BMC Next Generation Technology Utility Manager for DB2 for z/OS
Reference Manual

Supporting

Version 12.1 of BMC Next Generation Technology Utility Manager 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 or 1 713 918 8000</td>
<td>1 713 918 8000 or 1 800 841 2031</td>
</tr>
<tr>
<td>2103 CITYWEST BLVD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

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

© 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.

DB2, IBM, 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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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.
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 statements

This topic explains conventions for showing syntax statements.

A sample statement follows:

```
COMMAND KEYWORD1 [KEYWORD2 | KEYWORD3] KEYWORD4={YES | NO} fileName...
```

The following table explains conventions for syntax statements and provides examples:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items in italic type represent variables that you must replace with a name or value. If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words.</td>
<td>alias</td>
</tr>
<tr>
<td></td>
<td>databaseDirectory</td>
</tr>
<tr>
<td></td>
<td>serverHostName</td>
</tr>
</tbody>
</table>
### Convention

Brackets indicate optional items. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option.

\[ \text{tableName, columnName, field} \]
\[-full, -incremental, -level]\n
Braces indicate that at least one of the enclosed items is required. Do not type the braces when you enter the item.

\{DBDName | tableName\}
UNLOAD device={disk | tape, \textit{fileName} | deviceName}\n{-a | -c}\n
A vertical bar means that you can choose only one of the listed items. In the example, you would choose either \textit{commit} or \textit{cancel}.

\{commit | cancel\}\n
An ellipsis indicates that you can repeat the previous item or items as many times as necessary.

\textit{columnName}...

### Syntax diagrams

The following figure shows the standard format for syntax diagrams:

![Syntax Diagrams](image)

The following example illustrates the syntax for a hypothetical DELETE statement. Because the FROM keyword, \textit{alias} variable, and WHERE clause are optional, they appear below the main command line. In contrast, the \textit{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).

Summary of changes

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

Version 12.1, December 2016

This release of NGT Utility Manager includes the following product enhancements and changes.
Support for IBM DB2 Version 12

This release adds support for DB2 Version 12. This release supports:
- DB2 Version 12
- DB2 Version 11
- DB2 Version 10 in new-function mode (NFM) only

Integration between NGT Utility Manager and BMC Workbench

You can now manage the NGT Utility Manager Schedule, Criteria, and Exceptions tables from a view in the browser-based BMC Workbench for DB2 product.

Note
You can use this feature if you have a license for any of the following simplified solutions:
- BMC High Speed Utilities for DB2
- BMC Object Administration for DB2
- BMC Recovery for DB2
- BMC Performance for DB2SQL
- BMC Performance for DB2 Databases
- BMC Utility Management for DB2
- BMC Large Object Management for DB2

Running NGT Reorg under NGT Utility Manager

If you run NGT Reorg under NGT Utility Manager, NGT Reorg terminates if you specify REBALANCE or DDLIN.

RECOVER PLUS for DB2 name change

Starting with this release, the name of the RECOVER PLUS for DB2 product has changed to the BMC Next Generation Technology Recover for DB2 for z/OS (or NGT Recover) product.
Overview of NGT Utility Manager

The BMC Next Generation Technology Utility Manager for DB2 for z/OS (NGT Utility Manager) product helps you determine—from within an NGT utility product—whether to run the utility against your database objects, based on real-time statistics.

NGT Utility Manager maximizes application performance and saves resources by:

■ Enabling frequent Reorg executions (because the product can select fewer—or no—objects for processing

■ Reorganizing only objects that match your reorganization criteria, thus preventing unnecessary reorganizations

What NGT Utility Manager does

The NGT Utility Manager product uses criteria that you specify and IBM DB2 runtime statistics (RTS) to identify database objects that require processing.

NGT Utility Manager identifies the following table spaces and indexes:

■ Objects that need to be copied or reorganized
■ Objects for which you need to collect statistics

Then, depending on the options that you select, NGT Utility Manager takes one of the following actions:

■ Identifies and reports the objects that require processing—but without actually processing them
■ Identifies objects that require processing and runs the appropriate Next Generation Utility (NGT) product to process the identified objects

If you have purchased a license for NGT Utility Manager, you can invoke the product from any NGT utility product.
Note

To invoke NGT Utility Manager from the NGT Copy product, you must invoke NGTUTIL—not ACPMAIN—to drive your copies.

After setting up NGT Utility Manager, you can activate it by including an RTS keyword in your utility product's SYSIN statement.

In the following example, the RTS keyword invokes NGT Utility Manager to use real-time statistics to determine whether NGT Reorg should reorganize table space CENTRAL.ACCTGREC:

Example

```
//SYSIN DD *
REORG TABLESPACE CENTRAL.ACCTGREC RTS
```

Related Information

- “Setting up and running NGT Utility Manager” on page 19
- “Running an evaluation with NGT Utility Manager” on page 22

NGT Utility Manager requirements and restrictions

The NGT Utility Manager product has the following requirements and restrictions:

- NGT Utility Manager supports Version 10 and later of the IBM DB2 subsystem.
- NGT Utility Manager evaluates a table space, but it runs the specified utility only if the table space has read/write (RW) status and is not in a restrictive state.

Benefits of running NGT Utility Manager

Using NGT Utility Manager to evaluate the objects before you process them offers the following advantages:

- You can specify separate evaluation criteria for each object, or groups of objects.
- NGT Utility Manager can recommend an index for reorganization even if the corresponding table space does not require reorganization.
NGT Utility Manager can trigger a reorganization even if the product has not initialized Reorg values in RTS.

You can use WHERE clauses to specify individual job names, subsystem IDs, or utility IDs (UIDs).

You can use the Schedule table to fine-tune when to run NGT products (for example, by forcing or preventing reorganizations for a particular day of the week, date, or date range).

You can use the Exceptions table to force or prevent reorganizations that deviate from the schedule defined in the Schedule table records.

Overview of NGT Utility Manager tables

To evaluate objects, NGT Utility Manager requires Schedule, Criteria, and Exceptions tables. The tables are created when you install NGT Utility Manager, but you can edit them, as needed.

The Schedule table sets the normal processing time frame.

Use this table to avoid reorganizations on a particular day of the month, every month.

The Criteria table lists sets of table spaces or indexes to evaluate. NGT Utility Manager refers to fields in the Criteria table to determine whether to perform specified actions.

The Exceptions table defines exceptions to standard processing schedules. The values in these columns tell NGT Utility Manager whether to:

— Exclude objects from processing
— Force processing
— Ignore the Exceptions record for the specified utility

Example

If the Exceptions record applies to NGT Reorg, NGT Utility Manager ignores the Exceptions record for NGT Stats and NGT Copy.

If planning to implement application changes, you can edit the Exceptions table to prevent NGT Reorg from reorganizing objects on a specified day.
NGT Utility Manager evaluation process

When NGT Utility Manager is activated to run, it uses the following process to evaluate objects:

1. **Checks the Exceptions table** for a qualifying record (if you specified an application or record with the RTS keyword) or for an object name
   The Exceptions table includes a column for each applicable utility (NGT Reorg, NGT Copy, and NGT Stats):
   - If the value for a utility is F (Force) and an exception criterion exists, NGT Utility Manager forces the utility to run against the object.
   - If the value is X (Exclude) and an exception criterion exists, NGT Utility Manager excludes the utility from running against the object.
   - If the value for a utility is I (Ignore), NGT Utility Manager ignores the Exceptions record for the object. NGT Utility Manager then considers the next qualifying record.

2. **Checks the object against the Schedule table** if the Exceptions table did not resolve whether to process that object
   Like the Exceptions table, the Schedule table includes a column for each applicable utility to indicate whether the object applies to that utility (Y or N). If the time frame and the utility both apply, the object is included or excluded based on the value in the **Inclusive_Exclusive** column.

3. **Checks the object against the Criteria table** if the Schedule table did not exclude processing
   NGT Utility Manager selects the object for processing if the object meets any criterion in the Criteria table; otherwise, NGT Utility Manager prevents processing of the object.

Following the evaluation, NGT Utility Manager issues messages NGTZ307, NGTZ309, and NGTZ369 to confirm that it successfully read the RTS tables.
Example

NGTZ307 RTS CRITERIA DEFINITION SUCCESSFULLY READ
NGTZ307 TB=DB2USER.NGT_RTS_CRITERIA
NGTZ309 RTS EXCEPTION DEFINITION SUCCESSFULLY READ
NGTZ309 TB=DB2USER.NGT_RTS_EXCEPTIONS
NGTZ369 RTS SCHEDULE DEFINITION SUCCESSFULLY READ
NGTZ369 TB=DB2USER.NGT_RTS_SCHEDULE
Setting up and running NGT Utility Manager

This chapter explains how to set up the NGT Utility Manager product, modify the tables when necessary, and run the product following setup.

Setting up NGT Utility Manager

When installing NGT Utility Manager, the Installation System creates the required Criteria, Schedule, and Exceptions tables. The installation process also defines (registers) the tables to the NGT checkpoint data set on the specified subsystem.

After installation, no additional setup is required unless you need to:

- Define the tables to a checkpoint data set on a different subsystem
- Redefine the NGT Utility Manager schema because you deleted the NGT checkpoint data set and created a new one

If either situation applies to you, complete the appropriate procedure in “Defining NGT Utility Manager tables to the NGT checkpoint data set” on page 19.

Defining NGT Utility Manager tables to the NGT checkpoint data set

Use the following procedures (as relevant) to manually define (register) or redefine the NGT Utility Manager Criteria, Exceptions, and Schedule tables to the NGT checkpoint data set.
To manually define NGT Utility Manager tables to the checkpoint

1 In the DEFINE job, insert a SYSIN DD statement with a DEFINE statement to register the tables:

The following sample DEFINE statement is also available in the IDEFRTS member in the UBMCCNTL data set. For more information about the NGTRTS DEFINE statement, see “Keywords for manually defining the NGT Utility Manager tables” on page 52.

```
//SYSIN DD *
NGTRTS DEFINE DBNAME(databaseName) TSNAME(tablespaceName)
  BUFFERPOOL(bufferPoolName) STOGROUP(storageGroupName)
  CRITERIA_TABLE(creator.tableName)
  CRITERIA_INDEX(creator.indexName)
  EXCEPTIONS_TABLE(creator.tableName)
  EXCEPTIONS_INDEX(creator.indexName)
  SCHEDULE_TABLE(creator.tableName)
  SCHEDULE_INDEX(creator.indexName)
```

For the variable text, use the names that were generated when the Installation System created the NGT Utility Manager tablespace and tables.

**Example**

Following is a sample of a typical DEFINE job:

```
//STEP01 EXEC PGM=NGTUTIL,PARM='ssid,,RESTART'
//STEPLIB DD ...
//SYSPRINT DD SYSOUT=* 
//UTPRINT DD SYSOUT=* 
//SYSIN DD *
NGTRTS DEFINE DBNAME(NGTRTS) TSNAME(RTSCRIT)
  EXCP_DBNAME(NGTRTS)       EXCP_TSNAME(RTSEXCP)
  SCHEDULE_DBNAME(NGTRTS)   SCHEDULE_TSNAME(RTSSCHD)
  BUFFERPOOL(BP0) STOGROUP(DB2STGRP)
  CRITERIA_TABLE(DB2USER.NGT_RTS_CRITERIA)
  CRITERIA_INDEX(DB2USER.NGT_RTS_CRITERIA_IX)
  EXCEPTIONS_TABLE(DB2USER.NGT_RTS_EXCEPTIONS)
  EXCEPTIONS_INDEX(DB2USER.NGT_RTS_EXCEPTIONS_IX)
  SCHEDULE_TABLE(DB2USER.NGT_RTS_SCHEDULE)
  SCHEDULE_INDEX(DB2USER.NGT_RTS_SCHEDULE_IX)
/*
/*
```

2 Execute the DEFINE job.

Executing this DDL specifies a buffer pool and storage group for the new indexes.

3 Add default rows to the Criteria table (as applicable) by executing CNTL member CRITWILD with a native processor (such as IBM SPUFI or DSNTEP2).

The inserted default rows use starting values that the DSNACCOX stored procedure provides.
Note
You can edit the Criteria, Exceptions, and Schedule tables at any time. For more information, see “Running an evaluation with NGT Utility Manager” on page 22 and “Editing NGT Utility Manager tables” on page 23.

To redefine NGT Utility Manager tables to a new NGT checkpoint

1 In a DEFINE job, create a SYSIN DD statement that includes an NGTRTS DEFINE statement, as follows:

   a Insert an NGTRTS DEFINE statement that includes the REPLACE keyword.

   //SYSIN DD *
   NGTRTS DEFINE DBNAME(databaseName) TSNAME(tablespaceName)
   BUFFERPOOL(bufferPoolName) STOGROUP(storageGroupName) REPLACE

   Tip
   The IDEFRTS member in the UBMCCNTL data set provides a sample DEFINE statement with REPLACE.

   b To see which NGT Utility Manager tables are defined for the current DB2 subsystem, run the NGTRTS DISPLAY statement.

   c Using the results of the DISPLAY statement, add the displayed table and index names to your DEFINE command.

