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

Supporting

Version 11.2 of BMC Next Generation Technology LOBMaster for DB2 for z/OS
Version 11.2 of BMC Large Object Management for DB2 solution

May 2015
Contacting BMC Software

Several methods are available for contacting BMC Software.

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

United States and Canada

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

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

DB2, FlashCopy, IBM, MVS, RACF, 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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

Support website
You can obtain technical support from BMC 24 hours a day, 7 days a week at [http://www.bmc.com/support](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.
<table>
<thead>
<tr>
<th>Chapter 1</th>
<th>LOB Overview ............................................................................................................................. 1-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Basic LOB Components.................................................................................................................. 1-1</td>
</tr>
<tr>
<td>1.2</td>
<td>NGT Utilities for LOBs / LOBMaster ............................................................................................. 1-2</td>
</tr>
<tr>
<td>1.2.1</td>
<td>REORG........................................................................................................................................ 1-2</td>
</tr>
<tr>
<td>1.2.2</td>
<td>UNLOAD..................................................................................................................................... 1-4</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Summary ..................................................................................................................................... 2-1</td>
</tr>
<tr>
<td>2.1</td>
<td>Features .................................................................................................................................. 2-2</td>
</tr>
<tr>
<td>2.2</td>
<td>Usage Notes ............................................................................................................................ 2-3</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Input: DD Statements, Syntax, Keywords ................................................................................ 3-1</td>
</tr>
<tr>
<td>3.1</td>
<td>DD Statements.......................................................................................................................... 3-1</td>
</tr>
<tr>
<td>3.2</td>
<td>SYSIN Syntax Diagrams.............................................................................................................. 3-2</td>
</tr>
<tr>
<td>3.2.1</td>
<td>REORG TABLESPACE Syntax........................................................................................................ 3-2</td>
</tr>
<tr>
<td>3.2.2</td>
<td>REORG TABLESPACE Keywords.................................................................................................. 3-2</td>
</tr>
<tr>
<td>3.2.3</td>
<td>UNLD TABLE Syntax.................................................................................................................. 3-4</td>
</tr>
<tr>
<td>3.2.4</td>
<td>UNLD Keywords....................................................................................................................... 3-4</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Parameters .............................................................................................................................. 4-1</td>
</tr>
<tr>
<td>4.1</td>
<td>Parameter Overview .................................................................................................................. 4-1</td>
</tr>
<tr>
<td>4.2</td>
<td>LOBMaster specific Reorg parms (RRGPARMS)........................................................................... 4-2</td>
</tr>
<tr>
<td>4.2.1</td>
<td>+LOBERRORS(action,message)..................................................................................................... 4-2</td>
</tr>
<tr>
<td>4.2.2</td>
<td>+LOBMISSING(action,message)................................................................................................... 4-3</td>
</tr>
<tr>
<td>4.2.3</td>
<td>+LOBORPHANS(action,message)................................................................................................. 4-4</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Examples ................................................................................................................................. 5-1</td>
</tr>
<tr>
<td>5.1</td>
<td>Reorganize a table space with LOB column(s). .......................................................................... 5-1</td>
</tr>
<tr>
<td>5.2</td>
<td>Reorganize a table space and clean up orphan LOBs .............................................................. 5-1</td>
</tr>
<tr>
<td>5.3</td>
<td>Unload a LOB table space ........................................................................................................ 5-1</td>
</tr>
<tr>
<td>5.4</td>
<td>Unload a LOB table space to static unload file. ....................................................................... 5-2</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>LOBMaster Specific Reports .............................................................................................. 6-1</td>
</tr>
<tr>
<td>6.1</td>
<td>Reports to the LOBREPRT DD..................................................................................................... 6-1</td>
</tr>
<tr>
<td>6.1.1</td>
<td>LOB Report.............................................................................................................................. 6-1</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Inline Report ........................................................................................................................... 6-1</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Distribution of LOB Sizes......................................................................................................... 6-1</td>
</tr>
<tr>
<td>6.2</td>
<td>Reports to the SUMMARY DD.................................................................................................. 6-2</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Internal IDS Report.................................................................................................................. 6-2</td>
</tr>
</tbody>
</table>
# Table Of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3</td>
<td>Messages to the SYSERROR DD</td>
<td>6-3</td>
</tr>
<tr>
<td>6.3.1</td>
<td>+LOBERRORS Messages</td>
<td>6-3</td>
</tr>
<tr>
<td>6.3.2</td>
<td>+LOBMISSING Messages</td>
<td>6-3</td>
</tr>
<tr>
<td>6.3.3</td>
<td>+LOBORPHANS Messages</td>
<td>6-3</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Technical Details</td>
<td>7-1</td>
</tr>
<tr>
<td>7.1</td>
<td>Reasons for LOBMaster</td>
<td>7-1</td>
</tr>
<tr>
<td>7.2</td>
<td>When is LOBMaster required</td>
<td>7-2</td>
</tr>
</tbody>
</table>

Index .......................................................................................................................................... X-1
LOB Overview

Chapter 1

LOB Overview

LOBs, of whatever kind, Character (CLOB), Binary (BLOB), or Graphic (DBLOB), are an essential tool in Data Base design. With their growing popularity come some issues. The first is simply size; they can be massive. For each LOB column, each partition of a table space can have an associated LOB able space of up to 128T and there can be 4096 Base table space parts!!

