

# Reclaiming Space from Multidimensional Clustering (MDC) Tables – Enhancements for DB2 9.7



## Applies to:

This DB2 9.7 enhancement is applicable to any SAP products that use the multidimensional clustering (MDC) tables like NetWeaver BW. For more information, visit the [Landscape Design and Architecture homepage](#)

## Summary

To allow the reclamation of free space in MDC tables without sacrificing concurrency demands, DB2 9.7 introduced the "RECLAIM EXTENTS ONLY" option in the REORG TABLE command. This new feature can be run manually through the REORG TABLE CLP command or it can be configured to be run automatically through the existing automatic maintenance infrastructure. In this paper, we are going to talk about how this feature works and how to configure it to be run automatically.

**Author:** Beck Tang

**Company:** IBM Canada

**Created on:** 16 May 2009

## Author Bio



Beck Tang has eight years of experience as a member of the IBM SAP Integration and Support Center at the IBM Toronto Lab. His current role includes certification testing of SAP NetWeaver with IBM DB2 LUW and assisting customers with problem analysis and troubleshooting. Mr. Tang is also a customer advocate providing custom-tailored support for large customer accounts running SAP and DB2 LUW. He is a co-author of the red book, "SAP Solutions on IBM DB2 UDB V8.2.2 Handbook" and white papers, "Automatic table maintenance in DB2, Part 1: Automatic statistics collection in DB2 for Linux, UNIX, and Windows" and "Automatic table maintenance in DB2, Part 2: Automatic table and index reorganization in DB2 for Linux, UNIX, and Windows". Mr. Tang can be reached at [becktang@ca.ibm.com](mailto:becktang@ca.ibm.com) Beck Tang Beck Tang

## Table of Contents

Introduction .....	3
Checking the amount of reclaimable space of MDC tables with ADMINTABINFO administrative view and ADMIN_GET_TAB_INFO_V97 table function .....	4
Example of using the updated ADMINTABINFO administrative view and the ADMIN_GET_TAB_INFO_V97 table function to find out the reclaimable space of all MDC tables in a database .....	4
Manually reclaim free extents of MDC tables: RECLAIM EXTENTS ONLY option of the REORG TABLE CLP command .....	5
Syntax for Reclaim Reorg Option .....	5
Access Modes and Locking .....	6
Example for manually reclaiming free extents with the RECLAIM EXTENTS ONLY option of the REORG TABLE CLP command .....	6
Automatically reclaim free extents of MDC tables with the enhanced DB2 AUTO REORG feature .....	7
How automatic Reorg Reclaim works .....	7
Enabling/Maintaining automatic Reorg Reclaim .....	8
Enabling/Maintaining automatic Reorg Reclaim with the stored procedures AUTOMAINT_SET_POLICYFILE/ AUTOMAINT_GET_POLICYFILE .....	8
Diagnostic logging .....	14
Monitoring and Snapshot .....	14
Conclusion .....	16
Acknowledgement .....	16
Related Content .....	17
Disclaimer and Liability Notice .....	18

## Introduction

Multidimensional Clustering (MDC) provides an elegant method for clustering data in tables along multiple dimensions in a flexible, continuous, and automatic way. MDC tables are used extensively in SAP BW environments.

One of the benefits of storing data in an MDC table is, that it guarantees physical clustering with the trade off of reserving storage. Each storage unit in an MDC table is divided into blocks and each block is equivalent to an extent of pages in the tablespace. A block map keeps track of all the data extents belonging to a table and marks which blocks and extents have data on them and which ones do not. Blocks containing data are marked as being "in use". As the data in an MDC table goes through repeated deletions or rollouts, block entries within the block map are no longer marked "in use". These blocks are freed up for reuse by the MDC table to which they belong, but they cannot be used by other objects within the table space.

Before DB2 9.7, the only way to reclaim these free extents was to run an offline reorg. The offline table reorganization operation performed a full table reorganization to reconstruct the table object and then truncate any free space that was no longer needed. This operation can be time consuming and did not allow concurrent write access to the same table while the reorganization is being performed.

In DB2 9.7, the REORG RECLAIM extents option was introduced. It allows customers to reclaim these free extents within an MDC table with minimal time and resource efforts while allowing maximum concurrency. The available concurrency options are described in more detail further down.