   //SYSIN DD *
   NGTRTS DEFINE DBNAME(DBRTSCDB) TSNAME(RTSDSN1)
   BUFFERPOOL(BP0) STOGROUP(DB2STGRP)
   CRITERIA_TABLE(DB2USER.NGT_RTS_CRITERIA)
   CRITERIA_INDEX(DB2USER.NGT_RTS_CRITERIA_IX)
   EXCEPTIONS_TABLE(DB2USER.NGT_RTS_EXCEPTIONS)
   EXCEPTIONS_INDEX(DB2USER.NGT_RTS_EXCEPTIONS_IX)
   SCHEDULE_TABLE(DB2USER.NGT_RTS_SCHEDULE)
   SCHEDULE_INDEX(DB2USER.NGT_RTS_SCHEDULE_IX)

   Note
   If the new checkpoint data set already has schema objects registered, the REPLACE keyword tells NGT Utility Manager to replace those registrations with the objects listed in this DEFINE job.

2 Run the DEFINE job.

Related Information

- “Keywords for managing Criteria, Schedule, and Exceptions tables” on page 52
Running an evaluation with NGT Utility Manager

Use this procedure to tell NGT Utility Manager to evaluate objects on behalf of the specified NGT utility and determine whether the utility should process the identified objects.

To run an NGT Utility Manager evaluation

1. Enter a SYSIN DD statement that includes the RTS keyword, using the following syntax:
   ```plaintext
   //SYSIN DD *
   ngtUtilityStatement RTS[(option)]
   ```

   **Example**
   The following statement runs NGT Utility Manager to determine whether the NGT Reorg utility needs to process objects in table space MEDRECS.PATIENTS. Then (if the evaluation so indicates), the statement runs NGT Reorg to process the objects.
   ```plaintext
   REORG TABLESPACE MEDRECS.PATIENTS RTS
   ```

   The RTS keyword is required. After the keyword, you can include any of the following options. For a description of each option, see “Keywords for running NGT Utility Manager evaluations” on page 49:
   - **appCriteriaRecord**
   - **,GENERATESYSIN**
   - **,GENERATESYSINALL**
   - **,REPORTONLY**
   - **appCriteriaRecord** — NGT Utility Manager invokes an application criterion (based on the NAME field in a TYPE=AP Criteria record).

   **Example**
   In the following statement, the product processes against the named criteria record for application CRIT001:
   ```plaintext
   REORG TABLESPACE MEDRECS.PATIENTS RTS(CRIT001)
   ```

   - **,GENERATESYSIN** — NGT Utility Manager evaluates objects and generates a list of candidate objects for processing without processing the objects.
Managing NGT Utility Manager tables

The following topics explain how to edit NGT Utility Manager tables, display table definitions, and delete tables.

Editing NGT Utility Manager tables

The NGT Utility Manager tables are standard IBM DB2 tables. Use the following procedure to edit them.
1 Use either of the following methods to edit the tables:

- Use standard SQL statements to edit the tables.

- If available, use the BMC Workbench for DB2 graphical user interface (GUI) to edit the NGT Utility Manager tables. For more information, see the BMC Workbench for DB2 online Help.

---

**Note**

BMC Workbench for DB2 is available only as part of specific BMC solutions for DB2. It is not available as a stand-alone component.

---

### Displaying NGT Utility Manager table definitions

Use the following procedure to view the NGT Utility Manager table definitions that are currently registered in the checkpoint data set.

1 Enter a SYSIN DD statement that includes an NGTRTS DISPLAY statement, using the following syntax:

```
//SYSIN DD *
NGTRTS DISPLAY [ALL]
```

Include ALL if you want to see table definitions for all DB2 subsystems in the checkpoint. Omitting ALL displays definitions for the current DB2 subsystem only.

### Deleting NGT Utility Manager tables

Deleting the NGT Utility Manager tables is rarely needed. However, if necessary, you can use the DELETE statement to remove the registered NGT Utility Manager schema from the NGT checkpoint data set.

---

**Note**

DELETE does not remove the schema from the DB2 catalog.
NGT Utility Manager table-column reference

This chapter shows sample Schedule, Criteria, and Exceptions tables and describes the columns in each table.

NGT Utility Manager Schedule table

The Schedule table establishes the time frame for forcing or preventing table space processing.

**Figure 1: Schedule table column definitions**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Col No</th>
<th>Col Type</th>
<th>Length</th>
<th>Scale</th>
<th>Null</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>1</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>2</td>
<td>VARCHAR</td>
<td>257</td>
<td>0</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>PARTITION</td>
<td>3</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>O</td>
<td>0</td>
</tr>
<tr>
<td>UTILITY_ID</td>
<td>4</td>
<td>CHAR</td>
<td>16</td>
<td>O</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>REORG</td>
<td>5</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>COPY</td>
<td>6</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>STATS</td>
<td>7</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>INCLUSIVE_EXCLUSIVE</td>
<td>8</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>I</td>
</tr>
<tr>
<td>MONTH_WEEK</td>
<td>9</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>MONTH</td>
<td>10</td>
<td>INTEGER</td>
<td>4</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>11</td>
<td>INTEGER</td>
<td>4</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>START_TIME</td>
<td>12</td>
<td>TIME</td>
<td>3</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>END_TIME</td>
<td>13</td>
<td>TIME</td>
<td>3</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>REMARKS</td>
<td>14</td>
<td>VARCHAR</td>
<td>128</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>LAST_UPD_USERID</td>
<td>15</td>
<td>CHAR</td>
<td>8</td>
<td>0</td>
<td>N</td>
<td>User</td>
</tr>
<tr>
<td>LAST_UPD_TIMESTAMP</td>
<td>16</td>
<td>TIMESTAMP</td>
<td>10</td>
<td>6</td>
<td>N</td>
<td>Timestamp</td>
</tr>
<tr>
<td>VERSION</td>
<td>17</td>
<td>CHAR</td>
<td>8</td>
<td>0</td>
<td>N</td>
<td>12.1.00</td>
</tr>
</tbody>
</table>

Table 1 on page 26 describes each column that is listed under **Column Name** in the Schedule table.
### Table 1: Column names in the Schedule table

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| **TYPE**             | Type of RTS record. Valid values are:  
  - AP—Application  
  - TS—Table space  
  - IX—Index         |
| **NAME**             | Name of the RTS record. You can use wildcards in the name. Valid values are:  
  - Application—128 characters maximum  
  - Table space—17 characters maximum, in format `databaseName.tableSpaceName`  
  - Index—257 characters maximum, in format `creator.indexName` |
| **PARTITION**        | Partition number. 0 (zero) signifies all partitions.  
  Default value: 0 |
| **UTILITY_ID**       | Utility ID for this schedule record  
  To limit a schedule entry to a specific Utility ID, enter that ID. A blank field signifies all utility IDs. |
| **REORG**            | Whether this record applies to NGT Reorg operations. Valid values are:  
  - Y—The schedule entry applies to NGT Reorg.  
  - N—The schedule entry does not apply to NGT Reorg.  
  Default value: N |
| **COPY**             | Whether this record applies to NGT Copy operations. Valid values are:  
  - Y—The schedule entry applies to NGT Copy.  
  - N—The schedule entry does not apply to NGT Copy.  
  Default value: N |
| **STATS**            | Whether this record applies to NGT Stats operations. Valid values are:  
  - Y—The schedule entry applies to NGT Stats.  
  - N—The schedule entry does not apply to NGT Stats.  
  Default value: N |
| **INCLUSIVE_EXCLUSIVE** | Whether this record is processed. Valid values are:  
  - I—The schedule entry includes the object for processing.  
  - E—The schedule entry excludes the object from processing.  
  **Note:** If the specified MONTH and DAY columns include the object, and if you specify I in the INCLUSIVE_EXCLUSIVE column, the Criteria check continues to process the object. If you specify E in the INCLUSIVE_EXCLUSIVE column, the object is not processed.  
  Default value: I |
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| MONTH_WEEK    | Whether this record is a monthly or weekly schedule record. Valid values are:  
  - **M**—The schedule record is a monthly record.  
  - **W**—The schedule record is a weekly record.  
  Default value: **W**  
  The MONTH and DAY columns provide additional delimiters. |
| MONTH         | The month that the schedule record covers.  
  - This field is valid when MONTH_WEEK = **M**. Valid values are:  
    - 0 for all months  
    - 1–12 for January through December |
| DAY           | The day of the week or month that the schedule record covers.  
  - Valid values are:  
    - If MONTH_WEEK = **M**  
      - 0 for all days of the month  
      - 1–31 for the specified day of the month  
    - If MONTH_WEEK = **W**  
      - 0 for all days of the week  
      - 1–7 the specified day of the week, where 1 is Sunday |
| START_TIME    | When the product starts to honor the schedule record. The valid value is a time, entered in *hh:mm:ss* format. |
| END_TIME      | When the product stops honoring the schedule record. The valid value is a time, entered in *hh:mm:ss* format. |
| REMARKS       | Comments for the schedule record |
| LAST_UPD_USERID | ID of the last user to update the row  
  Default value: *userID* |
| LAST_UPD_TIMESTAMP | Time stamp of the most recent update to the row  
  Default value: *currentTimestamp* |
| VERSION       | Default value: 12.1.00 |

You can specify multiple schedule records for the same object if the records specify different time ranges. If the time ranges overlap, the product honors the first record that matches the criteria.

**NGT Utility Manager Criteria table**

You can edit the NGT Utility Manager Criteria table to set the criteria for selecting sets of table spaces or indexes for processing. The following topics describe the criteria and suggest where you might want to add new criteria.
Note

Member CRITWILD in the NGT CNTL library contains two INSERT statements with global (base) criteria. If you are configuring version 11.2 or earlier of NGT Utility Manager, contact BMC Customer Support for the relevant INSERT statements that reflect the values in the tables.

Reorganization criteria for table spaces

This topic describes the reorganization criteria applicable to table spaces.

Table spaces usually reach reorganization thresholds before they reach space reclamation thresholds.

When you set your INDREF criterion, consider whether you are configuring a data sharing DB2 subsystem when you select the minimum values.

The initial settings for minimum values are 100,000. The initial settings for insert and delete thresholds are 25%. Consequently, when these settings are applied, the product does not organize table spaces that have fewer than 400,000 rows.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Triggers a reorganization when</th>
<th>Starting (recommended) threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Criteria relating to organization</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS_REORG_INDREF</td>
<td>(REORGNEARINDREF + REORGFARINDREF) * 100 / TOTALROWS</td>
<td>10 (5 if data sharing)</td>
</tr>
<tr>
<td>TS_REORG_UNCLUST</td>
<td>REORGCLUSTERSENS &gt; 0 AND REORGUNCLUSTINS * 100 / TOTALROWS</td>
<td>10</td>
</tr>
<tr>
<td>TS_REORG_MASSDEL</td>
<td>REORGMASSDELETE</td>
<td>0</td>
</tr>
<tr>
<td>TS_REORG_DISORG</td>
<td>REORGDISORGBLOB * 100 / TOTALROWS</td>
<td>10</td>
</tr>
<tr>
<td>TS_REORG_CHKAREO</td>
<td>Advisory reorg status Default: N</td>
<td>Y</td>
</tr>
<tr>
<td><strong>Criteria relating to space reclamation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS_REORG_INSERTS</td>
<td>REORGINSERTS * 100 / TOTALROWS</td>
<td>25</td>
</tr>
<tr>
<td>TS_REORG_INSERTSMIN</td>
<td>Minimum number of inserts</td>
<td>100000</td>
</tr>
<tr>
<td>TS_REORG_DELETES</td>
<td>REORGDELETES * 100 / TOTALROWS</td>
<td>25</td>
</tr>
<tr>
<td>TS_REORG_DELETESMIN</td>
<td>Minimum number of deletes</td>
<td>100000</td>
</tr>
</tbody>
</table>
### Reorganization criteria for indexes

This topic describes the reorganization criteria applicable to indexes.

