

# CUDB Node Commands and Parameters

---

## USER GUIDE

**Copyright**

© Ericsson AB 2016. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

**Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark information.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Document Purpose and Scope	1
1.2	Target Groups	1
1.3	Revision Information	1
1.4	Typographic Conventions	3
<b>2</b>	<b>CUDB Commands</b>	<b>5</b>
2.1	CUDB Command	6
2.2	Management Commands	6
2.2.1	cudbAnalyser	6
2.2.2	cudbApplicationCounters	8
2.2.3	cudbApplyConfig	9
2.2.4	cudbCheckConsistency	15
2.2.5	cudbCheckReplication	18
2.2.6	cudbCollectInfo	20
2.2.7	cudbConsistencyMgr	22
2.2.8	cudbDataBackup	27
2.2.9	cudbDbDiskManage	30
2.2.10	cudbDsgMastershipChange	31
2.2.11	cudbDsgProvisioningManage	35
2.2.12	cudbGetLogs	37
2.2.13	cudbHaState	38
2.2.14	cudbLdapFeRestart	41
2.2.15	cudbManageBCServer	43
2.2.16	cudbManageLibDataDist	44
2.2.17	cudbManageStore	45
2.2.18	cudbMpstat	52
2.2.19	cudbPmJobReload	54
2.2.20	cudbPrepareStore	56
2.2.21	cudbReallocate	60
2.2.22	cudbReconciliationMgr	66
2.2.23	cudbRemoteTrust	68
2.2.24	cudbResumeProvisioningNotification	69
2.2.25	cudbServiceContinuity	70
2.2.26	cudbSwBackup	72
2.2.27	cudbSwVersionCheck	75
2.2.28	cudbSystemDataBackupAndRestore	79
2.2.29	cudbSystemStatus	84
2.2.30	cudbTakeAllMasters	89
2.2.31	cudbUnitDataBackupAndRestore	91
2.2.32	cudbUpdateUserInfo	94
2.3	Internal Commands	95
2.3.1	cudbBCServersRestart	95



2.3.2	cudbCopyAclsConfFile	95
2.3.3	cudbEvipConfigExtension	96
2.3.4	cudbEvipEncapsulator	96
2.3.5	cudbExecuteAllBlades	96
2.3.6	cudbFollowLdapfeLogs	96
2.3.7	cudbInstall	96
2.3.8	cudbLicensingTool	96
2.3.9	cudbManageSite	96
2.3.10	cudbOomConfigurator	96
2.3.11	cudbParallelCommandRun	96
2.3.12	cudbSdpInfo	96
2.3.13	cudbSdpInstall	97
2.3.14	cudbSdpReplace	97
2.3.15	cudbSdpUninstall	97
2.3.16	cudbSetDsgMaster	97
2.3.17	cudbSetPartitionStatus	97
2.3.18	cudbTpsStat	97
<b>Glossary</b>		<b>99</b>
<b>Reference List</b>		<b>101</b>



# 1 Introduction

This document describes the Ericsson Centralized User Database (CUDB) commands and command options, along with a complete reference on their options.

## 1.1 Document Purpose and Scope

This document describes the CUDB commands and command options with a complete reference on their options, syntax, output and some examples. The following topics are out of the scope of this document:

- Linux Operating System (OS) commands. Refer to *LDE Management Guide*, Reference [22] for information on the Linux Distribution Extension (LDE) commands.
- The process of updating the CUDB Configuration Model. Refer to *CUDB Node Configuration Data Model Description*, Reference [1] for the steps of updating the configuration model.

Most of the commands provide the output by means of logging information. Refer to *CUDB Node Logging Events*, Reference [2] for more details about logging events in CUDB.

## 1.2 Target Groups

This document is intended for system administrators and operators who are familiar with Linux systems. Users of this document must possess moderate knowledge of the CUDB system and its Operation and Maintenance (OAM) procedures.

## 1.3 Revision Information

### Rev. A

This document is based on 1/1553-CSH 109 067/9, and contains the following changes:

- Structural and content rearrangement throughout the document.
- Virtualization terminology updates throughout the document.
- Removed obsolete information.



