computes the volume count by dividing the size estimate by the value specified for the AVGVOLSP installation or command option (see page 693 or page 284).

If you specify a second value (*integer2*), REORG PLUS uses that value when the value for the THRESHLD installation or command option is exceeded (page 704 and page 286). This value applies to the secondary device type that you specify with the UNIT option (or, if applicable, through SMS).

DSNUTILB reorganization jobs

When invoking DSNUTILB, REORG PLUS includes the first parameter in the TEMPLATE control statement that it builds for the IBM DB2 REORG utility.

If building a secondary TEMPLATE control statement for your copy data sets, REORG PLUS includes the second VOLCNT parameter in that template. For information about when REORG PLUS builds a secondary template, see page 704.

Additional considerations

The following additional considerations apply to the VOLCNT option:

- This option is not available for DDTYPE SORTWORK. If you specify a value for this option for sort work files, REORG PLUS changes the value to 1.

- Specify only the number of volumes that you need.

- If the DDTYPE will be an SMS-managed data set, BMC recommends that you specify a value of 0 if your ACS routines are set up to provide a volume count.

Overriding this option

You can override the value for this option by using the VOLCNT command option (page 282).

DYNALOC installation option

The DYNALOC installation option provides information for dynamically allocating SORTWK data sets. BMCSORT deallocates these data sets at the end of each sort. The content of the $AUPSMAC macro in $C32SOPT follows, showing DYNALOC and the values that are shipped with BMCSORT.

```
$AUPSMAC DYNALOC=(SYSDA,3,ON,ON,6000000,3000000,3390,SC=,RETRY=(0,0)) X
                                      DYNAMIC ALLOC OPTIONS FOR SORT
```

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 108 on page 708 describes each parameter of the DYNALOC option. These parameters are positional. The values that you specify for these parameters should correspond to your site's standards for any system sort routine.

BMCSORT overrides the values that you supplied if BMCSORT determines that it can complete sorting more efficiently than the specified values allow. An invoking product's options might also override the BMCSORT options values that you specify when either of the following conditions exists:

- The values in the invoking product's dynamic allocation installation options or corresponding command options conflict with the values that you specify.
- You turn on BMCSORT SORTWK dynamic allocation from the product that invokes BMCSORT, and you specify OFF for the position 3 parameter.

  BMCSORT dynamically allocates SORTWK files as necessary.

Table 108 DYNALOC parameters (part 1 of 3)

Parameter name or position	Description	Initial value	Valid values
position 1	This parameter specifies the generic unit name from which REORG PLUS for DB2 should dynamically allocate SORTWK data sets. This parameter applies only when the Data Facility Storage Management Subsystem (DFSMS) product from IBM is not installed or is not active for temporary DASD work data sets. If DFSMS is active, use the SC parameter.	SYSDA	Use a unit name up to 8 characters.
position 2	Do not change this value. REORG PLUS for DB2 does not use this parameter, but the parameter is required for proper assembly of the installation options macro.	3	Do not change this value.

Table 108 DYNALOC parameters (part 2 of 3)

Parameter name or position	Description	Initial value	Valid values
position 3	This parameter tells REORG PLUS for DB2 whether to dynamically allocate SORTWK files. **Note**: BMC recommends that you not change this value.	ON	■ ON dynamically allocates SORTWK. ■ OFF does not dynamically allocate SORTWK.
position 4	Do not change this value. REORG PLUS for DB2 does not use this parameter, but the parameter is required for proper assembly of the installation options macro.	ON	Do not change this value.
position 5	Do not change this value. REORG PLUS for DB2 does not use this parameter, but the parameter is required for proper assembly of the installation options macro.	6000000	Do not change this value.
position 6	Do not change this value. REORG PLUS for DB2 does not use this parameter, but the parameter is required for proper assembly of the installation options macro.	3000000	Do not change this value.
position 7	This parameter specifies the DASD type with the smallest track capacity that a dynamically allocated SORTWK data set might encounter at your site.	3390	■ 3380, track capacity of 47968 ■ 3390, track capacity of 56664 ■ 9345, track capacity of 46456
SC	This parameter specifies the name of the DFSMS storage class from which to dynamically allocate SORTWK. If DFSMS is active and you do not specify a value for this parameter, REORG PLUS for DB2 uses the value from the first DYNALOC parameter. **Note**: If your installation has an automatic class selection (ACS) routine, it can override this specification.	blank	Use any valid DFSMS storage class.

Table 108 DYNALOC parameters (part 3 of 3)

Parameter name or position	Description	Initial value	Valid values
RETRY	This parameter specifies how you want REORG PLUS for DB2 to handle retry attempts for SORTWK dynamic allocation: ■ The first subparameter indicates the number of times that you want REORG PLUS for DB2 to retry the request. ■ The second subparameter indicates the number of minutes to wait between each retry. Using this parameter allows you to avoid a capacity-exceeded condition when disk space is not immediately available for a SORTWK dynamic allocation request. BMC recommends that you do not change this value because it can affect the elapsed time of your jobs. However, if you currently use SyncSort and rely on the retry function, BMC recommends that you use the same values as your SyncSort RETRY installation parameter.	(0,0)	If you use this parameter, BMC recommends that you specify the same values as your SyncSort RETRY installation parameter. The following values are valid for this parameter: ■ 0 through 16 for the first subparameter. 0 indicates that you do not want REORG PLUS for DB2 to retry the request. ■ 0 through 15 for the second subparameter. 0 indicates that you do not want REORG PLUS for DB2 to retry the request.

Appendix

B

Common utility tables

This appendix presents the following topics:

Overview

The BMC common utility tables contain information about the BMC utilities that you generate and submit through a BMC utility product. Table 109 on page 712 lists the tables that each utility uses and each table's default name and synonym.

Table 109 Common utility tables

Table	Default name	Synonym	BMCDSN[a]	CHECK PLUS	COPY PLUS	DASD MANAGER PLUS (BMCSTATS)	LOADPLUS	Log Master	RECOVER PLUS	RECOVERY MANAGER	REORG PLUS	UNLOAD PLUS
BMCDICT	CMN_BMCDICT	BMC_BMCDICT					X				X	
BMCHIST	CMN_BMCHIST	BMC_BMCHIST	X	X	X		X		X		X	X
BMCLGRNX	CMN_BMCLGRNX	BMC_BMCLGRNX			X			X	X	X		
BMCSYNC	CMN_BMCSYNC	BMC_BMCSYNC	X	X	X	X	X		X	X	X	X
BMCTRANS	CMN_BMCTRANS	BMC_BMCTRANS						X		X		
BMCUTIL	CMN_BMCUTIL	BMC_BMCUTIL	X	X	X	X	X		X	X	X	X
BMCXCOPY	CMN_BMCXCOPY	BMC_BMCXCOPY			X			X	X	X	X	X

[a] BMCDSN does not write to the tables, but does access them to display information from them.

Considerations and warnings

Note the following considerations when using the common utility tables:

- Some columns in the tables are present for compatibility with specific BMC utilities and are not used by all of the utilities.

- If you have applications that depend on the structure or content of these tables, be aware that these tables are subject to change.

- In general, the utility tables should not require maintenance, with the exception of BMCHIST.

- You should back up the BMC table spaces on a regular basis to enable recoveries. If you use COPY PLUS as the copy utility, you must use SHRLEVEL CHANGE for the following spaces:

  — BMCUTIL

  — BMCHIST

  — BMCSYNC

  — BMCXCOPY

WARNING

The following warnings apply:

- Do not run LOADPLUS, REORG PLUS, or UNLOAD PLUS against the BMC common utility tables or table spaces. Doing so can cause unpredictable results.

- Because RECOVER PLUS uses BMC tables during the recovery process, you cannot use RECOVER PLUS to recover the BMC tables, with the exception of the BMCHIST table.

- Do not run the RUNSTATS utility against the BMC common utility tables. Doing so can negatively impact utility performance.

- BMC strongly recommends that you use the ISOLATION (UR) bind option and issue SQL COMMIT statements when querying the tables in the BMC database. If objects in the BMC database are restricted for UPDATE, the executing BMC utilities might not be able to complete successfully.

Managing common utility tables

This section provides basic procedures for working with the common utility tables:

To determine your site's table names

The names of the common utility tables can be changed during installation. To determine the names that your site uses, perform one of the following actions:

- Use your utility to run a job with restart parameters of MAINT and MSGLEVEL(1).

  Specifying MSGLEVEL(1) with MAINT prints the names of the BMC tables that your utility uses and identifies the applied maintenance. The utility does not perform any other processing, and the job ends without affecting any utility that is running.

- Run the following SQL statement, replacing *tableName* with a BMC common utility table name (listed in Table 109 on page 712):

```
SELECT CREATOR,NAME FROM SYSIBM.SYSTABLES
 WHERE TSNAME='tableName';
```

- Get the names from your DB2 system administrator.

To query the tables

Run SQL statements similar to the following examples.

— *EXAMPLE*

This example queries the BMCXCOPY table to access information about the rows in an index space:

```
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

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

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    -- for LOADPLUS and REORG PLUS
 WHERE UTILID='utilityID';
```

The utility terminates with return code 8 when the next checkpoint is taken.

To clean up a BMC utility that is not executing, run the utility with the correct utility ID and specify TERM as the restart parameter.

BMCDICT table

Table 110 describes the BMCDICT table, which stores the compression dictionary during load or reorganization processing.

Table 110 BMCDICT table

Column name	Data type	Description
UTILID	CHAR(16)	utility identifier
DBNAME	CHAR(8)	database name
TSNAME	CHAR(8)	table space name
PARTITION	SMALLINT	partition number For a nonpartitioned table space, the value is 0.
SEQNO	SMALLINT	sequence number
DICTDATA	VARCHAR(4000)	dictionary data

Considerations

Note the following considerations:

- If you are processing a large number of compressed partitions, you might need to increase the size of the BMCDICT table space significantly from the standard size that was allocated during installation. To estimate the allocation, multiply 64 KB by the number of compressed partitions that you are processing concurrently (loading with LOADPLUS and reorganizing with REORG PLUS).

- LOADPLUS inserts rows into the BMCDICT table during the PRELOAD phase and deletes those rows following compression processing in the LOAD phase.

- REORG PLUS inserts rows into the BMCDICT table during the UNLOAD phase and deletes those rows following compression processing in the RELOAD phase.

Maintaining the BMCDICT table

If LOADPLUS or REORG PLUS abends during the time between building the compression dictionary and completing compression, rows might remain in the BMCDICT table. If you need to control the expansion of this table, use the following procedure:

1. Delete any rows in the BMCUTIL table that you know are no longer valid.

   Do not delete any rows for instances of utilities that are awaiting restart.

2. Use the following SQL statement to delete rows from the BMCDICT table:

```
DELETE
FROM creatorName.CMN_BMCDICT
 WHERE UTILID NOT IN
  (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
```

NOTE

The names of the BMCUTIL and BMCDICT tables might have been changed at your site during installation.

BMCHIST table

Table 111 describes the BMCHIST table, which contains information about completed executions of the BMC utilities for DB2. The following 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 an installation option), the utility inserts rows into the BMCHIST table during the UTILTERM phase.

Table 111 BMCHIST table (part 1 of 2)

Column name	Data type	Description
DBNAME	CHAR(8)	name of the database that contains the table or index space
SPNAME	CHAR(8)	name of the table or index space

Table 111 BMCHIST table (part 2 of 2)

Column name	Data type	Description
UTILNAME	CHAR(8)	name of the utility: ■ CHECK ■ COPY ■ LOAD ■ RECOVER ■ REORG ■ UNLOAD
UTILID	CHAR(16)	utility identifier
AUTHID	CHAR(8)	user ID that ran the utility
DATE	DATE	date that the utility completed
TIME	TIME	time that the utility completed
ELAPSED	TIME	elapsed time of the utility
PARTITION	LONG VARCHAR	ALL, or the partition numbers as specified by the DSNUM option (for COPY PLUS) or the PART option Note the following conditions: ■ This column lists only three-digit partitions (any loaded partitions 1 through 999). Four-digit partitions (any loaded partitions from 1000 through 4096) are not stored in this column. For jobs that load only four-digit partitions, this column is empty. ■ If the list of partitions exceeds 1011 bytes, the utility truncates the value that is stored in this column.
OBJNAME	VARCHAR(27)	fully qualified object name
PHASE_1	CHAR(8)	name of utility phase 1
ELAPSED_1	TIME	elapsed time of phase 1
PHASE_2	CHAR(8)	name of utility phase 2
ELAPSED_2	TIME	elapsed time of phase 2
PHASE_3	CHAR(8)	name of utility phase 3
ELAPSED_3	TIME	elapsed time of phase 3
PHASE_4	CHAR(8)	name of utility phase 4
ELAPSED_4	TIME	elapsed time of phase 4
PHASE_5	CHAR(8)	name of utility phase 5
ELAPSED_5	TIME	elapsed time of phase 5

Maintaining the BMCHIST table

When a utility completes successfully, it inserts a row into the BMCHIST table. Periodically, review BMCHIST and delete old rows to control its expansion.

To delete selected rows from the BMCHIST table based on the date that the utility completed, use the following sample SQL statement:

```
DELETE
FROM creatorName.CMN_BMCHIST
 WHERE DATE < 'yyyy-mm-dd';
```

You can also use the TERMEXIT user exit to control inserts into the BMCHIST table. For more information, see "TERMEXIT" on page 223.

BMCLGRNX table

Table 112 describes the contents of the BMCLGRNX table, which contains log ranges that show when a table space was open for updates.

Table 112 BMCLGRNX table

Column name	Data type	Description
LGRDBID	CHAR(2)	DBID of the modified object
LGRPSID	CHAR(2)	OBID of the modified object
LGRUCDT	CHAR(6)	modification date (*mmddyy*)
LGRUCTM	CHAR(8)	modification time (*hhmmssth*)
LGRSRBA	CHAR(6)	starting RBA
LGRSPBA	CHAR(6)	stopping RBA
LGRPART	SMALLINT	table space partition number
LGRSLRSN	CHAR(6)	starting LRSN of update log records for data sharing For non-data-sharing, the value is X'000000000000'.
LGRELRSN	CHAR(6)	ending LRSN of update log records for data sharing For non-data-sharing, the value is X'000000000000'.
LGRMEMBER	CHAR(2)	data sharing member ID of the modifying DB2 subsystem For non-data-sharing, the value is X'0001'.

BMCSYNC table

Table 113 describes the BMCSYNC table, which contains information about the status of the objects that the currently executing utilities are accessing. The BMCSYNC table synchronizes and controls access to DB2 spaces by concurrently executing BMC utility products. If you have more than one BMC utility installed, all of these utilities should share the same BMCSYNC table.

The utilities insert rows into the BMCSYNC table during the UTILINIT phase. While the job executes, the utilities update the table as the status of the object changes. The utilities delete rows from the BMCSYNC table during the UTILTERM phase.

Table 113 BMCSYNC table (part 1 of 3)

Column name	Data type	Description
UTILID	CHAR(16)	utility identifier For RECOVER PLUS, this column is blank when a RECOVER UNLOADKEYS command creates the row and then a RECOVER BUILDINDEX command reads and deletes the row.
NAME1	CHAR(8)	database name or creator name[a] For DASD MANAGER PLUS, the value is the database name.
NAME2	CHAR(18)	space, table, or index name[a] For DASD MANAGER PLUS, the BMCSTATS utility always inserts the space name (limited to a maximum of 8 characters).
KIND	CHAR(2)	type of object: ■ **IP** (index partition) ■ **IX** (index) ■ **TB** (table) ■ **TP** (table space partition) ■ **TS** (table space) ■ **DD, DW** (dynamic work file allocation) ■ **CI** (copy information) ■ **RD** (restart data set block)
PARTITION	SMALLINT	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 COPY PLUS, LOADPLUS, UNLOAD PLUS, CHECK PLUS, DASD MANAGER PLUS, and REORG PLUS use null or 0 for *any* nonpartitioned space.
BMCID	SMALLINT	internal identifier of the object DASD MANAGER PLUS does not use this column.

