BMC Next Generation Technology Reorg for DB2 for z/OS Reference Manual

Supporting

Version 12.1 of BMC Next Generation Technology Reorg for DB2 for z/OS

December 2016
Contacting BMC Software

Several methods are available for contacting BMC Software.

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

United States and Canada

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

Outside United States and Canada

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

© Copyright 2015-2016 BMC Software, Inc.

BMC, BMC Software, the BMC logo, the BMC Software logo, and other BMC marks are the exclusive properties of BMC Software, Inc. and are registered or may be registered in the U.S. and in other countries. BladeLogic and other BladeLogic marks are the exclusive properties of BladeLogic, Inc. and are registered or may be registered in the U.S. and in other countries. All other trademarks or registered trademarks are the property of their respective owners.

PIX is a trademark or registered trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

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

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

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2103 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

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

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

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

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

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

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

- Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)
- In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.
## Contents

### About this book
- Related publications ................................................................. 9
- Conventions .................................................................................. 10
- Syntax diagrams ........................................................................... 10

### Chapter 1 Overview of NGT Reorg
- NGT Reorg features ................................................................. 13
- Reasons to reorganize table spaces ............................................. 15
- Reasons not to reorganize table spaces ....................................... 15
- Index reorganization advantages ................................................. 16

### Chapter 2 Operational considerations for NGT Reorg
- NGT Reorg requirements and restrictions ................................. 17
  - NGT Reorg software requirements ............................................ 17
  - Required authorizations for NGT Reorg .................................... 17
  - Status requirements for NGT Reorg .......................................... 17
  - Storage requirements for NGT Reorg ........................................ 18
  - NGT Reorg restrictions ........................................................... 18
- Considerations for fine tuning NGT Reorg ................................. 18
  - Partition sizes and parallel processing ...................................... 19
  - Free space considerations ....................................................... 19
- Restart considerations .............................................................. 19
  - Serialization and concurrency .................................................. 20
  - Utility ID considerations ........................................................ 20
  - The checkpoint data set ........................................................... 21
- Considerations for rebalancing partitions ................................. 21
- Considerations for reorganizing partition-by-growth (PBG) table spaces .................................................. 23
- Considerations for objects that are in REORG-pending (REORP) status .................................................. 24
  - Minimizing downtime due to REORP ........................................ 24
  - DDL that causes REORP status ............................................... 25
- Considerations for processing DB2 indexes that are in a rebuild-pending status .................................... 25
  - Minimizing downtime due to RBDP or PSRBD status ............... 25
  - DDL that causes RBDP or PSRBD status ................................... 26
- How NGT Reorg handles pending schema changes .................. 26
- Considerations for reorganizing LOB data ................................. 28
- Considerations for reorganizing XML data ................................. 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reorganizations that invoke DSNUTILB</td>
<td>30</td>
</tr>
<tr>
<td>Additional NGT Reorg considerations</td>
<td>30</td>
</tr>
<tr>
<td>Considerations for image copies in NGT Reorg</td>
<td>30</td>
</tr>
<tr>
<td>Row count differences</td>
<td>31</td>
</tr>
<tr>
<td>Differences between REORG PLUS and NGT Reorg</td>
<td>31</td>
</tr>
<tr>
<td>Chapter 3 Input for NGT Reorg</td>
<td>33</td>
</tr>
<tr>
<td>DD statements for NGT Reorg</td>
<td>33</td>
</tr>
<tr>
<td>NGT Reorg syntax overview</td>
<td>36</td>
</tr>
<tr>
<td>Command syntax rules for NGT utilities</td>
<td>36</td>
</tr>
<tr>
<td>Alphabetical listing of REORG options</td>
<td>37</td>
</tr>
<tr>
<td>OUTPUT command</td>
<td>39</td>
</tr>
<tr>
<td>REORG TABLESPACE syntax</td>
<td>40</td>
</tr>
<tr>
<td>REORG TABLESPACE diagrams</td>
<td>40</td>
</tr>
<tr>
<td>REORG TABLESPACE options</td>
<td>45</td>
</tr>
<tr>
<td>REORG INDEX syntax</td>
<td>61</td>
</tr>
<tr>
<td>REORG INDEX diagrams</td>
<td>61</td>
</tr>
<tr>
<td>REORG INDEX options</td>
<td>62</td>
</tr>
<tr>
<td>Chapter 4 Parameters for NGT Reorg</td>
<td>67</td>
</tr>
<tr>
<td>Syntax rules for parameters</td>
<td>67</td>
</tr>
<tr>
<td>+ALLO</td>
<td>68</td>
</tr>
<tr>
<td>+BEFOREACC</td>
<td>70</td>
</tr>
<tr>
<td>+DBNAME</td>
<td>70</td>
</tr>
<tr>
<td>+FREEPAGE</td>
<td>70</td>
</tr>
<tr>
<td>+IGNDEFNO</td>
<td>71</td>
</tr>
<tr>
<td>+INITWINDOW</td>
<td>72</td>
</tr>
<tr>
<td>+LOB</td>
<td>73</td>
</tr>
<tr>
<td>+LOBPCT</td>
<td>73</td>
</tr>
<tr>
<td>+MAXTASKS</td>
<td>74</td>
</tr>
<tr>
<td>+MINTFSIZE</td>
<td>75</td>
</tr>
<tr>
<td>+PAGEUSE</td>
<td>76</td>
</tr>
<tr>
<td>+PARTINDEPEND</td>
<td>76</td>
</tr>
<tr>
<td>+PBGCHKDICT</td>
<td>77</td>
</tr>
<tr>
<td>+PCTFREE</td>
<td>78</td>
</tr>
<tr>
<td>+RUNSTATS</td>
<td>79</td>
</tr>
<tr>
<td>+VOLUMES</td>
<td>80</td>
</tr>
<tr>
<td>Chapter 5 Syntax examples for NGT Reorg</td>
<td>83</td>
</tr>
<tr>
<td>Reorganizing a table space</td>
<td>83</td>
</tr>
<tr>
<td>Reorganizing specific partitions of a table space</td>
<td>83</td>
</tr>
</tbody>
</table>
Reorganizing specific logical partitions of a table space .............................................. 84
Reorganizing a table space and generating an image copy ........................................ 84
Reorganizing a table space and sending discards to a data set ................................. 84
Reorganizing a set of table spaces and excluding certain table spaces ................... 85
Reorganizing an index ................................................................................................... 85
Reorganizing specific partitions of an index ............................................................... 85
Reorganizing a set of indexes and excluding certain indexes ................................. 86

Chapter 6         Recommended automation control points for NGT Reorg          87

Chapter 7          NGT Reorg reports                                          89

Reports to the SUMMARY DD ......................................................................................... 89
About this book

This book contains detailed information about the associated product or products. This preface explains the special conventions that the book uses, and how to access related publications.

If applicable, the preface also summarizes the major changes included in the latest release of the product.

Related publications

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


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


Products with online interfaces also offer online Help via the F1 key or, for graphical user interfaces (GUIs), via a Help button.
Tip
If you prefer hardcopy documentation, you can order it from your BMC sales representative or from Support Central. Also, from Support Central you can subscribe to receive proactive e-mail alerts when BMC issues notices.

Conventions

This document uses the following special conventions:

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

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

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

Syntax diagrams

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

The following guidelines provide additional information about syntax diagrams:

- Read diagrams from left to right and from top to bottom.
- A recursive (left-pointing) arrow above a stack indicates that you may choose more than one item in the stack.
- An underlined item is a default option.
- If a diagram shows punctuation marks, parentheses, or similar symbols, you must enter them as part of the syntax.
- In general, IBM commands, keywords, clauses, and data types are displayed in uppercase letters. However, if an item can be shortened, the minimum required portion might be shown in uppercase letters, with the remainder in lowercase (for example, CANcel).
- The following conventions apply to variables in syntax diagrams:
  - Variables are typically displayed in lowercase letters and are always italicized.
  - If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words (for example, databaseName).
Overview of NGT Reorg

This chapter provides an overview of the BMC Next Generation Technology Reorg for DB2 for z/OS (NGT Reorg) product.

NGT Reorg features

The following topics provide a highlight of NGT Reorg features.

RUNSTATS updates

NGT Reorg can optionally update the DB2 catalog with the RUNSTATS values calculated at the end of a reorganization.

If you reorganize only a subset of partitions, NGT Reorg updates statistics for those partitions and recomputes the composite statistics.

Concurrent copy support

Concurrent copies are made at the same time as a load or reorganization.

NGT Reorg will create concurrent DB2 image copies during a reorganization on any table space. A separate run is not required.

Concurrent copy support uses DASD, tape, or VTS. These copies are compatible with DSN1COPY. The following considerations apply for cartridge tapes needed for offsite storage:

- The recovery site copy should go to DASD that is immediately HSM-migrated to tape. This makes efficient use of tape and shortens recovery time by allowing for recall prior to starting the recovery.

- Use a virtual tape system that has external tapes that can be taken offsite and brought back. This makes efficient use of tapes and buffers them to DASD for faster recovery times.
**SYSCOPY posting**

NGT Reorg posts reorganizations and all concurrent copies in the DB2 SYSIBM.SYSCOPY catalog table. The reorganization is posted as LOG(NO).

**Row discards**

NGT Reorg deletes rows from the table during a reorganization when either of the following conditions exists:

- You specify DISCARD on the REORG command.
- The reorganization materializes pending limit key changes, and rows exist after the new last limit key.

NGT Reorg does not log deleted rows, but you can archive them to a discard data set, such as SYSDISC, and specify the format for that data set. For more information, see “ddname” on page 58.

**Discards due to a DISCARD specification**

The DISCARD option uses a WHERE clause that specifies which rows to delete during the REORG.

The WHERE clause supports the following functions:

- LIKE and IN predicates
- Subselects in an IN clause
- All DATE, TIME, and TIMESTAMP functions
- Boolean logic

**Note**

You cannot use wildcards on your DISCARD specification.

For more information about the DISCARD option, including restrictions, see “DISCARD FROM TABLE” on page 60. For more information about the WHERE clause, including syntax and restrictions, see the *BMC Next Generation Technology General User Guide*.

**Discards due to altered limit keys**

After materializing pending limit key changes, NGT Reorg deletes any rows found after the new last limit key. For more information, see “Pending limit key changes” on page 27.
Reasons to reorganize table spaces

Reorganize a table space for the following reasons:

■ To reclaim space from dropped tables

■ To eliminate pointer-overflow pairs
  When a row increases in size, it may no longer fit in the space reserved for it in the page. When this happens, DB2 moves the row to a new location and places in the original spot a pointer to this overflow location. This happens only if your rows contain variable columns that significantly increase in size as the result of SQL UPDATE statements, or if compression of the rows becomes less efficient after an update.

■ To re-establish free space
  If the table space is ordered by a clustering index, then the amount of free space assumes more significance. Availability of free space can allow DB2 to maintain clustering. Conversely, a free space shortage can cause inserted rows to be placed out of clustering order.

  Reestablishing free space may be counter-productive. If an application has heavy inserts and inadequate free space, it may seem at first glance that reorganizing the table space may improve performance by reestablishing free space for subsequent inserts. Often the opposite is the case. Table space scans take longer because there are more pages to scan. In either case, the benefit or penalty is small.

■ To collect segments in segmented table spaces
  Adjacency of segments is important if the table space is being scanned. However, avoiding table space scans is a much better idea than trying to make them more effective. Reorganizing an index may make it more attractive to DB2, thus providing a much more significant performance advantage.

■ To re-establish clustering
  Clustering indexes provide a performance advantage. However, in order to be useful, DB2 must use the index. If the clustering ratio is too low because of a disorganized table space, DB2 might not use the index. However, by reorganizing the table space, you can increase the cluster ratio.

Reasons not to reorganize table spaces

You might find that you should not reorganize table spaces, as for the following reasons:

■ Table space reorganizations cannot improve some access paths, such as index scans for non-clustering indexes and direct access to the table space via the index.
Table space reorganizations primarily improve the efficiency of table space scans. However, table space scans should be avoided, especially with large tables.

Table space reorganizations require more system resources than index reorganizations. Table space reorganizations reorganize every index in the table space, including indexes that do not need reorganizing.

Index reorganization advantages

Index reorganizations provide the following advantages:

- Improve the performance of your critical queries.
  After an index reorganization, queries take less time to complete and use less system resources, improving the overall performance of applications.

- Can reduce the number of levels in the index, improving the speed of index searches.

- Can reduce the number of leaf pages, improving the speed of index scans.

- Restore a reasonable free space specification. Over time, many indexes increase the amount of free space they hold, since page splits produce pages that are only 50% full.

- Can reduce the number of extents in the index data set, improving the physical access performance to the index.

- Require a fraction of the resources required to complete a full table space reorganization.

You should reorganize indexes as often as possible, as indexes are more volatile and disorganize faster than table spaces. As pages split, free space can rapidly increase to 50%. As a result, the number of index levels grows unnecessarily. Additionally, the distance between adjacent pages increases, a factor that does not affect most table spaces but is critical in index access paths.
Operational considerations for NGT Reorg

This chapter provides information that you need to know to run NGT Reorg.

NGT Reorg requirements and restrictions

This topic describes NGT Reorg requirements and restrictions.

NGT Reorg software requirements

The NGT Reorg product has the following requirements for supporting software:

- NGT Reorg requires a minimum of version 12.1.00 of the BMC DB2 Solution Common Code (SCC).
- To use any features that invoke DSNUTILB, you must be licensed to use the IBM DB2 REORG utility.

Required authorizations for NGT Reorg

NGT Reorg requires the following authorizations:

- REORG authority on the database
- SYSADM or DBADM authority

Status requirements for NGT Reorg

When running NGT Reorg, table spaces and indexes must be in read/write (RW) status.
In addition, table spaces and indexes cannot be in a restricted status.