- Section 2 on page 5: Updated the general description of commands and the available user groups.
- Section 2.2.1 on page 6: Updated syntax and examples of use of the `cudbAnalyser` command.
- Section 2.2.2 on page 8: Updated syntax information of `cudbApplicationCounters`.
- Section 2.2.3 on page 9 and Section 2.2.3.3 on page 10: Updated sections with information on command deprecation. Updated output in examples of use.
- Section 2.2.4 on page 14: Updated syntax, list of commands option, and examples of use.
- Section 2.2.5.6 on page 19: Updated syntax of the `--behind-limit` option of the `cudbCheckReplication` command.
- Section 2.2.7 on page 22: Updated syntax and command option descriptions of the `cudbConsistencyMgr` command.
- Section 2.2.8 on page 26: Updated syntax and command option information of `cudbDataBackup`.
- Section 2.2.9 on page 29: Updated syntax and command option descriptions of the `cudbDbDiskManage` command.
- Section 2.2.10 on page 31: Updated the command description, syntax, `--force` option description, and output examples of the `cudbDsgMastershipChange` command.
- Section 2.2.11.2 on page 35: Updated syntax of `cudbDsgProvisioningManage` command.
- Section 2.2.14 on page 40: Updated the command syntax, and updated the output examples of the `cudbLdapFeRestart` command.
- Section 2.2.15 on page 43: Updated syntax of the `cudbManageBCServer` command. Removed `-help` command option.
- Section 2.2.17.3 on page 46: Updated `kickstart` action of `cudbManageStore` command. Action is no longer restricted. Updated value of `-d | --ds <dsId>` command option.



- Section 2.2.20.3 on page 56: Updated value of `-d` | `--ds <dsId>` command option.
- Section 2.2.21.3 on page 61: Updated information regarding the `--nthreads` command.
- Section 2.2.21.4 on page 63, Section 2.2.26.3 on page 72, Section 2.2.26.5 on page 74, and Section 2.3.3 on page 95: Added information related to removed PLDB nodes.
- Section 2.2.23 on page 67: Added a new paragraph.
- Section 2.2.23.2 on page 68: Updated command.
- Section 2.2.23.3 on page 68: Updated the note of a command and added a new command with a note.
- Section 2.2.26 on page 72: Updated command syntax, and updated the examples of use of the `cudbSwBackup` command.
- Section 2.2.28 on page 79 and Section 2.2.28.4 on page 81: Updated information on command execution. Updated example information.
- Section 2.2.29.5 on page 86: Added `OAM1` to LDAP counter in the example.
- Section 2.2.31 on page 91: Updated command syntax, and updated the output example of the `cudbUnitDataBackupAndRestore` command.
- Section 2.2.32 on page 94: Updated with information on deprecation.

## 1.4 Typographic Conventions

Typographic Conventions can be found in the following document:

- *Typographic Conventions*







## 2 CUDB Commands

The following sections list and describe all CUDB commands that can be used within a Secure Shell (SSH) session in the CUDB nodes belonging to a CUDB system. The commands are grouped into the following categories:

- **Management commands:** These commands can be executed both by operators and Ericsson personnel. See Section 2.2 on page 6 for the alphabetical list and description of these commands.

**Note:** Some of the options of the management commands can be executed by Ericsson personnel only. These options are indicated with the “Restricted for Ericsson personnel” phrase.

- **Internal commands:** These commands can be executed by Ericsson personnel only. See Section 2.3 on page 95 for the alphabetical list of these commands.

The commands in this document can be executed from any CUDB node, unless specific conditions or restrictions are specified in the `Requisites` section of the related command.

For each command, the following information is included:

- Description and execution requisites
- Use and Options:
  - Options and their default values
  - Syntax of use
  - Output
- Examples of use

The commands described in this document can be used by connecting to any CUDB node through an SSH interface, and getting authenticated by the system. The commands can be executed with the `cudbOperator` or `cudbadmin` users through the `sudo` Linux command. The available rights of these users are as follows:

- The `cudbOperator` user can execute commands that provide information about the CUDB status without affecting the node or the system. The `Requisites` subsections of the commands always state if `cudbOperator` users can run the specific command. For more information about these users and the associated security groups, refer to *CUDB Security and Privacy Management*, Reference [3] and to *CUDB Users and Passwords*, Reference [4].



- The `cudbadmin` user can execute all the commands that the `cudbOperator` user can, and can also run commands that execute actions which may affect system status, behavior, or performance. Use of this user therefore requires OAM skills.

By default, the location of the commands described in the document is set on the `PATH` system variable during installation time.

Several commands offer a debug option. This option activates the `DEBUG` log level, whose `DEBUG` log messages are stored in the same location as the log messages belonging to other levels. The location of the log messages for each component are listed in *CUDB Node Logging Events*, Reference [2].