Table 113 BMCSYNC table (part 2 of 3)

Column name	Data type	Description
UTILNAME	CHAR(8)	name of the executing utility: ■ CHECK ■ COPY ■ STATS ■ LOAD ■ RECOVER ■ REORG ■ UNLOAD
SHRLEVEL	CHAR(1)	degree to which utilities can share this object: ■ Blank means that no status is requested, and any other utility can obtain any status. ■ **S** allows sharing among any number of SHRLEVEL S utilities. ■ **X** indicates that exclusive control is required. No other utility can run with SHRLEVEL X. For more information, see Table 11 on page 79.
STATUS	CHAR(1)	status of the utility or object: ■ blank (indicates no processing has been done) ■ **C** (for CHECK PLUS, indicates checked) ■ **L** (for LOADPLUS, indicates loaded) ■ **U** (for UNLOAD PLUS, indicates unloaded) ■ **R** (for REORG PLUS, indicates reloaded) DASD MANAGER PLUS does not use this column.
XCOUNT	INTEGER	number of rows or keys processed in the current phase DASD MANAGER PLUS does not use this column.
DDNAME	CHAR(8)	check, load, unload, or work ddname DASD MANAGER PLUS does not use this column.
BLOCKS	INTEGER	number of blocks for the check, load, unload, or work data set DASD MANAGER PLUS does not use this column.
ORIG_STATUS	CHAR(8)	encoded representation of the original DB2 status of the space For RECOVER PLUS, this column restores the DB2 status of a space after recovery, if necessary. DASD MANAGER PLUS does not use this column.
EXTRBA	CHAR(6)	*(RECOVER PLUS)* log point at which this space was externalized RECOVER PLUS serialization logic uses this column. The other utilities do not use this column.

Table 113 BMCSYNC table (part 3 of 3)

Column name	Data type	Description
STATE	LONG VARCHAR	restart information for the space For example, the STATE indicates the object state and sync information. DASD MANAGER PLUS does not use this column.
INSTANCE	SMALLINT	*(RECOVERY MANAGER and RECOVER PLUS)* the instance number of the current base objects (table and index) The default value is 1. The other utilities do not use this column.

[a] *(LOADPLUS, UNLOAD PLUS, CHECK PLUS, and REORG PLUS)* If the value for NAME1 would exceed 8 bytes or the value for NAME2 would exceed 18 bytes, NAME1 contains the DBID for the object; NAME2 contains the table OBID or index ISOBID of the object in hexadecimal format.

Considerations

Note the following considerations:

- You might need to increase the size of the BMCSYNC table space from the standard size that was allocated during installation when any of the following conditions exist:

  — You are processing a large number of partitions.

  Estimate this allocation based on the following factors:

  - number of utilities that you are executing concurrently
  - number of partitions that you are processing concurrently
  - number of files that you are allocating dynamically

  — You are loading or unloading XML data and the XML table space is partition-by-growth.

  Estimate this allocation based on the following factors:

  - number of utilities that you are executing concurrently
  - number of XML columns that you are loading or unloading
  - value of MAXPARTITIONS (a minimum of 256 partitions in this case)
  - number of files that you are allocating dynamically

— 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

- Do not run an IBM utility that attempts to manipulate data within the same objects on which a BMC utility is currently processing.

- If BMCSTATS is processing multiple objects and encounters an object that is held by another utility, the BMCSTATS job issues a warning. The warning identifies the object and the utility that is using it. BMCSTATS continues processing the next object.

- If BMCSTATS is processing an object and another utility requires exclusive control of that object, the other utility stops execution at initialization time.

Maintaining the BMCSYNC table

When a utility abends, rows might remain in the BMCSYNC table. If you need to control expansion of this table, use one of the following methods to delete rows:

- Use the TERM restart parameter on the EXEC statement to delete rows from the BMCUTIL and BMCSYNC tables. Do not delete any rows for instances of utilities that are awaiting restart.

- Delete invalid rows in the BMCUTIL table. Do not delete any rows for instances of utilities that are awaiting restart.

  Then use the following SQL statement to delete rows from the BMCSYNC table.

```
DELETE
FROM creatorName.CMN_BMCSYNC
 WHERE UTILID NOT IN
  (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
```

NOTE

The names of the BMCUTIL and BMCSYNC tables might have been changed at your site during installation.

BMCTRANS table

Table 114 describes the contents of the BMCTRANS table, which contains information that RECOVERY MANAGER and Log Master use for transaction recovery. The table contains one row for each execution of Log Master (that is, one row for each log scan performed).

Table 114 BMCTRANS table (part 1 of 2)

Column Name	Data type	Description
USERID	CHAR(8) NOT NULL	transaction creator
TRANID	VARCHAR(18) NOT NULL	transaction ID
STARTTIME	TIMESTAMP NOT NULL WITH DEFAULT	transaction start time
PITRBA	CHAR(6) NOT NULL FORBIT DATA	RBA for point-in-time recovery
OUTDSNAME	VARCHAR(35) NOT NULL	output data set prefix for SQL statements or the logical log
STATE[a]	SMALLINT NOT NULL	level of recovery analysis performed
PITTIME	TIMESTAMP NOT NULL WITH DEFAULT	timestamp for the PIT RBA
SEQNO	SMALLINT NOT NULL	sequence number of the filter text
PITWKEST	FLOAT NOT NULL	work estimate
FILTERLINE	VARCHAR(1040) NOT NULL	text of the filter (may span more than one row)
UNDONUMROWSUPD	FLOAT	number of unique rows (RIDs) that are selected by the filter of the log scan
UNDOSUBSEQUPDROWS	FLOAT	total number of anomaly log records relating to one of the rows (RIDs) selected by the log scan
UNDOLOGRECROWS	FLOAT	number of unique rows (RIDs) that are affected by an anomaly log record
UNDOJOBSTATUS	SMALLINT	code indicating the status of an UNDO log scan: ■ **0** (no action taken) ■ **1** (Log Master execution started) ■ **2** (Log Master execution completed successfully with return code 0,4) ■ **3** (Log Master execution completed unsuccessfully with return code 8,12) ■ **4** (Log Master execution abnormally ended)

Table 114 BMCTRANS table (part 2 of 2)

Column Name	Data type	Description
REDOJOBSTATUS	SMALLINT	code indicating the status of a REDO log scan: ■ **0** (no action taken) ■ **1** (Log Master execution started) ■ **2** (Log Master execution completed successfully with return code 0,4) ■ **3** (Log Master execution completed unsuccessfully with return code 8,12) ■ **4** (Log Master execution abnormally ended)
ENDTIME	TIMESTAMP NOT NULL WITH DEFAULT	transaction end time
ACTION	SMALLINT	code indicating what recovery, if any, has been performed on the transaction

[a] If STATE equals 0, only UNDO analysis has been performed. If STATE is between 1 and 9999, UNDO and PIT analysis have been performed. If STATE is greater than 10000, UNDO, PIT, and REDO analysis have been performed.

BMCUTIL table

Table 115 on page 725 describes the BMCUTIL table, which contains information about utilities that are currently running or started. The utilities use the table to control the use of utility IDs. Each BMC utility must have a unique ID for restart purposes. If you have more than one BMC utility installed, all of these utilities should share the same BMCUTIL table.

The utilities insert rows into the BMCUTIL table during the UTILINIT phase and update the table as the job status changes. The utilities delete rows from the BMCUTIL table during the UTILTERM phase.

Table 115 BMCUTIL table (part 1 of 2)

Column name	Data type	Description
UTILID	CHAR(16)	utility identifier
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 uses only X.
UTILNAME	CHAR(8)	name of the executing utility: ■ CHECK ■ COPY ■ STATS ■ LOAD ■ RECOVER ■ REORG ■ UNLOAD
PHASE	CHAR(8)	current phase of the utility 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
RESTART	CHAR(1)	restart option: ■ **N** (not restart) ■ **P** (RESTART(PHASE)) ■ **Y** (RESTART) DASD MANAGER PLUS does not use this column. Although UNLOAD PLUS accepts the RESTART, RESTART(PHASE), NEW / RESTART, and NEW / RESTART(PHASE) parameters, the utility executes as though you had specified the NEW parameter.
NOTEID	CHAR(8)	TSO user ID to be notified DASD MANAGER PLUS does not use this column.

Table 115 BMCUTIL table (part 2 of 2)

Column name	Data type	Description
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. If you need to control expansion of this table, use one of the following methods to delete rows:

- Use the TERM restart parameter on the EXEC statement to delete rows from the BMCUTIL and BMCSYNC tables. Do not delete any rows for instances of utilities that are awaiting restart.

- Delete invalid rows in the BMCUTIL table. Do not delete any rows for instances of utilities that are awaiting restart.

Then use the following SQL statement to delete rows from the BMCSYNC table.

```
DELETE
FROM creatorName.CMN_BMCSYNC
 WHERE UTILID NOT IN
  (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
```

NOTE

The names of the BMCUTIL and BMCSYNC tables might have been changed at your site during installation.

BMCXCOPY table

Table 116 on page 728 describes the contents of the BMCXCOPY table, which the BMC utilities use for tracking the following types of registered copies:

- indexes that COPY PLUS has copied:

  — COPY NO index copies
  — DSNUM *n* index (nonpartitioned) copies
  — incremental index copies
  — index copies that are made at data set level

- Instant Snapshots made by COPY PLUS with the BMC EXTENDED BUFFER MANAGER (XBM) product or 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 116 BMCXCOPY table (part 1 of 4)

Column name	Data type	Description
DBNAME	CHAR(8)	name of the database
IXNAME	CHAR(8)	name of the index space or table space for Instant Snapshots and associated copies
DSNUM	INTEGER	data set number within the index or table space
ICTYPE	CHAR(1)	operation type: ■ **F** (COPY FULL YES; for COPY PLUS version 8.1 and later, online consistent copies) ■ **I** (COPY FULL NO) ■ **W** (REORG LOG NO) ■ **X** (REORG LOG YES) ■ **B** (REBUILD INDEX) ■ **P** (POINT-IN-TIME RECOVERY) ■ **C** (for COPY PLUS version 7.3 and earlier, online consistent copies)
ICDATE	CHAR(6)	date of the entry (*yymmdd*)
START_RBA	CHAR(6)	a 48-bit positive integer containing the relative byte location of a point in the DB2 recovery log The indicated point as follows: ■ for ICTYPE F, the starting point for all updates since the image copy was taken ■ for COPY_TYPE O, the minimum of the consistent point and the oldest inflight URID ■ *(RECOVERY MANAGER)* for ICTYPE C, the consistent log point for the copy — RBA for non-data-sharing systems — LRSN for data sharing systems
FILESEQNO	INTEGER	tape file sequence number of the copy
DEVTYPE	CHAR(8)	type of device on which the copy resides
IBMREQD	CHAR(1)	whether the row came from the basic machine-readable material (MRM) tape: ■ **N** (NO) ■ **Y** (YES)
DSNAME	CHAR(44)	name of the data set If STYPE V, DSNAME is the name of the VSAM data component.
ICTIME	CHAR(6)	time at which this row was inserted (*hhmmss*) The insertion takes place after the completion of the operation that the row represents.

Table 116 BMCXCOPY table (part 2 of 4)

Column name	Data type	Description
SHRLEVEL	CHAR(1)	SHRLEVEL parameter on COPY if ICTYPE F: ■ **C** (change) ■ **R** (reference)
DSVOLSER	VARCHAR(1784)	volume serial numbers of the data set Commas separate items in a list of 6-byte numbers. This column is blank if the data set is cataloged.
TIMESTAMP	TIMESTAMP	date and time when the row was inserted This is the date and time that are recorded in ICDATE and ICTIME. The use of TIMESTAMP over ICDATE and ICTIME is recommended, because later DB2 releases might not support the latter two columns.
ICBACKUP	CHAR(2)	type of image copy contained in the data set: ■ **LB** (data set contains local backup data) ■ **RP** (data set contains recovery system main data) ■ **RB** (data set contains recovery system backup data) ■ **blank** (data set contains local system main data or is not one of multiple copies)
ICUNIT	CHAR(1)	media on which the image copy data set is stored: ■ **D** (DASD) ■ **T** (tape) ■ **blank** (medium is neither tape nor DASD)
STYPE	CHAR(1)	type of copy: ■ **blank** (for ICTYPE=F) ■ **V** (Instant Snapshot or a VSAM data set) ■ **e** (encrypted copy)
PIT_RBA	CHAR(6)	point-in-time recovery: ■ X'000000000000' (for ICTYPE=F) ■ consistent point (for COPY_TYPE=O)
GROUP_MEMBER	CHAR(8)	data-sharing group member (the name of the SSID where the copy was made) This column is blank if you are not using data sharing.
OTYPE	CHAR(1)	type of object: ■ **T** (table) ■ **I** (index) ■ **i** (compressed index)
LOWDSNUM	INTEGER	not used
HIGHDSNUM	INTEGER	not used
COPYPAGESF	FLOAT(8)	number of pages written to the copy data set

Table 116 BMCXCOPY table (part 3 of 4)

Column name	Data type	Description
NPAGESF	FLOAT(8)	high-used RBA divided by the page size
CPAGESF	FLOAT(8)	total number of changed pages
JOBNAME	CHAR(8)	job name
AUTHID	CHAR(8)	authorization ID
OLDEST_VERSION	SMALLINT	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.
LOGICAL_PART	INTEGER	logical partition number
LOGGED	CHAR(1)	logging attribute of the table space: ■ **Y** (logged) ■ **N** (not logged) ■ **blank** (row inserted prior to DB2 version 9) For a non-LOB table space or index space, blank indicates that the logging attribute is logged.
TTYPE	CHAR(8)	row format for the table space or partition: ■ **RRF** (reordered row format) ■ **BRF** (basic row format)
INSTANCE	SMALLINT	instance number of the current base objects (table and index) The default value is 1.
RELCREATED	CHAR(1)	DB2 release that created the object If the release is earlier than version 9, the value is blank.
COPY_TYPE	CHAR(1)	type of copy: ■ **C** (cabinet copy) ■ **O** (online consistent copy) ■ **blank** (default value)
NOTE_VALUE	CHAR(4)	encoded value that quickly locates data for a specific space in a cabinet copy The default value is blank.
NOTE_TYPE	CHAR(1)	type of NOTE (issued by COPY PLUS): ■ **A** (ABS— tape) ■ **R** (REL— disk) ■ **F** (frame) ■ **blank** (default value)
OCC_COPY_RBA	CHAR(6)	original START_RBA of an online consistent copy The default value is blank.

Table 116 BMCXCOPY table (part 4 of 4)

Column name	Data type	Description
OCC_LOCKRULE	CHAR(1)	locking rule for a table space (not used for indexes): ■ **A** (for page level) ■ **R** (for row level) ■ **blank** (default value)
OCC_SPACE_ALTERED	CHAR(1)	whether the space was altered: ■ **Y** (altered) ■ **N** (not altered) ■ **blank** (default value)
CAB_BLOCKS	INTEGER	total number of frames written for a cabinet copy

Maintaining the BMCXCOPY table

Periodically, you should review BMCXCOPY and delete old rows to control its expansion. To delete all rows from the BMCXCOPY table that are older than 30 days, use the following statement as an example:

```
DELETE
FROM creatorName.CMN_BMCXCOPY
 WHERE DAYS(CURRENT TIMESTAMP) - DAYS(TIMESTAMP) > 30;
```

REORG PLUS user exits

This appendix presents the following topics:

Overview

The REORG PLUS for DB2 product allows you to customize your reorganization processing by using user-written exits. Table 117 lists the following information:

- sample user exits that REORG PLUS provides
- corresponding exit point 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 117 User exits that REORG PLUS provides

Exit routine purpose	Exit point name	Language	See page
creating data set name patterns	DSNUEXIT	■ Assembler ■ COBOL II ■ LE COBOL ■ C ■ LE C	735
managing the redefinition of DB2® VSAM data sets	DSRSEXIT	REXX	780
controlling updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS updates	TERMEXIT	REXX	795

Accessing the sample user exits

Source code for the sample user exits is distributed in the REORG PLUS *HLQ.LLQ*SAMP library (where *HLQ* is the high-level qualifier specified during installation and *LLQ* is the low-level qualifier or prefix set during installation). Table 118 lists the library members that contain the sample user exits.