Storage requirements for NGT Reorg

NGT Reorg requires 2 GB of auxiliary storage to reorganize partition-by-growth (PBG) table spaces, regardless of the storage actually used.

NGT Reorg restrictions

The following restrictions apply to NGT Reorg:

■ If a table is added to or dropped from the table space during a reorganization, NGT Reorg terminates.

■ If a table is mass deleted from a segmented table space during a reorganization, NGT Reorg terminates.

■ If a table space is empty and is clustered by a nonpartitioned secondary index (NPI), NGT Reorg skips the table space.

■ If an index is in a rebuild-pending status (RBDP or PSRBD), NGT Reorg invokes DSNUTILB.

   For information about reorganization jobs that invoke DSNUTILB, see “Reorganizations that invoke DSNUTILB” on page 30.

■ NGT Reorg supports partition sizes (DSSIZE) greater than 64 GB except for the following PAGESIZE/DSSIZE combinations:

<table>
<thead>
<tr>
<th>PAGESIZE</th>
<th>DSSIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4KB</td>
<td>128GB</td>
</tr>
<tr>
<td></td>
<td>256GB</td>
</tr>
<tr>
<td>8KB</td>
<td>256GB</td>
</tr>
</tbody>
</table>

Considerations for fine tuning NGT Reorg

The default values shipped with NGT Reorg are appropriate for the majority of environments. To address specific needs, however, you can override these defaults by passing a set of parameters to NGT Reorg during run time.
Partition sizes and parallel processing

NGT Reorg processes segmented table spaces with sequential reads and sequential writes. To improve performance, BMC recommends converting segmented table spaces to partition-by-growth (PBG) table spaces and taking advantage of parallelism.

You can take advantage of parallelism by increasing the number of partitions and specifying DSSIZE values less than 32GB.

For example, if a PBG table space holds one 64-GB partition, NGT Reorg is still limited to one sequential read and sequential write. If the PBG table space holds eight 8-GB partitions instead of one 64-GB partition, NGT Reorg reads and writes the eight partitions in parallel, increasing the speed of the reorganization.

Free space considerations

NGT Reorg provides free space specification parameters:

- The +FREEPAGE parameter specifies the number of pages between empty pages in the table space.
- The +PCTFREE parameter specifies the percentage of free space in each data page of the table space.

These options are located in the RRGPARMS DD for table spaces and RRGPARMX DD for indexes. These parameters are equivalent to the DB2 parameters with the same names.

Restart considerations

If a reorganization fails, NGT Reorg requires only that you resubmit the job. You do not need to make changes. NGT Reorg determines that the prior execution terminated prematurely and resumes processing.

Note

You cannot make any changes to your REORG syntax on restart. NGT Reorg does not reread your SYSIN specification.
Serialization and concurrency

To prevent two incompatible utilities from executing on the same object concurrently, NGT Reorg uses the following mechanisms:

- An internal matrix of utility IDs and processing objects
- The common BMC utility tables, BMCSYNC and BMCUTIL

It is often necessary to address a previously failed utility ID prior to running a utility with a different utility ID on the same object. You can run the previous utility with a RESTART or with QUICKEXIT to clean up temporary files. If the utility is in a "must complete" state and a QUICKEXIT is run, the utility will simply complete. NGT Reorg ensures no damage can occur by resubmitting. NGT Reorg always runs with the object in RW status and leaves the original table space and indexes untouched until the reorganization is complete. For more information, see the *BMC Next Generation Technology General User Guide*.

Utility ID considerations

NGT Reorg accepts 16 characters or less in the utility ID as part of the PARM field on the JCL EXEC statement.

If NGT Reorg determines that a prior job using this same utility ID ended prematurely, then the current job proceeds as follows:

- If no restart option is specified, processing resumes from the most recent successful checkpoint, which is typically the last completed table space or partition.

- If a restart option is specified, then
  - If your choice is feasible for the reorganization, processing proceeds as dictated.
  - If the restart option is incompatible with the reorganization, then processing stops.

The key to this process is the utility ID. Note the following considerations:

- NGT Reorg resumes the reorganization of the original set of objects. If you change the control statements before resubmitting the utility, the new statements are ignored.

- Two NGT Reorg steps using the same utility ID in different jobs cannot run concurrently. This can happen when two steps appear in inadvertent duplicates of the same job or in unrelated jobs that use the same utility ID. The second step to begin execution fails.
The checkpoint data set

Every execution of NGT Reorg requires access to a linear VSAM data set named during the installation procedure and configured with the +NGTCKPT parameter or the //NGTCKPT DD statement. This checkpoint data set (NGTCKPT) contains all the restart information for utilities.

The checkpoint data set includes the following information:

- The list of objects to be reorganized and the status of each
- The internal tables required for restart
- The list of objects being renamed
- The RUNSTATS values

For more information about the checkpoint data set, see the BMC Next Generation Technology General User Guide.

Considerations for rebalancing partitions

NGT Reorg enables you to rebalance your partitions while reorganizing your table space. NGT Reorg redistributes the rows and adjusts the limit keys across partitions.

Partition rebalancing occurs during the reorganization in one of following ways (listed in the order in which they take precedence):

- Pending limit key ALTER operations exist on the table space.
  NGT Reorg ignores any DDLIN or REBALANCE specification and issues a message to that effect.

- No limit key ALTER operations are pending on the table space, but you specify DDLIN, or the default DDLIN DD exists in your JCL.
  NGT Reorg honors the DDLIN specification. NGT Reorg ignores any REBALANCE specification and issues a message to that effect.

- No limit key ALTER operations are pending on the table space, no DDLIN specification or DD exists, but you specify REBALANCE.
  NGT Reorg honors the REBALANCE specification.

For more information, see the following references:

- “Pending limit key changes” on page 27
- “DDLIN” on page 49
- “REBALANCE” on page 49
General partition rebalancing considerations

The following considerations apply to all partition rebalancing methods:

■ The value of the +FASTSWITCH parameter must be YES. If the value is NO, NGT Reorg issues a message and terminates.

■ NGT Reorg ignores any rebalance specification for nonpartitioned table spaces and for index reorganizations.

■ NGT Reorg invokes DSNUTILB when any of the following conditions exists:
  — A base table space contains LOB or XML columns.
  — You also specify AUX to reorganize both the base table space and its LOB table space in a single invocation.
  — The table space does not contain a partitioning index.

  **Note**

  NGT Reorg does not rebalance partitions in XML table spaces.

■ You cannot rebalance table spaces that have a limit key that contains a floating-point column.

■ The reorganization terminates if you request partition rebalancing and the number of duplicate keys will result in an empty partition. NGT Reorg cannot accurately determine an appropriate limit key in this case.

■ NGT Reorg issues a message and does not rebalance partitions if it determines that the number of rows is too small to rebalance.

■ You can have NGT Reorg write out rebalancing information by specifying DD statements as follows:
  — Specify DD PARTSDDL to have NGT Reorg write out DDL that you can use to create the table space, tables, and indexes that you are reorganizing.
  — Specify DD REPBLDAL to have NGT Reorg write out the ALTER statements for the object that you are reorganizing. These ALTER statements reflect the new keys following the rebalance operation.

If you do not include one or both of these DD statements for a rebalancing reorganization, NGT Reorg issues a warning message but continues the reorganization and rebalance operation.

Considerations for REBALANCE and DDLIN specifications

The following considerations apply specifically to REBALANCE and DDLIN specifications:
The reorganization terminates if your DDLIN contains a limit key ALTER operation that would result in discard processing. No discard processing occurs.

The reorganization terminates if you specify REBALANCE and a subset of partitions. If you specify REBALANCE, you must reorganize all partitions in the table space.

If you specify both REBALANCE and AUX (using the NGT LOBMaster product), NGT Reorg invokes DSNUTILB.

The reorganization terminates if you specify either the REBALANCE or DDLIN option and any of the following conditions exist:
— You are running NGT Reorg under the NGT Utility Manager product.
— You specify the DISCARD or MASSDELETE option.

Considerations for reorganizing partition-by-growth (PBG) table spaces

PBG table spaces can be large and can require significant resources to reorganize. The following considerations apply to reorganizations of PBG table spaces:

The MAXDISORG keyword cancels the reorganization if the percentage of disorganization is too high. This cancellation prevents unexpected, long-running reorganizations on large table spaces. If encountering such a PBG table space, NGT Reorg issues a message indicating that it did not reorganize this table space.

You can plan reorganizations on these table spaces by specifying a high MAXDISORG value when the extended elapsed time is more acceptable. The default value for MAXDISORG is 50%.

NGT Reorg currently does not support processing select partitions of a PBG table space. Reorganizing one partition of a PBG table space is of limited use because rows must stay in that partition when they really belong in another partition. Reorganizing one partition of a PBG table space would be the equivalent to reorganizing one data set of a classic segmented table space.

The NOCLIX keyword reorganizes a PBG table space as if there were no clustering index. This reclaims space only.

After an NGT Reorg job, all partitions of the PBG table space have the same compression dictionary. A new compression dictionary is produced and compared to the existing partition 1 dictionary; the better of the two will be used for all partitions. This means that NGT Reorg might have to decompress and compress each row.
Considerations for objects that are in REORG-pending (REORP) status

NGT Reorg uses RELOAD REORG to process objects that are in REORG-pending (REORP) status.

When processing objects that are in REORP status, NGT Reorg uses the NGT Load parameters that are specified in the LODPARMS DD, including common parameters that also exist in the RRGPARMS DD, such as +ALLO.

Note

Some restrictions apply when a partitioned table space in REORP status has associated LOB table spaces. For more information, see “Considerations for reorganizing LOB data” on page 28.

Minimizing downtime due to REORP

To minimize downtime due to REORP and maximize an object's availability, BMC recommends that you perform one of the following actions:

1. Reorganize the object that will be placed in REORP state due to an alter.

2. Perform the alter, placing the object in REORP state and making it unavailable.

3. Reorganize the object with the +SORT(NO) load parameter to eliminate the REORP status, making the object available again.

If downtime is not an issue, you can alter the object and run the reorganization with the +SORT(YES) load parameter to clear the REORP status and organize the data. The regularly scheduled reorganization job processes any other parts that meet the criteria for reorganization.
DDL that causes REORP status

The following list contains some of the SQL statements that put a table space in REORP status, making it unavailable. BMC recommends that you always test the DDL you plan to run to check for REORP status.

- ALTER TABLE ... ALTER COLUMN (of partitioning index, increase character length)
- ALTER TABLE ... ADD COLUMN (where the column is an identity column)
- *(DB2 Version 10)* ALTER TABLE ... ALTER PARTITION (to alter the limit key)
  Altering the last partition does not put the table space in REORP status.
- ALTER TABLE ... ADD PARTITION (if boundary for last partition not enforced)
- ALTER TABLE ... ROTATE PARTITION (if boundary for last partition not enforced)

Considerations for processing DB2 indexes that are in a rebuild-pending status

In most cases, NGT Reorg reorganizes indexes concurrently with its table spaces. However, if indexes are in a rebuild-pending status (RBDP or PSRBD), NGT Reorg invokes DSNUTILB.

For information about reorganization jobs that invoke DSNUTILB, see “Reorganizations that invoke DSNUTILB” on page 30.

Minimizing downtime due to RBDP or PSRBD status

BMC recommends the following process to minimize downtime caused by REBUILD-pending (RBDP) or page set REBUILD-pending (PSRBD) statuses:

1. Perform the alter that will place the index in a rebuild-pending state and make it unavailable.

2. Run a table space reorganization, ensuring that the utilities parameter +DSNUTILB is YES.
   This reorganization eliminates the rebuild-pending status, reorganizes the index, and makes it available again.
The regularly scheduled reorganization job processes other indexes, index partitions, or table space partitions that meet the NGT Utility Manager criteria for reorganization.

**DDL that causes RBDP or PSRBD status**

The following list contains some of the actions that put an index in REBUILD-pending status, making it unavailable. BMC recommends that you always test the DDL you plan to run to check for RBDP status.

- Altering a PIECESIZE value
- Altering an indexed variable length column between PADDED and NOT PADDED
- Adding a column to a table and index, and inserting rows, all in the same commit scope
- Altering an indexed column from SMALLINT, INTEGER, or DECIMAL to BIGINT
- Altering an indexed column from CHAR FOR BIT DATA or VARCHAR FOR BIT DATA to BINARY or VARBINARY
- Altering an indexed DECIMAL column

**How NGT Reorg handles pending schema changes**

NGT Reorg checks for pending schema changes and materializes them during a reorganization.

Processing the pending DDL requires NGT Reorg to process all partitions and post them simultaneously. This prevents the individual partition posting that occurs with partition independence when there are no nonpartitioned indexes.

During the reorganization, a utility posts in the SYSIBM.SYSUTILX table. Running the IBM -DIS UTIL command will show output similar to the following:

```
DSNU1001 -DB2P DSNUGDIS - USERID = userid1
          MEMBER = DB21
          UTILID = NGTReorgUtilityID.0002
          PROCESSING UTILITY STATEMENT 1
          UTILITY = RUNSTATS
```
you will see a stopped RUNSTATS utility with a utility ID starting with NGT followed by the NGT Reorg utility ID. *(Do not terminate this utility ID.)*

When pending schema changes exist on the table space, the object will be in UTRW status during the reorganization. This status ensures that no other alters can be run that add records to SYSPENDINGDDL for this object while NGT Reorg is processing the current pending DDL. When the reorganization starts, NGT Reorg processes the pending DDL it finds for an object. When the reorganization is complete, NGT Reorg clears all records for the object. The stopped RUNSTATS utility prevents further alters without restricting access to the object during the reorganization.

### General restrictions

The following restrictions apply to reorganizations that are materializing pending schema changes. For restrictions that apply specifically to pending limit key changes, see the section that follows this one.

- If there are pending changes and NGT Reorg attempts to process a subset of the partitions, the object is excluded from processing, and NGT Reorg issues the following message:

  **NGTR790 NO PART SELECTION ALLOWED ON A REORG THAT IS PROCESSING PENDING ALTERS**

  **Note**

  To avoid skipping needed reorganizations, you can perform the ALTER prior to reorganizing the entire object.