A REORG using the reclaim extents option can be triggered in two ways:

- Manually through the REORG TABLE CLP command
- Automatically by setting up an automatic maintenance policy for AUTO\_REORG policy. To achieve this, a new attribute "reclaimExtentsSizeForMDCTables" was added to the ReorgOptions element

The remainder of the paper describes these two methods of reclaiming free extents from an MDC table in detail.

## Checking the amount of reclaimable space of MDC tables with ADMINTABINFO administrative view and ADMIN\_GET\_TAB\_INFO\_V97 table function

In DB2 9.7, a new table function, `ADMIN_GET_TAB_INFO_V97` was introduced and the `ADMINTABINFO` administrative view was updated to allow customers to check the amount of reclaimable space for MDC tables.

The amount of reclaimable space for MDC tables can be viewed from the `RECLAIMABLE_SPACE` column of the output of the `ADMINTABINFO` administrative view and the `ADMIN_GET_TAB_INFO_V97` table function. The `RECLAIMABLE_SPACE` column indicates, for an MDC table in a DMS table space, the amount of disk space that can be reclaimed by running the REORG command with the reclaim extent option. The disk space is reported in KB. (For non-MDC tables, the value is reported as zero.)

**Note:** A query against the `ADMINTABINFO` view could take a while to finish.

### Example of using the updated ADMINTABINFO administrative view and the ADMIN\_GET\_TAB\_INFO\_V97 table function to find out the reclaimable space of all MDC tables in a database

- Scenario 1: Determine the table name, the associating table schema, and the disk space that can be reclaimed for all the MDC tables in the database with greater-than-zero reclaimable space in a single partition environment:

```
db2 => SELECT SUBSTR(TABSCHEMA,1,15) as "Table Schema", SUBSTR(TABNAME,1,15) as
"Table Name", RECLAIMABLE_SPACE FROM SYSIBMADM.ADMINTABINFO WHERE
RECLAIMABLE_SPACE > 0
```

Table Schema	Table Name	RECLAIMABLE_SPACE
SAPLR1	/BIZ/FNONCUM	188224
SAPLR1	/BIZ/FCUMULA	552064

2 record(s) selected.

- Scenario 2: Determine the amount of reclaimable space for a particular MDC table called “/BIZ/FCUMULA” under the schema, “SAPLR1”:

```
db2 => select RECLAIMABLE_SPACE from table (SYSPROC.ADMIN_GET_TAB_INFO_V97(
'SAPLR1', '/BIZ/FCUMULA' )) AS T
```

RECLAIMABLE_SPACE
552064

1 record(s) selected.

There is 552,064 KB of reclaimable space for the MDC table, “/BIZ/FCUMULA”.



## Access Modes and Locking

Different concurrent modes acquire various sets of table lock modes during the operation. Customer can choose the appropriate concurrent modes with the associated table lock modes to control the level of concurrent access other transactions have while the extents are being reclaimed.

- ALLOW WRITE ACCESS:
  - IS table/partition lock (default)
  - IX table space lock
  - X alter table lock
  - Conditional shared block locks on all the extents that are being freed
- ALLOW READ ACCESS:
  - S table/partition lock
  - IX table space lock
  - X alter table lock
- ALLOW NO ACCESS:
  - Z table lock
  - IX table space lock

### Note:

For ALLOW WRITE ACCESS, obtaining a conditional shared block locks on all the extents that are being freed is important in that it ensures further updates can be made to the blocks about to be freed from other transactions. On the other hand, if the REORG RECLAIM cannot obtain the proper lock on certain blocks because it is being updated, it means these blocks are still being (re)used. In this case, the REORG RECLAIM will skip these blocks and look for the next empty block to be freed.

### Example for manually reclaiming free extents with the RECLAIM EXTENTS ONLY option of the REORG TABLE CLP command

Scenario: The customer would like to find out all the MDC tables on the system with non-zero reclaimable space and then release the extents occupied by these tables.

1. Find out which MDC table(s) in the database have free extents that can be reclaimed:

```
db2 => SELECT SUBSTR(TABSCHEMA,1,15) as "Table Schema", SUBSTR(TABNAME,1,15) as
"Table Name", RECLAIMABLE_SPACE FROM SYSIBMADM.ADMINTABINFO WHERE
RECLAIMABLE_SPACE > 0
```

Table Schema	Table Name	RECLAIMABLE_SPACE
SAPLR1	/BIZ/FCUMULA	552064
SAPLR1	/BIZ/FNONCUM	188224

2 record(s) selected.