Table 118 Library member names of user exits

User exit	Language	Library member name
DSNUEXIT	Assembler	ARUEDSNA
	COBOL II and LE COBOL	ARUEDSN2
	C	ARUEDSC
	LE C	ARUEDSL
DSRSEXIT	REXX	DSRSREXX
TERMEXIT	REXX	TERMREXX

Using DSNUEXIT to construct data set name patterns

The user exit for data set name patterns, DSNUEXIT, provides REORG PLUS with user-defined variables that you can use in constructing data set name patterns for the DSNPAT command or installation option, and the SPILDSNP installation or SPILLDSNPAT command option. For information about the command options, see Chapter 3, “Syntax of the REORG command.” For information about the installation options, see Appendix A, “REORG PLUS installation options.”

To use a sample DSNUEXIT user exit from the *HQL.LLQ*SAMP library, copy the appropriate member as the base code for your exit and edit the code. Then, compile or assemble and link the edited code into an authorized library so that the exit can be loaded during execution.

DSNUEXIT requirements and considerations

The DSNUEXIT user exit has the following requirements and considerations:

- REORG PLUS supports DSNUEXIT user exits that are written in the following languages:

  — assembler

  — COBOL II

  — IBM Language Environment COBOL (LE COBOL)

  — C

  — IBM Language Environment C (LE C)

- The library in which this exit resides must be included in your system’s LINKLIST or in your JOBLIB or STEPLIB.

- For COBOL II and C programs, the appropriate runtime libraries must be authorized and must be in your LINKLIST, JOBLIB, or STEPLIB.

- For LE COBOL and LE C programs, the appropriate language environment runtime libraries must be authorized and must be in your LINKLIST, JOBLIB, or STEPLIB.

- Your routine must be reentrant.

- You cannot use these exit routines with a DSNUTILB reorganization.

- Dynamic data set allocation must be active.

- You can supply the module name and program language in the DSNUEXIT installation or command option. The DSNUEXIT command option overrides any value in the installation options. For more information, see page 220 or page 641.

- The phase in which REORG PLUS invokes your user-written exit depends on whether you run a two-phase or single-phase reorganization:

  — For a two-phase reorganization, REORG PLUS invokes your user-written exit at the beginning of the UNLOAD phase.

  — For a single-phase reorganization, REORG PLUS invokes your user-written exit at the beginning of the REORG phase.

DSNUEXIT return codes

REORG PLUS communicates with the DSNUEXIT exit by passing parameters *to* the exit and receiving a return code and user-defined variables *from* the exit. Table 119 lists where you can find the return code.

Table 119 DSNUEXIT return code location

Language	Return code location
assembler	contents of register 15
■ COBOL II ■ LE_COBOL	contents of the *RETURN-CODE* variable
■ C ■ LE_C	returned by the return function

Valid return codes and their results are as follows:

- 0 indicates a good return and processing continues.
- Other return codes indicate an error and the utility terminates the job.

DSNUEXIT user-defined variables

The DSNUEXIT user exit returns user-defined variables to REORG PLUS. You can use these variables with the DSNPAT installation or command option to construct data set name patterns. REORG PLUS uses these patterns and the values returned for any user-defined variables to create data set names during dynamic data set allocation. You can also use these variables with the SPILDSNP installation option or

SPILLDSNPAT command option to construct patterns for data set name prefixes. REORG PLUS uses these patterns and the values returned for any user-defined variables to create the data set name prefix for the spill data sets used when running a SHRLEVEL CHANGE reorganization.

The following rules apply for creating a user-defined exit to create your variables:

- variable names must meet the following requirements:
  - —begin with an underscore character
  - —begin in the leftmost byte of the XPUVNAME field
  - —be padded on the right with blanks if the variable name is shorter than 9 bytes
  - —not contain embedded blanks
  - —contain the following national characters for nonblank characters after the underscore:
    - A through Z
    - 0 through 9
    - #, @, $

- variable data must meet the following requirements:

  - —begin in the leftmost byte of the XPUVDATA field
  - —be padded on the right with blanks if the variable data is shorter than 8 bytes
  - —not contain embedded blanks
  - —consist of the following characters for nonblank characters:
    - A through Z
    - 0 through 9
    - #, @, $, –, .

DSNUEXIT assembler user exit

This section provides a sample exit parameter block, variable mapping structure, and exit that you can use as an example of how to write this user exit in assembler.

Exit parameter block DSECT

The exit parameter block DSECT (Figure 70 on page 738) contains both input and output fields. Input fields pass vital information to the user exit, such as the database name and user ID. If you make any modifications to these input fields, they are disregarded on return. The output fields pass information about your user variables back to REORG PLUS.

Figure 70 DSNUEXIT assembler exit parameter block

```
*---------------------------------------------------------------------*
*                                                                     *
* 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
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
XPGRPNM  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

Table 120 describes the major DSECT fields for the DSNUEXIT assembler user exit parameter block and their uses.

Table 120 Major DSECT fields for the DSNUEXIT assembler user exit parameter block

Field	Description	Length
XPJOBN	job name	up to 8 bytes
XPSTEP	step name	up to 8 bytes
XPDBNAME	database name	up to 8 bytes
XPSPNAME	name of the table space or index space from the REORG command	up to 8 bytes
XPRTYPE	type of reorganization being performed, table space (TS) or index (IX)	2 bytes
XPUSER	USERID of the user running REORG PLUS	up to 8 bytes
XPSSID	DB2 subsystem ID	4 bytes
XPDATE	date of the execution of the utility, in the format *MMDDYY*	6 bytes
XPTIME	time of the execution of the utility, in the format *HHMMSS*	6 bytes
XPUTILID	utility ID	up to 16 bytes
XPDATE8	date of the execution of the utility, in the format *MMDDYYYY*	8 bytes
XPGRPNM	DB2 data sharing group name In a non-data sharing environment, the field contains the DB2 SSID.	NA
XPVCAT	VCATNAME specified in the DB2 catalog for the table space being reorganized, or for the first partition if the table space is partitioned	NA
XPDATEJ	Julian date of the execution of the utility, in the format *CCYYDDD*	7 bytes
XPUSRWD1...	work space.	up to 4 bytes each
XPUSRWD8		
XPUVAREA	area containing user-defined variable information For information about establishing user-defined variables, see page 736.	NA
XPUVENT#	maximum number of entries in the output variable area	up to 100 entries

Variable mapping block DSECT

The variable mapping block DSECT, as shown in Figure 71, defines the output area of the exit parameter DSECT. Using this DSECT allows you to easily address the variable definition table. The number of entries in this table (each entry consists of both a variable name and its value) must not exceed 100.

Figure 71 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, #, @, $, -, AND .                   *
*                                                                     *
*---------------------------------------------------------------------*
*
XPUVARS  DSECT ,                  MAP A VARIABLE ENTRY
XPUVNAME DS    CL9                VARIABLE NAME
XPUVDATA DS    CL8                VARIABLE DATA
XPUVENT$ EQU   *-XPUVNAME         LENGTH OF EACH ENTRY
```

DSECT fields

Table 121 describes the major DSECT fields for the DSNUEXIT assembler variable mapping block and their uses.

Table 121 Major DSECT fields for the DSNUEXIT assembler variable mapping block

Field	Description
XPUVNAME	name of the variable you are defining
XPUVDATA	value of the variable you are defining
XPUVENT$	length of each of these variable entries in the table

Sample assembler user exit

Figure 72 shows an example of this user exit written in assembler.

Figure 72 DSNUEXIT sample assembler user exit (part 1 of 11)

```
ARUEDSNA TITLE 'ARUEDSNA - REORG PLUS USER EXIT USER EXAMPLE - V42'
ARUEDSNA CSECT
ARUEDSNA AMODE 31
ARUEDSNA RMODE 24
*---------------------------------------------------------------------*
*                        D I S C L A I M E R                          *
*---------------------------------------------------------------------*
*                                                                     *
*  THIS IS A SAMPLE REORG PLUS USER EXIT.  THIS EXIT WOULD BE USED    *
*  IN ORDER TO DEFINE USER VARIABLES FOR BUILDING DATASET NAMES FOR   *
*  DYNAMIC WORKFILE ALLOCATION WITH THE DSNPAT KEYWORD AND FOR        *
*  SHRLEVEL CHANGE WITH THE SPILLDSNPAT 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                               *
*                                                                     *
*---------------------------------------------------------------------*
         SPACE
*---------------------------------------------------------------------*
```

Figure 72 DSNUEXIT sample assembler user exit (part 2 of 11)

```
*                             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- *
*                   CATIONS TO THESE FIELDS WILL BE DISREGARDED UPON  *
*                   RETURN TO THE CALLER.  THE OUTPUT FIELDS ARE      *
*                   PROVIDED TO ALLOW THE EXIT TO PASS INFORMATION    *
*                   BACK TO THE CALLER.                               *
*                                                                     *
*---------------------------------------------------------------------*
         EJECT
*---------------------------------------------------------------------*
* INTERNAL REGISTER USAGE                                             *
*---------------------------------------------------------------------*
*                                                                     *
*  0 -                                                                *
*  1 - ON ENTRY TO MODULE = A(ARUDSNXP BLOCK) COPIED TO R10           *
*  2 -                                                                *
*  3 -                                                                *
*  4 -                                                                *
*  5 -                                                                *
*  6 -                                                                *
*  7 -                                                                *
*  8 -                                                                *
*  9 -                                                                *
* 10 -                                                                *
* 11 -                                                                *
* 12 - BASEREG                                                        *
* 13 - SAVEAREA                                                       *
* 14 -                                                                *
* 15 -                                                                *
```

Figure 72 DSNUEXIT sample assembler user exit (part 3 of 11)

```
*                                                                     *
*---------------------------------------------------------------------*
*
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
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
XPGRPNM  DS    CL4                DATA SHARING GROUP NAME
XPVCAT   DS    CL8                VCAT NAME (FROM 1ST PART IF PARTITNED)
```

Figure 72 DSNUEXIT sample assembler user exit (part 4 of 11)

```
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
*
*---------------------------------------------------------------------*
*                                                                     *
* 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 .                   *
*                                                                     *
*---------------------------------------------------------------------*
```

Figure 72 DSNUEXIT sample assembler user exit (part 5 of 11)

```
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    18F                LOCAL SAVE AREA
DWORK    DS    D                  DOUBLEWORD WORK AREA
CWORK    DS    CL8                CHARACTER WORK AREA
WRKAREA$ EQU   *-WORKAREA
         EJECT
*
*---------------------------------------------------------------------*
*                                                                     *
*                     P R O G R A M   S T A R T                       *
*                                                                     *
*---------------------------------------------------------------------*
*
ARUEDSNA CSECT
*
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
*                                                                     *
* UNCOMMENT THE NEXT TWO INSTRUCTIONS TO DISABLE THIS EXIT.           *
*                                                                     *
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
*
*        LA    R15,0                    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 ADDRESSABLITY
*
         LR    R10,R1                   GET A(PASSED BLOCK)
         USING ARUDSNXP,R10             MAP IT
*                                                                     *
*---------------------------------------------------------------------*
```

Figure 72 DSNUEXIT sample assembler user exit (part 6 of 11)

```
* GET SOME WORKING STORAGE                                            *
*                                                                     *
*---------------------------------------------------------------------*
*
         XR    R11,R11                  ZERO R11
         GETMAIN RC,LV=WRKAREA$,LOC=BELOW  GET WORK AREA
         LTR   R15,R15                  OK?
         BNZ   DSNXRC08                 ..NO, ERROR
         LR    R11,R1                   R11 = A(USER WORK AREA)
         ST    R11,XPUSRWD1             SAVE A(USER WORK AREA)
         USING WORKAREA,R11             ADDRESS WORKAREA DSECT
         LA    R15,SAVE                 MY SAVE AREA@
         ST    R15,8(,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                       *
*                                                                     *
*   _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
```

Figure 72 DSNUEXIT sample assembler user exit (part 7 of 11)

```
         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
*
         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
```

Figure 72 DSNUEXIT sample assembler user exit (part 8 of 11)

```
         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 DELIMETER
         BZ    DSNX0200                  NONE ->
*
         LA    R2,XPUTILID              @(UTIL ID)
         LR    R4,R1                    SAVE @(DELIMETER)
         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,DSNXMVCU              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 DELIMETER)
         LA    R1,XPUTILID+L'XPUTILID   @(END OF UTILID)
         SR    R1,R5                    LENGTH OF SECOND PART OF UTILID
         BZ    DSNXRC00                  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,DSNXMVCU              MOVE UTILID PREFIX
*
         B     DSNXRC00                 RETURN RC=0
*
DSNX0200 DS    0H
```

Figure 72 DSNUEXIT sample assembler user exit (part 9 of 11)

```
*
         MVC   XPUVDATA,XPUTILID        GET FIRST 8 BYTES
*
         LA    R3,XPUVENT$(,R3)         @(NEXT ENTRY)
         MVC   XPUVNAME(XPUVENT$),BLANKS CLEAR ENTRY
*
         MVC   XPUVNAME,=CL9'_UTILSFX'  SET VAR NAME
         MVC   XPUVDATA,XPUTILID+8      GET NEXT 8 BYTES
*
         B     DSNXRC00                 RETURN RC=0
*
DSNXMVCU MVC   XPUVDATA(0),0(R5)        *** EXECUTED INSTRUCTION ***
*
         DROP  R3
*
         EJECT
*
*----------------------------------------------------------------------*
*                                                                      *
* SET RETURN CODE AND EXIT                                             *
*                                                                      *
*----------------------------------------------------------------------*
*
DSNXRC00 DS    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                            *
*                                                                      *
```

Figure 72 DSNUEXIT sample assembler user exit (part 10 of 11)

```
*---------------------------------------------------------------------*
*
*                0 1 2 3 4 5 6 7 8 9 A B C D E F
*
TRTAB    DC    X'00000000000000000000000000000000' 0
         DC    X'00000000000000000000000000000000' 1
         DC    X'00000000000000000000000000000000' 2
         DC    X'00000000000000000000000000000000' 3
         DC    X'FF000000000000000000FF0000FFFF' 4 (SP) . + |
         DC    X'0000000000000000000000000000FF00' 5 ;
         DC    X'FFFF0000000000000000FFFF00FF0000' 6 - / ¦(X'6A') , _
         DC    X'00000000000000000000FF000000FF00' 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
*
*                0 1 2 3 4 5 6 7 8 9 A B C D E F
*
MONTHTAB DC    H'0'                    JANUARY
         DC    H'31'                   FEBRUARY
         DC    H'59'                   MARCH
         DC    H'90'                   APRIL
         DC    H'120'                  MAY
         DC    H'151'                  JUNE
         DC    H'181'                  JULY
         DC    H'212'                  AUGUST
         DC    H'243'                  SEPTEMBER
         DC    H'273'                  OCTOBER
         DC    H'304'                  NOVEMBER
         DC    H'334'                  DECEMBER
*
LEAPTAB  DC    H'0'                    JANUARY
         DC    H'31'                   FEBRUARY
         DC    H'60'                   MARCH
         DC    H'91'                   APRIL
         DC    H'121'                  MAY
         DC    H'152'                  JUNE
         DC    H'182'                  JULY
         DC    H'213'                  AUGUST
         DC    H'244'                  SEPTEMBER
         DC    H'274'                  OCTOBER
         DC    H'305'                  NOVEMBER
         DC    H'335'                  DECEMBER
*
PL4PAT   DC    X'4021202020202020'     EDIT PATTERN
*
```

Figure 72 DSNUEXIT sample assembler user exit (part 11 of 11)

```
BLANKS   DC    CL17' '                 SOME BLANKS
*
         EJECT
         LTORG ,
         SPACE
         END   ARUEDSNA
```

DSNUEXIT COBOL II and LE COBOL user exit

This section provides a sample exit parameter record, variable mapping record, and exit that you can use as an example of how to write the DSNUEXIT user exit in COBOL or LE COBOL.