- NGT Reorg does not reorganize a table space on which a pending ALTER exists that changes the data type from BINARY to VARBINARY or from VARBINARY to BINARY.

### Pending limit key changes

When a limit key ALTER operation is pending on a table space, the following considerations apply:

- You must ensure that the `+DBNAME` parameter is included in your RRGPARMS.
- NGT Reorg skips processing for a table space when any of the following conditions exists:
  - Other pending changes exist on the table space.
  - The table space contains LOB or XML columns.
NGT Reorg invokes DSNUTILB when a WHERE or WHEN clause results in discarded rows.

If NGT Reorg finds rows after the last new limit key, NGT Reorg discards them. The following considerations apply in this case:

- NGT Reorg terminates if a row is added during change apply processing that would be discarded in this case.
- NGT Reorg archives these rows to a discard file (such as SYSDISC). For more information, see “DISCARDDN” on page 58.
- If your JCL includes a SYSPUNCH DD statement, NGT Reorg generates a LOAD statement that enables you to load the discarded rows.

Considerations for reorganizing LOB data

NGT Reorg processes LOB table spaces and their base table spaces separately and puts LOB table spaces in UT status.

The NGT LOBMaster product checks and reorganizes base and LOB auxiliary table spaces at the same time. For more information about NGT LOBMaster, see the BMC Next Generation Technology LOBMaster for DB2 for z/OS Reference Manual.

When you specify a wildcard list of table spaces to process, you can omit LOB table spaces by specifying the +LOB(NO) parameter. +LOB(NO) is the default. With +LOB(YES), NGT Reorg processes all base and LOB table spaces included by the wildcard specified.

The following restrictions apply when reorganizing LOB table spaces or their base table spaces:

- NGT Reorg does not rebalance partitions in table spaces that contain LOB columns or in LOB table spaces.
- NGT Reorg does not natively process PBG table spaces that contain LOB columns. Note the following considerations:
  - You can reorganize PBG table spaces that contain LOB columns by specifying AUX to reorganize both the LOB table space and the base table space. This option requires a license for NGT LOBMaster. For more information, see the BMC Next Generation Technology LOBMaster for DB2 for z/OS Reference Manual.
  - Otherwise, NGT Reorg invokes DSNUTILB to process PBG table spaces that contain LOB columns.
NGT Reorg does not process a base table space with LOB columns when that table space is in REORP status.

A reorganization to remove the REORP status might require moving rows between partitions, including the LOB rows in the associated LOB table spaces.

This restriction applies to any ALTER statement that puts a table space in REORP status, such as ALTER ADD PARTITION or ALTER ROTATE.

During a reorganization with discards from the base table, NGT Reorg puts the LOB auxiliary table in ACHKP status. After the reorganization, you can run IBM CHECK utilities to find and delete orphaned LOB rows.

If you use NGT LOBMaster instead of NGT Reorg, this restriction does not apply.

Considerations for reorganizing XML data

NGT Reorg processes XML table spaces and their base table spaces independently of one another.

XML support in NGT Reorg has the following limitations:

- NGT Reorg skips processing of a table space that contains both XML and LOB columns.

- NGT Reorg invokes DSNUTILB when any of the following conditions exists:
  - You specify DISCARD.
  - You are rebalancing partitions.
  - You are reorganizing a versioned XML table space and all of the following conditions exist:
    - You specify RBALRSN_CONVERSION EXTENDED.
    - The START_TS and END_TS column is currently defined as BINARY(8).
    - You are reorganizing all partitions.

- If you are using an EDITPROC, NGT Reorg functions as follows:
  - If you are reorganizing the base table space, NGT Reorg skips EDITPROC processing.
  - If you are reorganizing the XML table space, NGT Reorg terminates processing of the table space.

- NGT Reorg does not produce statistics when reorganizing XML table spaces.
Reorganizations that invoke DSNUTILB

When NGT Reorg encounters an object that it does not support natively, it invokes the IBM DSNUTILB utility.

The UTLPARMS parameter +DSNUTILB must be YES to enable NGT Reorg to invoke DSNUTILB.

NGT Reorg converts or eliminates options in your REORG statement as needed, then passes it to DSNUTILB for processing. You can specify additional options on your NGT Reorg syntax statement that control the IBM DB2 REORG utility. The syntax diagrams for the REORG TABLESPACE command ("REORG TABLESPACE diagrams" on page 40) and REORG INDEX command ("REORG INDEX diagrams" on page 61) identify these options.

Additional NGT Reorg considerations

The following considerations also apply to NGT Reorg jobs.

Considerations for image copies in NGT Reorg

The following considerations apply to image copies in NGT Reorg:

For efficiency, NGT Reorg might initiate an extra copy server. This server job functions solely to produce image copies from objects that have been already reorganized and completed.

In addition to the syntax options described in “REORG TABLESPACE syntax” on page 40, you can use the following controls.

**Note**

NGT Reorg supports only dynamic allocation of copy data sets. If you specify copy DD statements in your JCL, NGT Reorg ignores them.

Global parameters

The following global parameters enable you to control copy processing. For more information about these parameters, see the *BMC Next Generation Technology General User Guide*:

- +FLASHCOPY enables you to specify whether FlashCopy image copies are required or preferred.
+SVRSPECIAL controls special server processing for image copies.

**NGT TAPE command**

The NGT TAPE command passes tape unit allocation data to NGT Reorg. For more information about this command, see the *BMC Next Generation Technology General User Guide*.

**OUTPUT command**

The OUTPUT command passes allocation information to NGT Reorg. For more information about this command, see the *BMC Next Generation Technology General User Guide*.

**Automation control points**

The following automation control points enable you to control copy processing. For more information, see the *BMC Next Generation Technology Automation Reference Manual*.
- XCPYDYNM provides image copy data set names for each image copy.
- XCPYINIT dynamically allocates tape drives.
- XCPYMIA controls routing of image copies to disk or tape devices.

**Row count differences**

NGT Reorg might report a differing number of keys before and after a reorganization. Row deletions performed before or during the reorganization might cause this difference. This is not indicative of a program error or data integrity issue.

**Differences between REORG PLUS and NGT Reorg**

Review this topic if you are an existing user of REORG PLUS and need to understand how NGT Reorg differs from the functionality you have been using.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>For syntax options that are available with NGT Reorg, see “Alphabetical listing of REORG options” on page 37.</td>
</tr>
<tr>
<td>zIIP offload processing</td>
<td>NGT Reorg does not require the BMC EXTENDED BUFFER MANAGER for DB2 (XBM) product or its SNAPSHOT UPGRADE FEATURE for DB2 (SUF). The NGT subsystem handles zIIP offload processing.</td>
</tr>
</tbody>
</table>
## Differences between REORG PLUS and NGT Reorg

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data sets: dynamic allocation</td>
<td>With NGT Reorg, you can dynamically allocate your discard, SYSPUNCH, and copy data sets by either using the appropriate automation control point or by using the OUTPUT command. For more information about the automation control point, see the <em>BMC Next Generation Technology General User Guide</em>. For more information about the OUTPUT command, see “OUTPUT command” on page 39.</td>
</tr>
<tr>
<td>Discard processing</td>
<td>NGT Reorg uses a DISCARD syntax option rather than DELETE. You can specify conditional information with DISCARD as you could with DELETE. For information about differences, see “DISCARD FROM TABLE” on page 60.</td>
</tr>
<tr>
<td>Wildcards</td>
<td>NGT Reorg enables you to use wildcards to reorganize multiple table spaces or indexes in a single job. For more information about wildcard use, see the <em>BMC Next Generation Technology General User Guide</em>.</td>
</tr>
<tr>
<td>Statistics</td>
<td>NGT Reorg does not update the BMC DASD MANAGER PLUS for DB2 statistics tables. To update DB2 catalog statistics, you specify the +RUNSTATS parameter. For information about the statistics that are updated, see the <em>BMC Next Generation Technology Stats for DB2 for z/OS Reference Manual</em>.</td>
</tr>
<tr>
<td>Conditional reorganization</td>
<td>NGT Reorg does not provide conditional reorganization syntax options. To perform a conditional reorganization, use NGT Reorg from NGT Utility Manager. These conditional reorganizations use only DB2 catalog tables, not the DASD MANAGER PLUS statistics tables. For more information, see the <em>BMC Next Generation Technology Utility Manager for DB2 for z/OS Reference Manual</em>.</td>
</tr>
<tr>
<td>Row selection</td>
<td>NGT Reorg does not provide the ability to select rows for processing based on a WHERE clause.</td>
</tr>
<tr>
<td>Column value update</td>
<td>NGT Reorg does not provide the ability to use REORG options to update column values during the reorganization.</td>
</tr>
<tr>
<td>Copies</td>
<td>NGT Reorg enables you to produce IBM FlashCopy image copies with the reorganization. NGT Reorg does not provide the ability to produce incremental copies with the reorganization.</td>
</tr>
<tr>
<td>Online reorganization</td>
<td>NGT Reorg always performs an online reorganization, leaving the object in RW status. You do not have to do anything special to run this reorganization and NGT Reorg does not require XBM or SUF.</td>
</tr>
<tr>
<td>Installation options</td>
<td>Instead of using the DOPTs installation options, NGT Reorg uses configuration parameters to provide default processing values. You can also specify these parameters in your SYSIN.</td>
</tr>
<tr>
<td>DB2 catalog objects</td>
<td>NGT Reorg does not reorganize DB2 catalog objects.</td>
</tr>
<tr>
<td>Partition-by-growth objects</td>
<td>NGT Reorg converts all partitions to use the same compression dictionary.</td>
</tr>
</tbody>
</table>
Input for NGT Reorg

This chapter describes the DD statements, SYSIN input (statement syntax), and keywords for the NGT Reorg product.

DD statements for NGT Reorg

This topic describes NGT Reorg DD statements.

**SYSIN**

This is the only DD statement required for use with NGT Reorg. Specify utility statements as follows:

```plaintext
//SYSIN DD *
statement
statement
statement
```

**RRGPARMS**

This DD statement is required only to override the NGT Reorg parameters specified during installation. You can specify RRGPARMS in a data set or in the JCL. For example:

```plaintext
//RRGPARMS DD DISP=SHR,DSN=NGT.REORG.PARMS
```

or

```plaintext
//RRGPARMS DD *
```
**RRGPARMX**

This DD statement is required only to override the NGT Reorg parameters for index processing specified during installation. You can specify RRGPARMX in a data set or in the JCL. For example:

```
//RRGPARMX DD DISP=SHR,DSN=NGT.REORG.PARMS
```

or

```
//RRGPARMX DD *
```

**UTLPARMS**

This DD statement is required only to override NGT utilities global parameters specified during installation. You can specify UTLPARMS in a data set or in the JCL. For example:

```
//UTLPARMS DD DISP=SHR,DSN=NGT.UTIL.PARMS
```

or

```
//UTLPARMS DD *
```

For more information about NGT utilities global parameters, see the *BMC Next Generation Technology General User Guide*.

**RUNSTATS**

*(optional)* This DD will contain RUNSTATS reports. You can send this DD statement to an output file or to a SYSOUT output class. For example:

```
//RUNSTATS DD DISP=SHR,DSN=NGT.RUNSTATS.REPORT
```

or

```
//RUNSTATS DD SYSOUT=* 
```

**SYSDISC**

*(optional)* This DD enables you to archive discarded rows. For example:

```
//SYSDISC DD DISP=(MOD,CATLG,DELETE),DSN=NGT.REORG.DISCARDS
```

The following considerations apply to SYSDISC:
To have NGT Reorg dynamically allocate your discard file, specify DISCARDDN on the REORG command instead of specifying a SYSDISC DD statement. For more information, see “DISCARDDN” on page 58.

If you specify both DISCARDDN and a SYSDISC DD statement, NGT Reorg uses the DISCARDDN option instead of the DD statement.

If you are reorganizing a partitioned table space and you specify a SYSDISC DD statement in your JCL, BMC makes the following recommendations:

—Specify DISP=(MOD,CATLG,DELETE). If you specify DISP=(NEW,CATLG,DELETE), SYSDISC will contain only the last partition's discarded rows.

—Do not specify the RLSE subparameter for the SPACE parameter. This specification can cause a B37 abend when the data set is reopened for a partition's discards.

**SYSPUNCH**

*(optional)* This DD will contain a generated LOAD statement if rows are discarded during the reorganization. For example:

```plaintext
//SYSPUNCH DD DISP=SHR, DSN=NGT.REORG.LOADSTMT
```

You must specify the DISCARDDN option with the FORMAT keyword. However, if you specify FORMAT UNLOAD, NGT Reorg does not write a LOAD statement to SYSPUNCH, (but discarded rows are still written to the discard file).

Instead of providing a data set name with the DD statement, you can dynamically allocate the data set by providing the data set name in an OUTPUT command in your NGT Reorg SYSIN. You must place the OUTPUT command before the REORG statement. For more information, see the *BMC Next Generation Technology General User Guide*.

**NGTAUTO**

If you are using automation control points with your job, you can specify an NGTAUTO DD statement to override the default data set (specified during configuration or with the +NGTAUTO1 or +NGTAUTO2 parameter).

This statement should reference a partitioned data set that contains the NGT automation control points. For example:

```plaintext
//NGTAUTO DD DISP=SHR, DSN=NGT.AUTOMATN.CTRL.PTS
```

For more information about NGT automation control points, see the *BMC Next Generation Technology Automation Reference Manual*. 
**DDLIN**

*(optional)* Include this DD in your JCL if you want to supply ALTER statements to update limit keys during your reorganization. NGT Reorg rebalances your partitions based on these limit key changes and updates the limit keys in the DBD and DB2 catalog.