The size and complexity of the LOB structure lends itself perfectly to the advanced techniques and algorithms of BMC Next Generation Technology Utilities. Users of NGT products are accustomed to significantly faster utilities that simplify the DB2 maintenance tasks. With the peculiar LOB structure and the need to run multiple DB2 utilities just to perform a validity check, there was much room for improvement. These improvements are delivered with BMC Next Generation Technology (NGT) LOBMaster.

1.1 Basic LOB Components

ID
Each separate LOB (of whatever type) has an ID. This is a 19-byte value that is comprised of a ROWID, unique to a table space row, and a version number identifying the last update of that LOB.

LOB IX
This is a unique index on the ID field that indexes the LOB table space, not the Base table.

BASE TS/TABLE
LOB columns. In a row of the Base Table with LOBS, there is always defined a ROWID column that has data. This is the customer table of rows that are defined with one or more LOBs unique to this row in the table.

Each LOB in a row can be NULL (no data) or contain a version number. The combination of the version number and the ROWID is the ID for this LOB in this row.

LOB Auxiliary TS
This is the DB2 dataset that contains the LOB data. It has a different format than any other kind of DB2 dataset. Each LOB table space contains he LOB data only for a specific column in the base table. If a table has three LOB columns defined, there will be three separate LOB table spaces.

LOB
A single LOB is any collection of data built by a user program and inserted into a DB2 row in a LOB-defined column. It can be any size from 1 byte to 2G.
So, there are three objects:

1. **BASE TABLE**
   ID per LOB per row

2. **LOB IX**
   Per defined LOB column
   Points to the LOB TS

3. **LOB TS**
   Repository of the LOB data for a single Base Table LOB column and partition.

### 1.2 NGT Utilities for LOBs / LOBMaster

#### 1.2.1 REORG

Run BMC Next Generation Technology (NGT) Reorg on the Base table space with the AUX keyword to cause ALL the LOB checking to be performed, errors corrected (where possible), and rows or LOBs in error to be unloaded to a sequential file and discarded from the TS.

- Simplifies the maintenance of LOBs, only Reorg needs run.
- All this runs in RW mode.
- FLASHCOPY is not required as it is for the IBM utilities.
- NO SORT Utility is invoked.
- Non-correctible errors will cause the Reorg to fail before any renames are done.

NGT Reorg classifies LOB errors into one of three categories:

1. LOB Errors
2. LOB Missing
3. LOB Orphans

Each of these categories has an associated parameter to control processing:

1. +LOBERRORS
2. +LOBMISSING
3. +LOBORPHANS

Each parm has two keywords, the first controls processing, the second controls error message printing. See Chapter 4 in this manual for parameter details.

Processing control:

1. ABEND Terminate processing when this situation occurs.
2. ERASE Simply eliminate the problem, if possible.
3. IGNORE Ignore the error, leave the data as it was, if possible.

Message control:

1. MESSAGE (or MSG or M) - Print up to 50 error messages.
2. NOMESSAGE (or NOMSG or NOM) – Do not print error messages.
LOB Overview

LOB ERRORS

There are many types of LOB Errors
1. The LOB is not complete.
2. The list of pages supposedly containing the LOB data is inconsistent.
3. One or more of the pages containing the LOB data is referenced by multiple LOBs.
4. A LOB data page is empty.
5. A LOB map page (the first page of a LOB and the page referenced by the LOB index) is not marked as a map page or is inconsistent.

All these errors are detected when LOBMaster scans the LOB TS. Action in these cases is controlled by the +LOBERRORS Reorg parameter. Detection can result in an ABEND, deletion of the LOB in error, or ignoring the error.

Understand that such a condition would most likely cause DB2 to abend during the read of the LOB, e.g. during a SQL fetch of the data DB2 would fail with an 04E abend.

✔️ NOTE:
LOB data is maintained in a DB2 LOB TS in a hierarchical structure. The “top” of the structure is the first map page, whose page number is in the LOB IX. The first map page contains a list of pages, some can be other map pages and some can be data pages. It also contains the total size of the LOB data. The map pages pointed to by the first map page can also point to other map pages and other data pages.

The inherent flaw in IBM’s design is that the data pages do not contain any information that connects that data page to the LOB to which it belongs. It is possible that the same data page would be involved in more than one LOB’s hierarchical structure. This is an error, but such errors can occur.

With LOBMaster the LOB TS scan checks that each data page is referenced by one and only one different LOB, that all data pages are accounted for, that the accumulated size of all the data pages for each LOB equals the total size of the LOB as specified in the first map page.

LOB MISSING

There are four different types of “missing” LOBs.
1. LOB is in the LOB IX, but does not exist in the LOB TS.
2. LOB is not in the LOB IX, but is in the LOB TS.
3. LOB is referenced by a row in the Base TS, but does not exist in the LOB TS.
4. LOB is referenced by a row in the Base TS and is in the LOB TS but does not exist in the LOB IX.

Types 1, 2 and 3 are detected after the LOB IX and LOB TS have completed their reads. Type 4 is detected during the Base TS read.

Types 2 and 4 can be corrected by “fixing” the LOB IX. +LOBMISSING(IGNORE) will cause this to happen.

Types 1 and 3 will most likely cause a DB2 SQL fetch to fail with an abend condition.