**Note:** The debugging option of the commands is used only when required by Ericsson personnel for troubleshooting purposes. This is because the amount of logging information can be very high during debugging, which can alter system working conditions.

All commands include a help option as well, used to provide information on the command and its available options.

For further information on CUDB system administration procedures, refer to *CUDB System Administrator Guide*, Reference [5].

## 2.1 CUDB Command

## 2.2 Management Commands

Commands in this section can be executed both by operators and Ericsson personnel.

### 2.2.1 **cudbAnalyser**

The `cudbAnalyser` command performs a log analysis on the logs collected by `cudbGetLogs`. For more information about this command, refer to *CUDB Logchecker*, Reference [6].

#### 2.2.1.1 **Requisites**

`cudbOperator` users have no access to run this command.

#### 2.2.1.2 **Syntax**

- Format (1): `cudbAnalyser -l | --logfile <logfile> -p | --previous-logfile <previous-logfile> [-s | --save-counter] [-q | --quiet] [-d | --debug] [-S | --send-alarm] [-w | --weight-threshold <threshold>]`



- Format (2): `cudbAnalyser -a | --auto-check [-s | --save-counter] [-q | --quiet] [-d | --debug] [-S | --send-alarm] [-w | --weight-threshold <threshold>]`
- Format (3): `cudbAnalyser -h | --help`

### 2.2.1.3

#### Command Options

The following command options can be used:

- `-h | --help`: Displays command help message and exits.
- `-l <logfile> | --logfile <logfile>`: Points to the newer log file.
- `-p | --previous-logfile <previous-logfile>`: Points to an older logfile.

**Note:** The timestamp of `<logfile>` must be newer than `<previous-logfile>` to receive a correct differential analysis.

- `-s | --save-counter`: Saves a counter under `/home/cudb/monitoring/preventiveMaintenance/cudb_analyser_error_counter.log` which shows the number of failing checks. This is needed for ESA integration.
- `-a | --auto-check`: The script checks the last two log files automatically. The use of the `--logfile` and `--previous-logfile` switches is not necessary in this case.
- `-q | --quiet`: Makes the error printouts less verbose.
- `-d | --debug`: Restricted for Ericsson personnel.
- `-S | --send-alarm`: Sends an alarm to ESA, based on the results.
- `-w | --weight-threshold <threshold>`: Prints errors if WEIGHT is greater or equal to the weight threshold (default value is 2). When performing a detailed system audit, set this value to 0 to print more errors.

### 2.2.1.4

#### Output

```
Started on ACTIVE SC...
[INFO] Checking files: //home/cudb/monitoring/preventiveMaintenance/ \
/CUDB_50_201307231233.log //home/cudb/monitoring/preventiveMaintenance/ \
/CUDB_50_201307231231.log logfile versions:0.0.60/0.0.60
[ERROR] OS: network stat shows errors (Severity: Minor)
> CUDB50 PL0 eth1/statistics/rx_dropped 27 K packets
> CUDB50 PL1 eth1/statistics/rx_dropped 27 K packets
> CUDB50 DS1_0 eth1/statistics/rx_dropped 27 K packets
> CUDB50 DS1_1 eth1/statistics/rx_dropped 27 K packets
```

### 2.2.1.5

#### Examples of Use

- `<CUDB_node_prompt> cudbAnalyser --help`



- `<CUDB_node_prompt> cudbAnalyser --auto-check --save-counter`
- `<CUDB_node_prompt> cudbAnalyser -l /home/cudb/monitoring/preventiveMaintenance/CUDB_28_201108190300.log -p /home/cudb/monitoring/preventiveMaintenance/CUDB_28_201108160300.log --send-alarm`
- `<CUDB_node_prompt> cudbAnalyser -l /home/cudb/monitoring/preventiveMaintenance/CUDB_28_201108190300.log -p /home/cudb/monitoring/preventiveMaintenance/CUDB_28_201108160300.log --save-counter`
- `<CUDB_node_prompt> cudbAnalyser --logfile /home/cudb/monitoring/preventiveMaintenance/CUDB_28_201108190300.log --previous-logfile /home/cudb/monitoring/preventiveMaintenance/CUDB_28_201108190300.log`

## 2.2.2 cudbApplicationCounters

The `cudbApplicationCounters` command executes application counters defined by the counter configuration file. Refer to *CUDB Application Counters*, Reference [7] for more information.