COBOL II and LE COBOL exit parameter record

The COBOL II and LE COBOL exit parameter record (Figure 73) contains both input and output fields. Input fields pass vital information to the user exit, such as the database name and user ID. If you make any modifications to these input fields, they are disregarded on return. The output fields pass information about your user variables back to REORG PLUS.

Figure 73 DSNUEXIT COBOL II and LE COBOL exit parameter record (part 1 of 2)

```
      *------------------------------------------------------------------
       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-DATE.
                   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).
```

Figure 73 DSNUEXIT COBOL II and LE COBOL exit parameter record (part 2 of 2)

```
                   15  EXIT-DATE8-DD          PIC 9(2).
                   15  EXIT-DATE8-YEAR        PIC 9(4).
                   15  FILLER REDEFINES EXIT-DATE8-YEAR.
                       20  EXIT-DATE8-CC      PIC 9(2).
                       20  EXIT-DATE8-YY      PIC 9(2).
               10  EXIT-GRPNM                 PIC X(4).
               10  EXIT-VCAT                  PIC X(8).
               10  EXIT-DATEJ.
                   15  EXIT-DATEJ-YEAR        PIC 9(4).
                   15  FILLER REDEFINES EXIT-DATEJ-YEAR.
                       20  EXIT-DATEJ-CC      PIC 9(2).
                       20  EXIT-DATEJ-YY      PIC 9(2).
                   15  EXIT-DATEJ-DDD         PIC 9(3).
               10  EXIT-FILLER2               PIC X(13).
05  WORK-AREA-ADDRESSES.
               10  WORK-AREA-1                PIC 9(4).
               10  WORK-AREA-2                PIC 9(4).
               10  WORK-AREA-3                PIC 9(4).
               10  WORK-AREA-4                PIC 9(4).
               10  WORK-AREA-5                PIC 9(4).
               10  WORK-AREA-6                PIC 9(4).
               10  WORK-AREA-7                PIC 9(4).
               10  WORK-AREA-8                PIC 9(4).
```

Parameter record fields

Table 122 describes the major fields for the DSNUEXIT COBOL II and LE COBOL user exit parameter record and their uses.

Table 122 Major parameter record fields for the DSNUEXIT COBOL II and LE COBOL user exit (part 1 of 2)

Field	Description	Length
EXIT-JOBNAME	job name	up to 8 bytes
EXIT-STEPNAME	step name	up to 8 bytes
EXIT-DBNAME	database name	up to 8 bytes
EXIT-TSNAME	name of the table space or index space from the REORG command	up to 8 bytes
EXIT-REORG-TYPE	type of reorganization being performed; table space (TS) or index (IX)	2 bytes
EXIT-USERID	USERID of the user running REORG PLUS	up to 8 bytes
EXIT-DB2-SSID	DB2 subsystem ID	4 bytes
EXIT-DATE	date of the execution of the utility, in the format *MMDDYY*	6 bytes
EXIT-TIME	time of the execution of the utility, in the format *HHMMSS*	6 bytes

Table 122 Major parameter record fields for the DSNUEXIT COBOL II and LE COBOL user exit (part 2 of 2)

Field	Description	Length
EXIT-UTILID-PARM	utility ID	up to 16 bytes
EXIT-DATE8	date of the execution of the utility, in the format *MMDDYYYY*	8 bytes
EXIT-GRPNM	DB2 data sharing group name In a non-data-sharing environment, the field contains the DB2 SSID.	NA
EXIT-VCAT	VCATNAME specified in the DB2 catalog for the table space being reorganized, or for the first partition, if the table space is partitioned	NA
EXIT-DATEJ	Julian date of the execution of the utility, in the format *CCYYDDD*	7 bytes
WORK-AREA-1...	8 parameters for work space	up to 4 bytes each
WORK-AREA-8		

COBOL ll and LE COBOL variable mapping record

The COBOL ll and LE COBOL variable mapping record (Figure 74) defines the output area of the exit parameter record. Using this record allows you to easily address the variable definition table. The number of entries in this table (each entry consists of both a variable name and its value) must not exceed 100.

Figure 74 DSNUEXIT COBOL II and LE COBOL variable mapping record

```
       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

Table 123 on page 754 describes the major variable mapping record fields for the DSNUEXIT COBOL ll and LE COBOL user exit and their uses.

Table 123 Major fields for the DSNUEXIT COBOL II and LE COBOL variable mapping record

Field	Description
USER-DEFINED-VARIABLE-TABLE	table containing user-defined variable information For information about establishing user-defined variables, see page 736.
VARIABLE-NAME	name of the variable you are defining
VARIABLE-VALUE	value of the variable you are defining

Sample COBOL II and LE COBOL user exit

Figure 75 shows an example of this user exit written in COBOL.

Figure 75 DSNUEXIT sample COBOL II and LE COBOL user exit (part 1 of 7)

```
      *----------------------------------------------------------------
      * ALL COBOL MODULES MUST BE COMPILED WITH DATA(31) AND DYNAM TO
      * EXECUTE PROPERLY!!!!!!!!
      *----------------------------------------------------------------
       IDENTIFICATION  DIVISION.
      *----------------------------------------------------------------
       PROGRAM-ID.     ARUEDSN2.
       AUTHOR.         BMC SOFTWARE
       DATE-WRITTEN.   AUGUST 1995.
       DATE-COMPILED.
      *----------------------------------------------------------------
      * ARUEDSN2 IS A SAMPLE DB2 COBOL II USER EXIT.

      * THIS IS A SAMPLE REORG PLUS USER EXIT. THIS EXIT WOULD BE USED
      * IN ORDER TO DEFINE USER VARIABLES FOR BUILDING DATA SET NAMES
      * FOR DYNAMIC WORKFILE ALLOCATION WITH THE DSNPAT KEYWORD AND FOR
      * SHRLEVEL CHANGE WITH THE SPILLDSNPAT 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 REOGANIZATION  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.
      *
      * PLEASE REVIEW ADDITIONAL DOCUMENTION IN THE REFERENCE MANUAL
      *----------------------------------------------------------------

       ENVIRONMENT     DIVISION.
       INPUT-OUTPUT    SECTION.
```

Figure 75 DSNUEXIT sample COBOL II and LE COBOL user exit (part 2 of 7)

```
 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.
```

Figure 75 DSNUEXIT sample COBOL II and LE COBOL user exit (part 3 of 7)

```
           05  FILLER REDEFINES YEAR-REMAINDER.
               10  YEAR-X                 PIC X(5).

      *-----------------------------------------------------------------
      *    THE FOLLOWING TWO TABLES ARE USED TO CALCULATE THE JULIAN
      *    DAY DEPENDING ON WHETHER THE YEAR IS A LEAP OR NOT.
      *-----------------------------------------------------------------

       01  NO-LEAP-MONTHS.
           05  JANUARY             PIC S9(3)  COMP-3  VALUE 00.
           05  FEBUARY             PIC S9(3)  COMP-3  VALUE 31.
           05  MARCH               PIC S9(3)  COMP-3  VALUE 59.
           05  APRIL               PIC S9(3)  COMP-3  VALUE 90.
           05  MAY                 PIC S9(3)  COMP-3  VALUE 120.
           05  JUNE                PIC S9(3)  COMP-3  VALUE 151.
           05  JULY                PIC S9(3)  COMP-3  VALUE 181.
           05  AUGUST              PIC S9(3)  COMP-3  VALUE 212.
           05  SEPTEMBER           PIC S9(3)  COMP-3  VALUE 243.
           05  OCTOBER             PIC S9(3)  COMP-3  VALUE 273.
           05  NOVEMBER            PIC S9(3)  COMP-3  VALUE 304.
           05  DECEMBER            PIC S9(3)  COMP-3  VALUE 334.
       01  FILLER REDEFINES NO-LEAP-MONTHS.
           05  MONTH-DAYS          PIC S9(3)  COMP-3  OCCURS 12 TIMES.

       01  LEAP-MONTHS.
           05  LEAP-JANUARY        PIC S9(3)  COMP-3  VALUE 00.
           05  LEAP-FEBRUARY       PIC S9(3)  COMP-3  VALUE 31.
           05  LEAP-MARCH          PIC S9(3)  COMP-3  VALUE 60.
           05  LEAP-APRIL          PIC S9(3)  COMP-3  VALUE 91.
           05  LEAP-MAY            PIC S9(3)  COMP-3  VALUE 121.
           05  LEAP-JUNE           PIC S9(3)  COMP-3  VALUE 152.
           05  LEAP-JULY           PIC S9(3)  COMP-3  VALUE 182.
           05  LEAP-AUGUST         PIC S9(3)  COMP-3  VALUE 213.
           05  LEAP-SEPTEMBER      PIC S9(3)  COMP-3  VALUE 244.
           05  LEAP-OCTOBER        PIC S9(3)  COMP-3  VALUE 274.
           05  LEAP-NOVEMBER       PIC S9(3)  COMP-3  VALUE 305.
           05  LEAP-DECEMBER       PIC S9(3)  COMP-3  VALUE 335.
       01  FILLER REDEFINES LEAP-MONTHS.
           05  LEAP-MONTH-DAYS     PIC S9(3)  COMP-3  OCCURS 12 TIMES.

      *-----------------------------------------------------------------
       LINKAGE SECTION.
      *-----------------------------------------------------------------
      *    THE TABLE CAN NOT OCCUR MORE THAN 100 TIME OR A STORAGE
      *    OVERLAY WILL OCCUR.
      *-----------------------------------------------------------------
       01  REORG-EXIT-PARMS.
           05  FIXED-PARM-VALUES.
               10  EXIT-JOBNAME              PIC X(8).
               10  EXIT-STEPNAME             PIC X(8).
               10  EXIT-DBNAME               PIC X(8).
```

Figure 75 DSNUEXIT sample COBOL II and LE COBOL user exit (part 4 of 7)

```
               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-DATE.
                   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(4).
                   15  FILLER REDEFINES EXIT-DATE8-YEAR.
                       20  EXIT-DATE8-CC     PIC 9(2).
                       20  EXIT-DATE8-YY     PIC 9(2).
               10  EXIT-GRPNM                PIC X(4).
               10  EXIT-VCAT                 PIC X(8).
               10  EXIT-DATEJ.
                   15  EXIT-DATEJ-YEAR       PIC 9(4).
                   15  FILLER REDEFINES EXIT-DATEJ-YEAR.
                       20  EXIT-DATEJ-CC     PIC 9(2).
                       20  EXIT-DATEJ-YY     PIC 9(2).
                   15  EXIT-DATEJ-DDD        PIC 9(3).
               10  EXIT-FILLER2              PIC X(13).

           05  WORK-AREA-ADDRESSES.
               10  WORK-AREA-1               PIC 9(4).
               10  WORK-AREA-2               PIC 9(4).
               10  WORK-AREA-3               PIC 9(4).
               10  WORK-AREA-4               PIC 9(4).
               10  WORK-AREA-5               PIC 9(4).
               10  WORK-AREA-6               PIC 9(4).
               10  WORK-AREA-7               PIC 9(4).
               10  WORK-AREA-8               PIC 9(4).


           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).
*-----------------------------------------------------------------
       PROCEDURE DIVISION USING REORG-EXIT-PARMS.
      *-----------------------------------------------------------------
```

Figure 75 DSNUEXIT sample COBOL II and LE COBOL user exit (part 5 of 7)

```
 0000-MAIN.
     MOVE ZERO TO RETURN-CODE.
     PERFORM 1000-PROCESS-DATE.
     GOBACK.

*-----------------------------------------------------------------
*    THE DATE IS PASSED IN A MMDDYYYY FORMAT AND CONVERTED TO
*    A JULIAN-DATE(WITH NO CENTURY) FORMAT OR A JULIAN-DATE
*    (WITH THE CENTURY) FORMAT.
*-----------------------------------------------------------------

 1000-PROCESS-DATE.
     MOVE EXIT-DATE8-DD TO JULIAN-DAYS.
     MOVE EXIT-DATE8-MM TO DAYS-SUBX.
     MOVE EXIT-DATE8-CC TO JULIAN-CC.
     MOVE EXIT-DATE8-YY TO JULIAN-YY.
     MOVE EXIT-DATE8-YEAR TO WORK-YEAR.

     DIVIDE WORK-YEAR BY FOUR
               GIVING YEAR-ANSWER
               REMAINDER YEAR-REMAINDER.

     IF YEAR-REMAINDER > ZERO
         THEN
             PERFORM 1100-NO-LEAP-YEAR
         ELSE
             DIVIDE WORK-YEAR BY ONE-HUNDRED
                      GIVING YEAR-ANSWER
                      REMAINDER YEAR-REMAINDER
             IF YEAR-REMAINDER > ZERO
                 THEN
                     PERFORM 1200-LEAP-YEAR
                 ELSE
                     DIVIDE WORK-YEAR BY FOUR-HUNDRED
                               GIVING YEAR-ANSWER
                          REMAINDER YEAR-REMAINDER
                     IF YEAR-REMAINDER > ZERO
                         THEN
                             PERFORM 1100-NO-LEAP-YEAR
                         ELSE
                             ADD LEAP-MONTH-DAYS(DAYS-SUBX)
                                              TO JULIAN-DAYS
                             PERFORM 1200-LEAP-YEAR
                     END-IF
             END-IF
     END-IF.

 1000-PROCESS-DATE-EXIT.
     EXIT.
```

Figure 75 DSNUEXIT sample COBOL II and LE COBOL user exit (part 6 of 7)

```
 1100-NO-LEAP-YEAR.
     ADD MONTH-DAYS(DAYS-SUBX) TO JULIAN-DAYS.
     PERFORM 1300-CENTURY.

 1100-NO-LEAP-YEAR-EXIT.
     EXIT.

 1200-LEAP-YEAR.
     ADD LEAP-MONTH-DAYS(DAYS-SUBX) TO JULIAN-DAYS.
     PERFORM 1300-CENTURY.

 1200-LEAP-YEAR-EXIT.
     EXIT.

 1300-CENTURY.
     PERFORM 1400-EDIT-UTILID.

 1300-CENTURY-EXIT.
     EXIT.

*-----------------------------------------------------------------
*    THE FULL 16 BYTES OF THE UTILITY ID PARM IS CHECKED FOR
*    A DELIMITER.  IF ONE IS FOUND AFTER THE FIRST 8 BYTES, IT
*    IS INCLUDED IN THE SUFFIX.  IF A DELIMITER IS FOUND IN THE
*    FIRST 8 BYTES, ONLY THOSE CHARACTERS/NUMBERS UP TO THE
*    DELIMITER, WILL BE MOVED INTO THE PREFIX FIELD.
*-----------------------------------------------------------------
*    THE DELIMITER CHARACTER IMMEDIATELY FOLLOWING THE FORWARD
*    SLASH ('/') IS A BROKEN VERTICAL BAR, HEX 6A.
*-----------------------------------------------------------------
 1400-EDIT-UTILID.
     MOVE ONE TO UTILID-POINTER, UTILID-TALLY.
     UNSTRING EXIT-UTILID-PARM
              DELIMITED BY ' ' OR '.' OR '+' OR '|' OR ';'
                 OR '-' OR '/' OR '¦' OR ',' OR '_' OR ':'
                 OR '=' OR '\' OR ' '
             INTO UTILID-PREFIX
                  COUNT IN UTILID-COUNTER
                  WITH POINTER UTILID-POINTER.
     IF UTILID-COUNTER = MAX-LENGTH
         THEN
             MOVE EXIT-SUFFIX TO UTILID-SUFFIX
         ELSE
             MOVE UTILID-COUNTER TO UTILID-POINTER
             ADD TWO TO UTILID-POINTER
             UNSTRING EXIT-UTILID-PARM
                 INTO UTILID-SUFFIX
                 WITH POINTER UTILID-POINTER
     END-IF.
     PERFORM 1500-UTILID-PARMS.
```

Figure 75 DSNUEXIT sample COBOL II and LE COBOL user exit (part 7 of 7)

```
       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-DATE(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

This section provides a sample exit parameter structure, variable mapping structure, and exit that you can use as an example of how to write the DSNUEXIT user exit in C.