When you set the PSEUDODEL threshold, consider whether you are configuring a data sharing DB2 subsystem.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Triggers a reorganization when</th>
<th>Starting (recommended) threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS_REORG_UPDATES</td>
<td>REORGUPDATES * 100 / TOTALROWS</td>
<td>NULL</td>
</tr>
<tr>
<td>TS_REORG_UPDATESMIN</td>
<td>Minimum number of updates</td>
<td>100000</td>
</tr>
<tr>
<td>TS_REORG_INSDELUPD</td>
<td>(REORGINSERTS + REORGDDELETES + REORGUPDATES) * 100 / TOTALROWS</td>
<td>20</td>
</tr>
<tr>
<td>TS_REORG_INSDELUPDMIN</td>
<td>Minimum number of INSDELUPD</td>
<td>100000</td>
</tr>
<tr>
<td>TS_REORG_EXTENTS</td>
<td>EXTENTS</td>
<td>50</td>
</tr>
<tr>
<td>TS_REORG_ALLOUSED</td>
<td>SPACE * 1024 / DATASIZE</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Triggers a reorganization when</th>
<th>Starting (recommended) threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria relating to organization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IX_REORG_PSEUDODEL</td>
<td>REORGPSEUDODELETES * 100 / TOTALENTRIES</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(5 if data sharing)</td>
</tr>
<tr>
<td>IX_REORG_LEAFLIMIT</td>
<td>REORGLEAFFAR * 100 / NACTIVE</td>
<td>10</td>
</tr>
<tr>
<td>IX_REORG_APPENDINS</td>
<td>REORGAPPENDINSERT * 100 / TOTALENTRIES</td>
<td>10</td>
</tr>
<tr>
<td>IX_REORG_NLEVELS</td>
<td>REORGNUMLEVELS</td>
<td>0</td>
</tr>
<tr>
<td>IX_REORG_MASSDEL</td>
<td>REORGMASSDELETE</td>
<td>0</td>
</tr>
<tr>
<td>IX_REORG_CHKAREO</td>
<td>Advisory reorg status</td>
<td>Y</td>
</tr>
<tr>
<td>Criteria relating to space reclamation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IX_REORG_EMPTY</td>
<td>NPAGES &gt; 5 AND NPAGES * 100 / NLEAF Pseudo-empty leaf pages</td>
<td>10</td>
</tr>
<tr>
<td>IX_REORG_INSERTS</td>
<td>REORGINSERTS * 100 / TOTALENTRIES</td>
<td>25</td>
</tr>
<tr>
<td>IX_REORG_INSERTSMIN</td>
<td>Minimum number of inserts</td>
<td>100000</td>
</tr>
</tbody>
</table>
**Criterion** | **Triggers a reorganization when** | **Starting (recommended) threshold**
--- | --- | ---
IX_REORG_DELETES | \( \text{REORGDELETES} \times 100 / \text{TOTALENTRIES} \) | 25
IX_REORG_DELETESMIN | Minimum number of deletes | 100000
IX_REORG_INSDEL | \( \left( \text{REORGINSETS} + \text{REORGDELETES} \right) \times 100 / \text{TOTALENTRIES} \) | 20
IX_REORG_INSDELMIN | Minimum number of changes | 100000
IX_REORG_EXTENTS | EXTENTS | 200

---

**Column definitions for the Criteria table columns**

The Criteria table tells NGT Utility Manager whether to perform actions specified in the Column Name column.

Figure 2 on page 30 describes each column that is listed under Column name in Table 1 on page 26.

Figure 2: Criteria table column definitions

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Col No</th>
<th>Col Type</th>
<th>Length</th>
<th>Scale</th>
<th>Null</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>1</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>2</td>
<td>VARCHAR</td>
<td>257</td>
<td>0</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>PARTITION</td>
<td>3</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>REMARKS</td>
<td>4</td>
<td>VARCHAR</td>
<td>128</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_REORG_NONE</td>
<td>5</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>TS_REORGCHKINDEX</td>
<td>6</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>TS_REORG_FORCECOPY</td>
<td>7</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>TS_REORG_MASSDEL</td>
<td>8</td>
<td>INTEGER</td>
<td>4</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_REORG_INSERTS</td>
<td>9</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_REORG_DELETES</td>
<td>10</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_REORG_UPDATES</td>
<td>11</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_REORG_INSDELUPD</td>
<td>12</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_REORG_INDREF</td>
<td>13</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_REORG_UNCLUST</td>
<td>14</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_REORG_DISORG</td>
<td>15</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_REORG_EXTENTS</td>
<td>16</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>IX_REORG_NONE</td>
<td>17</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>IX_REORG_MASSDEL</td>
<td>18</td>
<td>INTEGER</td>
<td>4</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>IX_REORG_INSERTS</td>
<td>19</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>IX_REORG_DELETES</td>
<td>20</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>IX_REORG_INSDEL</td>
<td>21</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>IX_REORG_PSEUDODEL</td>
<td>22</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>IX_REORG_LEAFLIMIT</td>
<td>23</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>IX_REORG_APPENDINS</td>
<td>24</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>IX_REORG_NLEVELS</td>
<td>25</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>IX_REORG_EXTENTS</td>
<td>26</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_COPY_NONE</td>
<td>27</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>TS_COPY_DESTRUCT</td>
<td>28</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>TS_COPY_INCREMENTAL</td>
<td>29</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>TS_COPY_CHKPENDING</td>
<td>30</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>TS_COPY_DAYSSINCE</td>
<td>31</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_COPY_UPDATED</td>
<td>32</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_COPY_CHANGES</td>
<td>33</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_INCR_DAYSSINCE</td>
<td>34</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TS_INCR_UPDATED</td>
<td>35</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
### Table 2: Column names in the Criteria table

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>Displays the type of RTS record. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ AP—Application</td>
</tr>
<tr>
<td></td>
<td>■ TS—Table space</td>
</tr>
<tr>
<td></td>
<td>■ IX—Index</td>
</tr>
<tr>
<td>NAME</td>
<td>Displays the name of the RTS record. You can use wildcards in the name.</td>
</tr>
<tr>
<td></td>
<td>Valid values are:</td>
</tr>
<tr>
<td></td>
<td>■ Application—128 characters maximum</td>
</tr>
<tr>
<td></td>
<td>■ Table space—17 characters maximum, in format databaseName.tableSpaceName</td>
</tr>
<tr>
<td></td>
<td>■ Index—257 characters maximum, in format creator.indexName</td>
</tr>
<tr>
<td>PARTITION</td>
<td>Displays the partition number. 0 (zero) signifies all partitions. Default</td>
</tr>
<tr>
<td></td>
<td>value: 0</td>
</tr>
<tr>
<td><strong>Column</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REMARKS</td>
<td>Displays Comments for the Criteria record</td>
</tr>
<tr>
<td>TS_REORG_NONE</td>
<td>Specifies whether to check for a previous reorganization:</td>
</tr>
<tr>
<td></td>
<td>■ If TS_REORG = Y, the product checks for a previous reorganization. If the table space has never been reorganized (REORGLASTTIME = NULL), the product reorganizes the table space.</td>
</tr>
<tr>
<td></td>
<td>■ If TS_REORG = B, the product checks for a previous reorganization or LOAD REPLACE.</td>
</tr>
<tr>
<td></td>
<td>■ If TS_REORG = null, the product reorganizes the table space.</td>
</tr>
<tr>
<td></td>
<td>Default value: N</td>
</tr>
<tr>
<td></td>
<td>Recommended value: Y</td>
</tr>
<tr>
<td>TS_REORG_CHKINDEX</td>
<td>Specifies whether the product passes indexes for RTS Criteria processing if a table space fails reorganization criteria</td>
</tr>
<tr>
<td></td>
<td>Default value: Y</td>
</tr>
<tr>
<td></td>
<td>Recommended value: Y</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If this field is set to Y and you reorganize table spaces, the product reorganizes indexes only if the table space does not require a reorganization.</td>
</tr>
<tr>
<td>TS_REORG_FORCECOPY</td>
<td>Specifies whether the product forces a copy of the table space if the table space fails Reorg criteria</td>
</tr>
<tr>
<td></td>
<td>Recommended value: N</td>
</tr>
<tr>
<td></td>
<td>Default value: N</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The criterion allows the product to copy all objects, whether or not they need to be reorganized.</td>
</tr>
<tr>
<td>TS_REORG_MASSDEL</td>
<td>Triggers a reorganization when the value of REORGMASSDELETE exceeds the value of this field</td>
</tr>
<tr>
<td></td>
<td>Recommended value: 0</td>
</tr>
<tr>
<td>TS_REORG_INSERTS</td>
<td>Triggers a reorganization when the value of REORGINserts * 100 / TOTALROWS exceeds the value of this field</td>
</tr>
<tr>
<td></td>
<td>Recommended value: 25</td>
</tr>
<tr>
<td>TS_REORG_DELETES</td>
<td>Triggers a reorganization when the value of REORGDELETES * 100 / TOTALROWS exceeds the value of this field</td>
</tr>
<tr>
<td></td>
<td>Recommended value: 25</td>
</tr>
<tr>
<td>TS_REORG_UPDATES</td>
<td>Triggers a reorganization when the value of REORgUPDATES * 100 / TOTALROWS exceeds the value of this field</td>
</tr>
<tr>
<td></td>
<td>Recommended value: NULL (no value)</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> IBM does not check this value for Reorg. The criterion is available but the initial setting is null.</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| TS_REORG_INSDELUPD   | Triggers a reorganization when the value of \( \left( \frac{\text{REORGINSERTS} + \text{REORGDELETES} + \text{REORGUPDATES}}{\text{TOTALROWS}} \right) \times 100 \) exceeds the value of this field  
Recommended value: 20 |
| TS_REORG_INDREF      | Triggers a reorganization when the value of \( \frac{\text{REORGNEARINDREF} + \text{REORGFARINDREF}}{\text{TOTALROWS}} \times 100 \) exceeds the value of this field  
Recommended value:  
- Non-data sharing environment: 10  
- Data sharing environment: 5 |
| TS_REORG_UNCLUST     | Triggers a reorganization when both of the following conditions exist:  
\- \( \text{REORGCLUSTERSENS} > 0 \)  
\- \( \frac{\text{REORGUNCLUSTINS}}{\text{TOTALROWS}} \times 100 \) exceeds the value of this field  
Recommended value: 10  
\* Note: If no SELECT commands are sensitive to cluster, the number of unclustered inserts is ignored. |
| TS_REORG_DISORG      | Triggers a reorganization when the value of \( \frac{\text{REORGDISORGLOB}}{\text{TOTALROWS}} \times 100 \) exceeds the value of this field  
Recommended value: 10 |
| TS_REORG_EXTENTS     | Triggers a reorganization when the value of EXTENTS exceeds the value of this field  
Recommended value: 50 |
| IX_REORG_NONE        | Checks for a previous reorganization as follows:  
\- If IX_REORG_NONE is set to Y:  
  \- Checks for a previous reorganization  
  \- If REORGLASTTIME = null (that is, the index has never been reorganized), reorganizes the index  
\- If IX_REORG_NONE is set to B (both), checks for a previous Reorg or Rebuild operation. If neither exists, NGT Utility Manager reorganizes the index.  
\- If IX_REORG_NONE is set to N, does nothing  
Default value: N  
Recommended value: Y |
| IX_REORG_MASSDEL     | Triggers a reorganization when REORGMASSDELETE exceeds the value of this field  
Recommended value: 0 |
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX_REORG_INSERTS</td>
<td>Triggers a reorganization when the value of ( \text{REORGINSERTS} \times 100 / \text{TOTALENTRIES} ) exceeds the value of this field. Recommended value: 25</td>
</tr>
<tr>
<td>IX_REORG_DELETES</td>
<td>Triggers a reorganization when the value of ( \text{REORGDELETES} \times 100 / \text{TOTALENTRIES} ) exceeds the value of this field. Recommended value: 25</td>
</tr>
<tr>
<td>IX_REORG_INSDEL</td>
<td>Triggers a reorganization when the value of ( \left( \text{REORGINSERTS} + \text{REORGDELETES} \right) \times 100 / \text{TOTALENTRIES} ) exceeds the value of this field. Recommended value: 20</td>
</tr>
<tr>
<td>IX_REORG_PSEUDODEL</td>
<td>Triggers a reorganization when the value of ( \text{REORGPSEUDODELETES} \times 100 / \text{TOTALENTRIES} ) exceeds the value of this field. Recommended value: 10 (5 if data sharing)</td>
</tr>
<tr>
<td>IX_REORG_LEAFLIMIT</td>
<td>Triggers a reorganization when the value of ( \text{REORGLEAFFAR} \times 100 / \text{NATIVE} ) exceeds the value of this field. Recommended value: 10</td>
</tr>
<tr>
<td>IX_REORG_APPENDINS</td>
<td>Triggers a reorganization when the value of ( \text{REORGAPPENDINSERT} \times 100 / \text{TOTALENTRIES} ) exceeds the value of this field. Recommended value: 10</td>
</tr>
<tr>
<td>IX_REORG_NLEVELS</td>
<td>Triggers a reorganization when ( \text{REORGNUMLEVELS} ) exceeds the value of this field. Recommended value: 0</td>
</tr>
<tr>
<td>IX_REORG_EXTENTS</td>
<td>Triggers a reorganization when Extents exceeds the value of this field. Recommended value: 200</td>
</tr>
<tr>
<td>TS_COPY_NONE</td>
<td>Checks for a previous Copy operation. If ( \text{COPYLASTTIME} = \text{null} ) (there is no copy), copy the table space. Default value: N Recommended value: Y</td>
</tr>
</tbody>
</table>
| TS_COPY_DESTRUCT    | Checks for a Reorg operation or a Load operation. If the criterion is set to Y, check for a:  
  - Reorg operation ( \( \text{REORGLASTTIME} > \text{COPYLASTTIME} \) )  
  - Load operation ( \( \text{LOADRLASTTIME} > \text{COPYLASTTIME} \) )  
  Default value: N Recommended value: Y                                                                                                 |
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| TS_COPY_INCREMENTAL  | Passes the table space on for testing against incremental copy RTS criteria, if *both* of the following conditions exist:  
  ■ The criterion is set to Y.  
  ■ The table space fails the RTS criteria for a full copy.  
  Default value: N  
  Recommended value: N  
  **Note:** The initial setting for this field is N. If you use incremental copies, you should review and set all of the incremental copy criteria. |
| TS_COPY_CHK_PENDING  | Passes the table space to preprocessing to check whether the table space has a Copy Pending status, if *both* of the following conditions exist:  
  ■ The criterion is set to Y.  
  ■ The table space fails the RTS criteria for a full copy.  
  Default value: N  
  Recommended value: Y |
| TS_COPY_DAYSSINCE    | Triggers a copy when the value of \((\text{CURRENT DATE} - \text{COPYLASTTIME})\) exceeds the value of this criteron  
  Recommended value: 7 |
| TS_COPY_UPDATED      | Triggers a copy when the value of \(\frac{\text{COPYUPDATEDPAGES} \times 100}{\text{NACTIVE}}\) exceeds the value of this field  
  Recommended value: 20 |
| TS_COPY_CHANGES      | Triggers a copy when the value of \(\frac{\text{COPYCHANGES} \times 100}{\text{TOTALROWS}}\) exceeds the value of this field  
  Recommended value: 10 |
| TS_INCR_DAYSSINCE    | Triggers a copy when the value of \((\text{CURRENT DATE} - \text{COPYLASTTIME})\) exceeds the value of this field  
  Recommended value: null (no value)  
  **Note:** BMC offers this field for installations that prefer periodic incremental copies rather than taking incremental copies if the table space has changed. If this value is set, you should set **TS_INCR_UPDATED** and **TS_INCR_CHANGED** to null. |
| TS_INCR_UPDATED      | Triggers a copy when the value of \(\frac{\text{COPYUPDATEDPAGES} \times 100}{\text{NACTIVE}}\) exceeds the value of this field  
  Recommended value: 1 |
| TS_INCR_CHANGES      | Triggers a copy when the value of \(\frac{\text{COPYCHANGES} \times 100}{\text{TOTALROWS}}\) exceeds the value of this field  
  Recommended value: 1 |
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX_COPY_NONE</td>
<td>Checks for a previous Copy operation. If COPYLASTTIME = null (there is no copy), the product copies the index. Default value: N Recommended value: Y</td>
</tr>
</tbody>
</table>
| IX_COPY_DESTRUCT       | Checks for any of the following:  
  - Reorg operation ( REORGLASTTIME > COPYLASTTIME )  
  - Load operation ( LOADRLASTTIME > COPYLASTTIME )  
  Default value: N  
  Recommended value: N | |
| IX_COPY_CHKPENDING    | If IX_COPY_CHKPENDING is set to Y and the index fails RTS criteria for copy, passes the index to preprocessing to check whether the index is in Copy Pending status  
  Default value: N  
  Recommended value: N | |
| IX_COPY_DAYSSINCE      | Triggers a copy when the value of (CURRENTDATE - (COPYLASTTIME) * 100 / TOTALROWS exceeds the value of this field  
  Recommended value: null | |
| IX_COPY_UPDATED        | Triggers a copy when the value of COPYUPDATEDPAGES * 100 / NACTIVE exceeds the value of this field  
  Recommended value: null | |
| IX_COPY_CHANGES        | Triggers a copy when the value of COPYCHANGES * 100 / TOTALROWS exceeds the value of this field  
  Recommended value: null | |
| IX_COPY_NACTIVE        | Triggers a copy when NACTIVE exceeds the value of this field  
  Recommended value: null | |
| TS_STATS_NONE          | Checks for a previous Stats operation:  
  - If none ( STATSLASTTIME = null), compiles statistics for the table space  
  - If A (all), checks for a previous Stats operation, NGT Reorg, and LOAD REPLACE  
  Default value: N  
  Recommended value: Y | c  |
| TS_STATS_INSDELUPD     | Triggers a statistics compilation when the value of (STATSINSERTS + STATSDELETES + STATSUPDATES) exceeds the value of this field  
  Recommended value: 20 | |
| TS_STATS_INDLUPTOT     | Triggers a statistics compilation when the value of (STATSINSERTS + STATSDELETES + STATSUPDATES) exceeds the value of this field  
  Recommended value: 0 | |
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| TS_STATS_MASSDEL    | Triggers a statistics compilation when the value of STATSMASSDELETE exceeds the value of this field  
Recommended value: 0                                                                                                                                                                                                                                                       |
| IX_STATS_NONE       | Checks for a previous run of NGT Stats. If NO previous run occurred, (STATSLASTTIME = null), NGT Utility Manager compiles statistics for the index.  
Default value: N  
Recommended value: Y  
**Note:** The product considers this criterion along with IX_STATS_INSDELTOT. The product triggers a statistics compilation when either of the following conditions exists:  
- (STATSINSERTS + STATSDELETES) * 100 / TOTALENTRIES exceeds the value of this field  
- STATSINSERTS + STATSDELETES exceeds the value of this field  
**Recommended value:** 20                                                                                                                        |
| IX_STATS_INSDEL     | Triggers a statistics compilation when the value of (STATSINSERTS + STATSDELETES) * 100 / TOTALENTRIES exceeds the value of this field  
**Recommended value:** 20  
**Note:** The product considers this criterion along with IX_STATS_INSDELTOT. The product triggers a statistics compilation when either of the following conditions exists:  
- (STATSINSERTS + STATSDELETES) * 100 / TOTALENTRIES exceeds the value of this field  
- STATSINSERTS + STATSDELETES exceeds the value of this field  
**Recommended value:** 20                                                                                                                        |
| IX_STATS_INSDELTOT  | Triggers a statistics compilation when the value of (STATSINSERTS + STATSDELETES) * 100 / TOTALENTRIES exceeds the value of this field  
**Recommended value:** 0  
**Note:** The product considers this criterion along with IX_STATS_INSDEL. The product triggers a statistics compilation when either of the following conditions exists:  
- (STATSINSERTS + STATSDELETES) * 100 / TOTALENTRIES exceeds the value of this field  
- STATSINSERTS + STATSDELETES exceeds the value of this field.                                                                                   |
| IX_STATS_MASSDEL    | Triggers a statistics compilation when the value of STATSMASSDELETE exceeds the value of this field  
**Recommended value:** 0  
**TS_REORG_WHERE** | Indicates the WHERE clause for table space reorganization processing  
**IX_REORG_WHERE** | Indicates the WHERE clause for index reorganization processing  
**TS_COPY_WHERE** | Indicates the WHERE clause for table space copy processing  
**TS_INCR_WHERE** | Indicates the WHERE clause for incremental table space copy processing  
**IX_COPY_WHERE** | Indicates the WHERE clause for index copy processing  
**TS_STATS_WHERE** | WHERE clause for table space statistics processing  |