DDLIN is the default DD name for this data set. Alternatively, you can specify the DDLIN option ("DDLIN" on page 49) on your REORG command to supply a different DD name.

For more information, see “Considerations for rebalancing partitions” on page 21.

**PARTSDDL**

*(optional)* Include this DD in your JCL if you are rebalancing partitions. NGT Reorg writes DDL to this data set that you can use to create the table space, tables, and indexes that you are reorganizing.

If you do not include this DD statement, NGT Reorg issues a warning message but continues the reorganization and rebalances your partitions.

**REPBLDAL**

*(optional)* Include this DD in your JCL if you are rebalancing partitions. NGT Reorg writes ALTER statements to this DD for the object that you are reorganizing. These ALTER statements reflect the new keys following the rebalance operation.

If you do not include this DD statement, NGT Reorg issues a warning message but continues the reorganization and rebalances your partitions.

---

**NGT Reorg syntax overview**

NGT Reorg enables you to specify either REORG TABLESPACE or REORG INDEX.

You can also specify the OUTPUT command to dynamically allocate certain data sets. For more information, see “OUTPUT command” on page 39.

---

**Command syntax rules for NGT utilities**

The following general rules apply to NGT utilities command syntax:
In an NGT utility command, the utility recognizes the following comment characters:

— An asterisk (*) in column 1
   The utility ignores the entire line.

— Two consecutive hyphens in a line
   The utility ignores the remainder of the line.

— REXX-like comments (starting with /* and ending with */)
   The utility ignores the comment text, including /* and */. You can specify these comments anywhere within the statement and you can nest these comments.

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, the utility 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.

Except where indicated in the syntax diagram, you cannot specify the same command option more than once. Duplicate command options cause the utility to terminate.

In the syntax diagrams in the following pages, underlined options indicate default options.

Alphabetical listing of REORG options

The following table lists the NGT Reorg syntax options alphabetically and indicates where to find each option description. The third column in the table indicates whether the option is valid for REORG INDEX.

<table>
<thead>
<tr>
<th>Option</th>
<th>Reference</th>
<th>Valid for REORG INDEX?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOESTSPACE</td>
<td>“AUTOESTSPACE” on page 50</td>
<td>No</td>
</tr>
<tr>
<td>AUX</td>
<td>“AUX” on page 48</td>
<td>No</td>
</tr>
<tr>
<td>Option</td>
<td>Reference</td>
<td>Valid for REORG INDEX?</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>CLONE</td>
<td>“CLONE” on page 50</td>
<td>Yes</td>
</tr>
<tr>
<td>COPYDDN</td>
<td>“COPYDDN (lp1,lb1)” on page 53</td>
<td>No</td>
</tr>
<tr>
<td>CURRENT_DATE</td>
<td>“CURRENT_DATE” on page 57</td>
<td>No</td>
</tr>
<tr>
<td>CURRENT_TIMESTAMP</td>
<td>“CURRENT_TIMESTAMP” on page 57</td>
<td>No</td>
</tr>
<tr>
<td>DDLIN</td>
<td>“DDLIN” on page 49</td>
<td>No</td>
</tr>
<tr>
<td>DEADLINE</td>
<td>“DEADLINE” on page 52</td>
<td>Yes</td>
</tr>
<tr>
<td>DELAY</td>
<td>“DELAY” on page 53</td>
<td>Yes</td>
</tr>
<tr>
<td>DISCARD FROM TABLE</td>
<td>“DISCARD FROM TABLE” on page 60</td>
<td>No</td>
</tr>
<tr>
<td>DISCARDDN</td>
<td>“DISCARDDN” on page 58</td>
<td>No</td>
</tr>
<tr>
<td>DRAIN</td>
<td>“DRAIN” on page 52</td>
<td>Yes</td>
</tr>
<tr>
<td>DRAIN_ALLPARTS</td>
<td>“DRAIN_ALLPARTS” on page 57</td>
<td>No</td>
</tr>
<tr>
<td>DRAIN_WAIT</td>
<td>“DRAIN_WAIT” on page 56</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNUM ALL</td>
<td>“DSNUM ALL” on page 47</td>
<td>No</td>
</tr>
<tr>
<td>EXCLUDE INDEX</td>
<td>“EXCLUDE INDEX” on page 64</td>
<td>Not valid for REORG TABLESPACE</td>
</tr>
<tr>
<td>EXCLUDE TABLESPACE</td>
<td>“EXCLUDE TABLESPACE” on page 46</td>
<td>No</td>
</tr>
<tr>
<td>FLASHCOPY YES</td>
<td>“FLASHCOPY YES” on page 54</td>
<td>No</td>
</tr>
<tr>
<td>FORCE</td>
<td>“FORCE” on page 51</td>
<td>Yes</td>
</tr>
<tr>
<td>INDEX</td>
<td>“REORG INDEX[ES] or INDEXSPACE” on page 62</td>
<td>Not valid for REORG TABLESPACE</td>
</tr>
<tr>
<td>LOBRECLAIMPERCENT</td>
<td>“LOBRECLAIMPERCENT integer” on page 47</td>
<td>No</td>
</tr>
<tr>
<td>LOGRANGES</td>
<td>“LOGRANGES” on page 57</td>
<td>Yes</td>
</tr>
<tr>
<td>LONGLOG</td>
<td>“LONGLOG” on page 53</td>
<td>Yes</td>
</tr>
<tr>
<td>LPART[S]</td>
<td>“PART[S] and LPART[S]” on page 46</td>
<td>Yes</td>
</tr>
<tr>
<td>MAPPINGDATABASE</td>
<td>“MAPPINGDATABASE” on page 52</td>
<td>No</td>
</tr>
<tr>
<td>MAPPINGTABLE</td>
<td>“MAPPINGTABLE” on page 52</td>
<td>No</td>
</tr>
<tr>
<td>MASSDELETE</td>
<td>“MASSDELETE” on page 55</td>
<td>No</td>
</tr>
</tbody>
</table>
### OUTPUT command

You can use the OUTPUT command, with your NGT Reorg SYSIN, to dynamically allocate discard, SYSPUNCH, and copy data sets. Place the OUTPUT command before your REORG statement.

For details, see the following information:

<table>
<thead>
<tr>
<th>Option</th>
<th>Reference</th>
<th>Valid for REORG INDEX?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXRO</td>
<td>“MAXRO” on page 52</td>
<td>Yes</td>
</tr>
<tr>
<td>NOCLIX</td>
<td>“NOCLIX” on page 47</td>
<td>No</td>
</tr>
<tr>
<td>PART[S]</td>
<td>“PART[S] and LPART[S]” on page 46 and “PART[S] and LPART[S]” on page 63</td>
<td>Yes</td>
</tr>
<tr>
<td>PREFORMAT</td>
<td>“PREFORMAT” on page 47</td>
<td>No</td>
</tr>
<tr>
<td>PUNCHDDN</td>
<td>“PUNCHDDN” on page 51</td>
<td>No</td>
</tr>
<tr>
<td>RBALRSN_CONVERSION</td>
<td>“RBALRSN_CONVERSION” on page 48</td>
<td>No</td>
</tr>
<tr>
<td>REBALANCE</td>
<td>“REBALANCE” on page 49</td>
<td>No</td>
</tr>
<tr>
<td>REBUILDDICT</td>
<td>“REBUILDDICT” on page 47</td>
<td>No</td>
</tr>
<tr>
<td>RECOVERYDDN</td>
<td>“RECOVERYDDN (rp1,rb1)” on page 54</td>
<td>No</td>
</tr>
<tr>
<td>REPORTONLY</td>
<td>“REPORTONLY” on page 47</td>
<td>Yes</td>
</tr>
<tr>
<td>RETRY</td>
<td>“RETRY” on page 56</td>
<td>Yes</td>
</tr>
<tr>
<td>RETRY_DELAY</td>
<td>“RETRY_DELAY” on page 56</td>
<td>Yes</td>
</tr>
<tr>
<td>ROWFORMAT</td>
<td>“ROWFORMAT” on page 48</td>
<td>No</td>
</tr>
<tr>
<td>SCOPE</td>
<td>“SCOPE” on page 50</td>
<td>No</td>
</tr>
<tr>
<td>SORTDEVT</td>
<td>“SORTDEVT” on page 51</td>
<td>Yes</td>
</tr>
<tr>
<td>SORTNPSI</td>
<td>“SORTNPSI” on page 51</td>
<td>No</td>
</tr>
<tr>
<td>SORTNUM</td>
<td>“SORTNUM” on page 51</td>
<td>Yes</td>
</tr>
<tr>
<td>SWITCHTIME</td>
<td>“SWITCHTIME” on page 57</td>
<td>Yes</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>“TIMEOUT” on page 56</td>
<td>Yes</td>
</tr>
<tr>
<td>WORKDDN</td>
<td>“WORKDDN” on page 64</td>
<td>Not valid for REORG TABLESPACE</td>
</tr>
</tbody>
</table>
REORG TABLESPACE syntax

This topic describes the NGT Reorg REORG TABLESPACE syntax.

REORG TABLESPACE diagrams

This topic contains syntax diagrams for the NGT Reorg REORG TABLESPACE command.

For option descriptions, see “REORG TABLESPACE options” on page 45.
REORG TABLESPACE syntax

Chapter 3  Input for NGT Reorg  41

1 Used only when invoking DSNUTILB
REORG TABLESPACE syntax

1 Used only when invoking DSNUTILB
copy specification block

\[
\text{COPYDDN} \quad ( \quad \text{SYSCOPY} \quad ,p1 \quad ,lb1 \quad ) \quad \text{FLASHCOPY YES}
\]

massdelete specification block

\[
\text{MASSDELETE} \quad ( \quad \text{PART} \quad ,\text{LPART} \quad ,\text{PARTS} \quad ,\text{LPARTS} \quad ) \quad \text{partitionNumber} \quad : \text{partitionNumber}
\]

\[
\text{DRAIN\_WAIT} \quad \text{integer}^1 \quad \text{RETRY} \quad \text{integer}^2 \quad \text{RETRY\_DELAY} \quad \text{integer}^2
\]

\[
\text{TIMEOUT}^1 \quad \text{TERM} \quad \text{ABEND} \quad \text{LOGRANGES}^1 \quad \text{YES} \quad \text{NO} \quad \text{DRAIN\_ALLPARTS}^1 \quad \text{NO} \quad \text{YES}
\]

\[
\text{SWITCHTIME}^1 \quad \text{NONE} \quad \text{timestamp} \quad \text{labeledDurationExpression} \quad \text{NEWMAXRO} \quad \text{NONE} \quad \text{integer}
\]

\[^1\text{Used only when invoking DSNUTILB}\]
**duration specification block**

CURRENT_DATE\(^1\)

CURRENT_TIMESTAMP\(^1\)

WITH TIME ZONE

+ constant

YEAR

YEARS

MONTH

MONTHS

DAY

DAYS

HOUR

HOURS

MINUTE

MINUTES

SECOND

SECONDS

MICROSECOND

MICROSECONDS

\(^1\) Used only when invoking DSNUTILB

**discard specification block**

DISCCARD

FROM TABLE creator.tableName

WHERE

(-- WHERE clause --)

UNLOAD

UNL

DSNTIAUL

DSN

EXTERNAL

EXT

REPORT

REP

ASCIIDELE

ASC

SYSDISC

ddbname

FORMAT

(-- PAD, NOPAD --)
REORG TABLESPACE options

This topic describes the options available to you when reorganizing a table space.

REORG TABLESPACE[S]

This specification tells NGT Reorg which table space or group of table spaces to reorganize.

databaseName.tableSpaceName

You must specify both the database name and the table space name. The default name is DSNDB04.

databasePattern.tableSpacePattern

You can use the percent (%) and underscore (_) symbols as wildcards to construct an SQL query that obtains all table spaces that match the specified pattern:

- Use % to represent any character string.
- Use _ to represent a single character.

A line will print on SYSPRINT and SUMMARY for each object found.

Specify databaseName.% to reorganize all the table spaces in a database. In the following example, every table space beginning with the character "A" in the database BIGDBASE will be reorganized:

REORG TABLESPACE BIGDBASE.A%

OBJECTSET objectSetName

Specify the name of an existing object set to have NGT Reorg reorganize all table spaces defined in that object set.

You can define object sets in the BMC DASD MANAGER PLUS for DB2 product. For more information about object sets, see the DASD MANAGER PLUS for DB2 User Guide.

You cannot use the following options when using an object set:

- PART[S] or LPART[S]
- EXCLUDE TABLESPACE
- MASSDELETE
- DISCARD
PART[S] and LPART[S]

Use these options to specify which partition or partitions to reorganize.

Use the PART or PARTS option to specify one or more physical partitions to reorganize. Use the LPART or LPARTS option to specify one or more logical partitions to reorganize.

The following restrictions apply to these options:

- NGT Reorg ignores this option when reorganizing a partition-by-growth (PBG) table space. All PBG partitions are reorganized simultaneously.
- You cannot use these options when you are reorganizing table spaces based on an object set.

Specifying partitions

The following information applies to all four options:

- You can specify partitions by number using integer values from 1 through 4096.
- You can specify a range of contiguous partitions as `partitionNumber : partitionNumber`.
- When you specify a range of partitions, the second number must be greater than the first.
- You can specify a mixture of individual partitions and ranges of partitions.
- If you specify REBALANCE, you cannot specify a subset of partitions.

Note

LPARTS does not support wildcards.

EXCLUDE TABLESPACE

When you specify a group of table spaces to reorganize, you can refine that list of table spaces by using the EXCLUDE TABLESPACE option, as shown in the following example:

```
REORG TABLESPACE ADMINDB.CAR%
   EXCLUDE TABLESPACE ADMINDB.CARMIDSZ
   EXCLUDE TABLESPACE ADMINDB.CARLUXRY
   EXCLUDE TABLESPACE ADMINDB.CARSUVS
```

You cannot use this option when you are reorganizing table spaces based on an object set.
REPORTONLY

The REPORTONLY option generates a summary report only. A reorganization of the object does not take place.