C exit parameter structure

The C exit parameter structure (Figure 76) contains both input and output fields. Input fields pass vital information to the user exit, such as the database name and user ID. If you make any modifications to these input fields, they are disregarded on return. The output fields pass information about your user variables back to REORG PLUS.

Figure 76 DSNUEXIT C exit parameter structure (part 1 of 2)

```
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)            */
```

Figure 76 DSNUEXIT C exit parameter structure (part 2 of 2)

```
     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       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           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

Table 124 describes the major structure fields for the DSNUEXIT C user exit and their uses.

Table 124 Major structure fields for the DSNUEXIT C exit parameter (part 1 of 2)

Field	Description	Length
xpjobn	job name	up to 8 bytes
xpstep	step name	up to 8 bytes
xpdbname	database name	up to 8 bytes
xpspname	name of the table space or index space from the REORG command	up to 8 bytes
xprtype	type of reorganization being performed, table space (TS) or index (IX)	2 bytes
xpuser	USERID of the user running REORG PLUS	up to 8 bytes
xpssid	DB2 subsystem ID	4 bytes
xpdate	date of the execution of the utility, in the format *MMDDYY*	6 bytes
xptime	time of the execution of the utility, in the format *HHMMSS*	6 bytes
xputilid	utility ID	up to 16 bytes
xpdate8	date of the execution of the utility, in the format *MMDDYYYY*	8 bytes
xpgrpnm	DB2 data sharing group name In a non-data sharing environment, the field contains the DB2 SSID.	NA

Table 124 Major structure fields for the DSNUEXIT C exit parameter (part 2 of 2)

Field	Description	Length
xpvcat	VCATNAME specified in the DB2 catalog for the table space being reorganized, or for the first partition if the table space is partitioned	NA
xpdatej	Julian date of the execution of the utility, in the format *CCYYDDD*	7 bytes
xpusrwd1...xpusrwd8	8 parameters of work space	up to 4 bytes each

C exit variable mapping structure

The C exit variable mapping structure, as shown in Figure 77, defines the output area of the exit parameter structure. Using this structure allows you to easily address the variable definition table. The number of entries in this table (each entry consists of both a variable name and its value) must not exceed 100.

Figure 77 DSNUEXIT C exit variable mapping structure

```
     struct XPUVAREA {
         char xpuvname(|9|);
         char xpuvdata(|8|);
     } xpuvars(|XPUVENT|) ;
```

Structure fields

Table 125 describes the major variable mapping structure fields for the DSNUEXIT C user exit and their uses.

Table 125 Major variable mapping structure fields for the DSNUEXIT COBOL II and LE COBOL user exit

Field	Description
XPUVAREA	area containing user-defined variable information For information about establishing user-defined variables, see page 736.
xpuvname	name of the variable you are defining
xpuvdata	value of the variable you are defining
XPUVENT	maximum number of entries in the output variable area (up to 100 entries)

Sample C user exit

Figure 78 on page 763 shows an example of this user exit written in C.

Figure 78 DSNUEXIT sample C user exit (part 1 of 7)

```
/*--------------------------------------------------------------------*/
/*                       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 SPILLDSNPAT 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 O T E S                               */
/*--------------------------------------------------------------------*/
/*                                                                    */
/* ARUEDSC  IS CALLED ONCE AT THE BEGINNING OF THE UNLOAD PHASE FOR A */
/* TWO-PHASE REORGANIZATION OR AT THE BEGINNING OF THE REORG PHASE    */
/* FOR A SINGLE-PHASE REORGANIZATION.                                 */
/*                                                                    */
/* WHEN INVOKED IT IS PASSED THE ADDRESS OF A USER EXIT BLOCK         */
/*                  DESCRIBED BY THE ARUDSNXP STRUCT.                 */
/*                  THIS CALL IS FROM ASEMBLER MODULE DYNWDSNX        */
/*                                                                    */
/* UPON EXIT    RETRUN CODE FROM THIS EXIT                            */
/*                                                                    */
/*                      0     GOOD RETURN                             */
/*                                                                    */
/*                      OTHER ERROR RETURN, TERMINATE UTILITY         */
/*                                                                    */
/* ARUDSNXP STRUCT  THIS STRUCT CONTAINS BOTH INPUT                   */
/*                  AND OUTPUT FIELDS.  INPUT FIELDS ARE PROVIDED TO  */
/*                  PASS VITAL INFORMATION TO THIS EXIT.  ANY MODIFI- */
/*                  CATIONS TO THESE FIELDS WILL BE DISREGARDED UPON  */
/*                  RETURN TO THE CALLER.  THE OUTPUT FIELDS ARE      */
/*                  PROVIDED TO ALLOW THE EXIT TO PASS INFORMATION    */
/*                  BACK TO THE CALLER.                               */
/*                                                                    */
```

Figure 78 DSNUEXIT sample C user exit (part 2 of 7)

```
/*--------------------------------------------------------------------*/
/*                                                                    */
/*--------------------------------------------------------------------*/
/*                                                                    */
/* XPUVARS DEFINES THE OUTPUT AREA OF THE ARUDSNXP STRUCT AT TAG      */
/* XPUVAREA. YOU MAY USE THIS STRUCT TO EASILY REFERENCE THE ARRAY    */
/* DEFINED AT XPUVAREA.                                               */
/*                                                                    */
/* NOTE:                                                              */
/*                                                                    */
/*  THE NUMBER OF ENTRIES IN THE XPUVAREA ARRAY MUST NOT EXCEED 100   */
/*  OR YOU WILL ADDRESS BEYOND THE END OF THE STRUCTURE STORAGE       */
/*  PROVIDED FOR THIS ROUTINE.                                        */
/*                                                                    */
/* VARIABLE NAME:                                                     */
/*                                                                    */
/*  1.  MUST BEGIN WITH AN UNDERSCORE CHARACTER C'_'  X'6D'           */
/*  2.  MUST BEGIN IN THE LEFTMOST BYTE OF XPUVNAME FIELD             */
/*  3.  IF LESS THAN 9 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS */
/*  4.  MAY NOT CONTAIN EMBEDDED BLANKS                               */
/*  5.  NON-BLANK CHARACTERS AFTER THE LEFTMOST BYTE MUST BE NATIONAL */
/*      CHARACTERS WHICH ARE A THROUGH Z, 0 THROUGH 9, #, @, AND $    */
/*  6.  MUST BE UPPER CASE.                                           */
/*                                                                    */
/* VARIABLE DATA:                                                     */
/*                                                                    */
/*  1.  MUST BEGIN IN THE LEFTMOST BYTE OF XPUVDATA FIELD             */
/*  2.  IF LESS THAN 8 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS */
/*  3.  MAY NOT CONTAIN EMBEDDED BLANKS                               */
/*  4.  NON-BLANK CHARACTERS MUST BE 'NATIONAL' CHARACTERS WHICH ARE  */
/*      A THROUGH Z, 0 THROUGH 9, #, @, $, -, AND .                   */
/*                                                                    */
/*--------------------------------------------------------------------*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define XPUVENT 100
void dynwdsnx(struct XPVAREA*);
static int monthtab(||) = {
             0,                  /*   january   */
             31,                 /*   february  */
             59,                 /*   march     */
             90,                 /*   april     */
             120,                /*   may       */
             151,                /*   june      */
             181,                /*   july      */
             212,                /*   august    */
             243,                /*   september */
             273,                /*   october   */
             304,                /*   november  */
             334                 /*   december  */
```

Figure 78 DSNUEXIT sample C user exit (part 3 of 7)

```
} ;
static int leaptab(||) = {
             0,                   /*   january   */
             31,                  /*   february  */
             60,                  /*   march     */
             91,                  /*   april     */
             121,                 /*   may       */
             152,                 /*   june      */
             182,                 /*   july      */
             213,                 /*   august    */
             244,                 /*   september */
             274,                 /*   october   */
             305,                 /*   november  */
             335                  /*   december  */
} ;
struct arudsnpx {
     char       xpjobn(|8|);     /* jobname                          */
     char       xpstep(|8|);     /* stepname                         */
     char       xpdbname(|8|);   /* database name                    */
     char       xpspname(|8|);   /* space name                       */
     char       xprtype(|2|);    /* reorg type (ts or ix)            */
     char       xpresrv1(|2|);   /* reserved for REORG PLUS          */
     char       xpuser(|8|);     /* user id                          */
     char       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       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           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|);
```

Figure 78 DSNUEXIT sample C user exit (part 4 of 7)

```
    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                           */
/*                                                                  */
/*--------------------------------------------------------------------*/
    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);
```

Figure 78 DSNUEXIT sample C user exit (part 5 of 7)

```
    sscanf(yr,"%d",&wyear);
    sscanf(day,"%d",&wday);
    sscanf(month,"%d",&imonth);
    imonth = imonth - 1;
    if (wyear % 4 == 0 && wyear % 100 != 0 || wyear % 400 == 0)
       totdays = wday + leaptab(|imonth|);
    else
       totdays = wday + monthtab(|imonth|);
    sprintf(yr,"%d",wyear);
    sprintf(day,"%d",totdays);
/*---------------------------------------------------------------------*/
/*   clear the user variable area to blanks                            */
/*---------------------------------------------------------------------*/
    memset(tp->xpuvname,' ', sizeof(struct XPUVAREA));
/*---------------------------------------------------------------------*/
/*   clear the user variable area to blanks                            */
/*---------------------------------------------------------------------*/
    memcpy(tp->xpuvname,"_JDATE", 6);
    memcpy(tp->xpuvdata,"D",1);
    memcpy(tp->xpuvdata+1,xparm->xpdate8+6,2);
    memcpy(tp->xpuvdata+3,day,3);
/*---------------------------------------------------------------------*/
/*  sprintf(tp->xpuvdata,"D%-2.2s%3.3s",xparm->xpdate8+6,day);        */
/*  do not use this format as the '\0' used by sprintf will cause      */
/*  BMC51239I USER VARIABLE '_JDATE' HAS INVALID DATA                  */
/*---------------------------------------------------------------------*/
    tp++;
/*---------------------------------------------------------------------*/
/*   clear the user variable area to blanks                            */
/*---------------------------------------------------------------------*/
    memset(tp->xpuvname,' ', sizeof(struct XPUVAREA));
    memcpy(tp->xpuvname,"_JCDATE", 7);
    memcpy(tp->xpuvdata,"D",1);
    memcpy(tp->xpuvdata+1,yr,4);
    memcpy(tp->xpuvdata+5,day,3);
/*---------------------------------------------------------------------*/
/*  sprintf(tp->xpuvdata,"D%-4.4s%-3.3s",yr,day);                     */
/*  do not use this format as the '\0' used by sprintf will cause      */
/*  BMC51239I USER VARIABLE '_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 deliminator character                               */
/*  limit _UTILPFX and UTILSFX to 8 characters -DSN node limit         */
/*---------------------------------------------------------------------*/
```

Figure 78 DSNUEXIT sample C user exit (part 6 of 7)

```
    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                           */
```

Figure 78 DSNUEXIT sample C user exit (part 7 of 7)

```
/*---------------------------------------------------------------------*/
    memset(tp->xpuvname,' ', sizeof(struct XPUVAREA));
    memcpy(tp->xpuvname, "_UTILSFX", 8);
    memcpy(tp->xpuvdata, xparm->xputilid+sfxstrt, sfxlen);
/*---------------------------------------------------------------------*/
/*  sprintf(tp->xpuvdata,"%-8.8s",xparm->xputilid+8);                  */
/*  do not use this format as the '\0' used by sprintf will cause      */
/*  BMC51239I USER VARIABLE '_UTILSPX' HAS INVALID DATA                */
/*---------------------------------------------------------------------*/
    return (0);
}
```

DSNUEXIT LE C user exit

This section provides a sample exit parameter structure, variable mapping structure, and exit that you can use as an example of how to write the DSNUEXIT user exit in LE C.

LE C exit parameter structure

The LE C exit parameter structure (Figure 79) contains both input and output fields. Input fields pass vital information to the user exit, such as the database name and user ID. If you make any modifications to these input fields, they are disregarded on return. The output fields pass information about your user variables back to REORG PLUS.

Figure 79 DSNUEXIT LE C exit parameter structure (part 1 of 2)

```
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       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           xpusrwd1;     /* user word 1*/
```

Figure 79 DSNUEXIT LE C exit parameter structure (part 2 of 2)

```
      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

Table 126 describes the major structure fields for the DSNUEXIT LE C user exit and their uses.

Table 126 Major structure fields for the DSNUEXIT LE C user exit parameter

Field	Description	Length
xpjobn	job name	up to 8 bytes
xpstep	step name	
xpdbname	database name	
xpspname	name of the table space or index space from the REORG command	
xprtype	type of reorganization being performed, table space (TS) or index (IX)	2 bytes
xpuser	USERID of the user running REORG PLUS	up to 8 bytes
xpssid	DB2 subsystem ID	4 bytes
xpdate	date of the execution of the utility, in the format *MMDDYY*	6 bytes
xptime	time of the execution of the utility, in the format *HHMMSS*	
xputilid	utility ID	up to 16 bytes
xpdate8	date of the execution of the utility, in the format *MMDDYYYY*	8 bytes
xpgrpnm	DB2 data sharing group name In a non-data sharing environment, the field contains the DB2 SSID.	not applicable
xpvcat	VCATNAME specified in the DB2 catalog for the table space being reorganized, or for the first partition if the table space is partitioned	
xpdatej	Julian date of the execution of the utility, in the format *CCYYDDD*	7 bytes
xpusrwd1...xpusrwd8	8 parameters of work space	up to 4 bytes each

LE C exit variable mapping structure

The LE C exit variable mapping structure (Figure 80 on page 771) 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 80 DSNUEXIT LE C exit variable mapping structure

```
    struct XPUVAREA {
        char xpuvname[9];
        char xpuvdata[8];
    } xpuvars[XPUVENT] ;
```

Structure fields

Table 127 describes the major variable mapping structure fields for the DSNUEXIT LE C user exit and their uses.

Table 127 Major variable mapping structure fields for the DSNUEXIT LE C user exit

Field	Description
XPUVAREA	area containing user-defined variable information For information about establishing user-defined variables, see page 736.
xpuvname	name of the variable you are defining
xpuvdata	value of the variable you are defining
XPUVENT	maximum number of entries in the output variable area (up to 100 entries)

Sample LE C user exit

Figure 81 shows an example of this user exit written in LE C.