Chapter 3 NGT Utility Manager table-column reference 37
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX_STATS_WHERE</td>
<td>Indicates the WHERE clause for index statistics processing</td>
</tr>
<tr>
<td>LAST_UPD_USERID</td>
<td>Displays the ID of the last user to update this row</td>
</tr>
<tr>
<td></td>
<td>Default value: userID</td>
</tr>
<tr>
<td>LAST_UPD_TIMESTAMP</td>
<td>Displays the TIMESTAMP of the last update to this row</td>
</tr>
<tr>
<td></td>
<td>Default value: currentTimeStamp</td>
</tr>
<tr>
<td>TS_REORG_CHKAREO</td>
<td>Triggers reorganization of the table space if the object has an Advisory Reorg Status</td>
</tr>
<tr>
<td></td>
<td>The utility performs the criterion check after NGT Utility Manager processing is completed.</td>
</tr>
<tr>
<td></td>
<td>Default value: N</td>
</tr>
<tr>
<td></td>
<td>Recommended value: Y</td>
</tr>
<tr>
<td>TS_REORG_INSERTSMIN</td>
<td>If this criterion is not null, performs an additional check after processing the insert percentage criterion. If the object meets the percentage criteria, the object must also meet this minimum number of inserts criterion.</td>
</tr>
<tr>
<td></td>
<td>Use this criterion to prevent reorganization of a very small table that triggers a criterion due to the low number of inserts.</td>
</tr>
<tr>
<td></td>
<td>Recommended value: 100000</td>
</tr>
<tr>
<td>TS_REORG_DELETESMIN</td>
<td>If this criterion is not null, performs an additional check after processing the delete percentage criterion. If the object meets the percentage criteria, the object must also meet this minimum number of deletes criterion.</td>
</tr>
<tr>
<td></td>
<td>Use this criterion to prevent reorganization of a very small table that triggers a criterion due to the low number of deletes.</td>
</tr>
<tr>
<td></td>
<td>Recommended value: 100000</td>
</tr>
<tr>
<td>TS_REORG_UPDATESMIN</td>
<td>If this criterion is not null, performs an additional check after processing the update percentage criterion. If the object meets the percentage criterion, the object must also meet this minimum number of updates criteria.</td>
</tr>
<tr>
<td></td>
<td>Use this criterion to prevent reorganization of a very small table that triggers a criterion due to the low number of updates.</td>
</tr>
<tr>
<td></td>
<td>Recommended value: 100000</td>
</tr>
<tr>
<td>TS_REORG_INSDELUPDMIN</td>
<td>If this criterion is not null, performs an additional check after processing the update percentage criterion. If the object meets the percentage criterion, the object must also meet this minimum number of updates criteria.</td>
</tr>
<tr>
<td></td>
<td>Use this criterion to prevent reorganization of a very small table that triggers a criterion due to the low number of updates.</td>
</tr>
<tr>
<td></td>
<td>Recommended value: 100000</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TS_REORG_ALLOUSED</td>
<td>If this criterion is not null, performs an additional check after processing the InsDelUpd percentage criterion. If the object meets the percentage criterion, the object must also meet this minimum size criterion. Use this criterion to prevent reorganization of a very small table that triggers a criterion due to the low number of updates. Recommended value: 2</td>
</tr>
<tr>
<td>TS_REORG_MAXSIZE</td>
<td>Triggers a reorganization of the table space if ( \text{Data set allocated size} \times 100 / \text{Maximum data set size} ) exceeds the value of this field. Use this criterion to trigger action or notification when an object is approaching its maximum size. Recommended value: NULL</td>
</tr>
</tbody>
</table>
| TS_REORG_HASHOVERFLOW  | Triggers a reorganization of the table space when both of these conditions exist:  
  - The overflow index for hash access is used.  
  - The overflow index \( \text{TOTALENTRY} \times 100 / \text{TOTALROWS} \) exceeds the value of this field.
Use this criterion to trigger a reorganization based on the percentage of rows that are overflow entries. Recommended value: NULL |
<p>| IX_REORG_CHKAREO       | Triggers reorganization of the index if the object has Advisory Reorg Status. The utility performs the criterion check after NGT Utility Manager processing is completed. Default value: N Recommended value: Y |
| IX_REORG_INSERTSMIN    | If this criterion is not null, performs an additional check after processing the insert percentage criterion. If the object meets the percentage criterion, the object must also meet this minimum number of inserts criterion. This prevents processing of very small objects. Recommended value: 100000 |
| IX_REORG_DELETESMIN    | If this criterion is not null, performs an additional check after processing the delete percentage criterion. If the object meets the percentage criterion, the object must also meet this minimum number of deletes criterion. This prevents processing of very small objects. Recommended value: 100000 |
| IX_REORG_INSDELMIN     | If this criterion is not null, performs an additional check after processing the InsDel percentage criterion. If the object meets the percentage criterion, the object must also meet this minimum number of changes criterion. This prevents processing of very small objects. Recommended value: 100000 |</p>
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| IX_REORG_EMPTY         | Triggers reorganization of the index if \( NPAGES > 5 \text{ AND } NPAGES \times 100 / \text{NLEAF} > \text{IX_REORG_EMPTY} \)
                          | This field triggers reorganization to reduce the number of pseudo-empty leaf pages. Pseudo deleted entries are entries that are marked deleted but are not physically deleted.
                          | Recommended value: 10                                                                                                                                  |
| IX_REORG_MAXSIZE       | Triggers reorganization of the index if \( \text{Data set Allocated size} \times 100 / \text{Maximum data set size} \) exceeds the value of this field
                          | Use this criterion to trigger action or notification based on the size of the object approaching its maximum size.
                          | Recommended value: NULL                                                                                                                                |
| TS_COPY_UPDATEDMIN     | If this criterion is not null, performs an additional check after processing the updated percentage criterion. If the object meets the updated percentage criterion, the object must also meet this minimum number of updates criteria. This prevents copying of objects with few updated pages.
                          | Recommended value: 0                                                                                                                                    |
| TS_COPY_CHANGESMIN     | If this criterion is not null, performs an additional check after processing the changes percentage criterion. If the object meets the updated percentage criterion, the object must also meet this minimum number of changes criteria. This prevents copying of objects with few changes.
                          | Recommended value: 0                                                                                                                                    |
| TS_INCR_UPDATEDMIN     | If this criterion is not null, performs an additional check after processing the updated percentage criterion. If the object meets the updated percentage criterion, the object must also meet this minimum number of updated pages criteria.
                          | Recommended value: 0                                                                                                                                    |
| TS_INCR_CHANGESMIN     | If this criterion is not null, performs an additional check after processing the changes percentage criterion. If the object meets the updated percentage criterion, the object must also meet this minimum number of changes criteria.
                          | Recommended value: 0                                                                                                                                    |
| IX_COPY_UPDATEDMIN     | If this criterion is not null, performs an additional check after processing the updated percentage criterion. If the object meets the updated percentage criterion, the object must also meet this minimum number of updates criteria.
<pre><code>                      | Recommended value: 0                                                                                                                                    |
</code></pre>
<p>| VERSION                | Default value: 12.1.00                                                                                                                                |</p>
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>This field should be set to Y when your IBM RTS tables are populated. BMC set the criterion to N to prevent a high number of Reorg jobs initially, in case the NGT Utility Manager tables are not yet populated.</td>
</tr>
<tr>
<td>b</td>
<td>This field should be set to Y when your IBM RTS tables are populated. BMC set the criterion to N to prevent an initial flood of Copy jobs, in case the NGT Utility Manager tables are not yet populated.</td>
</tr>
<tr>
<td>c</td>
<td>This field should be set to Y when your IBM RTS tables are populated. BMC set the criterion to N to prevent an initial flood of Stats jobs, in case the NGT Utility Manager tables are not yet populated.</td>
</tr>
</tbody>
</table>
| d      | For segmented multi-data-set table spaces:  
- IBM reports the extents of the last data set.  
- NGT Utility Manager reports the total extents of all data sets.  
Consequently, you might want to adjust this field. Alternatively, specify NULL to omit the criterion.  
For partition-by-growth universal table spaces (UTSs), **EXTENTS** are always significant. |
| e      | BMC does not recommend copying indexes. Initially, NGT Utility Manager sets all index copy criteria to N or null. If your installation copies indexes, you must review and manually set all of the index copy criteria. |
| f      | Perform these actions (in the order listed) to view space allocated to an object as a percentage of its maximum possible size:  
- For normal reorganization criteria, set this field to NULL.  
- Create an NGT Utility Manager **TYPE=AP** record with **NAME=MAXSIZE** that checks only this criterion and the previous **ALLOWUSED** criterion.  
- Disable the other criteria by setting them to N, 0, or NULL.  
- Run NGT Reorg **databaseName.% RTS(MAXSIZE,REPORTONLY)** on any database with an object that might reach the maximum size.  
Perform this procedure regularly and review the return codes of objects approaching their maximum sizes. |
| g      | This criterion incorporates NGT functionality. It is not derived from IBM RTS. |
| h      | For more information, see “WHERE clause for NGT Utility Manager” on page 57. |