+LOBMISSING(IGNORE) will not affect types 1 and 3.
LOB Overview

+LOBMISSING(ABEND) will cause the Reorg to fail.

+LOBMISSING(ERASE) in type 1 will remove the LOB IX reference. ERASE for type 3 is not currently implemented. The Base TS row will be updated to mark the LOB column NULL.

☑️ **NOTE:**
IGNORE on certain types of LOB errors can result in unpredictable results later in the reorg, e.g. during LOB write. It is strongly recommended that +LOBMISSING(ABEND, MSG) be used.

**LOB ORPHANS**

A LOB exists and is complete on the LOB TS, but is not referenced by any Base TS row.

An orphan may or may not have a LOB IX entry, it is still categorized as an orphan. There will first be a message indicating this is a “missing” LOB, since there is not LOB IX entry for the LOB TS entry.

Orphans are detected at the end of the Base TS read phase.

+LOBORPHANS(ERASE) will remove the orphans from the LOB IX and the LOB TS.

+LOBORPHANS(IGNORE) will leave the unreferenced LOB in the LOB TS, but it cannot be accessed.

1.2.2 **UNLOAD**

Run BMC Next Generation Technology (NGT) Unload on the Base table space with the SPANNED YES keyword to cause the LOB data to be unloaded along with the base table data. These unload record lengths can be quite large; they will be unloaded to Variable Blocked Spanned (VBS) format. This format is compatible with DB2 Load starting with DB2 10.
**Chapter 2**

**Summary**

**NGT LOBMaster**

LOBMaster is a utility for managing large objects (LOBs).

<table>
<thead>
<tr>
<th>When To Use It</th>
<th>When you need to reorganize table spaces that have LOB columns.</th>
</tr>
</thead>
</table>
| Authorizations Required | **Database:**  
REORG authority  
**User:**  
SYSADM authority  
DBADM authority |
| Restrictions | **UTS PBG table space with LOB columns.**  
NGT LOBMaster currently does not process PBG table spaces with LOB columns. They are detected and skipped.  
LOBMaster currently does not move data to the inline portion of the LOB.  
**Table space altered to change LIMITKEYS**  
NGT LOBMaster currently does not reorg to materialize altered limitkeys. |
| Statuses | **DB2 States**  
Table space: RW, Aux Check Pending, or Aux Warning states. |
2.1 Features

EASE OF USE

DB2 requires expert use of four different Utilities to process a LOB Reorg: CHECK INDEX, CHECK LOB, CHECK DATA and REPAIR. LOBMaster does all this by simply adding the AUX keyword to the Reorg statement for the Base table space. It is not necessary to identify which table spaces have LOBs defined because the AUX keyword is ignored for table spaces without LOBs. LOBMaster will discard or unload all the LOBs of whatever size for a Base table space row, in a format suitable for LOAD while all unload files are dynamically allocated.

SPEED

LOBMaster was designed specifically for the challenges of Big Data. One of the biggest challenges of Big Data is the necessity to be able to process this data at an incredible speed. Given the huge size of some LOBs, speed becomes a serious issue. For a 200G LOB, not particularly massive in relation to the potential maximum size, a sequential read and write could take three to four hours, ignoring any processing on the data.

LOBMaster employs many sophisticated techniques to enhance the speed of LOB Reorg:

1. LOB table spaces are read by parallel I/O tasks. Up to 8 different tasks are used, the actual number dynamically computed based on the size of the LOB.

2. Reads are done with sequential I/O, not random. Sequential I/O is significantly faster. Using 8 parallel read tasks, net I/O speeds can exceed 600 cylinders per second., or about 5 seconds per Gigabyte. A 200G LOB would take significantly less than twenty minutes.

3. All analysis of the integrity of the LOB data is done without invoking an external SORT and without the concomitant I/O. There are no calls to SORT or any part of LOBMaster's processing.

4. Writes of the output LOB table space are done in parallel subtasks, usually a separate subtask per output dataset (not table space, dataset). Therefore up to 254 tasks can be scheduled, though the number running in parallel at any one time is limited to 8.

5. All partitions run independently and use the unique NGT Master-Server technology. Each partition can be run simultaneously on different CPUs in different jobs, even on different members of a Sysplex.
**Summary**

**REORG.** NGT LOBMaster.

Has options to automatically handle Orphaned LOBs eliminating the need precede Reorg with:
- Check LOB
- Check Index
- Check Data
- Repair

Not restricted to Universal Table Spaces (UTS). Processes classic partitioned and single table segmented table spaces with LOB columns concurrent with their auxiliary table spaces.

With the Reorg REPORTONLY keyword, LOBMaster will produce DASD savings and LOB size reports without reorganizing. See Usage Notes below for more details.

Unless Reorg is run with the REPORTONLY keyword, LOBMaster will correct any issues encountered with the LOB IX. Corrections to the Base TS and/or LOB TS are controlled by the parameters described in Chapter 4.

Discards on the base table include archiving the LOB data to an UNLD file in spanned records which are suitable for LOAD.

**UNLOAD.** CDB/Auto-Online Unload.

With a LOBMaster license, NGT Unload can unload tables with LOB columns to spanned records. The unload dataset will have the regular table space data and the LOB column auxiliary table space data together in the spanned record.