The MDC table "/BIZ/FCUMULA" under the "SAPLR1" schema has 552,064 KB of reclaimable space while the MDC table "/BIZ/FNONCUM" under the "SAPLR1" schema has 188,224 KB of reclaimable space.

- The following REORG TABLE CLP command with the RECLAIM EXTENTS ONLY option allows concurrent access to the MDC table, /BIZ/FCUMULA, while reclaiming free extents:

```
db2 => REORG TABLE "SAPLR1"."/BIZ/FCUMULA" RECLAIM EXTENTS ONLY ALLOW WRITE ACCESS
DB20000I The REORG command completed successfully.
```

- Check that the space originally occupied by the MDC table, /BIZ/FCUMULA, is freed up:

```
db2 => select RECLAIMABLE_SPACE from table (SYSPROC.ADMIN_GET_TAB_INFO_V97(
'SAPLR1', '/BIZ/FCUMULA' )) AS T
```

```
RECLAIMABLE_SPACE
```

```
-----
```

```
0
```

```
1 record(s) selected.
```

The RECLAIMABLE\_SPACE column of the SYSPROC.ADMIN\_GET\_TAB\_INFO\_V97 table function shows a value of 0. The space originally occupied by the MDC table, /BIZ/FCUMULA, has been freed up.

Repeat step #2/#3 on the MDC table, "/BIZ/FNONCUM" to reclaim its free extents.

## Automatically reclaim free extents of MDC tables with the enhanced DB2 AUTO REORG feature

Manually keeping track of all the MDC tables with reclaimable space and issuing the REORG TABLE CLP command for each MDC table can be time consuming. In DB2 9.7, the AUTO REORG feature, which is part of the automatic table maintenance functionality in DB2, was enhanced with a new Reclaim option to provide support for reclaiming free extents of MDC tables automatically with minimal amount of time and resource. It allows concurrent write access to the table while RECLAIM REORG is running.

For more information on the AUTO REORG in general, refer to the white paper, [Automatic table maintenance in DB2, Part 2: Automatic table and index reorganization in DB2 for Linux, UNIX, and Windows](#).

### How automatic Reorg Reclaim works

Automatic Reorg Reclaim is built upon the existing automatic reorg infrastructure. As of DB2 9.7, a new reorg option attribute, `reclaimExtentsSizeForMDCTables`, can be specified in the automatic maintenance policy (the policy controls the behavior of automatic reorganization). Automatic reorganization will reclaim free extents for an MDC table when the space occupied by those free extents (in kilobytes) is larger than the value of the `reclaimExtentsForMDCTables` attribute. Once the `reclaimExtentsSizeForMDCTables` attribute is defined in the automatic maintenance policy (and the hierarchy of AUTO\_REORG database configuration parameters are turned on), automatic Reorg Reclaim is enabled for the database. By default the `reclaimExtentSizeForMDCTables` attribute is not defined in the automatic maintenance policy, so automatic reorg reclaim is disabled by default.

Similar to the existing automatic reorganization and automatic statistics collection maintenance types, the process of automatic Reorg Reclaim consists of two phases: the evaluation phase and execution phase.

The evaluation phase occurs for the first time two hours after a database has been activated, and again every two hours after that. The evaluation phase checks all the qualified MDC tables in DMS table spaces using the ADMIN\_GET\_TAB\_INFO\_V97 table function. Those MDC tables with the amount of reclaimable space exceeding the value defined by the reorg option, `reclaimExtentsSizeForMDCTables`, are marked for automatic Reorg Reclaim to be done in the execution phase. (For a DPF environment, the sum of the `reclaimable_space` value for the MDC table across all partitions is used).

By default, all MDC tables in DMS table spaces are candidates for evaluation. However, a “table exclude list” can be defined using the FilterClause element in the policy (e.g. “<FilterClause>TABNAME NOT LIKE 'Z%' </FilterClause>”) to narrow down a list of MDC tables for evaluation.

**Note:** the maxOfflineReorgTableSize attribute of the FilterClause element in the policy does not affect the automatic Reorg Reclaim. However, it can still be used for general automatic reorganization.

To enable automatic Reorg Reclaim, you need to configure an online automatic maintenance window. A default online maintenance window of 24 X 7 is enabled when the database is created. This means that without changing the default maintenance windows settings, auto Reorg Reclaim execution phase can take place at any time on any day.