### Examples of using table space and index criteria

Assume that the IBM RTS tables show base thresholds that would typically warrant reorganizing a table space and its index. Using NGT Utility Manager, you could automatically apply criteria similar to the following examples.

#### Sample of table space criteria

- **NACTIVE > 5**
- REORGMASSDELETE > 0
- (REORGINSERTS * 100) / TOTALROWS > 25 AND REORGINSERTS > 0
- (REORGDELETES * 100) / TOTALROWS > 25 AND REORGINSERTS > 0
- (REORGINSERTS + REORGDELETES + REORGUPDATES) * 100) / TOTALROWS > 20 AND (REORGINSERTS + REORGDELETES + REORGUPDATES > 0
- REORGNEARINDREF + REORGFARINDREF * 100 / TOTALROWS > 10
- REORGUNCLUSTINS * 100 / TOTALROWS > 10
- REORGDISORGLOB * 100 / TOTALROWS > 10
- EXTENTS > 50
- SPACE * 1024 / DATASIZE > 2
- TOTALENTRY*100 / TOTALROWS > 15

**Sample of index criteria**

- REORGMASSDELETE > 0
- REORGINSERTS * 100 / TOTALENTRIES > 25
- REORGDELETES * 100 / TOTALENTRIES > 25
- (REORGINSERTS + REORGDELETES)*100 / TOTALENTRIES > 20
- REORGPSUEDODELETES * 100 / TOTALENTRIES > 10
- REORGLEAFFAR * 100 / NACTIVE > 10
- REOR GAPPENDINSERT * 100 / TOTALENTRIES > 10
- REORGNUMLEVELS > 0 (levels added since last Reorg)
- EXTENTS > 50
- NPAGES > 5 AND NPAGES * 100 / NLEAF > 10

**Scenarios for applying object-specific criteria**

Although the criteria indicating disorganization are usually consistent across objects, some table spaces require customized thresholds. You can adjust reorganization frequency thresholds for specific table spaces. The following scenarios describe some of these cases.

**Hot spots**

If the majority of inserts to a table space are in specific locations, these "hot spots" become unclustered when their free space is exhausted. Even if RTS reports 5 percent (that is, below the threshold) unclustered inserts for the entire table space, exhaustion of free space in hot spots might affect readers. If access to the object is both random and sequential, the object might benefit from a lower overall unclustered threshold.
For table spaces with hot spots and sequential access, insert an object-specific record with a lower TS_REORG_UNCLUST threshold.

**Note**
Since IBM DB2 Version 10 introduced UNCLUSTERSENS, the cluster criterion takes the access method (random or sequential) into account. This simplifies the required criteria.

**Preventing reorganization: Example 1**

If rows are inserted and deleted randomly throughout a table space, the table space is less likely to require reorganization.

**Preventing reorganization: Example 2**

If a minimum criterion is not null and too few rows were deleted, the product might override the deletion criteria. Use the minimum criterion to prevent a reorganization if the following conditions apply:

- Only a few records were deleted.
- These records are a significant percentage of a very small table.

In this case, insert an object-specific record with insertion and deletion criteria thresholds set to NULL. The product ignores the table space.

The default values for insertion and deletion criteria trigger a reorganization at 25 percent of the table space. The default values for the minimum criteria are set to 100,000. Consequently, the product triggers a reorganization when more than 100,000 rows are affected. This implies a table space with 400,000 rows. (Although 400,000 sounds like a large number, 100,000 rows typically fill only about 8 cylinders.)

**Ensuring reorganization**

If all of the following conditions apply, the conditions maintain the cluster. However, you might need to reclaim space that the deletes released.

- A table space has an ascending key.
- Inserts are always ascending.
- You are doing deletes primarily on the oldest keys.

In this case, depending on the size of the table space, you might want to adjust the standard criteria to ensure a reorganization. Insert a row in the NGT Utility Manager Criteria table and add the appropriate criteria and thresholds.
Over-allocation

You might deliberately over-allocate objects because of anticipated growth. Specify the `TS_REORG_ALLOUSED` criterion \((\text{SPACE} \times 1024) / \text{DATASIZE} > \text{TS_REORG_ALLOUSED})\) only on objects where this is a consideration.

If you use NGT embedded automation (XSUTALLO) to allocate a DB2 data set of the correct size, you should not need this criterion.

Custom criteria and thresholds

Use the `WHERE` clause to specify custom thresholds on one, several, or all criteria records. The following examples show how you can select or exclude an object for processing by using the `WHERE` clause.

Example 1

To perform the following operation, update the `TS_REORG_WHERE` statement for all of your criteria records:
- Skip reorganization for objects with fewer than 100 pages.
- Reorganize objects smaller than 100 cylinders that are also in extents.

The following `WHERE` clause performs this operation:

```
DO_WHERE (A.NACTIVE < 18000 AND A.EXTENTS > 1)
OR SKIP_WHERE (A.NACTIVE < 100)
```

Example 2

To trigger an index for reorganization based on both RTS (`REORGLEAFNEAR` in RTS) criteria and the standard (`IX_REORG_LEAFLIMIT`) `REORGLEAFFAR` criteria, create a Criteria record for the object. Specify the following `WHERE` clause in the record:

```
DO_WHERE((A.REORGLEAFNEAR * 100) / A.TOTALENTRIES > 30
```

Related Information

- “Overview of the `WHERE` clause” on page 57

NGT Utility Manager Exceptions table

NGT Utility Manager uses the Exceptions table to handle deviations from periodic processing (for example, preventing reorganization during maintenance). Edit the
Exceptions table to define the exception conditions that force or prevent processing of a table space or index.

The following figure displays Exceptions table column definitions.

**Figure 3: Exceptions table column definitions**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Col No</th>
<th>Col Type</th>
<th>Length</th>
<th>Scale</th>
<th>Null</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>1</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>2</td>
<td>VARCHAR</td>
<td>257</td>
<td>0</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>PARTITION</td>
<td>3</td>
<td>SMALLINT</td>
<td>2</td>
<td>0</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>REORG</td>
<td>4</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>I</td>
</tr>
<tr>
<td>COPY</td>
<td>5</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>I</td>
</tr>
<tr>
<td>STATS</td>
<td>6</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>I</td>
</tr>
<tr>
<td>REMARKS</td>
<td>7</td>
<td>VARCHAR</td>
<td>128</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>REORG_WHERE</td>
<td>8</td>
<td>VARCHAR</td>
<td>512</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>COPY_WHERE</td>
<td>9</td>
<td>VARCHAR</td>
<td>512</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>STATS_WHERE</td>
<td>10</td>
<td>VARCHAR</td>
<td>512</td>
<td>0</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>LAST_UPD_USERID</td>
<td>11</td>
<td>CHAR</td>
<td>8</td>
<td>0</td>
<td>N</td>
<td>User</td>
</tr>
<tr>
<td>LAST_UPD_TIMESTAMP</td>
<td>12</td>
<td>TIMESTAMP</td>
<td>10</td>
<td>0</td>
<td>N</td>
<td>Timestamp</td>
</tr>
<tr>
<td>RULE_TYPE</td>
<td>13</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>X</td>
</tr>
<tr>
<td>VERSION</td>
<td>14</td>
<td>CHAR</td>
<td>8</td>
<td>0</td>
<td>N</td>
<td>12.1.00</td>
</tr>
</tbody>
</table>

Member EXCAPP in the CNTL data set contains sample SQL that you can add to an Exceptions table row to skip reorganization of a table space. You might want to do this to prevent reorganization of the table space while another event is running. You can invoke member EXCAPP instead of overriding an existing job schedule.

Table 1 on page 28 describes each column that is listed under **Column Name** in the Exceptions table.

**WARNING**

When the product processes Exceptions records, specific conditions override general or wildcard conditions.

NGT Utility Manager sorts Exceptions records according to their specificity and honors the first Exceptions record that it encounters. If you defined multiple records for the same object, you cannot know which Exceptions record the product honored and which it ignored.

Table 3: Column names in the Exceptions table

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| TYPE     | Type of RTS record. Valid values are:  
|          | - AP—Application  
|          | - TS—Table space  
<p>|          | - IX—Index |</p>
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>Name of the RTS record. You can use wildcards in the name. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- Application—128 characters maximum</td>
</tr>
<tr>
<td></td>
<td>- Table space—17 characters maximum, in format <code>databaseName.tableSpaceName</code></td>
</tr>
<tr>
<td></td>
<td>- Index—257 characters maximum, in format <code>creator.indexName</code></td>
</tr>
<tr>
<td>PARTITION</td>
<td>Partition number. 0 (zero) signifies all partitions. Default value: 0</td>
</tr>
<tr>
<td>REORG</td>
<td>Forces or excludes a reorganization by NGT Reorg. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- F—Forces a reorganization</td>
</tr>
<tr>
<td></td>
<td>- X—Excludes a reorganization</td>
</tr>
<tr>
<td></td>
<td>- I—Ignors (not a reorganization Exceptions record)</td>
</tr>
<tr>
<td></td>
<td>Default value: I</td>
</tr>
<tr>
<td>COPY</td>
<td>Forces or excludes copying by NGT Copy. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- F—Forces copying</td>
</tr>
<tr>
<td></td>
<td>- X—Excludes copying</td>
</tr>
<tr>
<td></td>
<td>- I—Ignors (not a copy Exceptions record)</td>
</tr>
<tr>
<td></td>
<td>Default value: I</td>
</tr>
<tr>
<td>STATS</td>
<td>Forces or excludes statistics collection by NGT Stats. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- F—Forces statistics collection</td>
</tr>
<tr>
<td></td>
<td>- X—Excludes statistics collection</td>
</tr>
<tr>
<td></td>
<td>- I—Ignors (not a statistics Exceptions record)</td>
</tr>
<tr>
<td></td>
<td>Default value: I</td>
</tr>
<tr>
<td>REMARKS</td>
<td>Comments for an Exception entry</td>
</tr>
<tr>
<td>REORG_WHERE</td>
<td>WHERE clause for NGT Reorg table space processing</td>
</tr>
<tr>
<td>COPY_WHERE</td>
<td>WHERE clause for NGT Copy table space processing</td>
</tr>
<tr>
<td>STATS_WHERE</td>
<td>WHERE clause for NGT Stats table space processing</td>
</tr>
<tr>
<td>LAST_UPD_USERID</td>
<td>USERID of the last user to update the row</td>
</tr>
<tr>
<td></td>
<td>Default value: <code>userID</code></td>
</tr>
<tr>
<td>LAST_UPD_TIMESTAMP</td>
<td>Time stamp of the most recent update to the row.</td>
</tr>
<tr>
<td></td>
<td>Default value: <code>currentTimeStamp</code></td>
</tr>
<tr>
<td>RULE_TYPE</td>
<td>Forces or excludes object processing. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- F—Forces object processing by the checked utilities</td>
</tr>
<tr>
<td></td>
<td>- X—Excludes object processing by the checked utilities</td>
</tr>
<tr>
<td></td>
<td>- I—Ignors the object</td>
</tr>
<tr>
<td></td>
<td>Default value: X</td>
</tr>
<tr>
<td>VERSION</td>
<td>Default value: 12.1.00</td>
</tr>
</tbody>
</table>
Related Information

- “Overview of the WHERE clause” on page 57
NGT Utility Manager syntax reference

This chapter describes the syntax for running NGT Utility Manager evaluations, and for managing the Criteria, Schedule, and Exceptions tables.