### 2.2 Usage Notes

**Reorg LOB Report.** With a LOBMaster license, NGT Reorg produces LOB reports in the LOBREPRT DD. These reports describe the distribution of LOBs by size and report how many pages will be saved by the Reorg. By specifying REPORTONLY on the Reorg statement, you can obtain this report by only reading the table spaces, not reorganizing them.

**Reorg LOB Discard.** When Reorging and discarding from a table space with LOBs, the optional discard datasets and Syspunch datasets must be dynamically allocated. There will be a separate dataset for each partition for increased parallel processing and these datasets will only be allocated for partitions with discarded rows.

**Unload Data.** With a LOBMaster license, NGT Unload will unload the LOB data along with the base table data when SPANNED YES is specified. The unload file will be Variable Blocked Spanned (VBS) which is compatible with DB2 Load starting with DB2 10.
Input: DD Statements, Syntax, Keywords

This chapter covers the DD statements, SYSIN input (statement syntax) keywords and parameters specific to LOBMaster.

3.1 DD Statements

LOBREPRT
To get the LOB reports, specify a LOBREPRT DD statement. This statement only needs SYSOUT=* to get the reports returned to the job. An example of this DD statement is shown here:

//LOBREPRT DD SYSOUT=*  

For more information and sample reports, see the Chapter 6 in this manual.
3.2 SYSIN Syntax Diagrams

3.2.1 REORG TABLESPACE Syntax

```
REORG TABLESPACE dbname.spacename
  [$dbmask.spacemask]
  [AUX]
  [reorg options]
``` 

3.2.2 REORG TABLESPACE Keywords

**AUX**
Default YES. With AUX or AUX(YES), the LOB Auxiliary table space will be processed along with the Base TS. Without AUX or with AUX(NO) only the base TS will be processed.

**NOTE:**
If there is a pending DDL change to drop a LOB column and AUX(YES) is specified, AUX will automatically be changed to (NO). The AUX(YES) reorg should be run after the dropping of the LOB column is materialized.

**reorg options**
See the *BMC Next Generation Technology Reorg for DB2 for z/OS Reference Manual* for the standard Reorg keywords. Below are some noteworthy keywords when processing objects with LOBs:

**REPORTONLY**
Analyze and write LOB reports to LOBREPRRT without reorganizing.

The DISCARD keyword clause functions somewhat differently when AUX is specified on a table space with LOBs.

1. Any specified DISCARDDN will be ignored.
2. The discard files will always be dynamically allocated as VBS (variable blocked spanned) record format. For partitioned table spaces a separate discard file will be allocated for each partition that has discards. This spanned discard file is compatible with IBM Load starting with DB2 10.
3. The XULDDYNM automation routine must exist and must provide a dataset name for the discard file. This name must be unique for each partition.
4. The SYSPUNCH file (containing a generated Load statement) will be dynamically allocated. There will be a separate file for each partition with discards.
5. The UNLDDYNM automation routine must provide a unique dataset name for the SYSPUNCH file for each partition. Note: XULDDYNM has stem variable URDDS indicating whether the file being allocated is for DISCARD or SYSPUNCH.

6. The generated LOAD statement will specify the SPANNED keyword. This is specified with the DB2 Load utility starting with DBB2 10 for z/OS.

7. All LOB data for discarded base TS rows will be included in the spanned record on the unload file.

**NOTE:**
If DISCARD and REPORTONLY are both used on the Reorg statement, the discard file will be written even though no rows are discarded. This can be used to selectively unload data from a table with LOBs defined.
3.2.3 UNLD TABLE Syntax

UNLD DATA

unload options

SPANNED YES

3.2.4 UNLD Keywords

**SPANNED YES**

When present, the LOB Auxiliary table space will be processed along with the Base table space and the data will be unloaded to a variable blocked spanned (VBS) dataset.

Without SPANNED YES only the Base table data will be unloaded.

**Unload options**

See the *BMC Next Generation Technology Unload for DB2 for z/OS Reference Manual* for the standard Unload keywords.

When unloading data from a table with LOB columns defined AND any of the LOB columns are specified on the UNLD statement (or if all columns are selected) the SPANNED keyword MUST be used if the LOB data is to be unloaded with the base table data.

If SPANNED YES is not specified, LOB data is not unloaded; it is ignored.

If SPANNED YES is specified, MODE DB2SQL may not be specified.

Otherwise, the NGT Unload utility works as defined and described in the *BMC Next Generation Technology Unload for DB2 for z/OS Reference Manual*. 
Chapter 4  Parameters

There is not a LOBMaster DD for parameters. LOBMaster introduces new parameters to Reorg and can be overridden in the RRGPARMS DD in the job stream. If not overridden, the default values will be used from the specific system configuration. For more about the system configuration see the NGT Installation Guide.

For more about NGT parameters including the parameters listed below, see the respective product manuals.

4.1  Parameter Overview

Parameters coded in the global parameters DD statement UTLPARMS or in any of the utility parameter files follow a few simple syntax rules.