### 2.2.2.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.2.2 Syntax

```
cudbApplicationCounters -C | --Counter-file <counter_configuration_file> -U | --Unique <uniqueNumber> [-f | --file-config <configuration_file>] [-u | --user-facility <facility_level>] [-l | --licensing] [-D | --Debug] [-h | --help]
```

### 2.2.2.3 Command Options

The following command options can be used:

- `-C | --Counter-file <counter_configuration_file>`: Selects the file which contains the information required for generating counters and storing them in PLDB.
- `-f | --file-config`: Sets the XML file with the data of the entire CUDB system.
- `-U | --Unique`: Unique number greater than 0 between process instances.



- -u | --user-facility: Restricted for Ericsson personnel.
- -l | --licensing: Restricted for Ericsson personnel.
- -D | --Debug: Restricted for Ericsson personnel.
- -h | --help: Displays command help message and exits.

#### 2.2.2.4 Output

The command output shows the command version and OAM library version when executed successfully:

```
cudbApplicationCounters Ver.(1.3.1) OAM Lib.Version
(1.1.0)
```

#### 2.2.2.5 Examples of Use

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the System Controller (SC) of the node where the PLDB master is located:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the cudbApplicationCounters command as follows:

```
<CUDB_node_prompt> cudbApplicationCounters -C
/cluster/home/cudb/oam/performanceMgmt/appCounters/config/<app_counter>.conf -U 1 -u LOCAL1
```

The expected output must look similar to the below example:

```
cudbApplicationCounters Ver.(1.3.1) OAM Lib.Version
(1.1.0)
```

### 2.2.3 cudbApplyConfig

---



---

#### Attention!

The use of this command is restricted to installation only. To make configuration model changes persistent in regular operation and maintenance procedures, use the administrative operation applyConfig. Refer to the “applyConfig” section of *CUDB Node Configuration Data Model Description*, Reference [1] for more information on applyConfig.

---



---



The `cudbApplyConfig` command automatically makes any configuration model change persistent. If no successful result is returned by the command, the configuration updates do not take effect, even if they are stored in the model and are visible through new CUDB Configuration CLI sessions. Refer to *CUDB Node Configuration Data Model Description*, Reference [1] for more information on using the command.

### 2.2.3.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.3.2 Syntax

```
cudbApplyConfig [-s|--scope <scope>] [-v|--verbose]
[-h|--help]
```

### 2.2.3.3 Command Options

The following command options can be used:

- `-s | --scope <scope>`: Defines the scope of applying configuration changes. The available values of `<scope>` are as follows:
  - `all`: Requests configuration change application to all concerned entities. It is the default value.

---

---

### Attention!

This command option is deprecated. To make configuration model changes persistent, use the administrative operation `applyConfig`. Refer to the “`applyConfig`” section of *CUDB Node Configuration Data Model Description*, Reference [1] for more information on `applyConfig`.

---

---

- `init`: Restricted for Ericsson personnel.
- `init_c2l`: Restricted for Ericsson personnel.
- `-v | --verbose`: Restricted for Ericsson personnel.
- `-h | --help`: Displays command help message and exits.

### 2.2.3.4 Output

The command output is a log of the performed changes. The command result is provided at the end of the log.



### 2.2.3.5 Examples of Use

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC of the node where the PLDB master is located:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the `cudbApplyConfig` command as follows:

```
<CUDB_node_prompt> cudbApplyConfig
```