Keywords for running NGT Utility Manager evaluations

This topic explains the keywords that you use in a SYSIN DD statement to request that NGT Utility Manager evaluate objects for processing.

\[ \text{ngtUtilityStatement} \rightarrow \text{RTS} \]

\[ (\text{app}, \text{GENERATESYSIN}) \]

\[ (\text{app}, \text{GENERATESYSINALL}) \]

\[ (\text{app}, \text{REPORTONLY}) \]

\[ \text{ngtUtilityStatement} \]

Specifies the utility for which you want to evaluate objects (for example, REORG to run an evaluation for NGT Reorg, or COPY for NGT Copy)

\[ \text{RTS} \]

Triggers NGT Utility Manager to check the Exceptions table, Schedule table, and Criteria table to determine whether to run the specified NGT utility
Example

**app**

Specifies the **NAME** field in a **TYPE=AP** criterion record

When specified, NGT Utility Manager processes the object against the named criterion record for application **app**.

**Example**

The following example specifies an application-specific record for a utility run:

```
//SYSIN DD *
RUNSTATS TABLESPACE CENTRAL.ACCTMSTR RTS(ACCOUNTING)
```

**GENERATESYSIN**

Prevents the NGT statement from being executed

NGT Utility Manager expands the object of the statement (which can be a wildcard pattern) and runs it past the Exception and Selection criteria. Then NGT Utility Manager generates a list of objects selected for processing.

Use this keyword if you are not using NGT utilities.

**Example**

This example excludes an object from reorganization while generating Reorg statements that would have been processed during reorganization:

```
//SYSIN DD *
REORG OBJECTSET PROD.ACCOUNTING RTS(.GENERATESYSIN)
```

**Note**

When you use the GENERATESYSIN keyword, all processing takes place in the master job. No server job is used.

**GENERATESYSINALL**

Like GENERATESYSIN, prevents the NGT statement from being executed, but also returns ("passes back") objects excluded from processing with the keyword **PASSED**. Then, NGT Utility Manager uses the XRTSSYIN
automation routine to perform additional checks not available to NGT Utility Manager.

For more information about XRTSSYIN, see “Recommended automation routines for NGT Utility Manager” on page 67.

**REPORTONLY**

Tells NGT Utility Manager to report which objects should be processed, but without actually processing them.

NGT Utility Manager expands the object of the statement (which can be a wildcard pattern) and analyzes it against the Exception and Selection criteria. NGT Utility Manager then generates a report (Figure 4 on page 51) that lists the objects that would have been processed or skipped.

--- **Example** ---

This example generates an NGT Utility Manager report while excluding the objects from processing:

```sql
//SYSIN DD *
REORG OBJECTSET PROD.ACCOUNTING RTS(,REPORTONLY)
```

--- **Figure 4: Sample summary report** ---

```
--- BMC Next Generation Technology V vv.rr.mm
RTS REORG EXCLUDED TABLESPACE REPORT (STATEMENT 1)
NAME   TABLESPACE   PART(S)   REASON
-------  -----------  ---------  --------------------------------
DSN8DCAT ABC         0001       EMPTY_TABLESPACE
DSN8DCAT ABC         0002:0004 PASSED ALL RTS CRITERIA

RTS REORG EXCLUDED INDEX REPORT (STATEMENT 1)
INDEX  NAME       PART(S)   REASON
------  ----------  ---------  -----------------------
DSN8DCAT ABCX1     0001:0004 PASSED ALL RTS CRITERIA
```

--- **Related Information** ---

- “NGT Utility Manager routine ” on page 67
- “Using RTS to determine whether to run an NGT utility using NGT Utility Manager ” on page 65
Keywords for managing Criteria, Schedule, and Exceptions tables

This topic describes the keywords that you use in a SYSIN DD statement if manually defining (registering) the NGT Utility Manager tables. This topic also describes the keyword for displaying your current table definitions.

For more information about when to define tables manually, see “Defining NGT Utility Manager tables to the NGT checkpoint data set” on page 19.

NGT Utility Manager definition modes

The following table describes the NGT Utility Manager definition modes.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPLACE</td>
<td>If the tables and indexes exist, the REPLACE keyword causes NGT Utility Manager to replace them with new tables. Then, NGT Utility Manager removes the old NGT Utility Manager definitions from the NGT checkpoint dataset and replaces them with the new definitions.</td>
</tr>
<tr>
<td>Not specified</td>
<td>If REPLACE is not specified, processing assumes that the NGT Utility Manager definitions are new and do not exist in the NGT checkpoint data set.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If NGT Utility Manager finds a previous definition of the NGT Utility Manager tables, the product issues an error message.</td>
</tr>
</tbody>
</table>

Related Information

- “Defining NGT Utility Manager tables to the NGT checkpoint data set” on page 19

Keywords for manually defining the NGT Utility Manager tables

Following is the syntax for the RTS DEFINE statement. You can use it in a SYSIN DD statement to manually define Criteria, Schedule, and Exceptions tables to the checkpoint:

For more information about when to use the DEFINE statement, see “Defining NGT Utility Manager tables to the NGT checkpoint data set” on page 19.
Syntax for defining tables

Example

```sql
//SYSIN DD *
NGTRTS DEFINE DBNAME(DBRTSCDB) TSNAME(RTSDSN1)
BUFFERPOOL(BP0) STOGROUP(DB2STGRP)
CRITERIA_TABLE(DB2USER.NGT_RTS_CRITERIA)
CRITERIA_INDEX(DB2USER.NGT_RTS_CRITERIA_IX)
EXCEPTIONS_TABLE(DB2USER.NGT_RTS_EXCEPTIONS)
EXCEPTIONS_INDEX(DB2USER.NGT_RTS_EXCEPTIONS_IX)
SCHEDULE_TABLE(DB2USER.NGT_RTS_SCHEDULE)
SCHEDULE_INDEX(DB2USER.NGT_RTS_SCHEDULE_IX)
```

The syntax includes the following keywords (listed alphabetically):

**BUFFERPOOL**

Name of the buffer pool. Use this keyword when defining NGT Utility Manager indexes

**CRITERIA_INDEX**

Name of the index for the NGT Utility Manager Criteria table 
*(creator.indexName)*

**CRITERIA_TABLE**

Name of the NGT Utility Manager Criteria table *(creator.tableName)*

**DBNAME**

Name of the database to contain NGT Utility Manager tables
EXCEPTIONS_INDEX

Name of the index on NGT Utility Manager Exceptions table (creator.name)

EXCEPTIONS_TABLE

Name of the NGT Utility Manager Exceptions table (creator.tableName)

EXCP_DBNAME

(optional) Keyword used to name a separate database for the NGT Utility Manager Exceptions table

EXCP_TSNAME

(optional) Keyword used to name a separate table space for the NGT Utility Manager Exceptions table

SCHEDULE_DBNAME

(optional) Keyword used to name a separate database for the NGT Utility Manager Schedule table

SCHEDULE_TSNAME

(optional) Keyword used to name a separate table space for the NGT Utility Manager Schedule table

STOGROUP

DB2 storage group to be used when defining NGT Utility Manager indexes

TSNAME

Name of the table space to contain NGT Utility Manager tables

Keywords for displaying NGT Utility Manager table definitions

You can use NGTRTS DISPLAY anytime to see which NGT Utility Manager tables are defined in the checkpoint data set. You can display definitions for the current DB2 subsystem, or for all DB2 subsystems in the data set.

The figure below displays the syntax for statements that display NGT Utility Manager tables.
NGTRTS DISPLAY

Displays the table definitions for the current DB2 subsystem in the NGT checkpoint data set

NGTRTS DISPLAY ALL

Displays the table definitions for all DB2 subsystems in the NGT checkpoint data set
WHERE clause for NGT Utility Manager

This chapter describes the NGT Utility Manager WHERE clause and how to use it.

Overview of the WHERE clause

Use WHERE clause processing to add criteria that trigger or prevent processing.

NGT Utility Manager provides two sets of column variables:

- Global (qualifier G)
- DB2 RTS (qualifier A)

WHERE clauses are applicable to Exceptions and Criteria tables. The Exceptions table WHERE clause has only global variables because the product checks the Exceptions table before it checks the criteria.

For SQL statements in which you ordinarily use the IBM MOD function, you must use the operator `//` instead.

---

Example

Valid WHERE clause:
```
WHERE DAYS(CURRENT DATE) // 2 = 1
```

Invalid WHERE clause:
```
WHERE MOD(DAYS(CURRENT DATE), 2) = 1
```

---

For more information, see the chapter that discusses NGT SQL language in the *BMC Next Generation Technology General User Guide*. 
DO_WHERE and SKIP_WHERE functions

NGT Utility Manager provides two functions for working with WHERE clauses: DO_WHERE and SKIP_WHERE. You can use these functions instead of using G.NEEDS_REORG or using AND NOT.

DO_WHERE

If the standard criteria do not select the object, the product checks the DO_WHERE function and honors it if TRUE.

\[
\text{DO\_WHERE}(\text{A.NACTIVE} < 540 \text{ AND EXTENTS} > 1)
\]

SKIP_WHERE

If the standard criteria select the object, the product checks the SKIP_WHERE function and honors it if TRUE.

\[
\text{SKIP\_WHERE}(\text{A.NACTIVE} < 8)
\]

Use of WHERE clauses with the Criteria table

NGT Utility Manager uses standard (Criteria table) criteria together with WHERE clause criteria to exclude or select objects for processing:

- **Order**: The product checks the standard criteria before it checks WHERE clause criteria. If the standard criteria exclude an object, you can use the WHERE clause to select the object for processing. If the standard criteria select an object, you can use the WHERE clause to exclude it.

- **TRUE and FALSE**: When the WHERE clause is TRUE, NGT Utility Manager selects the object for processing. When the WHERE clause is FALSE, NGT Utility Manager excludes the object.

- **Priority**: If the standard criteria exclude the object while the WHERE clause is TRUE, the product selects the object. If the standard criteria select the object while the WHERE clause is FALSE, the product excludes the object.
Use of WHERE clauses with the Exceptions table

NGT Utility Manager uses the Exceptions table together with WHERE clause criteria to exclude or force objects for processing.

When invoking the Exceptions table, the product always checks the WHERE clause first. When the WHERE clause is TRUE, the corresponding product option (Force or Exclude) is honored.

Example

The following example uses DB2 functions to force or exclude a utility on Sunday or Saturday:

```
WHERE DAYOFWEEK(CURRENT_DATE) IN (1,7)
```

The following example uses DB2 functions to force or exclude a utility on a specific date:

```
WHERE CURRENT_TIMESTAMP BETWEEN '2016-02-25-06.00.00.000000'
    AND '2016-02-25-11.00.00.000000'
```

The following example uses global columns to force or exclude a utility if the job name starts with the string PAYR:

```
WHERE G.JOBNAME LIKE 'PAYR%'
```
Column names for use with WHERE clauses

This topic describes the column names that you can use in WHERE clauses.