1. All parameters must start with a plus sign (+).
2. Parameters can start anywhere, but must be contained within the first 72 character positions of the line. Columns 73-80 are ignored and may contain sequence numbers.
3. Parameters may be coded on multiple lines. Continuation is automatic.
4. Multiple parameters may be coded on the same line.
5. Specify comments by embedding them between /* */ pairs. Comments can be anywhere, even in the middle of a word. All comment strings are removed before syntax checking begins. If specifying parameters via a DD * card in your JCL, be careful not to begin a comment start (/*) in column 1. If you do so, MVS will internally generate a //SYSIN for any subsequent records in the dataset unless you use DLM= to change the JCL comment delimiter.
6. Comments may be nested. There is no (practical) limit to the nesting level of the comments, however, for each /*, there must be a corresponding */.
7. You may specify no value for a parameter by just coding open and close parenthesis with no value, such as +DISP(). This will cause the default value to be used.
8. Blanks, literally coded anywhere, have no significance. All blanks are removed before syntax checking begins. This means that a series of values must be separated by commas.
9. If the same parameter is specified multiple times within the same input dataset, the latest specification prevails.
10. Do not use the plus sign inside the parenthesis for any global or product parameter.
4.2 **LOBMaster specific Reorg parms (RRGPARMS)**

4.2.1 **+LOBERRORS(action,message)**
Specify what to do if NGT integrity checking finds an error while processing a table space with an associated LOB.

**Syntax**

```
+LOBERRORS(action,message)
```

**Description**

Use this parm to specify what is to be done if Reorg finds an integrity error while processing LOBs.

If errors in the LOB structure are found and MSG is specified, messages will be written to SYSERROR for each error followed by messages for total LOB error counts. See Section 6.3 in this manual for a sample report.

**Operands**

**Action**

<table>
<thead>
<tr>
<th>Action</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABEND or A</td>
<td>Abend the Reorg.</td>
</tr>
<tr>
<td>ERASE or E</td>
<td>For future use.</td>
</tr>
<tr>
<td>IGNORE, IGN, or I</td>
<td>Leave both as is and complete the Reorg.</td>
</tr>
</tbody>
</table>

**Message**

<table>
<thead>
<tr>
<th>Message</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESSAGE, MSG or M</td>
<td>Write message to SYSERROR.</td>
</tr>
<tr>
<td>NOMESSAGE, NOMSG or NOM</td>
<td>Don't write message to SYSERROR.</td>
</tr>
</tbody>
</table>

**Example**

```
+LOBERRORS(ABEND,MSG)
```

Abend if errors in the structure and issue a message.

**NOTE:**
For sample messages see “Reports to the SYSERROR DD” in Chapter 6 of this manual.
4.2.2  +LOBMISSING(*action*, *message*)

Specify what to do if LOB auxiliary row is missing.

Syntax

```
+LOBMISSING( *action*, *message* )
```

Description

Use this parm to specify what is to be done if Reorg finds a row in the Base table space whose corresponding row in the LOB auxiliary table space is missing.

A LOB is only considered missing if the base row has a version number in the LOB field indicating that there should be a corresponding LOB row in the auxiliary table space.

If missing LOBs are found and MSG is specified, messages will be written to SYSERROR for each missing LOB followed by messages for total missing LOB counts. See Section 6.3 in this manual for a sample report.

Operands

**Action**

- **ABEND** or **A**  Default: Abend the Reorg.
- **ERASE** or **E**  *For future use.*
- **IGNORE**, **IGN**, or **I**  Leave both as is and complete the Reorg.

**Message**

- **MESSAGE**, **MSG** or **M**  Default: Write message to SYSERROR.
- **NOMESSAGE**, **NOMSG** or **NOM**  Don’t write message to SYSERROR.

Example

```
+LOBMISSING(ABEND,MSG)
```

Abend if any missing LOB entries and write a message.

**NOTE:**

For sample messages see “Reports to the SYSERROR DD” in Chapter 6 of this manual.
4.2.3  \texttt{+LOBORPHANS}(\textit{action},\textit{message})

Specify what to do if LOB base table row is missing.

Syntax

\[ \texttt{+LOBORPHANS}(\textit{action},\textit{message}) \]

Description

Use this parm to specify what is to be done if Reorg finds that a row from the LOB auxiliary table space does not have a corresponding row in the Base table.

If an orphan is found and MSG is specified, messages will be written to SYSERROR for each orphan followed by messages for total orphan counts. See Section 6.3 in this manual for a sample report.

Operands

Action

\begin{itemize}
\item \texttt{ABEND} or \texttt{A}  Default: Abend the Reorg.
\item \texttt{ERASE} or \texttt{E}  Erase the LOB entry
\item \texttt{IGNORE}, \texttt{IGN} or \texttt{I}  Leave both as is and complete the Reorg.
\end{itemize}

Message

\begin{itemize}
\item \texttt{MESSAGE}, \texttt{MSG} or \texttt{M}  Default: Write message to SYSERROR.
\item \texttt{NOMESSAGE}, \texttt{NOMSG} or \texttt{NOM}  Don't write message to SYSERROR.
\end{itemize}

Example

\[ \texttt{+LOBORPHANS(ERASE,MSG)} \]

Delete the orphan LOBs and issue a message.

\textbf{ADVANTAGE:}

The \texttt{+LOBORPHANS(ERASE,MSG)} option is the equivalent of:

\begin{itemize}
\item CHECK INDEX
\item CHECK LOB
\item CHECK DATA...SCOPE AUXONLY...LOBERROR REPORT
\end{itemize}

Then for each orphan found:

\begin{itemize}
\item REPAIR...LOCATE...ROWID...DUMP
\end{itemize}
Chapter 5  Examples

This chapter covers the syntax for specific LOBMaster examples.