The execution of the automatic reorg reclaim issues an db2Reorg API equivalent to the following REORG TABLE CLP command with the RECLAIM EXTENTS ONLY option underneath the cover for each qualified MDC table:

```
REORG TABLE <MDC table Name> RECLAIM EXTENT ONLY ALLOW WRITE
```

The “ALLOW WRITE” option is used to allow concurrent write access to the MDC table during the REORG TABLE operation.

### Enabling/Maintaining automatic Reorg Reclaim

Enabling automatic Reorg Reclaim involves enabling automatic reorganization in general by:

1. Configuring database configuration parameters related to automatic reorganization
2. Setup the required automatic maintenance policy

On SAP systems running with DB2 V9.7 databases, these configurations can be maintained through DB2 interface by updating the database configuration parameters (AUTO\_MAINT, AUTO\_TBL\_MAINT and AUTO\_REORG) with CLP and maintaining the automatic maintenance policy with the stored procedure: AUTOMAINT\_SET\_POLICYFILE/ AUTOMAINT\_GET\_POLICYFILE. Support for handling the policy through the DBACockpit will be part of the upcoming release of NetWeaver and Solution Manager"

### Enabling/Maintaining automatic Reorg Reclaim with the stored procedures AUTOMAINT\_SET\_POLICYFILE/ AUTOMAINT\_GET\_POLICYFILE

To enable automatic Reorg Reclaim through DB2, the database configuration parameters: AUTO\_MAINT, AUTO\_TBL\_MAINT and AUTO\_REORG have to be turned on. If the default online maintenance window of 24X7 does not fit the need, the online maintenance windows can be redefined with the AUTOMAINT\_SET\_POLICYFILE stored procedure. The AUTOMAINT\_SET\_POLICYFILE stored procedure is also used to setup the new Reclaim option, reclaimExtentsSizeForMDCTables, which is part of the automatic reorganization policy

**Note:** The default policy for the AUTO\_REORG policy type does not include reclaimExtentsSizeForMDCTables reorg option attribute. To use the Auto Reorg Reclaim feature, you should explicitly include reclaimExtentsSizeForMDCTables in a policy file, and use AUTOMAINT\_SET\_POLICYFILE to override the default policy.

## Configuration Parameters

The procedure to enable the new Reclaim option of Automatic Reorganization is similar to enabling Automatic Reorganization in general. First of all, the hierarchy of database configuration parameters that relates to the AUTO\_REORG (off by default) parameters has to be enabled. In other words, the database configuration parameters: AUTO\_MAINT, AUTO\_TBL\_MAINT and AUTO\_REORG have to be set to "ON". (In SAP environment on DB2 9.7, AUTO\_MAINT and AUTO\_TBL\_MAINT are set to "ON" by default already to enable Automatic Statistics collection, but AUTO\_REORG is not turned on by default.). For example, in the database below, automatic reorganization is enabled.

```
sapsx2:db2lr1 7% db2 get db cfg for lr1 | grep "Automatic "
```

<b>Automatic maintenance</b>	<b>(AUTO_MAINT) = ON</b>
Automatic database backup	(AUTO_DB_BACKUP) = OFF
<b>Automatic table maintenance</b>	<b>(AUTO_TBL_MAINT) = ON</b>
Automatic runstats	(AUTO_RUNSTATS) = ON
Automatic statement statistics	(AUTO_STMT_STATS) = ON
Automatic statistics profiling	(AUTO_STATS_PROF) = OFF
Automatic profile updates	(AUTO_PROF_UPD) = OFF
<b>Automatic reorganization</b>	<b>(AUTO_REORG) = ON</b>

To turn on these database configuration parameters (i.e. AUTO\_MAINT, AUTO\_TBL\_MAINT, AUTO\_REORG), you can run a command similar to:

```
sapsx2:db21r1 8% db2 update db cfg for 1r1 using AUTO_MAINT ON AUTO_TBL_MAINT ON
AUTO_REORG ON
DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully.
SQL1363W One or more of the parameters submitted for immediate modification
were not changed dynamically. For these configuration parameters, all
applications must disconnect from this database before the changes become
effective.
```

**Note:** On a DB2 instance where the database partitioning feature is enabled, this command must be issued on the catalog partition. Setting the configuration parameter on any other partition has no effect.