Global condition names

Use the following global column names (G.column_name) with WHERE clauses:

<table>
<thead>
<tr>
<th>Column name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.ACCOUNT</td>
<td>Job accounting data</td>
</tr>
<tr>
<td>G.DB2_VERSION</td>
<td>DB2 version</td>
</tr>
<tr>
<td>G.JOBNAME</td>
<td>Job name</td>
</tr>
<tr>
<td>G.NEEDS_COPY 'Y'</td>
<td>Whether the criterion specifies that Copy is required</td>
</tr>
<tr>
<td>G.NEEDS_REORG 'Y'</td>
<td>Whether the criterion specifies that Reorg is required</td>
</tr>
<tr>
<td>G.NEEDS_STATS 'Y'</td>
<td>Whether the criterion says that Stats is required</td>
</tr>
<tr>
<td>G.PLAN</td>
<td>Plan name</td>
</tr>
<tr>
<td>G.PROGRAMMER</td>
<td>Job programmer</td>
</tr>
<tr>
<td>G.SSID</td>
<td>DB2 subsystem ID</td>
</tr>
<tr>
<td>G.STEPNAME</td>
<td>Step name</td>
</tr>
<tr>
<td>G.UID</td>
<td>Utility ID</td>
</tr>
</tbody>
</table>

NGT Utility Manager columns for table spaces

To create criteria when you use the WHERE clause, use the following NGT Utility Manager column names (A.column.name) with table spaces:

- A.COPYCHANGES
- A.COPYLASTTIME
- A.COPYUPDATEDPAGES
- A.COPYUPDATELRSN
- A.COPYUPDATETIME
- A.DATASIZE
- A.DBID
- A.DBNAME

Note

The DB2 RTS tables use these column names. For more information about DB2 RTS tables and columns, see the IBM documentation.
A.DRIVETYPE
A.EXTENTS
A.GETPAGES
A.HASHLASTUSED
A.IBMREQD
A.INSTANCE
A.LOADRLASTTIME
A.LPFACILITY
A.NACTIVE
A.NAME
A.NPAGES
A.PARTITION
A.PSID
A.REORGCLUSTERSENS
A.REORGDELETES
A.REORGDISORGLOB
A.REORGFARINDREF
A.REORGHASHACCESS
A.REORGINSERTS
A.REORGLASTTIME
A.REORGMASSDELETE
A.REORGNEARINDREF
A.REORGSIGSCANACCESS
A.REORGSIGSCANACCESS
A.REORGUNCLUSTINS
A.REORGUPDATES
A.SPACES
A.STATS01
A.STATSDELETE
A.STATSINSERTS
A.STATSINSERTS
A.STATSLASTTIME
A.STATSMASSDELETE
A.STATSMUPDATE
A.SYS_START
A.SYS_END
A.TOTALROWS
A.TRANS_START
A.UNCOMPRESSEDDATASIZE
A.UPDATESTATSTIME
NGT Utility Manager columns for indexes

To create criteria when you use the WHERE clause, use the following NGT Utility Manager column names \( (A.column.name) \) with indexes:

**Note**
The DB2 RTS tables use these column names. For more information about DB2 RTS tables and columns, see the IBM documentation.

- A.COPYCHANGES
- A.COPYLASTTIME
- A.COPYUPDATERLSN
- A.COPYUPDATERSN
- A.COPYUPDATERSN
- A.COPYUPDATERSN
- A.CREATOR
- A.DBID
- A.DBNAME
- A.DRIVETYPE
- A.EXTENTS
- A.GETPAGES
- A.IBMREQD
- A.INDEXSPACE
- A.INSTANCE
- A.ISOBID
- A.LASTUSED
- A.LOADRLASTTIME
- A.NACTIVE
- A.NAME
- A.NLEAF
- A.NLEVELS
- A.NPAGES
- A.PARTITION
- A.PSID
- A.REBUILDLASTTIME
- A.REORGAPPENDINSERT
- A.REORGDELETES
- A.REORGINDEXACCESS
- A.REORGINSETS
- A.REORGLASTTIME
- A.REORGLEAFFAR
- A.REORGLEAFNEAR
- A.REORGMASSDELETE
- A.REORGNUMLEVELS
- A.REORGPSUDEDELETE
- A.SPACE
Column names for use with WHERE clauses

A.STATS101
A.STATSDELETES
A.STATSINSERTS
A.STATSLASTTIME
A.STATSMASSDELETE
A.SYS_START
A.SYS_END
A.TOTALENTRIES
A.TRANS_START
A.UPDATESTATSTIME
Syntax examples for NGT Utility Manager

This chapter provides examples of tasks that you can perform by using NGT Utility Manager.

Using RTS to determine whether to run an NGT utility using NGT Utility Manager

To use RTS to determine whether to run an NGT utility, include the RTS keyword in the statement.

--- Example ---

```sql
//SYSIN DD *
REORG TABLESPACE WORKDB.INVENTRY RTS

//SYSIN DD *
COPY TABLESPACE CENTRAL.ACCTGREC RTS

//SYSIN DD *
RUNSTATS TABLESPACE CENTRAL.ACCT% RTS
```

--- Related Information ---

- “Keywords for running NGT Utility Manager evaluations” on page 49

Specifying an application-specific record

You can use the RTS keyword to specify an application-specific record for a utility run.
Generating SYSIN for NGT Reorg

To exclude an object from reorganization while generating Reorg statements that would have been processed during reorganization, use the following statement:

```
//SYSIN DD *
REORG OBJECTSET PROD.ACCOUNTING RTS(.GENERATESYSIN)
```

Generating an NGT Utility Manager report

To generate an NGT Utility Manager report while excluding the target objects from processing, use the following statement:

```
//SYSIN DD *
REORG OBJECTSET PROD.ACCOUNTING RTS(.REPORTONLY)
```

Related Information

- “NGT Utility Manager reports” on page 69
Recommended automation routines for NGT Utility Manager

BMC Next Generation Technology automation (NGT) automation routines are integrated into NGT utilities. You can incorporate automation routines into your processing.

The NGTAUTO DD statement triggers automation routine processing.

---

**Example**

```bash
//NGTAUTO DD DISP=SHR,DSN=CDB.AUTO.EXITS
```

---

The automation routine data set contains one member. The member contains the routines that you select. To use NGT automation routines, include the RTS keyword with the relevant NGT utilities (NGT Reorg, NGT Copy, or NGT Stats).

In addition to the automation control points that are available with these products, NGT Utility Manager includes the NGT Utility Manager routine. For more information, see “NGT Utility Manager routine” on page 67.

---

NGT Utility Manager routine

When you include RTS(GENERATESYSIN), control point XRTSSYIN contains the list of statements that match your RTS criteria. You can use this information to generate:

- What-if lists of objects that would be processed
- Utility statements
- Jobs that process the objects by using another vendor’s utilities
Example
The following example displays the original statement, original statement number, and generated statements:

```-rexx
/* REXX */
Say RTS_STATEMENT
Say STMT
Do I = 1 to STATEMENTS.0
   Say STATEMENTS.I
End
```

For more information about automation routine XRTSSYIN, see the *BMC Next Generation Technology Automation Reference Manual*.

Related Information

- “Keywords for running NGT Utility Manager evaluations” on page 49
NGT Utility Manager reports

This chapter describes the reports that NGT Utility Manager generates.

Summary reports

NGT Utility Manager generates reports that provide the following information:
- Table spaces selected for processing
- Table spaces excluded from processing
- Indexes selected for processing
- Indexes excluded from processing

Examples of Reorg table space reports

The following report examples are for one two-part table space, where the product selects the second partition of the table space and the second partition of a partitioned index.

Reorg table space report

The following example shows a reorg table space report:

```
RTS_REORG_TABLESPACE REPORT (STATEMENT 1)
DATABASE  NAME   TABLESPACE NAME   PART(S)   SELECTION REASON
----------- ------- --------- ---------- ----------
DBASE001   TSPACE41  0002       INSERTS_PERCENTAGE
```

The following criteria selections could have generated this report:
- MASS_DELETE_VALUE
- INSERTS_PERCENTAGE
- DELETES_PERCENTAGE
- UPDATES_PERCENTAGE
- INSERT/DELETE/UPDATE_PERCENTAGE
- INDREF_PERCENTAGE
Reorg excluded table space report

The following example shows a reorg excluded table space report:

```
RTS REORG EXCLUDED TABLESPACE REPORT (STATEMENT 1)

DATABASE  TABLESPACE  EXCLUDE
NAME       NAME       PART(S)   REASON
----------- ---------- --------- -----------------------
DBASE001   TSPACE41   0001      PASSED ALL RTS CRITERIA
```

Examples of Reorg index reports

The following report examples are for one two-part table space, where the product selects the second partition of the table space and the second partition of a partitioned index.

Reorg index report

The following example shows a reorg index report:

```
RTS REORG INDEX REPORT (STATEMENT 1)

INDEX           INDEX                     SELECTION
CREATOR         NAME    PART(S)   REASON
--------------- --------------- --------- -------------------
PRODDBA         CO41IX03        0001      INSERTS_PERCENTAGE
```

The following criteria selections could have generated this report:

- MASS_DELETE_VALUE
- INSERTS_PERCENTAGE
- DELETES_PERCENTAGE
- INSERT/DELETE_PERCENTAGE
- PSEUDO-DELETED_PERCENTAGE
- LEAF_LIMIT_PERCENTAGE
- APPENDED_INSERTS_PERCENTAGE
- NUMBER_OF_LEVELS_VALUE
- EXTENTS_VALUE
- NO DB2 RTS VALUES FOR OBJECT
- AREO
Reorg excluded index report

The following example shows a reorg excluded index report:

<table>
<thead>
<tr>
<th>INDEX</th>
<th>INDEX</th>
<th>EXCLUDE</th>
<th>REASON</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODDBA</td>
<td>C041IX01</td>
<td>0001</td>
<td>PASSED ALL RTS CRITERIA</td>
</tr>
</tbody>
</table>

Journal messages

The following journal messages display the criteria details for the reports.

If you used WHERE clauses or Exceptions records, the journal messages display information about them, also. The product reports the details in the journal only.

**Note**

To display the journal, take one of the following actions:

- Add `//JOURNAL DD DUMMY` to the NGTUTIL JCL. The journal is sent to `//CDBPRINT DD`).
- Select `+JOURNAL(YES)` in `//UTLPARMS` The journal is sent to `//JRNLOUT DD`).

Figure 5: Journal messages

NGTZ315 RTS TABLESPACESTATS TABLE SUCCESSFULLY READ (17,966 ROWS FOUND).
NGTZ307 RTS CRITERIA DEFINITION SUCCESSFULLY READ:
NGTZ307 TB=NGTUSER.NGT_RTS_CRITERIA
NGTZ232 RTS CRITERIA TABLE SUCCESSFULLY READ (4 ROWS FOUND).
NGTZ234 RTS EXCEPTION DEFINITION SUCCESSFULLY READ:
NGTZ234 TB=NGTUSER.NGT_RTS_EXCEPTIONS
NGTZ235 RTS CRITERIA TABLE DOES NOT EXIST OR IS EMPTY.
NGTZ236 OBJECT SELECTED FOR RTS PROCESSING:
NGTZ236 TS=DBASE001.TSPACE41
NGTZ242 RTS CRITERIA TABLE ENTRY (TYPE TS) FOUND FOR OBJECT:
NGTZ242 NAME=%, %
NGTZ242 PARTITION=0
NGTZ244 RTS VALUES FOUND FOR OBJECT (PART 1). RTS TIMESTAMP = 2014-03-24-10.09.42.101218.
NGTZ253 RTS REORG TS CRITERIA (LAST_PERFORMED_REORG): NOT CHECKED.
NGTZ253 RTS REORG TS CRITERIA (LAST_PERFORMED_LOAD): NOT CHECKED.
NGTZ253 RTS REORG TS CRITERIA (EMPTY_TABLESPACE): NOT EMPTY (500 ROWS).
NGTZ253 RTS REORG TS CRITERIA (MAXIMUM_SIZE_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (10 %).
NGTZ253 RTS REORG TS CRITERIA (INSERTS_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (25 %).
NGTZ253 RTS REORG TS CRITERIA (DELETES_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (25 %).
NGTZ253 RTS REORG TS CRITERIA (UPDATES_PERCENTAGE): NOT CHECKED.
NGTZ253 RTS REORG TS CRITERIA (INSERT/DELETE/UPDATE_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (20 %).
NGTZ253 RTS REORG TS CRITERIA (CLUSTER SENSITIVITY): <= 0
NGTZ253 RTS REORG TS CRITERIA (DISORGED_LB_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (10 %).
NGTZ253 RTS REORG TS CRITERIA (INOREP_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (10 %).
NGTZ253 RTS REORG TS CRITERIA (MASS_DELETE_VALUE): VALUE (0) <= CRITERIA (100).
NGTZ253 RTS REORG TS CRITERIA (EXTENTS_VALUE): VALUE (1) <= CRITERIA (50).
NGTZ253 RTS REORG TS CRITERIA (ALLOCATED/USED_RATIO): NOT CHECKED.
NGTZ253 RTS REORG TS CRITERIA (HASH_OVERFLOW_RATIO): NOT CHECKED.
NGTZ253 RTS REORG TS CRITERIA (WHERE_CLAUSE): NOT CHECKED.
NGTZ245 OBJECT (PART 1) WILL NOT BE PROCESSED BY REORG: PASSED ALL RTS CRITERIA.
NGTZ269 CHKINDEX=Y SPECIFIED. INDEXES WILL BE CHECKED AGAINST RTS CRITERIA.
NGTZ316 RTS INDEXSPACESTATS TABLE SUCCESSFULLY READ (17,614 ROWS FOUND).
NGTZ221 2 INDEXES ADDED FOR RTS CRITERIA PROCESSING.
NGTZ311 FORCECOPY=N SPECIFIED. COPIES WILL NOT BE FORCED.