PREFORMAT

The PREFORMAT option tells NGT Reorg to preformat a table space page when it is allocated. If PREFORMAT is specified, NGT Reorg preformats all pages up to the high allocated RBA. Preformatting writes zero pages from the high used RBA to the high allocated RBA.

REBUILDICT

The REBUILDICT option tells NGT Reorg to rebuild the compression dictionary.

NOCLIX

The NOCLIX option tells NGT Reorg to ignore the clustering index and process the table space as if there were no clustering index. This will cause the reorganization to reclaim space only.

This keyword is valid only for segmented nonpartitioned table spaces or partition-by-growth table spaces.

LOBRECLAIMPERCENT integer

The LOBRECLAIMPERCENT option specifies the percentage of savings that you want to achieve by performing the reorganization.

If the specified savings cannot be met, NGT Reorg skips the reorganization. The savings is calculated by subtracting the number of pages needed for the output data set from the current high used RBA.

You can specify a value from 0 through 99. The default value is 20. Specifying 0 forces the reorganization to run. This option overrides the +LOBPCT parameter.

DSNUM ALL

The presence of DSNUM ALL tells NGT Reorg that you want to reorganize all partitions of a PBG table space. In addition, NGT Reorg relocates rows between partitions and adds new partitions, if needed.
**Note**
By default, NGT Reorg processes all partitions in a partition-by-growth table space. You can specify DSNUM ALL, but it is not required.

**MAXDISORG integer**

The MAXDISORG keyword prevents unexpected reorganizations of large, highly disorganized partition-by-growth table spaces. The integer value specifies the percentage of disorganized rows. The default value is 50, or 50% disorganized. This keyword is ignored if the NOCLIX keyword is specified.

**ROWFORMAT**

The ROWFORMAT option tells NGT Reorg to convert the row format during a reorganization.

**BRF**

Converts the row format from reordered row format (RRF) to basic row format (BRF).

**RRF**

Converts the row format from BRF to RRF.

**AUX**

The AUX option tells the BMC NGT LOBMaster for DB2 for z/OS product whether to process both the base table space and any LOB auxiliary table spaces at the same time.

If you specify both AUX and REBALANCE, the utility invokes DSNUTILB.

AUX requires an NGT LOBMaster license. For more information, see the *BMC Next Generation Technology LOBMaster for DB2 for z/OS Reference Manual*.

**RBALRSN_CONVERSION**

The RBALRSN_CONVERSION option tells NGT Reorg to convert the DB2 RBAs and LRSNs between basic (6 byte) and extended (10 byte) format.

The default behavior depends on the UTILITY_OBJECT_CONVERSION subsystem parameter:
If you do not specify RBALRSN_CONVERSION and have set UTILITY_OBJECT_CONVERSION to BASIC, then RBALRSN_CONVERSION defaults to BASIC.

If you have set UTILITY_OBJECT_CONVERSION to NOBASIC or EXTENDED, then RBALRSN defaults to EXTENDED.

**BASIC**

Converts from extended format to basic format. If you specify BASIC and set the UTILITY_OBJECT_CONVERSION subsystem parameter to NOBASIC, the object fails the reorganization.

**EXTENDED**

Converts from basic format to extended format. If you specify EXTENDED and set the UTILITY_OBJECT_CONVERSION subsystem parameter to BASIC, the object fails the reorganization.

NGT Reorg invokes DSNUTILB when you specify RBALRSN_CONVERSION EXTENDED and all of the following conditions exist:

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

**NONE**

Does not perform conversions. If you specify NONE, have set the UTILITY_OBJECT_CONVERSION subsystem parameter to NOBASIC, and the object is in basic format, the object fails the reorganization.

**DDLIN**

The DDLIN option enables you to override the default DD name of the DDLIN data set. Use the DDLIN data set to supply ALTER statements to NGT Reorg to rebalance partitions.

For more information, see “Considerations for rebalancing partitions” on page 21 and “DDLIN” on page 36.

**REBALANCE**

The REBALANCE option enables you to rebalance partitions during your reorganization. NGT Reorg uses this option to rebalance your partitions when no limit key ALTER operations are pending on the table space and no DDLIN specification or DD exists.
NGT Reorg terminates when any of the following conditions exist:

- You specify DISCARD or MASSDELETE.
- You are running NGT Reorg under NGT Utility Manager.
- You specify a subset of partitions. (When you specify REBALANCE, you must rebalance all partitions in the table space.)

NGT Reorg invokes DSNUTILB if you are reorganizing an XML table space or its base table space.

For additional partition rebalancing requirements and limitations, see “Considerations for rebalancing partitions” on page 21.

**SORTCLUSTER**

(Used only when invoking DSNUTILB) SORTCLUSTER tells the utility whether to sort the data records in clustering order during a rebalance operation. NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

**CLONE**

(Used only when invoking DSNUTILB) CLONE indicates that you want to reorganize only clone tables from the table spaces that you specify.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

**SCOPE**

(Used only when invoking DSNUTILB) SCOPE indicates which table spaces or partitions you want to be reorganized.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

**AUTOESTSPACE**

(Used only when invoking DSNUTILB) AUTOESTSPACE tells the utility to calculate the size of the fixed hash space for hash-organized table spaces.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.
FORCE

*(Used only when invoking DSNUTILB)* FORCE specifies what thread cancelation to perform when draining the table space.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.

SORTNPSI

*(Used only when invoking DSNUTILB)* SORTNPSI specifies whether to sort all keys of a nonpartitioned secondary index.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.

PUNCHDDN

*(Used only when invoking DSNUTILB)* PUNCHDDN specifies the DD statement to contain LOAD utility control statements when generated.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.

SORTDEVT

*(Used only when invoking DSNUTILB)* SORTDEVT specifies the device type to use for temporary data sets used by the sort process.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.

SORTNUM

*(Used only when invoking DSNUTILB)* SORTNUM specifies the number of temporary sort data sets to be allocated.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.  

Chapter 3  Input for NGT Reorg  51
MAPPINGTABLE

(Used only when invoking DSNUTILB) MAPPINGTABLE specifies the name of the table to use to map RIDs between the original object and the shadow copy of the object.

You cannot specify both MAPPINGTABLE and MAPPINGDATABASE.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

MAPPINGDATABASE

(Used only when invoking DSNUTILB) MAPPINGDATABASE specifies the name of the database to contain the implicitly created mapping table.

You cannot specify both MAPPINGDATABASE and MAPPINGTABLE.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

DEADLINE

(Used only when invoking DSNUTILB) DEADLINE specifies the latest time by which the switch phase must begin.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

MAXRO

(Used only when invoking DSNUTILB) MAXRO specifies the maximum amount of time for the read-only phase of log processing.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

DRAIN

(Used only when invoking DSNUTILB) DRAIN specifies the drain behavior during the log phase.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.
LONGLOG

(Used only when invoking DSNUTILB) LONGLOG specifies the action to take when the utility encounters a longlog condition.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

DELAY

(Used only when invoking DSNUTILB) DELAY specifies the minimum amount of time between sending the LONGLOG message to the console and performing the action specified by the LONGLOG option.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

COPYDDN (/lp1,lb1)

COPYDDN specifies local primary (LP) and local backup (LB) copy destinations.

These parameters do not specify a DD name; they only signify the intention to produce a copy to this destination. The actual unit name and options for the copy destinations are defined by one of the following methods. The value of the +OVERRIDEOUTPUT parameter determines the method that NGT Reorg uses.

- The XCPYDYNM automation control point (either directly or by using the NGTTAPE command)
- The OUTPUT command
  You must place the OUTPUT command before the REORG statement in your SYSIN.

For more information about XCPYDYNM, see the BMC Next Generation Technology Automation Reference Manual. For more information about the OUTPUT command and the +OVERRIDEOUTPUT parameter, see the BMC Next Generation Technology General User Guide.

The following requirements and considerations apply to the COPYDDN option:

- You must specify the OUTPUT command, or the XCPYDYNM automation control point must be available to provide the image copy data set name for each image copy.
IBM SHRLEVEL CHANGE REORG requires that a copy be generated with the reorganization. BMC NGT is following IBM's standards. If this clause is not specified, the default is to produce a local primary (LP1) copy only.

The parentheses are optional if you specify only one destination name. The parentheses are required if you specify two DD names.

**RECOVERYDDN \((rp1, rb1)\)**

RECOVERYDDN specifies remote primary (RP) and remote backup (RB) copy destinations.

These parameters do not specify a DD name; they only signify the desire to produce a copy to this destination. The actual unit name and options for the copy destinations are defined by one of the following methods. The value of the +OVERRIDEOUTPUT parameter determines the method that NGT Reorg uses.

- The XCPYDYNM automation control point
- The NGTTAPE command
- The OUTPUT command

You must place the OUTPUT command before the REORG statement in your SYSIN.

For more information about XCPYDYNM, see the *BMC Next Generation Technology Automation Reference Manual*. For more information about the OUTPUT command and the +OVERRIDEOUTPUT parameter, see the *BMC Next Generation Technology General User Guide*.

The following requirements and considerations apply to the RECOVERYDDN option:

- You must specify the OUTPUT command, or the XCPYDYNM automation control point must be available to provide the image copy data set name for each image copy.

- The parentheses are optional if you specify only one destination name. The parentheses are required if you specify two DD names.

**FLASHCOPY YES**

FLASHCOPY YES requests a FlashCopy image copy from NGT Reorg.

NGT Reorg fails if the volumes do not support FlashCopy.
Note

NGT Reorg outputs FlashCopy image copies to VSAM data sets. Because of this, the copy data set cannot be a generation data group (GDG) data set. Also, there will be a cluster and data entry, so the DSNAME must leave two bytes for the .D to be added. XCPYDYNM can produce a different name for FlashCopy data sets.

MASSDELETE

This option specifies that all rows of the partitions specified by the PART, PARTS, LPART, or LPARTS keywords will be deleted from the table.

Specifying partitions

The following information applies to all four keywords:

- You can specify partitions by number using integer values from 1 through 4096.
- You can specify a range of contiguous partitions as `partitionNumber:partitionNumber`.
- When you specify a range of partitions, the second number must be greater than the first.
- You can specify a mixture of individual partitions and ranges of partitions.
- When you specify a range of partitions for MASSDELETE, this range must be a subset of the partitions specified for the reorganization. For example, if you are reorganizing a 32-partition table space, and you have specified that you want to reorganize partitions 18 through 24, any of the following partition range specifications are valid for MASSDELETE:
  - 18:19
  - 20:23
  - 21:22

On the other hand, you cannot specify MASSDELETE PARTS(18:26) in this example because partitions 25 and 26 are outside the range of partitions that you specified to reorganize.

Considerations

NGT Reorg terminates if you specify any of the following options with MASSDELETE:

- DISCARD
- REBALANCE
- DDLIN
The following additional considerations apply to MASSDELETE:

- In one input stream, you can specify only one REORG TABLESPACE statement for a table space. (Multiple statements referring to the same table space will NOT be grouped under one unit of work and will result in processing the object multiple times.)

- When a partition is subject to MASSDELETE, the compression dictionary will be maintained or removed based on the COMPRESS catalog flag. If the COMPRESS flag is on but the partition does not have a dictionary, a dictionary is built from the data currently in the partition.

- During the reorganization, any update or delete to a row in a partition that is subject to MASSDELETE will cause NGT Reorg to terminate. Newly inserted rows are not subject to MASSDELETE processing.

**DRAIN_WAIT**

(Used only when invoking DSNUTILB) DRAIN_WAIT tells the utility how long to wait when draining the object.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

**RETRY**

(Used only when invoking DSNUTILB) RETRY indicates the maximum number of times that you want the utility to retry attempting a drain.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

**RETRY_DELAY**

(Used only when invoking DSNUTILB) RETRY_DELAY specifies the duration between drain retries.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

**TIMEOUT**

(Used only when invoking DSNUTILB) TIMEOUT specifies the action to take if the utility times out while attempting to drain an object.
NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.

**LOGRANGES**

*(Used only when invoking DSNUTILB)* LOGRANGES specifies whether to use SYSLGRNX information for the log phase.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.

**DRAIN_ALLPARTS**

*(Used only when invoking DSNUTILB)* DRAIN_ALLPARTS specifies the action to take when a nonpartitioned secondary index is defined on the partitioned table space that you are reorganizing.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.

**SWITCHTIME**

*(Used only when invoking DSNUTILB)* SWITCHTIME specifies the time for the final (read-only) iteration of the log phase to begin.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.

**CURRENT_DATE**

*(Used only when invoking DSNUTILB)* CURRENT_DATE tells the utility to calculate the deadline or switchtime based on the current date.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.

**CURRENT_TIMESTAMP**

*(Used only when invoking DSNUTILB)* CURRENT_TIMESTAMP tells the utility to calculate the deadline or switchtime based on the current timestamp.

NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 *Utility Guide and Reference*.
DISCARDDN

The DISCARDDN option tells NGT Reorg to write all deleted rows to a sequential file.

**ddname**

Specify a valid DD name for this file. Data control block characteristics will be set internally. NGT Reorg writes each deleted row as a separate record in the file. The default format is UNLOAD, which is the same as the DB2 REORG UNLOAD format, suitable for being loaded into a similar table using the LOAD ... FORMAT UNLOAD statement. See the *BMC Next Generation Technology Load for DB2 for z/OS Reference Manual* for details.

A special case exists if NGT Reorg is discarding rows due to a DISCARD specification, but neither the DISCARDDN option nor a SYSDISC DD is present. In this case, any discarded rows are *permanently lost*, regardless of whether you have requested dynamic allocation of a discard data set. The discarded rows are not sent to any data set, and no data set is dynamically allocated.

If either SYSDISC or DISCARDDN is specified, NGT Reorg writes the discarded rows to the specified data set. If you specify DISCARDDN but a data set is not defined, NGT Reorg dynamically allocates the data set by using one of the following methods. The value of the +OVERRIDEOUTPUT parameter determines the method that NGT Reorg uses.