### The stored procedures: AUTOMAINT\_SET\_POLICYFILE/AUTOMAINT\_GET\_POLICYFILE

The configuration settings of automatic Reorg Reclaim and the maintenance windows can be maintained through the stored procedures: AUTOMAINT\_SET\_POLICYFILE/AUTOMAINT\_GET\_POLICYFILE which requires the associated policy files in XML format (there are two other similar stored procedures, AUTOMAINT\_SET\_POLICY/AUTOMAINT\_GET\_POLICY. They will not be discussed in this paper).

```
SYSPROC.AUTOMAINT_SET_POLICYFILE (--policy_type--, --policy_file_name--):
```

This stored procedure replaces the configuration settings for a particular policy type (defined by the `policy_type` argument) used by the database with the ones specified in the policy file (defined by the `policy_file_name` argument).

The `policy_type` argument can be of one of the following values:

- AUTO\_BACKUP,
- AUTO\_REORG,
- AUTO\_RUNSTATS
- MAINTENANCE\_WINDOW.

The policy file to be set, should be put in `$HOME/sqllib/tmp` of the instance owner (i.e. `db2<sid>`) on Unix/Linux platforms. (On Windows®, the instance directory name can be determined from the values of the `DB2INSTPROF` registry variable and the `DB2INSTANCE` environment variable. For a policy file named 'automaint\policy.xml', if `db2set` gives `DB2INSTPROF=F:\DB2\DB2HIA` and `%DB2INSTANCE%=db2hia`, then the file name will be `F:\DB2\DB2HIA\tmp\automaint\policy.xml`) The configuration settings for the `MAINTENANCE_WINDOW` and `AUTO_REORG` policy types are used for enabling automatic Reorg Reclaim.

```
SYSPROC.AUTOMAINT_GET_POLICYFILE (--policy_type--, --policy_file_name--):
```

This stored procedure retrieves the policy of the `policy_type` (i.e. `AUTO_BACKUP`, `AUTO_REORG`, `AUTO_RUNSTATS` or `MAINTENANCE_WINDOW`) from the database and writes the policy in xml format to a file name defined by the value of the argument, `policy_file_name`. This stored procedure can be used to check the current configuration settings of the policy type being used in the database. The retrieved policy file is put in `$HOME/sqllib/tmp` of the instance owner (i.e. `db2<sid>`).

## Configuring the automatic maintenance window(s)

As stated earlier, enabling automatic Reorg Reclaim a suitable online maintenance window. If the default online maintenance window (24 X 7) fits the requirement, the maintenance windows do not need to be explicitly redefined. To check the default maintenance windows, you can run, as db2<sid>, the stored procedure SYSPROC.AUTOMAINT\_GET\_POLICYFILE. The retrieved default policy file is stored in \$HOME/sqllib/tmp of db2<sid>. For example:

```

augsburg1:db2hia 17> db2 "call SYSPROC.AUTOMAINT_GET_POLICYFILE('MAINTENANCE_WINDOW',
'def_main_win.txt')"

Return Status = 0
augsburg1:db2hia 18> cd $HOME/sqllib/tmp
augsburg1:db2hia 19> cat def_main_win.txt
<?xml version="1.0" encoding="UTF-8"?>
<DB2MaintenanceWindows
xmlns="http://www.ibm.com/xmlns/prod/db2/autonomic/config" >

  <!-- Online Maintenance Window -->
  <OnlineWindow Occurrence="During" startTime="00:00:00" duration="24" >
    <DaysOfWeek>All</DaysOfWeek>
    <DaysOfMonth>All</DaysOfMonth>
    <MonthsOfYear>All</MonthsOfYear>
  </OnlineWindow>
</DB2MaintenanceWindows>

```

If you would like to change the policy file, you can use the SYSPROC.AUTOMAINT\_SET\_POLICYFILE stored procedure. Refer to the comment of the sample maintenance policy (i.e. as db2<sid>, view the \$HOME/sqllib/samples/automaintcfg/DB2MaintenanceWindowPolicySample.xml file) on how to modify the maintenance policy to fit your need.