Figure 81 DSNUEXIT sample LE C user exit (part 1 of 7)

```
/*--------------------------------------------------------------------*/
/*                       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 SPILLDSNPAT 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.       */
/*                                                                    */
```

Figure 81 DSNUEXIT sample LE C user exit (part 2 of 7)

```
/* 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                               */
/*--------------------------------------------------------------------*/
/*                                                                    */
/* ARUEDSC  IS CALLED ONCE AT THE BEGINNING OF THE UNLOAD PHASE FOR A */
/* TWO-PHASE REORGANIZATION OR AT THE BEGINNING OF THE REORG PHASE    */
/* FOR A SINGLE-PHASE REORGANIZATION.                                 */
/*                                                                    */
/* WHEN INVOKED IT IS PASSED THE ADDRESS OF A USER EXIT BLOCK         */
/*                  DESCRIBED BY THE ARUDSNXP STRUCT.                 */
/*                  THIS CALL IS FROM ASEMBLER MODULE DYNWDSNX        */
/*                                                                    */
/* UPON EXIT    RETRUN CODE FROM THIS EXIT                            */
/*                                                                    */
/*                      0     GOOD RETURN                             */
/*                                                                    */
/*                      OTHER ERROR RETURN, TERMINATE UTILITY         */
/*                                                                    */
/* ARUDSNXP STRUCT  THIS STRUCT CONTAINS BOTH INPUT                   */
/*                  AND OUTPUT FIELDS.  INPUT FIELDS ARE PROVIDED TO  */
/*                  PASS VITAL INFORMATION TO THIS EXIT.  ANY MODIFI- */
/*                  CATIONS TO THESE FIELDS WILL BE DISREGARDED UPON  */
/*                  RETURN TO THE CALLER.  THE OUTPUT FIELDS ARE      */
/*                  PROVIDED TO ALLOW THE EXIT TO PASS INFORMATION    */
/*                  BACK TO THE CALLER.                               */
/*                                                                    */
/*--------------------------------------------------------------------*/
/*                                                                    */
/*--------------------------------------------------------------------*/
/*                                                                    */
/* XPUVARS DEFINES THE OUTPUT AREA OF THE ARUDSNXP STRUCT AT TAG      */
/* XPUVAREA. YOU MAY USE THIS STRUCT TO EASILY REFERENCE THE ARRAY    */
/* DEFINED AT XPUVAREA.                                               */
/*                                                                    */
/* NOTE:                                                              */
/*                                                                    */
/*  THE NUMBER OF ENTRIES IN THE XPUVAREA ARRAY MUST NOT EXCEED 100   */
/*  OR YOU WILL ADDRESS BEYOND THE END OF THE STRUCTURE STORAGE       */
/*  PROVIDED FOR THIS ROUTINE.                                        */
/*                                                                    */
/* VARIABLE NAME:                                                     */
/*                                                                    */
/*  1.  MUST BEGIN WITH AN UNDERSCORE CHARACTER C’_’  X’6D’           */
/*  2.  MUST BEGIN IN THE LEFTMOST BYTE OF XPUVNAME FIELD             */
/*  3.  IF LESS THAN 9 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS */
/*  4.  MAY NOT CONTAIN EMBEDDED BLANKS                               */
```

Figure 81 DSNUEXIT sample LE C user exit (part 3 of 7)

```
/*  5.  NON-BLANK CHARACTERS AFTER THE LEFTMOST BYTE MUST BE NATIONAL */
/*      CHARACTERS WHICH ARE A THROUGH Z, 0 THROUGH 9, #, @, AND $     */
/*  6.  MUST BE UPPER CASE.                                            */
/*                                                                     */
/* VARIABLE DATA:                                                      */
/*                                                                     */
/*  1.  MUST BEGIN IN THE LEFTMOST BYTE OF XPUVDATA FIELD              */
/*  2.  IF LESS THAN 8 BYTES, MUST BE PADDED ON THE RIGHT WITH BLANKS */
/*  3.  MAY NOT CONTAIN EMBEDDED BLANKS                                */
/*  4.  NON-BLANK CHARACTERS MUST BE ’NATIONAL’ CHARACTERS WHICH ARE   */
/*      A THROUGH Z, 0 THROUGH 9, #, @, $, -, AND .                    */
/*                                                                     */
/*-------------------------------------------------------------------*/
#pragma runopts(PLIST(MVS))
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define XPUVENT 100
static int monthtab[] = {
             0,                  /*   january   */
             31,                 /*   february  */
             59,                 /*   march     */
             90,                 /*   april     */
             120,                /*   may       */
             151,                /*   june      */
             181,                /*   july      */
             212,                /*   august    */
             243,                /*   september */
             273,                /*   october   */
             304,                /*   november  */
             334                 /*   december  */
} ;
static int leaptab[] = {
             0,                   /*   january   */
             31,                  /*   february  */
             60,                  /*   march     */
             91,                  /*   april     */
             121,                 /*   may       */
             152,                 /*   june      */
             182,                 /*   july      */
             213,                 /*   august    */
             244,                 /*   september */
             274,                 /*   october   */
             305,                 /*   november  */
             335                  /*   december  */
} ;
struct arudsnpx {
     char       xpjobn[8];       /* jobname                         */
     char       xpstep[8];       /* stepname                        */
     char       xpdbname[8];     /* database name                   */
     char       xpspname[8];     /* space name                      */
```

Figure 81   DSNUEXIT sample LE C user exit (part 4 of 7)

```
     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       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           xpusrwd1;     /* user word 1*/
     int           xpusrwd2;     /* user word 2*/
     int           xpusrwd3;     /* user word 3*/
     int           xpusrwd4;     /* user word 4*/
     int           xpusrwd5;     /* user word 5*/
     int           xpusrwd6;     /* user word 6*/
     int           xpusrwd7;     /* user word 7*/
     int           xpusrwd8;     /* user word 8*/
     struct XPUVAREA {
         char xpuvname[9];
         char xpuvdata[8];
     } xpuvars[XPUVENT] ;
} ;
int main    (int argc, char *argv[])
{
    char yr[5];
    char day[4];
    char month[4];
    char wuid[17];
    int i;
    int x;
    int l;
    int wday;
    int wyear;
    int iyear;
    int imonth;
    int totdays;
    int xdisable;
    int pfxlen;
    int sfxstrt;
    int sfxlen;
    struct XPUVAREA *tp;
    struct arudsnpx *xparm;
    xparm = (void *)argv[1];
    tp = xparm->xpuvars;
    x = l = wday = wyear = iyear = imonth = totdays = 0;
    xdisable = i = pfxlen = sfxstrt = sfxlen = 0;
/*-------------------------------------------------------------------*/
```

Figure 81 DSNUEXIT sample LE C user exit (part 5 of 7)

```
/*                                                                    */
/* 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                            */
/*                                                                    */
/*--------------------------------------------------------------------*/

    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,"%03.03d",totdays);
/*--------------------------------------------------------------------*/
/*   clear the user variable area to blanks                           */
/*--------------------------------------------------------------------*/
    memset(tp->xpuvname,’ ’, sizeof(struct XPUVAREA));
/*--------------------------------------------------------------------*/
/*   clear the user variable area to blanks                           */
/*--------------------------------------------------------------------*/
```

Figure 81 DSNUEXIT sample LE C user exit (part 6 of 7)

```
    memcpy(tp->xpuvname,"_JDATE", 6);
    memcpy(tp->xpuvdata,"D",1);
    memcpy(tp->xpuvdata+1,xparm->xpdate8+6,2);
    memcpy(tp->xpuvdata+3,day,3);
/*-------------------------------------------------------------------*/
/*  sprintf(tp->xpuvdata,"D%-2.2s%3.3s",xparm->xpdate8+6,day); */
/*  do not use this format as the ’\0’ used by sprintf will cause    */
/*  BMC51239I USER VARIABLE ’_JDATE’ HAS INVALID DATA                */
/*-------------------------------------------------------------------*/
    tp++;
/*-------------------------------------------------------------------*/
/*   clear the user variable area to blanks                          */
/*-------------------------------------------------------------------*/
    memset(tp->xpuvname,’ ’, sizeof(struct XPUVAREA));
    memcpy(tp->xpuvname,"_JCDATE", 7);
    memcpy(tp->xpuvdata,"D",1);
    memcpy(tp->xpuvdata+1,yr,4);
    memcpy(tp->xpuvdata+5,day,3);
/*-------------------------------------------------------------------*/
/*  sprintf(tp->xpuvdata,"D%-4.4s%-3.3s",yr,day);                    */
/*  do not use this format as the ’\0’ used by sprintf will cause    */
/*  BMC51239I USER VARIABLE ’_JCDATE’ HAS INVALID DATA               */
/*-------------------------------------------------------------------*/
/*                                                                   */
/* build _utilpfx and _utilsfx user variables                        */
/*                                                                   */
/*-------------------------------------------------------------------*/
    tp++;
    sprintf(wuid,"%16.16s",xparm->xputilid);
/*-------------------------------------------------------------------*/
/*  get the length of xputilid                                       */
/*  find the first deliminator 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 ’=’:
```

Figure 81 DSNUEXIT sample LE C user exit (part 7 of 7)

```
            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                            */
/*---------------------------------------------------------------------*/
    memset(tp->xpuvname,’ ’, sizeof(struct XPUVAREA));
    memcpy(tp->xpuvname, "_UTILSFX", 8);
    memcpy(tp->xpuvdata, xparm->xputilid+sfxstrt, sfxlen);
/*---------------------------------------------------------------------*/
/*  sprintf(tp->xpuvdata,"%-8.8s",xparm->xputilid+8);                  */
/*  do not use this format as the ’\0’ used by sprintf will cause      */
/*  BMC51239I USER VARIABLE ’_UTILSPX’ HAS INVALID DATA                */
/*---------------------------------------------------------------------*/
    return (0);
}
```

Using a DSRSEXIT or TERMEXIT user exit

You can use the following user exits written in REXX:

- DSRSEXIT to manage the redefinition of DB2 VSAM data sets (see page 780)
- TERMEXIT to control updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS (see page 795)

This section describes the following information common to all REXX exits that you can use with REORG PLUS:

- requirements and restrictions (next section)
- variables and return codes (see page 779)

To use a sample REXX user exit from the *HQL.LLQ*SAMP library, copy the appropriate member as the base code for your exit, edit the code, and save the modified exit in your SYSEXEC library.

To run a REXX exit, add the following DD statements to the JCL for your REORG PLUS job:

```
//SYSEXEC DD DSN=name,DISP=SHR
//SYSTSPRT DD SYSOUT=*
```

In the first DD statement, *name* is the library that contains your REXX exit. The second DD statement specifies where all of the output from the REXX 'SAY' statements are to be routed.

DSRSEXIT and TERMEXIT common restrictions

The REXX user exits have the following restrictions:

- You cannot use these exit routines with a DSNUTILB reorganization.
- 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 and TERMEXIT common variables and return codes

REORG PLUS communicates with REXX exits through REXX variables. Table 128 lists the variables that are common to all REXX exits. For the list of variables unique to a particular REXX exit, see the section on 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.

REORG PLUS communicates with REXX exits through REXX variables. Table 128 lists the variables that REORG PLUS passes to *all* REXX user exits. You cannot change the value of any of these variables. Each exit also has variables unique to it, which are described in the section for that user exit.

Table 128 Variables common to all REXX exits (part 1 of 2)

Variable name	Description
BMC_EXIT_POINT	name of the exit point
BMC_UTILITY_NAME	REORG
BMC_REORG_TYPE	reorganization type, either 'TABLESPACE or INDEX
BMC_UTILITY_SHRLEVEL	the SHRLEVEL of the reorganization: ■ N for SHRLEVEL NONE ■ U for SHRLEVEL REFERENCE UNLOADONLY ■ R for SHRLEVEL REFERENCE ■ C for SHRLEVEL CHANGE
BMC_UTIL_ID	UTILID from the EXEC parameters REORG PLUS truncates longer utility IDs to eight characters.
BMC_UTILPFX	BMC utility ID prefix
BMC_UTILSFX	BMC utility ID suffix
BMC_DB2_RELEASE	three-character version, release, and maintenance number (VRM) of the DB2 release
BMC_JOBNAME	JOB name used in the JCL
BMC_STEPNAME	STEP name used in the JCL
BMC_DBNAME	database containing the space being used for this data set allocation
BMC_TSIX	table space or index space name
BMC_USERID	job user ID You must have a security package to use this variable.

Table 128 Variables common to all REXX exits (part 2 of 2)

Variable name	Description
BMC_VCATNAME	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
BMC_DB2_SSID	DB2 subsystem ID
BMC_DB2_GROUPNAME	DB2 data sharing group name In a non-data sharing environment, this variable contains the DB2 SSID.
BMC_DATE	current date (in the form *MMDDYY*)
BMC_DATEJ	current Julian date (in the form *YYYYDDD*)
BMC_DATE8	current date (in the form *MMDDYYYY*)
BMC_TIME	current time (in the form *HHMMSS*)
BMC_TIME4	current time (in the form *HHMM*)

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 781)

- update the DB2 catalog with the changed values for use with subsequent allocations unless you indicate otherwise (see “Resizing DB2 objects” on page 781)

- selectively tell REORG PLUS to specify REDEFINE NO for an object (see “Setting REDEFINE NO” on page 781)

- specify the order of the storage group volumes to use for each object (see “Ordering storage group volumes” on page 782)

- restrict the volume list to a subset of the original storage group volume list (see “Ordering storage group volumes” on page 782)

- add the DATACLAS, MGMTCLAS, and STORCLAS for storage groups (see "Adding SMS classes" on page 783)

NOTE

Within the exit, if you specify REDEFINE NO for an object and you also change the primary or secondary quantity for that same object, REORG PLUS ignores REDEFINE NO and issues message BMC50307I.

Resizing DB2 objects

Resizing your DB2 objects during a reorganization allows you to

- reclaim unused space
- provide additional space for growth
- manage the object size at the partition level
- use the new size for subsequent allocations by altering the DB2 catalog to reflect the changes

After it has completed object redefinition, REORG PLUS issues an SQL ALTER TABLESPACE or ALTER INDEX statement to update the DB2 catalog with the modified primary and secondary quantities for use with subsequent allocations. You can modify a variable to prevent REORG PLUS from issuing this statement. For information about how to resize objects, see "Modifying the variables for resizing the data sets" on page 788.

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 789.

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
```

Ordering 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 789.

Adding 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

You can use the DSRSEXIT user exit only for storage-group-defined objects. If you want to alter the size of VCAT-defined objects, use the SYSIDCIN data set. For more information, see “SYSIDCIN data set” on page 345.

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 195.

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.

To use the DSRSEXIT exit

1 Specify the name of your user exit in one of the following ways:

  - On the DSRSEXIT installation option, replace NONE with the name of your REXX user exit. For information about the DSRSEXIT installation option, see Appendix A, “REORG PLUS installation options.”

  - On the REORG command, specify the DSRSEXIT option with the name of your REXX user exit. For information about the DSRSEXIT command option, see “exitPoint” on page 220.

2 Include the following DD statement in your JCL, where *dataSetName* is the name of the library containing the REXX exit:

```
//SYSEXEC DD DSN=dataSetName,DISP=SHR
```

3 Include the following DD statement, where all output from REXX 'SAY' statements will be routed:

```
//SYSTSPRT DD SYSOUT=*
```

In a RACF® environment, REORG PLUS uses the DB2 RACF ID when writing to SYSTSPRT from the REXX exit if OPNDB2ID=YES. For more information, see "Using RACF and OPNDB2ID=YES" on page 67 and the OPNDB2ID=YES installation option on page 662.

DSRSEXIT variables

In addition to the variables common to all REXX exits (Table 128 on page 779), REORG PLUS passes other variables to the DSRSEXIT user exit. Table 129 on page 785 describes the variables specific to this exit, grouping the variables according to whether they apply to any object type, a table space only, or an index only. Table 129 also lists the following information about these variables:

- variable name
- brief description of the variable

  If the variable is found in the DB2 catalog or in the integrated catalog facility (ICF), the table lists the corresponding DB2 or ICF column name. If the variable is specific to REORG PLUS, the table lists the values that you can enter.

- whether you can update the variable within the exit (yes or no)

For a more complete explanation of a variable and its valid values, refer to the IBM documentation. For more information about the variables that you can change, see "Modifying the variables for resizing the data sets" on page 788.

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 129 Variables that REORG PLUS passes to DSRSEXIT (part 1 of 3)

Variable name	Description and DB2 or ICF column name, or valid values	Update?
Variables that apply to all objects		
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
BMC_HIGH_USED_RBN	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.	no
BMC_HIGH_ALLOC_RBN	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.	no
BMC_PRIMARY_QUANTITY	primary quantity, in 4-KB pages, to be allocated (PRIQTY)	**yes**
BMC_SECONDARY_QUANTITY	secondary quantity, in 4-KB pages, to be allocated (SECQTY)	**yes**
BMC_PRIMARY_SPACE_ICF	primary space currently being used (PRIMSPAC)	no
BMC_SECONDARY_SPACE_ICF	secondary space currently being used (SCONSPAC)	no
BMC_NBR_EXTENTS	number of extents (NOEXTNT)	no
BMC_ICF_ALLOC_TYPE	allocation type in tracks (TRK), cylinders (CYL), or blocks (BLK) (SPACOPTN)	no
BMC_DB2_STORTYPE	VCAT (E) or STOGROUP (I) indicator (STORTYPE)	no
BMC_ESTIMATED_CARDINALITY	estimate of the number of rows involved in the reorganization, from ANALYZE (integer)	no
BMC_OBJECT_PART_NUMBER	current partition number of the table space or index (PARTITION)	no
BMC_OBJECT_COMPONENT_NUMBER	data set number from the last node of the DB2 linear data set name (for example, 001 from .A001)	no
BMC_DB2_FREEPAGE	number of pages loaded before a page is left as free space (FREEPAGE)	no
BMC_DB2_PCTFREE	percentage of each page left as free space (PCTFREE)	no
BMC_DB2_VCATNAME	data set name HLQ node of storage group or ICF catalog used for space allocation (VCAT NAME)	no
BMC_DB2_STORNAME	storage group name (STORNAME)	no

Table 129 Variables that REORG PLUS passes to DSRSEXIT (part 2 of 3)

Variable name	Description and DB2 or ICF column name, or valid values	Update?
BMC_DB2_DATACLAS	name of the SMS DATACLAS, taken from the DSN6SPRM macro in DSNZPARM	no
BMC_DB2_MGMTCLAS	specify the name of the SMS MGMTCLAS to add **Note**: You cannot use the DSRSEXIT to change an existing specified SMS MGMTCLAS.	no
BMC_DB2_STORCLAS	specify the name of the SMS STORCLAS to add **Note**: You cannot use the DSRSEXIT to change an existing specified SMS STORCLAS.	no
BMC_DATASET_NAME	fully qualified data set name	no
BMC_OBJECT_TYPE_IND	table space (TS) or index space (IX)	no
BMC_ALTER_DB2_CATALOG	whether to update the DB2 catalog (YES or NO) The default is YES.	yes
BMC_REDEFINE_OBJECT	whether REORG PLUS should use REDEFINE YES (YES) or REDEFINE NO (NO) for the current object The default is YES.	yes
BMC_REORDER_STOGROUP_VOLUMES	whether REORG PLUS should use the storage group volume list returned from the exit (YES) or ignore it (NO) The default is NO.	yes
BMC_STOGROUP_VOLUME.0 BMC_STOGROUP_VOLUME.*n*	stem variable that contains the number of volumes in the storage group list. BMC_STOGROUP_VOLUME.0=*n*, where *n* is the number of volumes. BMC_STOGROUP_VOLUME.*1 through* BMC_STOGROUP_VOLUME.*n* contain the actual names of volumes in the storage group	yes
Variables that apply only to a table space		
BMC_SYSTABLESPACE_PARTITIONS	number of partitions of the table space or index (PARTITIONS)	no
BMC_SYSTABLESPACE_PGSIZE	size of pages in the table space in KB (PGSIZE)	no
BMC_SYSTABLESPACE_SEGSIZE	number of pages in each segment of a segmented table space (0 if not segmented) (SEGSIZE)	no
BMC_SYSTABLESPACE_NTABLES	number of tables defined in the table space (NTABLES)	no
BMC_SYSTABLESPACE_MAXROWS	maximum number of rows that DB2 will place on a data page (MAXROWS)	no
BMC_SYSTABLESPACE_DSSIZE	maximum size of a data set in KB (DSSIZE)	no

Table 129 Variables that REORG PLUS passes to DSRSEXIT (part 3 of 3)

Variable name	Description and DB2 or ICF column name, or valid values	Update?
BMC_SYSTABLESPACE_NACTIVE	number of active pages in the table space (NACTIVEF)	no
BMC_SYSTABLEPART_COMPRESSED	indicator that table space is compressed (COMPRESS)	no
Variables that apply only to an index		
BMC_SYSINDEXES_NAME	name of the index (NAME)	no
BMC_SYSINDEXES_DBNAME	name of the data base that contains the index (DBNAME)	no
BMC_SYSINDEXES_CREATOR	authorization ID of the owner of the index (CREATOR)	no
BMC_SYSINDEXES_INDEXSPACE	name of the index space (INDEXSPACE)	no
BMC_SYSINDEXES_CLUSTERING	whether CLUSTER was specified when the index was created (CLUSTERING)	no
BMC_SYSINDEXES_CLUSTERED	whether the table is actually clustered by the index (CLUSTERED)	no
BMC_SYSINDEXES_FULLKEYCARD	number of distinct values of the key (FULLKEYCARD(F))	no
BMC_SYSINDEXES_UNIQUERULE	whether the index is unique (UNIQUERULE)	no
BMC_SYSINDEXES_PIECESIZE	maximum size of a data set in KB for nonpartitioned indexes (PIECESIZE)	no
BMC_SYSINDEXPART_LEAFDIST	100 times the average number of leaf pages between successive active leaf pages of the index (LEAFDIST)	no

If you are redefining a nonpartitioned DB2 object, for more information, see “Sample DSRSEXIT REXX user exit” on page 790.

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 788)

- bypass redefining the data set (see “Modifying the variable for selective REDEFINE” on page 789)

- order the storage group volumes (see “Modifying the variables for the storage group volumes” on page 789)

Modifying the variables for resizing the data sets

You can use the information in the variables to help you determine the primary and secondary quantities to allocate. Because DB2 stores the primary and secondary values in 4-KB pages in its catalog, the values passed to the exit represent 4-KB pages. To modify the variables BMC_PRIMARY_QUANTITY or BMC_SECONDARY_QUANTITY within the exit, set the quantity to the number of 4-KB pages that you want allocated. REORG PLUS uses the values that you set when it redefines the DB2 VSAM data sets.

REORG PLUS also issues an SQL ALTER TABLESPACE or ALTER INDEX statement to update the DB2 catalog with the modified primary and secondary quantities unless you set the variable BMC_ALTER_DB2_CATALOG to NO. The DB2 catalog tables that REORG PLUS updates are SYSIBM.SYSTABLEPART and SYSIBM.SYSINDEXPART.

NOTE

If you are reorganizing a partition-by-growth table space, REORG PLUS does not issue the ALTER statement to update the DB2 catalog.

If you use the default of YES for the BMC_ALTER_DB2 _CATALOG variable to have REORG PLUS update the DB2 catalog, you will also need the following authorities:

- For the ALTER TABLESPACE statement, you will need *one* of these authorities:

  — ownership of the table space
  — DBADM authority for its database
  — SYSADM or SYSCTRL authority

- For the ALTER INDEX statement, you will need *one* of these authorities:

  — ownership of the index
  — ownership of the table on which the index is defined
  — DBADM authority for the database that contains the table
  — SYSADM or SYSCTRL authority

Modifying the variable for selective REDEFINE

If you do not want REORG PLUS to perform VSAM DELETE / DEFINE processing for a particular object, set BMC_REDEFINE_OBJECT to NO. Bypassing this processing significantly improves performance and can be especially useful in an SAP environment, where many indexes exist but most of them have not grown.

If you use the DSRSEXIT user exit to alter the primary or secondary quantity for an object and also set BMC_REDEFINE_OBJECT to NO, REORG PLUS ignores the REDEFINE NO request and issues message BMC50307I.

If you set BMC_REDEFINE_OBJECT to NO for a staging data set, you must have predefined the data set. Otherwise, REORG PLUS terminates the job.

Modifying the variables for the storage group volumes

When the DSRSEXIT gets control, the variable BMC_STOGROUP_VOLUME.0 contains the number of volumes in the storage group (and therefore the number of stem variables containing volume names). The stem variables BMC_STOGROUP_VOLUME.1 through n contain the volume names in the order that DB2 returns them from SYSIBM.SYSVOLUMES. You can change this order by assigning new values to the list of stem variables. Any changes that you make to the storage group order are in effect only for the current object.

You can reorder and return the entire original list or any subset of the list, but you cannot add any volume that is not in the original storage group list. If you add a volume name that is not in the original list, REORG PLUS terminates the job.

To change the volume order

1. Set the BMC_REORDER_STOGROUP_VOLUMES variable to YES.

   If you do not set this variable to YES, REORG PLUS will ignore any changes that you make to the original volume list.

2. Set a stem variable BMC_STOGROUP_VOLUME.*n* to the name of a storage group volume.

   You will have one BMC_STOGROUP_VOLUME.*n* variable for each volume in your list, incrementing *n* for each volume.

For example, suppose REORG PLUS returned these volumes in the original storage group list:

- BMC_STOGROUP_VOLUME.0=5
- BMC_STOGROUP_VOLUME.1=BMC001
- BMC_STOGROUP_VOLUME.2=BMC002
- BMC_STOGROUP_VOLUME.3=BMC003
- BMC_STOGROUP_VOLUME.4=BMC004
- BMC_STOGROUP_VOLUME.5=BMC005

To have REORG PLUS allocate the data sets on volume BMC005 first, and then BMC004, followed by BMC001, BMC003, and BMC002, you would leave the stem variable BMC_STOGROUP_VOLUME.0 set to 5 and change the values in the remaining stem variables as follows:

- BMC_STOGROUP_VOLUME.1=BMC005
- BMC_STOGROUP_VOLUME.2=BMC004
- BMC_STOGROUP_VOLUME.3=BMC001
- BMC_STOGROUP_VOLUME.4=BMC003
- BMC_STOGROUP_VOLUME.5=BMC002

To use a subset of the original volume list, set the stem variable BMC_STOGROUP_VOLUME.0 to the number of volumes that you want REORG PLUS to consider. (The maximum value for this stem variable is the total number of volumes in the STOGROUP.)

To use volumes BMC002, BMC003, and BMC004 (a subset of the volumes listed in the preceding example), you would change the values in the stem variables as follows:

- BMC_STOGROUP_VOLUME.0=3
- BMC_STOGROUP_VOLUME.1=BMC002
- BMC_STOGROUP_VOLUME.2=BMC003
- BMC_STOGROUP_VOLUME.3=BMC004

Sample DSRSEXIT REXX user exit

Figure 82 on page 791 provides a sample REXX exit that shows one way of setting the variables in the DSRSEXIT user exit.

Figure 82 Sample DSRSEXIT REXX user exit (part 1 of 5)

```
/* 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'. THUS  */
/*    ALL PRIMARY/SECONDARY QUANTITY CHANGES WILL AUTOMATICALLY BE  */
/*    STAGED FOR SQL ALTER AFTER THE RELOAD (OR REORG) PHASE        */
/*    IS COMPLETE.                                                  */
/* 5. DISPLAYS THE VOLUME LIST FOR THE CURRENT STOGROUP.            */
/*                                                                  */
/* **************************************************************** */
/*                                                                  */
```

Figure 82 Sample DSRSEXIT REXX user exit (part 2 of 5)

```
/* IMPORTANT NOTE REGARDING REXX EXITS:                             */
/*    YOU MUST *NOT* PERFORM A CALL TO ANOTHER REXX PROGRAM !!      */
/*    DOING SO WILL CAUSE VALUES IN YOUR REXX VARIABLES TO BE LOST. */
/*    IT *IS* OK TO USE 'CALL' TO REFERENCE LABELED SUBROUTINES     */
/*    LOCATED WITHIN THE SAME REXX PROGRAM.                         */
/*                                                                  */
/* **************************************************************** */
SAY '**** START *********** ' BMC_EXIT_POINT ' ************'
SAY 'BMC_EXIT_POINT               ='     BMC_EXIT_POINT
SAY 'BMC_UTIL_ID                  ='     BMC_UTIL_ID
SAY 'BMC_UTILITY_NAME             ='     BMC_UTILITY_NAME
SAY 'BMC_DB2_RELEASE              ='     BMC_DB2_RELEASE
SAY 'BMC_JOBNAME                  ='     BMC_JOBNAME
SAY 'BMC_STEPNAME                 ='     BMC_STEPNAME
SAY 'BMC_DBNAME                   ='     BMC_DBNAME
SAY 'BMC_TSIX                     ='     BMC_TSIX
SAY 'BMC_USERID                   ='     BMC_USERID
SAY 'BMC_VCATNAME                 ='     BMC_VCATNAME
SAY 'BMC_DB2_SSID                 ='     BMC_DB2_SSID
SAY 'BMC_DB2_GROUPNAME            ='     BMC_DB2_GROUPNAME
SAY 'BMC_DATE                     ='     BMC_DATE
SAY 'BMC_TIME                     ='     BMC_TIME
SAY 'BMC_DATE8                    ='     BMC_DATE8
SAY 'BMC_TIME4                    ='     BMC_TIME4
SAY 'BMC_DATEJ                    ='     BMC_DATEJ
SAY 'BMC_UTILPFX                  ='     BMC_UTILPFX
SAY 'BMC_UTILSFX                  ='     BMC_UTILSFX
SAY 'BMC_HIGH_USED_RBA            ='     BMC_HIGH_USED_RBA
SAY 'BMC_HIGH_ALLOC_RBA           ='     BMC_HIGH_ALLOC_RBA
SAY 'BMC_HIGH_USED_RBN            ='     BMC_HIGH_USED_RBN
SAY 'BMC_HIGH_ALLOC_RBN           ='     BMC_HIGH_ALLOC_RBN
SAY 'BMC_PRIMARY_QUANTITY         ='     BMC_PRIMARY_QUANTITY
SAY 'BMC_SECONDARY_QUANTITY       ='     BMC_SECONDARY_QUANTITY
SAY 'BMC_PRIMARY_SPACE_ICF        ='     BMC_PRIMARY_SPACE_ICF
SAY 'BMC_SECONDARY_SPACE_ICF      ='     BMC_SECONDARY_SPACE_ICF
SAY 'BMC_NBR_EXTENTS              ='     BMC_NBR_EXTENTS
SAY 'BMC_ICF_ALLOC_TYPE           ='     BMC_ICF_ALLOC_TYPE
SAY 'BMC_DB2_STORTYPE             ='     BMC_DB2_STORTYPE
SAY 'BMC_ESTIMATED_CARDINALITY    ='     BMC_ESTIMATED_CARDINALITY
SAY 'BMC_OBJECT_PART_NUMBER       ='     BMC_OBJECT_PART_NUMBER
SAY 'BMC_OBJECT_COMPONENT_NUMBER  ='     BMC_OBJECT_COMPONENT_NUMBER
SAY 'BMC_DB2_FREEPAGE             ='     BMC_DB2_FREEPAGE
SAY 'BMC_DB2_PCTFREE              ='     BMC_DB2_PCTFREE
SAY 'BMC_DB2_VCATNAME             ='     BMC_DB2_VCATNAME
SAY 'BMC_DB2_STORNAME             ='     BMC_DB2_STORNAME
SAY 'BMC_DB2_DATACLAS             ='     BMC_DB2_DATACLAS
SAY 'BMC_DB2_STORCLAS             ='     BMC_DB2_STORCLAS
SAY 'BMC_DB2_MGMTCLAS             ='     BMC_DB2_MGMTCLAS
SAY 'BMC_DATASET_NAME             ='     BMC_DATASET_NAME
SAY 'BMC_OBJECT_TYPE_IND          ='     BMC_OBJECT_TYPE_IND
SAY 'BMC_UTILITY_SHRLEVEL         ='     BMC_UTILITY_SHRLEVEL

IF BMC_OBJECT_TYPE_IND = 'TS' THEN DO    /*   IF    TS OBJECT   */
```

Figure 82 Sample DSRSEXIT REXX user exit (part 3 of 5)

```
SAY 'BMC_SYSTABLESPACE_PARTITIONS ='     BMC_SYSTABLESPACE_PARTITIONS
SAY 'BMC_SYSTABLESPACE_PGSIZE     ='     BMC_SYSTABLESPACE_PGSIZE
SAY 'BMC_SYSTABLESPACE_SEGSIZE    ='     BMC_SYSTABLESPACE_SEGSIZE
SAY 'BMC_SYSTABLESPACE_NTABLES    ='     BMC_SYSTABLESPACE_NTABLES
SAY 'BMC_SYSTABLESPACE_MAXROWS    ='     BMC_SYSTABLESPACE_MAXROWS
SAY 'BMC_SYSTABLESPACE_DSSIZE     ='     BMC_SYSTABLESPACE_DSSIZE
SAY 'BMC_SYSTABLESPACE_NACTIVE    ='     BMC_SYSTABLESPACE_NACTIVE
SAY 'BMC_SYSTABLEPART_COMPRESSED  ='     BMC_SYSTABLEPART_COMPRESSED
END