Chapter 8  NGT Utility Manager reports  71
NGTZ244 RTS VALUES FOUND FOR OBJECT (PART 2). RTS TIMESTAMP = 2014-03-24-10.32.41.072851.
NGTZ253 RTS REORG TS CRITERIA (LAST_PERFORMED_REORG): NOT CHECKED.
NGTZ253 RTS REORG TS CRITERIA (LAST_PERFORMED_LOAD): NOT CHECKED.
NGTZ253 RTS REORG TS CRITERIA (EMPTY_TABLESPACE): NOT EMPTY (500 rows).
NGTZ253 RTS REORG TS CRITERIA (MAXIMUM_SIZE_PERCENTAGE): NOT CHECKED.
NGTZ253 RTS REORG TS CRITERIA (INSERTS_PERCENTAGE): PERCENTAGE (60 %) > CRITERIA (25 %).
NGTZ248 OBJECT (PART 2) WILL BE PROCESSED BY REORG: RTS CRITERIA SATISFIED.
NGTZ235 OBJECT SELECTED FOR RTS PROCESSING:
NGTZ235 IX=PRODDBA.C041IX01
NGTZ242 RTS CRITERIA TABLE ENTRY (TYPE IX) FOUND FOR OBJECT:
NGTZ242 PARTITION=0
NGTZ244 RTS VALUES FOUND FOR OBJECT (PART 1). RTS TIMESTAMP = 2014-03-24-10.09.40.574555.
NGTZ253 RTS REORG IX CRITERIA (LAST_PERFORMED_REORG): NOT CHECKED.
NGTZ253 RTS REORG IX CRITERIA (LAST_PERFORMED_REBUILD): NOT CHECKED.
NGTZ253 RTS REORG IX CRITERIA (INSERTS_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (25 %).
NGTZ253 RTS REORG IX CRITERIA (DELETES_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (25 %).
NGTZ253 RTS REORG IX CRITERIA (APPENDED_INSERTS_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (10 %).
NGTZ253 RTS REORG IX CRITERIA (DELETED_INSERTS_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (10 %).
NGTZ253 RTS REORG IX CRITERIA (LEAF_LIMIT_PERCENTAGE): PERCENTAGE (0 %) <= CRITERIA (10 %).
NGTZ253 RTS REORG IX CRITERIA (NUMBER_OF_LEVELS_VALUE): VALUE (0) <= CRITERIA (0).
NGTZ253 RTS REORG IX CRITERIA (EXTENTS_VALUE): VALUE (1) <= CRITERIA (50).
NGTZ253 RTS REORG IX CRITERIA (PSEUDO_DELETED_PERCENTAGE): NOT CHECKED.
NGTZ253 RTS REORG IX CRITERIA (MAXIMUM_SIZE_PERCENTAGE): NOT CHECKED.
NGTZ253 RTS REORG IX CRITERIA (WHERE_CLAUSE): NOT CHECKED.
NGTZ245 OBJECT (PART 1) WILL BE PROCESSED BY REORG: PASSED ALL RTS CRITERIA.
NGTZ267 REORG WILL NOT BE PERFORMED ON OBJECT: ALL PARTITIONS EXCLUDED BY RTS PROCESSING.
NGTZ235 OBJECT SELECTED FOR RTS PROCESSING:
NGTZ235 IX=PRODDBA.C041IX03
NGTZ242 RTS CRITERIA TABLE ENTRY (TYPE IX) FOUND FOR OBJECT:
NGTZ242 PARTITION=0
NGTZ244 RTS VALUES FOUND FOR OBJECT (PART 1). RTS TIMESTAMP = 2014-03-24-10.09.40.574555.
NGTZ253 RTS REORG IX CRITERIA (LAST_PERFORMED_REORG): NOT CHECKED.
NGTZ253 RTS REORG IX CRITERIA (LAST_PERFORMED_REBUILD): NOT CHECKED.
NGTZ253 RTS REORG IX CRITERIA (INSERTS_PERCENTAGE): PERCENTAGE (40 %) > CRITERIA (25 %).
NGTZ248 OBJECT (PART 1) WILL BE PROCESSED BY REORG: RTS CRITERIA SATISFIED.
NGTZ249 RTS CRITERIA PROCESSING COMPLETED (2 OBJECTS TO BE PROCESSED, 2 OBJECTS EXCLUDED).
Index

A

A.COPYCHANGES 60, 62
A.COPYLASTTIME 60, 62
A.COPYUPDATEDPAGES 60, 62
A.COPYUPDATELSN 60, 62
A.COPYUPDATETIME 60, 62
A.CREATOR 62
A.DATASIZE 60
A.DBID 60, 62
A.DBNAME 60, 62
A.DRIVETYPE 61, 62
A.EXTENTS 61, 62
A.GETPAGES 61, 62
A.HASHLASTUSED 61
A.IBMREQD 61, 62
A.INDEXSPACE 62
A.INSTANCE 61, 62
A.ISOBID 62
A.LASTUSED 62
A.LOADRLASTTIME 61, 62
A.LPFACILITY 61
A.NACTIVE 61, 62
A.NAME 61, 62
A.NLEAF 62
A.NLEVELS 62
A.NPAGES 61, 62
A.PARTITION 61, 62
A.PSID 61, 62
A.REBUILDLASTTIME 62
A.REORGAPPENDINSERT 62
A.REORGCLUSTERSENS 61
A.REORGDELETES 61, 62
A.REORGDISORGLOB 61
A.REORGFARINDREF 61
A.REORGHASHACCESS 61
A.REORGINDEXACCESS 62
A.REORGINSERTS 61, 62
A.REORGLASTTIME 61, 62
A.REORGLEAFFAR 62
A.REORGLEAFNEAR 62
A.REORGMASSDELETE 61, 62
A.REORGNEARINDREF 61
A.REORGNUMLEVELS 62
A.REORGPSEUDODELETES 62
A.REORGSCANACCESS 61
A.REORGNUCLUSTINS 61
A.REORGUPDATES 61
A.SPACE 61, 62
A.STATS01 61
A.STATS101 63
A.STATSDELETES 61, 63
A.STATSPS 61, 63
A.STATSLASTTIME 61, 63
A.STATSMASSDELETE 61, 63
A.STATSUMUPDATES 61
A.SYS_END 61, 63
A.SYS_START 61, 63
A.TOTALENTRIES 63
A.TOTALROWS 61
A.TRANS_START 61, 63
A.UNCOMPRESSEDDATASIZE 61
A.ULONGSTATSTIME 61, 63
app 50
application-specific record 50, 65
automation exit 67
automation routine 67

B

BUFFERPOOL 53

C

conventions, documentation 8
COPY 26, 46
COPY_WHERE 46
Criteria table 16, 30
CRITERIA_INDEX 53
CRITERIA_TABLE 53

D
DAY 27
DB2 RTS 57
DBNAME 53
DEFINE 20
DEFINE statement 20
DO_WHERE 58
documentation summary of changes 10
documentation information 7

E
END_TIME 27
EXCAPP 45
Exceptions table 16
with WHERE clauses 59
EXCEPTIONS_INDEX 54
EXCEPTIONS_TABLE 54
EXCP_DBNAME 54
EXCP_TSNAME 54

G
G.ACCOUNT 60
G.DB2_VERSION 60
G.JOBNAME 60
G.NEEDS_COPY 'Y' 60
G.NEEDS_REORG 'Y' 60
G.NEEDS_STATS 'Y' 60
G.PLAN 60
G.PROGRAMMER 60
G.SSID 60
G.STEPNAME 60
G.UID 60
GENERATESYSIN 50
GENERATESYSINALL 50
Generating SYSIN 50, 66
Global 57

I
INCLUSIVE_EXCLUSIVE 26

Index criteria 42
index reports 70
Index Term 52
IX_COPY_CHANGES 36
IX_COPY_CHKAREO 36
IX_COPY_DAYSSINCE 36
IX_COPY_DESTRUCT 36
IX_COPY_NACTIVE 36
IX_COPY_NONE 36
IX_COPY_UPDATED 36
IX_COPY_UPDATEDMIN 40
IX_COPY_WHERE 37
IX_REORG_APPENDINS 29, 34
IX_REORG_CHKAREO 29, 39
IX_REORG_DELETES 30, 34
IX_REORG_DELETESMIN 30, 39
IX_REORG_EMPTY 29, 40
IX_REORG_EXTENTS 30, 34
IX_REORG_INSDEL 30, 34
IX_REORG_INSDELMIN 30, 39
IX_REORG_INSERTS 29, 34
IX_REORG_INSERTSMIN 29, 39
IX_REORG_LEAFLIMIT 29, 34
IX_REORG_MASSDEL 29, 33
IX_REORG_MAXSIZE 40
IX_REORG_NLEVELS 29, 34
IX_REORG_NONE 33
IX_REORG_PSEUDODEL 29, 34
IX_REORG_WHERE 37
IX_STATS_INSDEL 37
IX_STATS_INSDELMIN 30, 39
IX_STATS_INSDELTOT 37
IX_STATS_MASSDEL 37
IX_STATS_NONE 37
IX_STATS_WHERE 38

L
LAST_UPD_TIMESTAMP 27, 38
LAST_UPD_USERID 27, 38

M
MOD function 57
MONTH 27
MONTH_WEEK 27
N
NAME 26, 31, 46
NGT Copy 49, 65
NGT Reorg 49, 65
NGT Stats 49, 65
NGT Utility Manager report 51, 66
NGT Utility Manager routine 67
NGTAUTO DD statement 67
NGTRTS DISPLAY 54
NGTRTS DISPLAY ALL 55
NGTUtilityStatement 49

P
PARTITION 26, 31
PASSED 50
publications, related 7

R
related publications 7
REMARKS 27, 32, 46
REORG 26, 46
REORG_WHERE 46
REPLACE 52
REPORTONLY 51
revision bars 10
RTS 49

S
S_INCR_CHANGES 35
Schedule table 16
SCHEDULE_DBNAME 54
SCHEDULE_TSNAMESPACE 54
SKIP_WHERE 58
START_TIME 27
STATS 26, 46
STATS_WHERE 46
STOGROUP 54
summary of product changes 10
syntax statement conventions 8
SYSIN 20
generating 50, 66

T
Table space criteria 41
table space report 69
TS_COPY_CHANGES 35
TS_COPY_CHANGESMIN 40
TS_COPY_CHKAREO 28, 32
TS_COPY_CHKINDEX 32
TS_COPY_DELETES 28, 32
TS_COPY_DELETESMIN 28, 38
TS_COPY_DISORG 28, 33
TS_COPY_EXENTS 29, 33
TS_COPY_FORCECOPY 32
TS_COPY_HASHOVERFLOW 39
TS_COPY_INDXREF 28, 33
TS_COPY_INSDELUPD 29, 33
TS_COPY_INSDELUPDMIN 29, 38
TS_COPY_INSERTS 28, 32
TS_COPY_INSERTSMIN 28, 38
TS_COPY_MASSDEL 28, 32
TS_COPY_MAXSIZE 39
TS_COPY_NONE 32
TS_COPY_UNCLUST 28, 33
TS_COPY_UPDATES 29, 32
TS_COPY_UPDATESMIN 29, 38
TS_COPY_WHERE 37
TS_INCR_CHANGESMIN 40
TS_INCR_DAYSSINCE 35
TS_INCR_UPDATEDMIN 40
TS_INCR_WHERE 37
TS_REORG_ALLOUSED 29, 39
TS_REORG_CHKAREO 28, 32
TS_REORG_CHKINDEX 32
TS_REORG_DELETES 28, 32
TS_REORG_DELETESMIN 28, 38
TS_REORG_DISORG 28, 33
TS_REORG_EXENTS 29, 33
TS_REORG_FORCECOPY 32
TS_REORG_HASHOVERFLOW 39
TS_REORG_INDXREF 28, 33
TS_REORG_INSDELUPD 29, 33
TS_REORG_INSDELUPDMIN 29, 38
TS_REORG_INSERTS 28, 32
TS_REORG_INSERTSMIN 28, 38
TS_REORG_MASSDEL 28, 32
TS_REORG_MAXSIZE 39
TS_REORG_NONE 32
TS_REORG_UNCLUST 28, 33
TS_REORG_UPDATES 29, 32
TS_REORG_UPDATESMIN 29, 38
TS_REORG_WHERE 37
TS_STATS_INDLUPTOT 36
TS_STATS_INSDELUPD 36
TS_STATS_MASSDEL 37
TS_STATS_NONE 36
TS_STATS_WHERE 37
TSNAME 54
TYPE 26, 31, 45
with Exceptions table 59

U
UTILITY_ID 26

V
VERSION 27

W
WHERE clause 57, 58
WHERE clauses

X
XRTSSYIN 50, 67