For example, you would like to restrict the online maintenance window to be between 8:00 am and 6:00 pm (and specify no offline window). You can use the following maintenance window policy file:

```

<?xml version="1.0" encoding="UTF-8"?>
<DB2MaintenanceWindows
xmlns="http://www.ibm.com/xmlns/prod/db2/autonomic/config" >

  <!-- Online Maintenance Window -->
  <OnlineWindow Occurrence="During" startTime="00:08:00" duration="10" >
    <DaysOfWeek>All</DaysOfWeek>
    <DaysOfMonth>All</DaysOfMonth>
    <MonthsOfYear>All</MonthsOfYear>
  </OnlineWindow>
</DB2MaintenanceWindows>

```

Create the customized maintenance window file (i.e. "cus\_main\_win.txt" in this example) as db2<sid> in \$HOME/sqllib/tmp. Then run the SYSPROC.AUTOMAINT\_SET\_POLICYFILE stored procedure to redefine the online maintenance window:

```

augsburg1:db2hia 45> db2 "call SYSPROC.AUTOMAINT_SET_POLICYFILE('MAINTENANCE_WINDOW',
'cus_main_win.txt')"

Return Status = 0

```

Make sure the return status is 0, which means DB2 accept this new policy. If there is any problem, check the syntax of the XML policy file.

## Enabling the reclaim extents option in automatic reorg

After setting up an online maintenance window, the last step to enable auto Reorg Reclaim is setting a new policy for the auto\_reorg policy type with the new Reclaim option attribute, `reclaimExtentsSizeForMDCTables`, defined. This can be done through the `SYSPROC.AUTOMAINT_SET_POLICYFILE` stored procedure as well. A sample XML policy file for the AUTO\_REORG policy type, `$HOME/sqllib/samples/automaintcfg/DB2AutoReorgPolicySample.xml`, explains how to setup the `reclaimExtentsSizeForMDCTables` reorg option as well as the other reorg options.

Here is an example of how to setup the configuration of the AUTO\_REORG policy type for the automatic maintenance policy with the `reclaimExtentsSizeForMDCTables` reorg option attribute on.

1. Make sure the database configuration parameters: `AUTO_MAINT`, `AUTO_TBL_MAINT` and `AUTO_REORG` are turned on and a proper online maintenance window has been setup.
2. As `db2<sid>`, create a XML policy file named `myautoreorgpolicy.xml` in `$HOME/sqllib/tmp/` of the AUTO\_REORG policy type. A value of 102400 KB is used for `reclaimExtentsSizeForMDCTables` which means only MDC tables of DMS type with more than 102,400 KB amount of reclaimable space will be marked for automatic Reorg Reclaim.

```
<?xml version="1.0" encoding="UTF-8"?>
<DB2AutoReorgPolicy
xmlns="http://www.ibm.com/xmlns/prod/db2/autonomic/config" >

  <ReorgOptions dictionaryOption="Rebuild" indexReorgMode="Online"
useSystemTempTableSpace="true" reclaimExtentsSizeForMDCTables="102400"
/>

  <ReorgTableScope maxOfflineReorgTableSize="1000000">
    <FilterClause>TABSCHEMA = 'SAPLR1'</FilterClause>
  </ReorgTableScope>
</DB2AutoReorgPolicy>
```

3. Setup the policy by running the SYSPROC.AUTOMAINT\_SET\_POLICYFILE stored procedure:

```
augzburg1:db2hia 84> db2 "call SYSPROC.AUTOMAINT_SET_POLICYFILE('AUTO_REORG',  
'myautoreorgpolicy.xml')"
```

```
Return Status = 0
```

Note: 1) The reorg option, `indexReorgMode`, specifies whether the index reorganization should be performed online or offline. Once the `reclaimExtentsSizeForMDCTables` reorg option is set to a value, automatic Reorg Reclaim will be activated whether `indexReorgMode` reorg option is set to Online or Offline. I think the options for `indexReorgMode` are not really explained.

2) After the evaluation period, if it is determined that an MDC table requires index reorg:

i) If the MDC table reclaimable size is greater than the mentioned policy size for `reclaimExtentsSizeForMDCTables`, the AUTO REORG execution will trigger with the RECLAIM option. In this case no further Index reorg is required.

ii) If the MDC table reclaimable size is less than the mentioned policy size for `reclaimExtentsSizeForMDCTables`, this table will be considered as a normal table and DB2 follows the normal AUTO REORG procedure.