The expected output must be similar to the below example:

```
INFO - Starting...
INFO - ***** PREPARATION PHASE
STARTS *****
INFO - ***** PREPARATION PHASE
STARTS *****
INFO - ***** PREPARATION PHASE
STARTS *****
INFO - ***** PREPARATION PHASE
STARTS *****
INFO - ***** PREPARATION PHASE
STARTS *****

INFO - Waiting [30] seconds for CUDB OI to commit changes.

INFO - Checking if OI changes file exists. INFO - Done.
INFO - Reading OI changes file [/home/cudb/oam/configMgmt/
commands/config/cudbOiImmChanges.txt]...

INFO - OI changes file successfully read, parsing
changes...
INFO - Processing non info operations start.
INFO - Processing info operations start.
INFO - Processing info operations end.
INFO - OI changes file successfully parsed.
INFO - Done.

INFO - Retrieving current redundancy level.
INFO - Done.

INFO - Checking live hosts...

PING OAM1 (10.22.49.1) 56(84) bytes of data.

--- OAM1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time
0ms
```



```
rtt min/avg/max/mdev = 0.004/0.038/0.121/0.048 ms,  
ipg/ewma 0.235/0.084 ms  
PING OAM2 (10.22.49.2) 56(84) bytes of data.  
  
--- OAM2 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time  
3ms  
rtt min/avg/max/mdev = 0.156/0.272/0.605/0.192 ms,  
ipg/ewma 1.274/0.458 ms  
PING 10.22.49.3 (10.22.49.3) 56(84) bytes of data.  
  
--- 10.22.49.3 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time  
0ms  
rtt min/avg/max/mdev = 0.099/0.184/0.379/0.115 ms,  
ipg/ewma 0.269/0.292 ms  
PING 10.22.49.4 (10.22.49.4) 56(84) bytes of data.  
  
--- 10.22.49.4 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time  
0ms  
rtt min/avg/max/mdev = 0.118/0.162/0.280/0.068 ms,  
ipg/ewma 0.224/0.228 ms  
PING 10.22.49.5 (10.22.49.5) 56(84) bytes of data.  
  
--- 10.22.49.5 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time  
0ms  
rtt min/avg/max/mdev = 0.128/0.149/0.209/0.037 ms,  
ipg/ewma 0.201/0.183 ms  
PING 10.22.49.6 (10.22.49.6) 56(84) bytes of data.  
  
--- 10.22.49.6 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time  
0ms  
rtt min/avg/max/mdev = 0.102/0.127/0.191/0.037 ms,  
ipg/ewma 0.191/0.162 ms  
PING 10.22.49.7 (10.22.49.7) 56(84) bytes of data.  
  
--- 10.22.49.7 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time  
0ms  
rtt min/avg/max/mdev = 0.101/0.173/0.300/0.076 ms,  
ipg/ewma 0.234/0.243 ms  
PING 10.22.49.8 (10.22.49.8) 56(84) bytes of data.  
  
--- 10.22.49.8 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time  
0ms  
rtt min/avg/max/mdev = 0.120/0.143/0.196/0.030 ms,  
ipg/ewma 0.175/0.172 ms
```





```
PING 10.22.49.9 (10.22.49.9) 56(84) bytes of data.

--- 10.22.49.9 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time
0ms
rtt min/avg/max/mdev = 0.100/0.129/0.209/0.047 ms,
ipg/ewma 0.176/0.174 ms
PING 10.22.49.10 (10.22.49.10) 56(84) bytes of data.

--- 10.22.49.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time
0ms
rtt min/avg/max/mdev = 0.096/0.170/0.244/0.055 ms,
ipg/ewma 0.184/0.186 ms
INFO - Hosts checked. Live control blades [2] - live PL
blades [4] live DS blades [4].
INFO - Done.

INFO - Reading configuration file [/home/cudb/oam/configMg
mt/commands/config/sqlSentences.cfg]...
INFO - Configuration file parsing started.
INFO - Configuration file parsing finished.
INFO - Generating SQL sentences...
INFO - SQL sentences generated.
INFO - Done.

INFO - Reading configuration file [/home/cudb/oam/configMg
mt/commands/config/commands2Launch.cfg]...
INFO - Configuration file parsing started....
INFO - Configuration file parsing finished.
INFO - Generating commands to launch...
INFO - Commands to launch generated.
INFO - Reading files to detect...
INFO - Reading configuration file [/home/cudb/oam/configMg
mt/commands/config/files2Detect.cfg]...
INFO - Configuration file parsing started....
INFO - Configuration file parsing finished.
INFO - Adding commands to execute...
INFO - Done.

INFO - Reading configuration file [/home/cudb/oam/configMg
mt/commands/config/procs2Stop.cfg]...
INFO - Configuration file parsing started.
INFO - Configuration file parsing finished.
INFO - Building list of processes affected...
INFO - List of processes affected built.
INFO - Done.

INFO - ***** PREPARATION PHASE ENDS
*****
```



```
INFO - ***** PREPARATION PHASE ENDS
*****
INFO - ***** PREPARATION PHASE ENDS
*****
INFO - ***** PREPARATION PHASE ENDS
*****

INFO - ***** EXECUTION PHASE STARTS
*****
INFO - ***** EXECUTION PHASE STARTS
*****
INFO - ***** EXECUTION PHASE STARTS
*****
INFO - ***** EXECUTION PHASE STARTS
*****

INFO - SQL queries executed.
INFO - Done.

INFO - Done.

INFO - Done.

INFO - Launching commands...
INFO - Commands executed.
INFO - Done.

INFO - Executing processes restart procedure...
INFO - Processes restart procedure executed.
INFO - Done.

INFO - OI changes file moved to backup.
INFO - If it is needed to execute this command again for
the same OI changes set rename the file [/home/cudb/oam/
configMgmt/commands/config/cudbOiImmChanges.txt.bak] to
[/home/cudb/oam/configMgmt/commands/config/cudbOiImmChan
ges.txt].

INFO - ***** EXECUTION PHASE ENDS
*****
INFO - ***** EXECUTION PHASE ENDS
*****
INFO - ***** EXECUTION PHASE ENDS
*****
INFO - ***** EXECUTION PHASE ENDS
*****
```