5.1  Reorganize a table space with LOB column(s)
To reorganize a table space, add the AUX keyword to the REORG TABLESPACE statement.

//SYSIN DD *
REORG TABLESPACE NWDB.TSWLOB AUX

With the AUX keyword, NGT Reorg will reorganize all associated LOB table space at the same

time it is reorganizing the Base TS.

If the Base table space does not have LOB columns defined, the AUX keyword is ignored.

5.2  Reorganize a table space and clean up orphan LOBs
To reorganize a table space, add the AUX keyword to the REORG TABLESPACE statement.
To clean up orphaned LOBs set +LOBORPHANS (ERASE,MSG). This could be configured as

the system default.

//RPGPARMS DD *
+LOBORPHANS(ERASE,MSG)
//SYSIN DD *
REORG TABLESPACE NWDB.TSWLOB AUX

By specifying +LOBORPHANS(ERASE,MSG) CDB Reorg will perform the equivalent of Check

LOB, Check Index, Check Data, and Repair to delete the orphans.

5.3  Unload a LOB table space
To unload a table with LOB columns and include the LOB data with the base table data, specify

SPANNED YES on the UNLD statement.

//SYSIN DD *
UNLD SPANNED YES
FROM TABLE OWNRID.Table_With_LOB

By not specifying an OUTDDN the unload file is dynamically allocated by the XULDDYNM

automation routine.
5.4 **Unload a LOB table space to static unload file.**

This unload names an unload file. Having one unload file will prevent some parallelism and the use of Servers which spreads the workload and speeds processing.

```
//SYSREC DD UNIT=SYSDA,DISP=(NEW,CATLG,DELETE),
// DSN=HLQ.UNLD.LOBTABLE,
// SPACE=(CYL,(1000,500),RLSE)
//SYSIN DD *
   UNLD FORMAT DSNTIAUL OUTDDN SYSREC
   SPANNED YES
   FROM TABLE OWNRID.Table_With_LOB
```

The dynamic allocation of unload files by XULDDYNM is recommended, especially when unloading a lot of data from multiple partitions.
LOBMaster Specific Reports

Chapter 6

With a LOBMaster license, NGT Reorg produces a series of reports to the LOBREPT DD and additional information to existing SUMMARY DD for LOB table spaces.

6.1 Reports to the LOBREPT DD

6.1.1 LOB Report

The Object Space Report documents the type and size of the objects being processed.

LOB REPORT FOR BASE TABLESPACE DBPROD1.PBRLOB01

<table>
<thead>
<tr>
<th>PART</th>
<th>COL#</th>
<th>TSNAME</th>
<th># LOBS</th>
<th>SIZE</th>
<th>AVERAGE</th>
<th>SMALLEST</th>
<th>LARGEST</th>
<th>INPUT</th>
<th>OUTPUT</th>
<th>SAVED</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26</td>
<td>LJ9BRPDQ</td>
<td>2</td>
<td>90</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>23</td>
<td>23</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>LJ9BRWHD</td>
<td>2</td>
<td>90</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>23</td>
<td>23</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>LJ9BRXN1</td>
<td>2</td>
<td>90</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>23</td>
<td>23</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>LJ9BRXBW</td>
<td>2</td>
<td>90</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>23</td>
<td>23</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
<td></td>
<td>8</td>
<td>360</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>92</td>
<td>92</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

6.1.2 Inline Report

The Inline LOB Report shows the number and size of the inline portion of the LOBs.

INLINE REPORT FOR BASE TABLESPACE BI12DWWA.PBRLOB01

<table>
<thead>
<tr>
<th>PART</th>
<th>COL#</th>
<th>TSNAME</th>
<th># LOBS</th>
<th>COMPLETE</th>
<th>TOTAL SIZE</th>
<th>AVG SIZE</th>
<th># LOBS</th>
<th>TOTAL SIZE</th>
<th>AVG SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26</td>
<td>LK23C675</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>LK23DJNH</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>LK23DJZ8</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>LK23ESYZ</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
<td></td>
<td>4</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

6.1.3 Distribution of LOB Sizes

This report shows, for each LOB table space, the distribution of the LOBs by size.

DISTRIBUTION OF LOB SIZES

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSNAME</th>
<th>PART</th>
<th>COL#</th>
<th>INLINE</th>
<th>RANGE LOW VALUE</th>
<th>RANGE HIGH VALUE</th>
<th># LOBS</th>
<th>PCT OF TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBPROD1</td>
<td>PBRLOB01 1</td>
<td>26</td>
<td>LJ9BRPDQ</td>
<td>0</td>
<td>100</td>
<td>--</td>
<td>2 LOBS, 100.0%</td>
<td></td>
</tr>
<tr>
<td>DBPROD1</td>
<td>PBRLOB01 2</td>
<td>26</td>
<td>LJ9BRWHD</td>
<td>0</td>
<td>100</td>
<td>--</td>
<td>2 LOBS, 100.0%</td>
<td></td>
</tr>
<tr>
<td>DBPROD1</td>
<td>PBRLOB01 3</td>
<td>26</td>
<td>LJ9BRXN1</td>
<td>0</td>
<td>100</td>
<td>--</td>
<td>2 LOBS, 100.0%</td>
<td></td>
</tr>
<tr>
<td>DBPROD1</td>
<td>PBRLOB01 4</td>
<td>26</td>
<td>LJ9BRXBW</td>
<td>0</td>
<td>100</td>
<td>--</td>
<td>2 LOBS, 100.0%</td>
<td></td>
</tr>
<tr>
<td>DISTRIBUTION OF LOB SIZES FOR ALL</td>
<td>All</td>
<td>ALL</td>
<td>All</td>
<td>8 LOBS</td>
<td>--</td>
<td>--</td>
<td>8 LOBS, 100.0%</td>
<td></td>
</tr>
</tbody>
</table>
6.2 Reports to the SUMMARY DD