## Diagnostic logging

Automatic Reorg Reclaim writes log points to the db2diag.log as tables are processed during the execution phase (i.e. when the reorganization takes place). An entry will be written when the reorganization of a table has started and when the reorganization has completed. These log entries will always be written, regardless of the value of DIAGLEVEL. In the case of an error, the "stop" log point will include the sqlcode for the error that was encountered. Here is an example of db2diag.log entries of a successful automatic Reorg Reclaim on an MDC table:

```

2009-05-08-20.09.01.704022-240 I1174128E377      LEVEL: Event
PID      : 17103                TID   : 339                PROC  : db2acd 0
INSTANCE: db21r1                NODE  : 000
EDUID   : 339                  EDUNAME: db2acd 0
FUNCTION: DB2 UDB, Automatic Table Maintenance, db2AutoReorgExec, probe:10
START    : Automatic reorg has started on table LR1      ."SAPLR1  "."/BIZ/FNONCUM"

2009-05-08-20.09.10.650708-240 I1174506E429      LEVEL: Event
PID      : 17103                TID   : 339                PROC  : db2acd 0
INSTANCE: db21r1                NODE  : 000
APPID   : *LOCAL.db21r1.090509000911
EDUID   : 339                  EDUNAME: db2acd 0
FUNCTION: DB2 UDB, Automatic Table Maintenance, db2AutoReorgExec, probe:10
STOP    : Automatic reorg has completed successfully on table LR1      ."SAPLR1
        "."/BIZ/FNONCUM"

```

## Monitoring and Snapshot

The main mechanism to monitor and perform problem determination for auto Reorg Reclaim is the health indicator (db.tb\_reorg\_req). It requires the database manager configuration parameter, HEALTH\_MON, to be set to ON (On SAP system, HEALTH\_MON is by default set to OFF). The health indicator enters an "attention" state whenever a table requires manual Reorg Reclaim (this means that automatic Reorg Reclaim of a table has failed or it could be that any of the AUTO\_MAINT, AUTO\_TBL\_MAINT, AUTO\_REORG switches are set to off). There are other interfaces to retrieve snapshot information including "GET HEALTH SNAPSHOT" CLP command, db2GetSnapshot API, and the health indicators table functions (HEALTH\_DB\_HI, HEALTH\_DB\_HIC).

Here is an example output of the "GET HEALTH SNAPSHOT" command:

```

Database Health Snapshot

Snapshot timestamp           = 05/14/2009 23:41:43.267647
Database name                = LR1
Database path                = /db2/LR1/db21r1/NODE0000/SQL00001/
...
Indicator Name              = db.tb_reorg_req
Value                       = 1
Evaluation timestamp         = 05/14/2009 22:00:50.922068
Alert state                  = Attention
...
Collection:

Name                        = "SAPLR1"."/BIZ/FCUMULA"
Detail                      = RECLAIM EXTENTS ONLY ALLOW WRITE
State                       = Attention
Evaluation timestamp        = 05/14/2009 22:01:05.000000

Collection:

Name                        = "SAPLR1"."/BIZ/FNONCUM"
Detail                      = RECLAIM EXTENTS ONLY ALLOW WRITE
State                       = Attention
Evaluation timestamp        = 05/14/2009 22:01:05.000000

```

In this example, the SAPLR1./BIZ/FCUMULA and SAPLR1./BIZ/FNONCUM tables were in Attention state because the database configuration parameter, AUTO\_REORG had not been set to ON. In a situation like this, the user can get explanations and recommendations on what the problem was and how to recover from it by running the CLP command, "db2 get recommendations for health indicator db.tb\_reorg\_req for db on <sid>".

## Conclusion

The new DB2 V9.7 reclaim extents option on reorg allows free extents of MDC tables to be reclaimed and freed up to be used by other tables in the same table space while allowing concurrent access by other transactions. This feature could be used manually with the REORG TABLE command with the RECLAIM EXTENTS ONLY option; or the automatic maintenance mechanism, which can be configured through DB2 stored procedures. Regardless of how this feature is used, it provides extra flexibility on managing the storage space of MDC tables.

## Acknowledgement

Special Thanks for Joachim Pfefferle, Scott Walkty, Siju T Thomas, and Kayalvizhi Ganesan for reviewing the paper.

## Related Content

[DB2 Information Center Version 9.7](#)

[Automatic table maintenance in DB2, Part 2: Automatic table and index reorganization in DB2 for Linux, UNIX, and Windows](#)

For more information, visit the [Landscape Design and Architecture homepage](#)

## Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.