- **The XULDDYNM automation control point**
  
  In XULDDYNM, the *dsname* variable sets the data set name.

- **The OUTPUT command**
  
  You must place the OUTPUT command before the REORG statement in your SYSIN.

For details about XULDDYNM, see the *BMC Next Generation Technology Automation Reference Manual*. For more information about the OUTPUT command and the +OVERRIDEOUTPUT parameter, see the *BMC Next Generation Technology General User Guide*.

If you specify a SYSDISC DD *and* the DISCARDDN option, then NGT Reorg writes the discarded rows only to the dynamically-allocated data set and not to any data set associated with the SYSDISC DD.

**FORMAT**

Specifies the format of the discard data set. You must specify DISCARDDN to use this option. If you do not specify a format, the default format is UNLOAD. Note the following considerations:
- If FORMAT UNLOAD is in effect, NGT Reorg does not write generated LOAD statements to SYSPUNCH.

- If you are specifying DISCARDDN to archive rows being discarded due to altered limit keys, BMC recommends that you specify FORMAT(DSN,NOPAD).

**DSNTIAUL**

Uses the IBM DB2 internal format DSNTIAUL. You can abbreviate this as DSN.

If you specify more than one table on a DISCARD clause, the output file will contain a 2-byte OBID value at the beginning of each row. If you specify only one table on a DISCARD clause, this 2-byte OBID will be absent.

**UNLOAD**

*(default)* Uses the DB2 REORG UNLOAD format. You can abbreviate this as ULD.

SYSPUNCH output will not be written if FORMAT UNLOAD is specified.

**EXTERNAL**

Uses EXTERNAL format. You can abbreviate this as EXT.

**REPORT**

Uses REPORT format. You can abbreviate this as REP.

**ASCIIDEL**

Uses ASCIIDEL format (ASCII, with values delimited by double quotes). You can abbreviate this as ASC.

**PAD**

*(default)* Pads VARCHAR columns to their full length. You cannot use PAD with the EXTERNAL, REPORT, ASCIIDEL formats.

**NOPAD**

The NOPAD option will cause the output file *not* to pad VARCHAR columns to their full length - that is, to the actual length of the data only, *minus any trailing blanks*. NOPAD is not valid for the following formats: EXTERNAL, REPORT, ASCIIDEL.
DISCARD FROM TABLE

DISCARD FROM TABLE specifies that rows are to be deleted as part of the reorganization process, based on a condition or set of conditions.

*DISCARD FROM TABLE must appear last on the REORG statement.* The following considerations apply to the DISCARD option:

- You cannot use this option when any of the following conditions exists:
  - You are reorganizing objects based on an object set.
  - You are rebalancing partitions.
  - The table space contains an archive table.
  - You are reorganizing system-maintained temporal table spaces when versioning is active.
    - On such table spaces, deleted rows go to the history table.
  - NGT Reorg invokes DSNUTILB when any of the following conditions exists:
    - You are reorganizing an XML table space or its base table space.
    - Pending limit key changes exist on the table space.

- You cannot use wildcards with the DISCARD option.

- If the table space contains multiple tables, you can repeat the FROM TABLE keywords and table name, as in the following example:

```
REORG TABLESPACE A.B DISCARDDN DELETES DISCARD FROM TABLE A.TABLE1 WHERE (COL01 = 'M') FROM TABLE A.TABLE2 WHERE (CUSTNO IN (SELECT CUSTOMER FROM A.TABLE3 WHERE INVOICE_DATE < CURRENT DATE - 30 DAYS))
```

- If you specify DISCARD and you are using NGT Utility Manager, NGT Reorg discards rows only from the partitions chosen by RTS for reorganization.

`creatorName.tableName`

Names the table whose rows are to be deleted.

WHERE

This keyword (or its equivalent, WHEN) begins the specification of an SQL WHERE clause. The following requirements and restrictions apply to this keyword:

- You must enclose the WHERE clause in parentheses.
- You cannot use wildcards.

For more information about NGT WHERE clause options, see the topic about WHERE clause support in the *BMC Next Generation Technology General Reference Manual*. 

---

**DISCARD FROM TABLE syntax**

**60**  
*BMC Next Generation Technology Reorg for DB2 for z/OS Reference Manual*
User Guide. The most efficient WHERE clauses are the ones that NGT utilities can process natively without sending the SQL to DB2.

NGT WHERE also supports WHERE columnName IN FILE(ddname), which can be very efficient and useful when discarding the same keyed rows from multiple tables. The file specified in the WHERE clause is allocated to the Master job so this job must run with +SVRMODE(INTERNAL).

If a SYSPUNCH DD statement exists, the LOAD statement for the discarded rows will be written to the data set. In addition, you must specify the FORMAT keyword. However, if FORMAT UNLOAD is in effect, NGT Reorg does not write generated LOAD statements to SYSPUNCH.

**REORG INDEX syntax**

This topic describes the NGT Reorg REORG INDEX command syntax.

**REORG INDEX diagrams**

This topic contains syntax diagrams for the NGT Reorg REORG INDEX command.

For option descriptions, see “REORG INDEX options” on page 62.
**REORG INDEX options**

This topic describes the options available to you when reorganizing an index.

**REORG INDEX[ES] or INDEXSPACE**

Use this option to specify which index or group of indexes you want to reorganize.

**INDEX creator.indexName**

Specifies the index to reorganize.

---

1 Used only when invoking DSNUTILB
As an alternative to specifying a particular index, you can specify an object set. For more information, see “OBJECTSET objectSetName” on page 63.

INDEXES creatorPattern.indexPattern

Specifies a set of indexes to reorganize. Use the INDEXES keyword along with a naming pattern that contains wildcard characters. For details about the use of wildcards, see the BMC Next Generation Technology General User Guide.

As an alternative to specifying a set of indexes by using a pattern, you can specify an object set. For more information, see “OBJECTSET objectSetName” on page 63.

INDEXSPACE

creator.indexSpace

Specifies the index space to reorganize.

creatorPattern.indexSpacePattern

Specifies a set of index spaces to reorganize. Use the INDEXSPACE keyword along with a naming pattern that contains wildcard characters. For details about the use of wildcards, see the BMC Next Generation Technology General User Guide.

As an alternative to specifying a particular index space or specifying a set of index spaces by using a pattern, you can specify an object set. For more information, see “OBJECTSET objectSetName” on page 63.

OBJECTSET objectSetName

Specify the name of an existing object set to have NGT Reorg reorganize all indexes defined in the object set.

You can define object sets in the BMC DASD MANAGER PLUS for DB2 product. For more information about object sets, see the DASD MANAGER PLUS for DB2 User Guide.

You cannot use the following options when using an object set:

- PART[S] or LPART[S]
- EXCLUDE INDEX

PART[S] and LPART[S]

Use these options to specify which index partitions you want to reorganize.
Use the PART or PARTS option to specify one or more physical partitions to reorganize. Use the LPART or LPARTS option to specify one or more logical partitions to reorganize.

You cannot use this option when you are reorganizing indexes based on an object set.

### Specifying partitions

The following information applies to all four options:

- You can specify partitions by number using integer values from 1 through 4096.
- You can specify a range of contiguous partitions as `partitionNumber:partitionNumber`.
- When you specify a range of partitions, the second number must be greater than the first.
- You can specify a mixture of individual partitions and ranges of partitions.

---

**Note**

LPARTS does not support wildcards.

---

**REPORTONLY**

The REPORTONLY option generates a summary report only. A reorganization of the object does not take place.

**EXCLUDE INDEX**

When you specify a group of indexes to reorganize, you can refine that list of indexes by using the EXCLUDE INDEX option, as illustrated in the following examples:

```sql
REORG INDEXES TABLESPACE SOMEDB.% EXCLUDE TABLESPACE SOMEDB.ACCTS
REORG INDEXES TABLESPACE SOMEDB.% EXCLUDE INDEX SOMEDB.IXACCTS
```

You cannot use this option when you are reorganizing indexes based on an object set.

**WORKDDN**

*(Used only when invoking DSNUTILB)* WORKDDN specifies the DD statement for the unload data set.
NGT Reorg passes this option to the IBM DSNUTILB utility. For more information about this option, see the IBM DB2 Utility Guide and Reference.

**reorg options**

Many of the options described in the REORG TABLESPACE syntax topic also apply to REORG INDEX.

For information about which options apply to REORG INDEX and a reference to a description of each topic, see “Alphabetical listing of REORG options” on page 37.
Parameters for NGT Reorg

This chapter describes parameters that apply when using NGT Reorg. You can specify these parameters in the RRGPARMS DD or RRGPARMX DD statement in your NGT job stream. NGT Reorg uses default values from the system configuration unless you override them in your job stream.

All of the parameters that this chapter describes apply when reorganizing a table space. In addition, the following parameters apply when reorganizing an index:

- +ALLO
- +FREEPAGE
- +PCTFREE
- +VOLUMES

Syntax rules for parameters

The following rules apply to the syntax for parameters for the Next Generation Technology (NGT) products:

- All parameters must start with a plus symbol (+).
- Parameters can start anywhere, but must be contained within the first 72 character positions of the line. Columns 73-80 are ignored and can contain sequence numbers.
- You can code parameters over multiple lines. Continuation is automatic.
- You can code multiple parameters on the same line.
- You can code comments by enclosing the comment between /* (beginning) and */ (end). You can place comments anywhere, including in the middle of a word. The utility removes all comment strings before checking syntax.
**Note**

If you specify parameters by using a DD statement in your JCL, do not begin a comment with /* in column one. Placing /* in column one causes IBM MVS to generate a //SYSIN statement for any subsequent records in the data set unless you use DLM= to change the JCL comment delimiter.

- You can nest comments, with no limit to the number of nested levels that you can use. Ensure that each /* has a corresponding */.

- When specifying a series of values, you must separate them with commas. Blanks, wherever they are coded, have no significance. The utility removes all blanks before checking syntax.

- If the same parameter is specified multiple times within the same input data set, the latest specification prevails.

- Do not use the plus symbol inside parentheses for any parameter.

**+ALLO**

The +ALLO parameter is a data set PRIQTY value. It controls the value that NGT Reorg uses for a new table space primary allocation quantity. Use this parameter in RRGPARMS to control table space allocations and in RRGPARMX to control index data set allocations.

**USED**

Specifies that NGT Reorg should use the space amount from the existing data set as the primary quantity for the new data set. This value is obtained from the high-used RBA in the VSAM catalog. This is the default value.
ALLO

Specifies that NGT Reorg should use the currently allocated space amount from the existing data set as the primary quantity for the new data set. The value is obtained from the high-allocated RBA in the VSAM catalog.

DB2

Specifies that NGT Reorg should use primary quantity and secondary quantities from the DB2 catalog as the values for the new data set. If the object is not DB2-managed or if the PQTY value is less than the high-used RBA for the data set, NGT Reorg uses the high-allocated value.

KEEPDB2

Specifies that NGT Reorg should use values from the primary quantity (PQTY) DB2 catalog for the new data set. This is different from +ALLO (DB2) in that +ALLO(DB2) is a conditional switch, whereas the KEEPDB2 parameter forces NGT Reorg to use the DB2 values.

If the dataset is user-managed, NGT Reorg uses the values in use for primary quantity as it is defined to AMS for the new dataset.

If a value of -1 is set for any object in the catalog for either PRIQTY or SECQTY, NGT Reorg correctly keeps the DB2 primary and secondary allocations.

MXIG

This parameter has a similar meaning to the MVS JCL SPACE parameter. MXIG requests that space allocated to the data set be the smaller of the PQTY needed or the space available on the volume. This parameter affects only primary space allocation. Secondary extents should be available when you use this parameter.

REUSE

Specifies that NGT Reorg should write the newly loaded data set over the original data set. +ALLO (REUSE) can be honored only if the VSAM cluster has the REUSE attribute and the existing size is at least large enough for the data set to be loaded.

Example

In the following example, NGT Reorg issues the currently allocated space to determine the primary allocation for the new data set.

+ALLO(ALLO)
**+BEFOREACC**

The presence of +BEFOREACC(UT) tells NGT Reorg to check for a concurrently running IBM REORG utility. The only value that you can specify with this parameter is UT.

**Note**

During installation, the configuration process adds +BEFOREACC() to your NGT Reorg parameter data set. This specification does not check for a concurrently running IBM REORG utility. You must change the value to +BEFOREACC(UT) to have NGT Reorg perform this check.

NGT Reorg checks at the beginning of the reorganization and before the change apply or rename stage begins. If it finds that the object is in any UT status, NGT Reorg skips the object. NGT Reorg fails with a return code 4 for UTRW and UTUT statuses and with a return code 8 for UTRO status.

**+DBNAME**

The +DBNAME parameter provides the name of a database that NGT Reorg will use as a work database for internal processing.

Specify the name of an existing database.

**+FREEPAGE**

The +FREEPAGE parameter specifies the number of pages between empty pages in the table space.
The default value is specified at installation time. The value specified must be between 0 and 255.

**Note**
The DB2 catalog is *not* updated with the new value. This will have no effect on DB2 or the table space unless you run the IBM RECOVER, LOAD, or REORG utilities on the object. In that case, DB2 will revert to the original value.

-1

Continue to use the FREEPAGE value currently in use.

**integer**

An integer representing the new FREEPAGE value.

**Example**

In the following example, NGT Reorg inserts an empty page after every 35 pages of the table space.

```
+FREEPAGE(35)
```

**+IGNDEFNO**

The +IGNDEFNO parameter specifies whether to ignore indexes specified with DEFINE NO and to process the underlying table space. Use +IGNDEFNO when you have segmented multi-table table spaces where some tables are empty and have indexes that are defined as DEFINE NO or DEFER YES.
YES

Ignores deferred indexes on empty tables and processes the table space for the other tables in the table space.