6.2.1 Internal IDS Report

The LOBS ON BASE TBL information has been added to the existing Internal IDS report for each defined LOB column to show the auxiliary TS and IX for each LOB column.

```
- INTERNAL IDS REPORT
- TABLESPACE B11ZDWWA.PBRLOB01 DB=0112 PS=008C FILE=008B PTNS=4 SGSZ=16
- TBL 008D TEST_PBRLOB01_TBL1
- GTIX DBPROD1.FAKE PIX FANSET=0000 PGSET=0000 TYPE=DF KYLEN=4 KYCOLS=1
- C NPI DBPROD1.RLOBIXG2 FANSET=00A8 PGSET=00A9 TYPE=DF KYLEN=4 KYCOLS=1
- NPI DBPROD1.RLOBIX26 FANSET=00EB PGSET=00EC TYPE=DF KYLEN=13 KYCOLS=2
- LOBS ON BASE TBL COL #26
-     FILE FSET AUX      AUX
-      PART OBID OBID TSNAME IXNAME
-     ---- ---- ---- -------- --------
-       1 008E 0092 LJYQNF9A ITES1TCR0
-       2 0099 009C LJYONQGT1 ITES1IUN
-       3 009E 00A1 LJYQNGEO ITES1ITCO
-       4 00A3 00A6 LJYQR1ITJ ITES1IYO
```
6.3 Messages to the SYSERROR DD

6.3.1 +LOBERRORS Messages

If MSG is specified on the +LOBERRORS parameter, messages like the following will be written to the SYSERROR if LOB structure errors are encountered.

```
CDBLB910 MISSING ROWID 1990A7FACE3D318-01-0001 (80), MPPG 00000218, 148 BYTES
CDBLB913 - NO LOB IX FOR LOB, LOB IN LOB TS LOB1A IS COMPLETE
CDBLB913 - LOB INDEX WILL BE CORRECTED.
CDBLB910 MISSING ROWID 3DEF918CCED33017-01-0001 (80), MPPG 0001B295, 6,092,923 BYTES
CDBLB913 - NO LOB IX FOR LOB, LOB IN LOB TS LOB1A IS COMPLETE
CDBLB913 - LOB INDEX WILL BE CORRECTED.
CDBLB910 MISSING ROWID CD72A98CCED3301C-01-0001 (80), MPPG 0001B895, 56,895,894 BYTES
CDBLB913 - NO LOB IX FOR LOB, LOB IN LOB TS LOB1A IS COMPLETE
CDBLB913 - LOB INDEX WILL BE CORRECTED.
CDBLB910 MISSING ROWID DA49E7F4CED33011-01-0001 (80), MPPG 0000FF85, 6,683,221 BYTES
CDBLB913 - NO LOB IX FOR LOB, LOB IN LOB TS LOB1A IS COMPLETE
CDBLB913 - LOB INDEX WILL BE CORRECTED.
```

6.3.2 +LOBMISSING Messages

If MSG is specified on the +LOBMISSING parameter, messages like the following will be written to the SYSERROR if missing LOBs are encountered.

```
CDBLB8901 - LOB 1E3C3CF35D3320F-1904012F0700-0001 C 5 (2) TSROW 00000206) MISSING - NOT IN LOBTS
CDBLB8901 - LOB 3071012F35D3320C-1904012F0700-0001 C 5 (2) TSROW 00000209) MISSING - NOT IN LOBTS
CDBLB8901 - LOB 4737B22F35D33210-1904012F0700-0001 C 5 (2) TSROW 00000208) MISSING - NOT IN LOBTS
CDBLB8901 - LOB 802E982F35D33014-1904012F0700-0001 C 5 (2) TSROW 00000207) MISSING - NOT IN LOBTS
CDBLB8901 - LOB 5DB33218-1904012F0700-0001 C 5 (2) TSROW 00000209A) MISSING - NOT IN LOBTS
CDBLB903 - LOB GWOLB01, COL 5 (2) - 5 TOTAL ERRORS
CDBLB904 - LOB GWOLB01, COL 5 (2) - 5 MISSING
CDBLB905 - RUN CONTINUES BECAUSE +LOBMISSING = IGNORE
```