ELSE DO                                  /*   ELSE  IX OBJECT   */

SAY 'BMC_SYSINDEXES_NAME          ='     BMC_SYSINDEXES_NAME
SAY 'BMC_SYSINDEXES_DBNAME        ='     BMC_SYSINDEXES_DBNAME
SAY 'BMC_SYSINDEXES_CREATOR       ='     BMC_SYSINDEXES_CREATOR
SAY 'BMC_SYSINDEXES_INDEXSPACE    ='     BMC_SYSINDEXES_INDEXSPACE
SAY 'BMC_SYSINDEXES_CLUSTERING    ='     BMC_SYSINDEXES_CLUSTERING
SAY 'BMC_SYSINDEXES_CLUSTERED     ='     BMC_SYSINDEXES_CLUSTERED
SAY 'BMC_SYSINDEXES_FULLKEYCARD   ='     BMC_SYSINDEXES_FULLKEYCARD
SAY 'BMC_SYSINDEXES_UNIQUERULE    ='     BMC_SYSINDEXES_UNIQUERULE
SAY 'BMC_SYSINDEXES_PIECESIZE     ='     BMC_SYSINDEXES_PIECESIZE
SAY 'BMC_SYSINDEXPART_LEAFDIST    ='     BMC_SYSINDEXPART_LEAFDIST
END

/* **************************************************************** */
/*                                                                  */
/* CHANGING 'BMC_PRIMARY_QUANTITY' OR 'BMC_SECONDARY_QUANTITY' IS   */
/* ALL YOU HAVE TO DO TO CHANGE THE ACTUAL VSAM ALLOCATION WHEN     */
/* 'REDEFINE YES' (DEFAULT) IS SPECIFIED IN YOUR REORG SYNTAX.      */
/*                                                                  */
/* SQL ALTER WILL ALSO BE PERFORMED AUTOMATICALLY SO THE PRI/SEC    */
/* QUANTITY CHANGE(S) ARE REFLECTED IN THE DB2 CATALOG TABLES       */
/* SYSIBM.SYSTABLEPART AND/OR SYSIBM.SYSINDEXPART.                  */
/*                                                                  */
/* IF YOU *DO NOT* WANT SQL ALTER PERFORMED THEN SET                */
/* 'BMC_ALTER_DB2_CATALOG = NO' FOR THAT OBJECT.                    */
/*                                                                  */
/* **************************************************************** */
/*                                                                  */
/* NOTE FOR NON-PARTITIONED DB2 DATASETS:                           */
/*                                                                  */
/* FOR MULTIDATASET INDEX COMPONENTS (.A002 -> .A00N) THAT ARE NOT  */
/* COMPONENTS OF A PARTITIONED INDEX (PERHAPS CREATED BECAUSE OF   */
/* A PIECESIZE SPECIFICATION)                                       */
/*                                                                  */
/* - OR -                                                           */
/*                                                                  */
/* FOR MULTIDATASET TABLESPACE COMPONENTS (.A002 -> .A00N) THAT ARE */
/* NOT PART OF A PARTITIONED TABLESPACE                             */
/*                                                                  */
/* YOU *CAN* CHANGE THE PRIMARY_QUANTITY AND/OR SECONDARY_QUANTITY  */
/* VALUES FOR THESE COMPONENTS.                                     */
/*                                                                  */
/* YOU *CAN NOT* SPECIFY 'BMC_ALTER_DB2_CATALOG = YES' FOR THEM     */
/* SINCE THEY DO NOT HAVE ENTRIES IN SYSIBM.SYSTABLEPART OR         */
```

Figure 82 Sample DSRSEXIT REXX user exit (part 4 of 5)

```
/* 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

/* **************************************************************** */
/*                                                                  */
/* THE FOLLOWING SECTION DISPLAYS ALL VOLUMES IN THE                */
/* CURRENT STOGROUP IN THE ORDER IN WHICH THEY WILL BE USED.        */
/*                                                                  */
/* THIS PROGRAM CAN BE MODIFIED TO CHANGE THE ORDER OF THE          */
/* VOLUMES IN THE  "BMC_STOGROUP_VOLUME."  STEM VARIABLE.           */
/*                                                                  */
/* YOU MAY ALSO RETURN ONLY A *SUBSET* OF THE ORIGINAL VOLUME LIST  */
/* IF YOU WISH.  TO DO THIS YOU SHOULD UPDATE AS MANY OF THE        */
/* BMC_STOGROUP_VOLUME.1 THROUGH BMC_STOGROUP_VOLUME.N VARIABLES    */
/* AS YOU WISH, AND THEN CHANGE THE BMC_STOGROUP_VOLUME.0 VARIABLE  */
/* TO REFLECT THE NBR OF VOLUMES (1 THROUGH BMC_STOGROUP_VOLUME.0)  */
/* YOU WISH TO BE CONSIDERED FOR ALLOCATION.  FOR INSTANCE, IF      */
/* YOU SET "BMC_STOGROUP_VOLUME.0 = 3" THEN ONLY THE FIRST THREE    */
/* VOLUMES IN THE BMC_STOGROUP_VOLUME. STEM VARIABLE WILL BE USED   */
/* FOR ALLOCATION (IN THE ORDER YOU RETURN THEM).                   */
/*                                                                  */
/* YOU MAY *NOT* ADD VOLUME NAMES THAT WEREN'T IN THE ORIGINAL LIST!*/
/*                                                                  */
/* IF YOU CHANGE THE ORDER OF VOLUMES IN YOUR STOGROUP BE SURE TO   */
/* SET  BMC_REORDER_STOGROUP_VOLUMES = 'YES'  OR REORG PLUS WILL    */
/* IGNORE YOUR CHANGES.  ANY CHANGES TO THE STOGROUP VOLUME ORDER   */
/* WILL BE IN EFFECT ONLY FOR THE CURRENT OBJECT.                   */
```

Figure 82 Sample DSRSEXIT REXX user exit (part 5 of 5)

```
/*                                                                  */
/* **************************************************************** */

SAY '**** START *********** STOGROUP SECTION   ************'

SAY 'BMC_REORDER_STOGROUP_VOLUMES  = '  BMC_REORDER_STOGROUP_VOLUMES
SAY 'NUMBER OF VOLUMES IN STOGROUP = '  BMC_STOGROUP_VOLUME.0

INDX = 1

DO WHILE INDX <= BMC_STOGROUP_VOLUME.0

  SAY 'BMC_STOGROUP_VOLUME.'INDX '   = ' BMC_STOGROUP_VOLUME.INDX
  INDX = INDX + 1

END                                     /* END DO WHILE */

SAY '**** END   *********** STOGROUP SECTION   ************'

SAY '**** END   *********** ' BMC_EXIT_POINT ' ************'

RETURN
```

Using TERMEXIT to control BMCHIST and statistics updates

TERMEXIT is the user exit for controlling BMCHIST and statistics functions in the UTILTERM phase. TERMEXIT provides REORG PLUS with user-defined variables that allow you to dynamically control processing of updates to BMCHIST, BMCSTATS, real-time statistics, and UPDATEDB2STATS at execution time.

For information about the BMCHIST and TERMEXIT command options, see Chapter 3, “Syntax of the REORG command.” For information about the corresponding installation options, see Appendix A, “REORG PLUS installation options.”

Within this exit, you can also use SQL to maintain your BMCHIST tables. The example in Figure 83 deletes all rows over 90 days old.

Figure 83 Example maintaining BMCHIST tables using SQL

```
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;
```

TERMEXIT variables

In addition to the variables common to all REXX exits (Table 128 on page 779), REORG PLUS passes additional variables to the TERMEXIT user exit. Table 130 describes the variables specific to this exit and indicates whether you can update the variable (yes or no).

Table 130 Variables that REORG PLUS passes to TERMEXIT

Variable name	Description	Update?
BMC_OPT_BMCHIST	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.	yes
BMC_OPT_RTS	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.	yes
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

Figure 84 provides a sample REXX exit that shows one way of setting the variables in the TERMEXIT user exit.

Figure 84 Sample TERMEXIT REXX user exit (part 1 of 6)

```
/* REXX  */
/* ******************************************************************
 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.

 ****************************************************************** */

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?  */
```

Figure 84 Sample TERMEXIT REXX user exit (part 2 of 6)

```
IF RC THEN S_RC = RXSUBCOM('ADD','DSNREXX','DSNREXX')

/* ****************************************************************
   CONNECT TO DB2 USING VARIABLE PASSED FROM REORG
******************************************************************* */

CURR_FUNC = 'CONNECT TO DB2'
ADDRESS DSNREXX 'CONNECT' BMC_DB2_SSID

IF SQLCODE = 0 THEN DO
   SAY '*** CONNECTED TO SUBSYSTEM = ' BMC_DB2_SSID
   END
ELSE DO
   SAY '*** CONNECT TO SUBSYSTEM = ' BMC_DB2_SSID ' *** FAILED ***'
   SIGNAL BAD_SQLCODE
   END

/* ****************************************************************
   PREPARE STMT / DECLARE CURSOR / OPEN CURSOR
******************************************************************* */
SQLHIST1 = 'SELECT COUNT(*), MAX(DATE) FROM ' ,
           STRIP(BMC_TBCREATOR_BMCHIST)||'.'||BMC_TBNAME_BMCHIST ,
           ' WHERE UTILID = ? '

SQLHIST2 = 'SELECT MAX(TIME) FROM ' ,
           STRIP(BMC_TBCREATOR_BMCHIST)||'.'||BMC_TBNAME_BMCHIST ,
           ' WHERE UTILID = ?  AND  DATE = ? '

CURR_FUNC = 'PREPARE SQL HIST1'             /* PREPARE SQLHIST1  */
ADDRESS DSNREXX ,
   'EXECSQL PREPARE S1 FROM :SQLHIST1'
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE

CURR_FUNC = 'PREPARE SQL HIST2'             /* PREPARE SQLHIST2  */
ADDRESS DSNREXX ,
   'EXECSQL PREPARE S2 FROM :SQLHIST2'
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE

CURR_FUNC = 'DECLARE CURSOR C1'             /* DECLARE C1 FOR HIST1 */
ADDRESS DSNREXX ,
   'EXECSQL DECLARE C1 CURSOR FOR S1'
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE

CURR_FUNC = 'DECLARE CURSOR C2'             /* DECLARE C2 FOR HIST2 */
ADDRESS DSNREXX ,
   'EXECSQL DECLARE C2 CURSOR FOR S2'
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE

CURR_FUNC = 'OPEN CURSOR C1'                /* OPEN CURSOR C1    */
ADDRESS DSNREXX ,
    'EXECSQL OPEN C1 USING :BMC_UTIL_ID'
```

Figure 84 Sample TERMEXIT REXX user exit (part 3 of 6)

```
IF SQLCODE /= 0 THEN SIGNAL BAD_SQLCODE

/* ****************************************************************
   FETCH PRIOR BMCHIST UTILITY HISTORY
******************************************************************* */

  SAY
  SAY '** BMCHIST UTILITY HISTORY **'

  CURR_FUNC = 'FETCH CURSOR C1'       /* GET UTILCOUNT AND LAST-DATE */
    ADDRESS DSNREXX ,
      'EXECSQL FETCH C1 INTO :UTIL_COUNT, :LAST_UTIL_DATE:NULLIND'
  IF SQLCODE /= 0 THEN SIGNAL BMCHIST_SELECT
  IF UTIL_COUNT = 0 THEN SIGNAL BMCHIST_SELECT

  CURR_FUNC = 'OPEN CURSOR C2'        /* OPEN CURSOR C2       */
  ADDRESS DSNREXX ,
      'EXECSQL OPEN C2 USING :BMC_UTIL_ID, :LAST_UTIL_DATE'
  IF SQLCODE /= 0 THEN SIGNAL BMCHIST_SELECT

  CURR_FUNC = 'FETCH CURSOR C2'       /* GET LAST RUNTIME ON MAX_DATE */
    ADDRESS DSNREXX ,
      'EXECSQL FETCH C2 INTO :LAST_UTIL_TIME:NULLIND'

BMCHIST_SELECT:
     SELECT
       WHEN (SQLCODE = 0) & (UTIL_COUNT > 0) THEN DO
         SAY '** FOR UTILID                 =' BMC_UTIL_ID
         SAY '** LAST RUN DATE              =' LAST_UTIL_DATE
         SAY '** LAST RUN TIME              =' LAST_UTIL_TIME
         SAY '** TOTAL ROWS FOR THIS UTILID =' UTIL_COUNT

         END      /* END SQLCODE = 0   */

       WHEN (SQLCODE = 100) | (UTIL_COUNT = 0) THEN DO

         SAY '** NO PRIOR HISTORY FOR UTILID = ' BMC_UTIL_ID

         END      /* END SQLCODE = 100 */

       OTHERWISE SIGNAL BAD_SQLCODE

     END                         /*  END SELECT    */

         SAY ' '

  ADDRESS DSNREXX  'EXECSQL CLOSE C1'
  ADDRESS DSNREXX  'EXECSQL CLOSE C2'

  ADDRESS DSNREXX  'EXECSQL  COMMIT'
```

Figure 84 Sample TERMEXIT REXX user exit (part 4 of 6)

```
/* ****************************************************************
   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 STMTS'
    ADDRESS DSNREXX ,
       'EXECSQL PREPARE S3 FROM :SQLSTMTS'
    END
  ELSE DO                                      /* ELSE REORG INDEX   */
    CURR_FUNC = 'PREPARE SQL STMIX'
    ADDRESS DSNREXX ,
       'EXECSQL PREPARE S3 FROM :SQLSTMIX'
    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.
```

Figure 84 Sample TERMEXIT REXX user exit (part 5 of 6)

```
******************************************************************** */
  CURR_FUNC = 'FETCH CURSOR C3'
    ADDRESS DSNREXX ,
      'EXECSQL FETCH C3 INTO :REORG_DATE:NULLIND'

     SELECT
       WHEN (SQLCODE = 0) & (NULLIND = 0) THEN DO

        SAY '** LAST REORG RTS TIMESTAMP FOR ',
            STRIP(BMC_DBNAME)||.||STRIP(BMC_TSIX),
            ' PART ' CURR_PART ' = ' REORG_DATE

        END      /* END SQLCODE = 0   */

       WHEN (SQLCODE = 100) | (NULLIND /= 0)  THEN DO

        SAY '** LAST REORG RTS TIMESTAMP FOR ',
            STRIP(BMC_DBNAME)||.||STRIP(BMC_TSIX),
            ' PART ' CURR_PART ' =  * NO REORG RTS HISTORY *'

        END      /* END SQLCODE = 100 */

       OTHERWISE SIGNAL BAD_SQLCODE
     END                           /*  END SELECT    */

  ADDRESS DSNREXX  'EXECSQL CLOSE C3'
  ADDRESS DSNREXX  'EXECSQL  COMMIT'

END     /* END  DO PART_IX = 1 TO BMC_REORG_PART_NUMBERS.0  */

/* *****************************************************************
   DISCONNECT DB2 THREAD
******************************************************************** */

  ADDRESS DSNREXX DISCONNECT            /*  DISCONNECT INTERFACES */

SAY
SAY '**** END   *********** ' BMC_EXIT_POINT ' ************'

RETURN     /*  NORMAL EXIT FOR THIS PROGRAM */


/* *****************************************************************
   SQL ERROR - IF YOU GET HERE SOMETHING BAD HAPPENED
******************************************************************** */
BAD_SQLCODE:
   SAY
   SAY '*** ERROR DURING DB2 '''CURR_FUNC''' SQLCODE ' SQLCODE

SAY
SAY '********************  SQLCA CONTENTS ********************'
```

Figure 84 Sample TERMEXIT REXX user exit (part 6 of 6)

```
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
SAY '****** TERMINATING DUE TO SQL ERROR IN ' BMC_EXIT_POINT

EXIT 8
```

Index

Notes