NO

Does not process table spaces that have deferred indexes.

Example

In the following example, NGT Reorg processes table spaces that have deferred indexes on empty tables.

```
+IGNDEFNO(YES)
```

+INITWINDOW

The `+INITWINDOW` parameter specifies the maximum amount of memory that each server task can use. This allows NGT Reorg to limit memory usage.

```
+INITWINDOW ( integer )
```

NGT Reorg issues a message if the object being reorganized is too big to run.

The value specified must be between 64 and 4000. All values are expressed in megabytes.

`integer`

Specifies the size of the data space in megabytes.

`4000`

Indicates there is no limit on the amount of memory to use.

Example

In the following example, each server task is allowed to use 2G of memory.

```
+INITWINDOW(2000)
```
**+LOB**

The +LOB parameter specifies whether to include LOB auxiliary table spaces when processing a wildcard list of table spaces.

```
+LOB (NO YES)
```

**NO**

Excludes associated LOB auxiliary table spaces. NO is the default.

**YES**

Processes associated LOB auxiliary table spaces.

*Note*

The NGT LOBMaster AUX keyword overrides the +LOB parameter. If you specify +LOB(NO) and AUX, the LOB table space will be processed. For more information, see the *BMC Next Generation Technology LOBMaster for DB2 for z/OS Reference Manual*.

**Example**

In the following example, NGT Reorg processes all LOB table spaces when reorganizing the base table space.

```
+LOB(YES)
```

**+LOBPCT**

The +LOBPCT parameter specifies a space saving trigger for processing LOB auxiliary table spaces.

```
+LOBPCT (integer)
```

Specify +LOBPCT to set a trigger that tells NGT Reorg whether to reorganize the LOB table space based on the percentage of LOB space that the reorganization will
save. If the reorganization will not reclaim the specified percent, NGT Reorg excludes the object. Use this parameter to prevent unnecessary costly reorganizations.

You can override this parameter by using the LOBRECLAIMPERCENT option.

20

*(default)* Specifies that 20 percent of space must be saved in order for NGT Reorg to process LOB auxiliary table spaces.

*integer*

Specifies the percentage of space that must be saved in order for NGT Reorg to process LOB auxiliary table spaces. Set to 0 to reorganize the object, regardless of space savings.

**Example**

In the following example, NGT Reorg processes LOB table spaces when 30 percent of the space can be saved.

```
+LOBPCT(30)
```

**+MAXTASKS**

The `+MAXTASKS` parameter specifies the maximum number of concurrently executing tasks per server. The value specified must be between 1 and 15.

```
+MAXTASKS (10)
```

`+MAXTASKS` can have an "absolute" maximum of 15 tasks per server. You can specify a lower maximum number of server tasks. For example, if you specify `+SVRDISK(3)` and `+MAXTASKS(7)`, no more than 21 server tasks can run at once. If you specify `+MAXTASKS(15)`, the highest valid value possible, up to 45 tasks could run at once.

10

*(default)* Runs up to 10 tasks simultaneously, per server.
**integer**

An integer between 1 and 15 representing the maximum number of tasks per server.

**Example**

In the following example, parallel processing is restricted to five concurrent server tasks.

```plaintext
+MAXTASKS(5)
```

**+MINTSSIZE**

The +MINTSSIZE parameter specifies the minimum table space size, in pages, for which a reorganization performs. NGT Reorg bypasses table spaces smaller than the specified value.

```plaintext
+MINTSSIZE(integer)
```

**0**

Processes all table spaces.

**integer**

A numeric value from 1024 to 2147483648 (2G).

**Examples**

In the following example, NGT Reorg bypasses a reorganization of table spaces smaller than 65536 pages (or 64K pages) in size and considers all table spaces of at least 64K pages for processing.

```plaintext
+MINTSSIZE(65536)
```

In the following example, NGT Reorg bypasses a reorganization of table spaces smaller than 1G pages in size.

```plaintext
+MINTSSIZE(1073741824)
```
In the following example, NGT Reorg reorganizes all table spaces specified in SYSIN.

+MINTSSIZE(0)

**+PAGEUSE**

The +PAGEUSE parameter specifies the percentage of available memory NGT Reorg uses.

50

(default) Specifies that NGT Reorg can use up to 50 percent of available memory.

+PAGEUSE(0) defaults to +PAGEUSE(50).

integer

An integer from 1 to 99.

**Example**

In the following example, NGT Reorg uses 75 percent of available pages, leaving 25 percent blank.

+PAGEUSE(75)

**+PARTINDEPEND**

The +PARTINDEPEND parameter controls the order that partitions are processed when only partitioned indexes are being reorganized.
By default, when there are only partitioned indexes, NGT Reorg performs reorganization on each partition independently of the other partitions. This parameter lets you override this behavior and complete all the partitions together when the last partition completes. The values are defined as follows:

**YES**

*(default)* When only partitioned indexes are being reorganized, switch and post partitions as they complete.

**NO**

Switch and post all partitions together, when all partitions are completed.

---

**Note**

If you specify `+PARTINDEPEND(NO)`, no partition is reorganized if *any* partition fails.

---

**+PBGCHKDICT**

The `+PBGCHKDICT` parameter specifies whether NGT Reorg performs a PBG compression dictionary check.

**YES**

*(default)* Checks the compression dictionary for all partitions. YES tells NGT Reorg to make the compression dictionary the same for all partitions if it is not already.
NO

Skips the compression dictionary check and assumes that all partitions have the same compression dictionary. NO saves the time needed to check the compression dictionary, but will result in an abend during the reorganization if the partitions do not use the same dictionary.

Example

In the following example, NGT Reorg skips the dictionary check.

+PBGRCHDKICT(NO)

+PCTFREE

The +PCTFREE parameter specifies the percentage of free space in each data page of the table space.

The value specified must be between 0 and 99. The default value is specified at installation time.

-1

Continue to use the PCTFREE value currently in use.

integer

An integer representing the new PCTFREE value.

Note

The DB2 catalog is NOT updated with the new value. This will have no effect on DB2 or the table space unless you run the IBM RECOVER, LOAD, or REORG utilities on the object. In that case, DB2 reverts to the original value.

Example

In the following example, NGT Reorg uses the PCTFREE value from the DB2 catalog for each object.

+PCTFREE(-1)
**+RUNSTATS**

The +RUNSTATS parameter activates statistics gathering and updates the DB2 catalog.

### NO

NGT Reorg does not update the statistics.

### YES

NGT Reorg updates the statistics according to the STTPARMS. The STTPARMS DD is the location in which the parameters for are specified. For more information, see the *BMC Next Generation Technology Stats for DB2 for z/OS Reference Manual*.

### ALL

Same as YES. NGT Reorg updates all statistics, including the history tables.

**Restrictions**

The +RUNSTATS parameter is subject to the following restrictions:

- This parameter computes aggregate statistics using catalog values for partitions that are not processed. To ensure that meaningful values are computed when processing a subset of a partitioned table space, you must run NGT Stats on the entire table space when, or immediately after, the table space has loaded to collect statistics for all partitions.

- NGT Reorg skips non-indexed tables.

- The statistics that NGT Reorg gathers update SYSCOLUMNS for the first column only of each index. Run NGT Stats to obtain statistics for all columns of all indexes.

---

**Note**

For partitioned table spaces, NGT Reorg does not update table statistics unless there is an index on the partitioning columns or there is a DPSI defined.
Example

In the following example, NGT Reorg gathers statistics and updates the DB2 catalog with the new values.

```+RUNSTATS(YES)
```

Note

NGT utilities always gather KEYCARD statistics when any statistics are requested.

Additional considerations

Column grouping (COLGROUP) statistics are updated for every column combination of every index. Frequency distributed statistics can also be collected. Frequency statistics collection is controlled by the NGT Stats parameters in the STTPARMS DD. If the STTPARMS DD is not supplying these parameters in the reorganization job, NGT Reorg obtains them from the NGT configuration.

For a list of the DB2 catalog statistics that are updated, see the *BMC Next Generation Technology Stats for DB2 for z/OS Reference Manual*.

In the following example, NGT Reorg gathers column frequency distributed statistics.

```RRGPARMS DD *
+RUNSTATS(YES)
STTPARMS DD *
+FREQ(MOST)
+NUMCOLS(4)
+COUNT(2)
```

Note

If one of the indexes has only three columns, frequency statistics is gathered for that group of three columns.

+VOLUMES

The +VOLUMES parameter specifies a list of volumes to be used in the DEFINE statement for each new data set. You can abbreviate this parameter as +VOLS.
You can specify up to nine volume serials (volsers), separated by commas. (NGT Reorg ignores any volsers after the ninth one.) You can continue the volser list onto multiple lines.

You can specify a default list during installation. Specifying +VOLUMES at execution time overrides the default. If you do not specify a volume override list, the new data set is defined on the same list of volumes used for the original data set, which is obtained from a LISTCAT of the original data set.

+VOLUMES does not represent a list of volume serials. It represents a value that is substituted into a model AMS DEFINE statement.

You can modify this model DEFINE statement during installation, for example, to support the use of SMS. If the variable &VOLUMES is used in a DEFINE keyword other than VOLUMES(&VOLUMES), for example MANAGEMENTCLASS(&VOLUMES), the value specified on the +VOLUMES parameter is substituted for it.

`volserList`

A list of volume serials.

**Note**

You can use '*' for a single volume request and '*','*' for a multi-volume request. Use +VOL('*,','*') for three volumes.

To allocate volumes as they are needed, specify +VOLS(*) and specify VOLUMES="*" in XSUTXTND to add volumes when extend is necessary. XSUTALLO overrides this parameter.

**Example**

In the following example, +VOLUMES substitutes '*" for the &VOLUMES variable in the DEFINE statement.

```
+VOLUMES(*)
```

**Conditional volumes**

A conditional form of the +VOLUMES parameter specifies that a subset of the volumes are to be chosen from the list based on the space available on the volume just before allocation occurs.

When the first value specified in the +VOLUMES parameter is "/ n" (n being a number from 1 to 8), then n volumes are chosen from the list based on the space available on the volume. n is the number of volumes to include in the final list.
In the following example, +VOLUMES requests a check of the space available on the four volumes specified to find one volume that contains contiguous space for the PRIQTY value of the data set.

+VOLUMES(/1,AAAA,BBBB,CCCD,DDDD)

The volume with the largest contiguous free space will be used. Only the one volume will be used for the data set because "/1" was specified.

To include multiple volumes in the final allocation list, specify "/n", where n is more than 1. Additional volumes will be specified as candidate volumes starting at the beginning of the list, not including the one picked for the primary volume, until the count is reached.

In the following example, volume CCCC has the largest contiguous free space of any of the named five volumes.

+VOLUMES(/4,AAAA,BBBB,CCCD,DDDD,EEED)

In this case, the volumes list passes as follows:

(CCCC,AAAA,BBBB,DDDD)
Syntax examples for NGT Reorg

This chapter provides examples of specific tasks you can perform by using NGT Reorg.

Reorganizing a table space

To reorganize a table space, use the REORG TABLESPACE statement.

```
REORG TABLESPACE NWDB.ALLACCTS
```

This syntax works for simple, segmented, and partitioned table spaces.

Reorganizing specific partitions of a table space

To reorganize specific (physical) partitions of a partitioned table space, use the PART or PARTS option.

You can specify a single partition, range of partitions, or a mixture of both, as shown in the following examples. You can use the PART and PARTS options interchangeably.

```
REORG TABLESPACE NWDB.ALLACCTS PART(7)
```

When specifying a range of partitions, specify the starting and ending partition numbers, separated by a colon:

```
REORG TABLESPACE NWDB.ALLACCTS PARTS(2:8)
REORG TABLESPACE NWDB.ALLACCTS PART(2,5:9,12,15:22,25,30)
```
Reorganizing specific logical partitions of a table space

To reorganize specific logical partitions of a partitioned table space, use the LPART or LPARTS keyword.

You can specify a single partition, range of partitions, or a mixture of both, as shown in the following examples. You can use the LPART and LPARTS options interchangeably.

```
REORG TABLESPACE NWDB.ALLACCTS   LPART(7)
```

When specifying a range of partitions, specify the starting and ending partition numbers, separated by a colon:

```
REORG TABLESPACE NWDB.ALLACCTS   LPART(2:8)
REORG TABLESPACE NWDB.ALLACCTS   LPARTS(2,5:9,12,15:22,25,30)
```

Reorganizing a table space and generating an image copy

NGT Reorg automatically produces an image copy after each reorganization. You must use either the OUTPUT command or the automation control point XCPYDYNM to allocate image copy data sets. For more information about XCPYDYNM, see the BMC Next Generation Technology Automation Reference Manual. For more information about the OUTPUT command, see the BMC Next Generation Technology General User Guide.

Reorganizing a table space and sending discards to a data set

To reorganize a table space and send discards to a data set based on specific criteria, use the DISCARD FROM TABLE clause and include WHERE.

```
//OBSOLETE DD DISP=SHR,DSN=ADMIN.CENTRAL.DISCARDS
//SYSIN DD *
REORG TABLESPACE DB9900.CENTRAL
   DISCARDDN(OBSOLETE)
   DISCARD FROM TABLE DB9900.CENTRAL_INV_TBL WHERE (SKU > '592032')
```
Reorganizing a set of table spaces and excluding certain table spaces

You can reorganize a set of table spaces using wildcards and use EXCLUDE TABLESPACE to exclude certain table spaces.

```sql
REORG TABLESPACE ADMINDB.CAR%
  EXCLUDE TABLESPACE ADMINDB.CARMIDSZ
  EXCLUDE TABLESPACE ADMINDB.CARLUXRY
  EXCLUDE TABLESPACE ADMINDB.CARSUVS
```

Reorganizing an index

To reorganize an index, use the REORG INDEX statement.

```sql
REORG INDEX NWDB.XACCT912
```

This syntax works for indexes on simple, segmented, and partitioned table spaces.