6.3.3 +LOBORPHANS Messages

If MSG is specified on the +LOBORPHANS parameter, messages like the following will be written to the SYSERROR if orphaned LOBs encountered.

```
G201.LBMLRGLB CDBLB901 - LOB L8998C404F5D3320E-1904012F0700-0001 C 5 (1) (3,156,468 BYTES) = ORPHAN - NO BASE REFC
G201.LBMLRGLB CDBLB901 - LOB 225E1B04F5D33008-1904012F0700-0001 C 5 (1) (48 BYTES) = ORPHAN - NO BASE REFC
G201.LBMLRGLB CDBLB901 - LOB 339AC804F5D33016-1904012F0700-0001 C 5 (1) (6,683,121 BYTES) = ORPHAN - NO BASE REFC
G201.LBMLRGLB CDBLB901 - LOB 4737B22F35D33210-1904012F0700-0001 C 5 (1) (6,683,121 BYTES) = ORPHAN - NO BASE REFC
G201.LBMLRGLB CDBLB901 - LOB 5DB33218-1904012F0700-0001 C 5 (1) (6,683,121 BYTES) = ORPHAN - NO BASE REFC
G201.LBMLRGLB CDBLB901 - LOB G2W0LB01, COL 5 (1) - 5 TOTAL ERRORS
G201.LBMLRGLB CDBLB901 - LOB G2W0LB01, COL 5 (1) - 5 ORPHANS
G301.LBMLRGLB CDBLB901 - LOB 12145584F5D33213-1904012F0700-0001 C 5 (2) (548,645 BYTES) = ORPHAN - NO BASE REFC
G301.LBMLRGLB CDBLB901 - LOB 459BB904F5D33009-1904012F0700-0001 C 5 (2) (56,895,794 BYTES) = ORPHAN - NO BASE REFC
G301.LBMLRGLB CDBLB901 - LOB 7A91284F5D33002-1904012F0700-0001 C 5 (2) (56,895,794 BYTES) = ORPHAN - NO BASE REFC
G301.LBMLRGLB CDBLB901 - LOB A575884F5D33218-1904012F0700-0001 C 5 (2) (6,992,823 BYTES) = ORPHAN - NO BASE REFC
G301.LBMLRGLB CDBLB901 - LOB A575704F5D33020-1904012F0700-0001 C 5 (2) (56,895,794 BYTES) = ORPHAN - NO BASE REFC
G301.LBMLRGLB CDBLB901 - LOB G3W0LB01, COL 5 (2) - 5 TOTAL ERRORS
G301.LBMLRGLB CDBLB901 - LOB G3W0LB01, COL 5 (2) - 5 ORPHANS
```
7.1 Reasons for LOBMaster

NGT LOBMaster automatically checks the validity of the Base TS, LOB IX, and LOB TS replacing the need for Check LOB, Check Index, Check Data, and REPAIR utilities. This both simplifies and speeds up the management of objects with LOB columns. NGT LOBMaster has parameters that provide the user with options to automatically handle conditions such as orphaned LOBs.

**Reclaim LOB auxiliary space from deleted rows.** Always specify a LOBREPRT DD to get reports showing how much space is saved from reorganizing the LOB auxiliary table spaces. This report can be obtained without Reorganizing by specifying REPORTONLY on the Reorg statement.

**Avoid orphaned LOB rows when discarding.** When using Reorg to discard rows from a table with LOB column, specify AUX so the associated LOB auxiliary table spaces will be reorganized concurrently therefore discarding the LOB data along with the base row. If you request to save the discarded data, the LOB data will be written to the UNLD file along with the Base table data in spanned record format suitable for LOAD.
7.2 **When is LOBMaster required**

Since Base table spaces with LOB columns and their associated LOB table spaces must be kept in sync, the following conditions require LOBMaster and the AUX keyword to Reorg.

**Reorg with Discarding of a TS with LOB columns.**

Discarding rows from the Base table requires discarding the associated rows from the LOB table spaces. To avoid orphaning LOB data from discarded base table rows, specify the AUX keyword on the Reorg statement. With the AUX keyword the LOB data will be discarded concurrently with the base row.
Index

+ +LOBERRERS, 4-2
+LOBERRORS, 1-2, 1-3, 6-3
+LOBMISSING, 1-2, 1-3, 4-3, 6-3
+LOBORPHANS, 1-2, 1-4, 4-4, 5-1, 5-2, 6-3

A
Administrator, 2-1
authority, 2-1
Authorizations, 2-1
AUX, 3-2, 5-1

B
Base Table, 1-1
Base Tablespace, 1-1

D
DB2 States, 2-1
DD Statements, 3-1
Diagram, 3-2
DISCARD, 3-2
Distribution of LOB Sizes, 6-1

F
Features, 2-2

I
Inline Report, 6-1
input, 3-1
Internal IDS Report, 6-2

L
LOB, 1-1
LOB Auxiliary Tablespace, 1-1
LOB Errors, 1-2, 1-3
LOB ID, 1-1
LOB IX, 1-1
LOB Missing, 1-2
LOB Orphans, 1-2, 1-4
LOB Report, 6-1
LOBREPRT, 2-3, 3-1, 6-1

R
Reasons to Reorg, 7-1
Reorg, 2-3
REORG TABLESPACE, 3-2
RW, 2-1

S
sequential datasets, 2-1
SPANNED YES, 1-4, 2-3, 3-4
Statuses, 2-1
Summary, 2-1
Syntax, 3-1, 3-2
SYSADM, 2-1
SYSIN, 3-1
SYSPUNCH, 3-2

U
UNLD, 3-4
Unload, 2-3

V
Variable Blocked Spanned, 3-2, 3-4
VBS, 1-4, 2-3, 3-2, 3-4

X
XULDDYNM, 3-2