Reorganizing specific partitions of an index

To reorganize specific partitions of a partitioned index, use the PART or PARTS keyword.

You can specify a single partition, range of partitions, or a mixture of both, as shown in the following examples. You can use the PART and PARTS options interchangeably.

```sql
REORG INDEX NWDB.ARIX1 PARTS 7
```

When specifying a range of partitions, specify the starting and ending partition numbers, separated by a colon:

```sql
REORG INDEX NWDB.ARIX1 PARTS 2:8
REORG INDEX NWDB.ARIX1 LPARTS(2,5:9,12,15:22,25,30)
```
Reorganizing a set of indexes and excluding certain indexes

You can reorganize a set of indexes using wildcards and use EXCLUDE INDEX to exclude certain indexes.

```
REORG INDEXES ADMINDB.CAR%
   EXCLUDE INDEX ADMINDB.XMIDSZ1
   EXCLUDE INDEX ADMINDB.XLUXRY2
   EXCLUDE INDEX ADMINDB.XSUVS8
```
Recommended automation control points for NGT Reorg

NGT automation control points, which are integrated into all NGT utilities, can enhance and expand processing options. This chapter describes recommended automation control points to use with NGT Reorg.

The presence of the NGTAUTO DD statement in your JCL triggers automation control point processing, as follows:

```
//NGTAUTO DD DISP=SHR,DSN=NGT.AUTO.CTRL.POINTS
```

The automation control point data set contains one member with each of the automation control points selected for use.

**Note**

If you do not include an NGTAUTO DD statement in your JCL, NGT Reorg uses the values specified in the +NGTAUTO1 or +NGTAUTO2 parameters (either specified in the configuration or overridden in the UTLPARMS DD).

BMC recommends that you use the following NGT automation control points with NGT Reorg. However, you can also use other automation control points. For more information about these and other automation control points, see the *NGT Automation Reference Manual*.

**XRRGTERM**

NGT Reorg invokes XRRGTERM after REORG processing finishes for each object. You can use XRRGTERM to save statistics to a file or in a DB2 table.

**XRRGRWRO**

NGT Reorg invokes XRRGRWRO just before performing the drain in preparation for the Change Apply phase. You can use this automation control point to use a specific time or event to trigger the Change Apply phase, or to set a delay between drain attempts during Change Apply.
XSUTGLOB

XSUTGLOB sets global variables that other automation points can then use.

XSUT0000

Alternatively, you can use XSUT0000 to set global variables.

XSUTSYIN

The utility calls XSUTSYIN before the parser reads SYSIN. XSUTSYIN has full access to the SYSIN and can modify it. You can use XSUTSYIN to override or disallow some input parameters.

XSUTTIME

You can use XSUTTIME to cancel a utility based on the time of day.

XSUTALLO

You can use XSUTALLO to set allocation values and volumes for DB2 object VSAM data sets.

XSUTDBMG

You can use XSUTDBMG to set volumes for initial allocation and extend processing of work data sets.

XSUTTERM

The utility calls XSUTTERM after processing finishes. You can use XSUTTERM to automatically insert run statistics into a DB2 table after processing, for example, or to automatically send an audit report.

XSVRXERR

XSVRXERR is called if processing ends with an error condition. You can use XSVRXERR to alert a user or to raise an error flag.
NGT Reorg reports

This chapter describes the reports and messages that the NGT Reorg product produces.

Reports to the SUMMARY DD

This topic describes the information that NGT Reorg sends to the SUMMARY DD.

Internal IDS report

The Internal IDS report correlates the DB2 internal object identifiers to their full external names. You can use this to interpret product messages with internal identifiers. The following figure shows an example of this report:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FANSET</td>
<td>OBID</td>
</tr>
<tr>
<td>PGSET</td>
<td>ISOBID</td>
</tr>
</tbody>
</table>

Each line of this report represents part of the object being processed:

- The first line specifies the database.tableSpace, followed by the hexadecimal value for the DBID (DB), PSID (PS), OBID (FILE), and the number of partitions (PTNS).
- The second line specifies the table OBID and name.
- The third line in this case is a dummy partitioning index for internal use. It is used on table-partitioned table spaces.
- The remaining lines show the following index information:
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>First character: Dup or Unique, second character: Fixed or Variable.</td>
</tr>
<tr>
<td>KYLEN</td>
<td>Key length in bytes</td>
</tr>
<tr>
<td>KYCOLS</td>
<td>Number of table columns in the index</td>
</tr>
</tbody>
</table>

### Object space report

The Object space report documents the type and size of the objects being processed as shown in the following example:

<table>
<thead>
<tr>
<th>OBJECTS BEING REORGED</th>
<th>PHYS</th>
<th>LOGL</th>
<th>TYPE</th>
<th>CX?</th>
<th>I/J</th>
<th>PTNS</th>
<th>(# 4K PAGES)</th>
<th>ALLOCATED</th>
<th>USED SPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>SPCNAME</td>
<td>OBID</td>
<td>PSID</td>
<td>PART</td>
<td>PART</td>
<td>TYPE</td>
<td>CX?</td>
<td>I/J</td>
<td>PTNS</td>
</tr>
<tr>
<td>TESTDBAS</td>
<td>TESTTSPC</td>
<td>0001</td>
<td>0002</td>
<td>1</td>
<td>1</td>
<td>I</td>
<td>2</td>
<td>180</td>
<td>22</td>
</tr>
<tr>
<td>TESTDBAS</td>
<td>TESTTSPC</td>
<td>2</td>
<td>2</td>
<td>I</td>
<td>2</td>
<td>180</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TESTDBAS</td>
<td>TESTTSPC</td>
<td>TOTS</td>
<td>360</td>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TESTDBAS</td>
<td>TESTIXO1</td>
<td>0004</td>
<td>0005</td>
<td>1</td>
<td>1</td>
<td>PIX</td>
<td>CLX</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>TESTDBAS</td>
<td>TESTIXO1</td>
<td>2</td>
<td>2</td>
<td>I</td>
<td>2</td>
<td>180</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TESTDBAS</td>
<td>TESTIXO1</td>
<td>TOTS</td>
<td>360</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TESTDBAS</td>
<td>TESTIX02</td>
<td>0006</td>
<td>0007</td>
<td>NPI</td>
<td>I</td>
<td>AVGKL</td>
<td>13</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>TESTDBAS</td>
<td>TESTIX03</td>
<td>0008</td>
<td>0009</td>
<td>1</td>
<td>1</td>
<td>DPSI</td>
<td>AVGKL</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>TESTDBAS</td>
<td>TESTIX03</td>
<td>TOTS</td>
<td>360</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS</td>
<td>1,260</td>
<td>74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Data set summary report

The data set summary report documents the size, by partition of the table space being reorganized, as shown in the following example:

<table>
<thead>
<tr>
<th>PARTITION DATASET SUMMARY</th>
<th>CONT PAGE</th>
<th>EMPTY PAGE</th>
<th>ROWS</th>
<th>PTR ROWS</th>
<th>OFLOW ROWS</th>
<th>DISCARDS</th>
<th>VERSIONED ROWS</th>
<th>NULL CLIX ROWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PART</td>
<td>PAGES</td>
<td>CNTL PAGES</td>
<td>EMPTY PAGES</td>
<td>ROWS</td>
<td>PTR ROWS</td>
<td>PAGES</td>
<td>OFLOW ROWS</td>
<td>DISCARDS</td>
</tr>
<tr>
<td>1</td>
<td>65,509</td>
<td>5</td>
<td>28</td>
<td>65,472</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>65,509</td>
<td>5</td>
<td>28</td>
<td>65,472</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>65,509</td>
<td>5</td>
<td>28</td>
<td>65,472</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>65,509</td>
<td>5</td>
<td>28</td>
<td>65,472</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>65,509</td>
<td>5</td>
<td>28</td>
<td>65,472</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>65,509</td>
<td>5</td>
<td>28</td>
<td>65,472</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Before-After report (BF AF)

The Before-After report documents rows, pages, and index metrics prior to and after the reorganization, as shown in the following example:

<table>
<thead>
<tr>
<th>BF AF</th>
<th>SPNAME</th>
<th>TYP</th>
<th>PART</th>
<th>ROWS</th>
<th>KEYS</th>
<th>DELETED ROWS</th>
<th>PTR ROWS</th>
<th>PAGES</th>
<th>TS-DATA</th>
<th>IX-LEAF</th>
<th>IX</th>
<th>IX</th>
<th>IX</th>
<th>NEAR OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>TESTTSPC</td>
<td>PTS</td>
<td>1</td>
<td>500</td>
<td>500</td>
<td>0</td>
<td>22</td>
<td>22</td>
<td>20</td>
<td>20</td>
<td>95.8</td>
<td>22</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>TESTIX01</td>
<td>PIX</td>
<td>1</td>
<td>500</td>
<td>500</td>
<td>AVGKL= 13</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>TESTIX03</td>
<td>DPS</td>
<td>1</td>
<td>500</td>
<td>9</td>
<td>AVGKL= 10</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

90  BMC Next Generation Technology Reorg for DB2 for z/OS Reference Manual
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>TESTTSPC PTS</td>
<td>2</td>
<td>500</td>
<td>0</td>
<td>22</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>500</td>
<td></td>
<td>22</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>TESTIX01 PIX</td>
<td>2</td>
<td>500</td>
<td>500</td>
<td>7</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>500</td>
<td>500</td>
<td></td>
<td>7</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>TESTIX03 DPS</td>
<td>2</td>
<td>500</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>500</td>
<td>1</td>
<td></td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

----------------------------------------------------------------------------------------------------------------------

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>TESTIX02 NPI</td>
<td>2</td>
<td>1,000</td>
<td>22</td>
<td>6</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>1,000</td>
<td>22</td>
<td></td>
<td>6</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

----------------------------------------------------------------------------------------------------------------------

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>** TOTALS **</td>
<td>2</td>
<td>1,000</td>
<td>1,010</td>
<td>0</td>
<td>74</td>
<td>50</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>1,000</td>
<td>1,010</td>
<td></td>
<td>74</td>
<td>50</td>
<td><strong>XRIDS =</strong></td>
</tr>
</tbody>
</table>
Index

+BEFOREACC(UT) 70
+DBNAME 70

A
ALLO 68
AMS 68
ASCIIDEL 58
authorizations 17
AUTOESTSPACE option 50
automation exit processing 87
AUX option 48

B
basic format 48
basic row format 48
BRF 48

C
CLONE option 50
command options
   rules for syntax 36
conventions
   rules for specifying command options 36
conventions, documentation 10
COPYDDN option 53
CURRENT_DATE option 57
CURRENT_TIMESTAMP option 57

D
DB2 68
DD statements 33
DSDLIN 21, 33
DDLIN option 49
DEADLINE option 52
DELAY option 53
delimited tokens 36
differences between REORG PLUS and NGT Reorg 31
DISCARD option 60
DISCARDDN option 58
documentation information 9
DRAIN option 52
DRAIN_ALLPARTS option 57
DRAIN_WAIT option 56
DSNUTILB 58
DSNUM ALL option 47
DSNUTILB, invoking 30

E
EXCLUDE INDEX option 64
EXCLUDE TABLESPACE option 46
extended format 48
EXTERNAL 58

F
FLASHCOPY YES option 54
FORCE option 51
FORMAT 58

H
high-used RBA 68

I
INDEX option 62
Index Term 76
INDEXES option 63
INDEXSPACE option 63

K
KEEPDB2 68

L
LOB data, reorganizing 48
LOBRECLAIMPERCENT option 47
logical partitions 46
LOGRANGES option 57
long object names 36
LONGLOG option 53
LPART 84
LPART option 46, 63
LPARTS 84
LPARTS option 46, 63

M
MAPPINGDATABASE option 52
MAPPINGTABLE option 52
MASSDELETE option 55
MAXDISORG keyword 47
MAXRO option 52
MVS JCL SPACE 68
MXIG 68

N
names of objects, long 36
NGTAUTO 33, 87
NOCLIX option 47
NOPAD 58

O
object names, long 36
object sets 45, 62
OUTPUT command 35

P
PAD 58
PART 83, 85
PART option 46, 63
partition rebalancing 21
partitions, specifying 46
PARTS 83, 85
PARTS option 46, 63
PARTSDDL 33
PQTY 68
PREFORMAT option 47
PSRBD status 17, 25
publications, related 9
PUNCHDDN option 51

R
RBALRSN_CONVERSION option 48
RBDP status 17, 25
REBALANCE option 49
rebalancing partitions 21
rebuild pending 17
REBUILD pending 25
REBUILDICT option 47
REBUILDIX option 17
RECOVERYDDN option 54
RECP status 17
related publications 9
reordered row format 48
REORG INDEX 62
REORG INDEX statement 85
REORG INDEXES 62
REORG INDEXESPACE 62
REORG PLUS 31
REORG TABLESPACE 40
REORG TABLESPACE statement 83, 84
REPBLDAL 33
REPORT 58
REPORTONLY option 47, 64
restrictions 18
RETRY option 56
RETRY_DELAY option 56
REUSE 68
ROWFORMAT option 48
RRF 48
RRGPARMS 33
RRGPARMS DD 67
RRGPARMX 33
rules for command syntax 36
RUNSTATS 33
S

SCOPE option 50
SORTDEVT option 51
SORTNPSI option 51
SORTNUM option 51
specific partitions 83, 85
SWITCHTIME option 57
syntax
  rules for specifying 36
SYSDISC 33
SYSIN 33
SYSIN input 33
SYSPUNCH 33

T

TIMEOUT option 56
tokens in syntax, rules for 36

U

UNLOAD 58

V

VSAM catalog 68
VSAM cluster 68

W

WHEN 60
WHERE 60
work database 70
WORKDDN option 64

X

XML data 29
XSUT0000 87
XSUTGLOB 87
XSUTSYIN 87
XSUTTERM 87
XSUTXERR 87
*481572*