## 2.2.4 cudbCheckConsistency

The `cudbCheckConsistency` command checks if the slave DS Units hold approximately as many rows in their tables as the corresponding master DS Unit in the same DSG. The lightweight consistency check relates to PLDB units as well. The check is considered to be failed if the difference of the row counts of any SQL table exceeds a certain value, given in percentage. Tables with only a few rows can be skipped.

`cudbCheckConsistency` can be executed manually, but it also runs regularly as a `cron` job. When executed manually, the default configuration values can be overridden with command line options. When it finds a failure while running as a `cron` job, it raises an alarm on the node which hosts the failed slave DS Unit. For further information about alarms related to this command, refer to *Storage Engine, Potential Data Inconsistency between Replicas Found in DS*, Reference [8] and *Storage Engine, Potential Data Inconsistency between Replicas Found in PLDB*, Reference [9].

**Note:** If any changes are made in the configuration file on a CUDB node, then the exact same configuration changes must be performed on all the other CUDB nodes as well. This is needed because `cudbCheckConsistency` operates at the CUDB system level instead of the CUDB node level. After the configuration is updated in the whole system, the following command must be issued on every SC of every CUDB node:

```
/etc/init.d/cudbCheckConsistencySrv restart
```

### 2.2.4.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.4.2 Syntax

- Format (1): `cudbCheckConsistency [-l | --locked] [-a | --alarms] [-m | --maxdiff-limit <LIMIT>] [-c | --check-limit <LIMIT>] [-v | --verbose <LEVEL>]`
- Format (2): `cudbCheckConsistency -h | --help`
- Format (3): `cudbCheckConsistency -o | --object-tables`

### 2.2.4.3 Command Options

The following command options can be used:

- `-h | --help`: Displays command help message and exits.
- `-l | --locked`: Refuses to run if started anywhere else than on the PLDB master CUDB node, on the HA active SC. This option is useful and mandatory for running the program as a `cron` job.



- `-a | --alarms`: Allows to raise alarms if the test fails. This option is useful and mandatory when the program runs as a `cron` job. When started interactively, there is no need to raise alarms.
- `-v | --verbose <LEVEL>`: Sets the verbosity level to the defined `<LEVEL>`. Accepted values: '1' or '2'. A value of '1' means that detailed information about the progress of the program is printed to the console, but these extra details do not go into the system log. Level '2' means that extra details are written both to the console and to the system log.
- `-m | --maxdiff-limit <LIMIT>`: Uses the defined value of `<LIMIT>` as the maximal tolerable difference in percentage (`MAXDIFF_LIMIT` configuration value). `<LIMIT>` must be a non-negative floating point value rounded to two decimal digits in the [ 0.00 ; 100.00 ] closed interval. For example, "0.58" means 0.58% difference is allowed. The test fails if the difference is greater than the value defined for `<LIMIT>`.
- `-c | --check-limit <LIMIT>`: Does not check tables that have fewer rows than the value defined for `<LIMIT>` in the master DS Unit (`CHECK_LIMIT` configuration value). `<LIMIT>` must be a non-negative integer.
- `-o | --object-tables`: Checks for any possible differences in the `OBJECT_CLASSES` and the `MULTI_VALUE_OBJECTS` tables.

**Note:** This option must not be run as a Cron job, but only from the command line.

#### 2.2.4.4 Output

The following example output is displayed:

```
[info] cudbCheckConsistency is running with options: '', MAXDIFF_LIMIT: 1.00%, CHECK_LIMIT: 100
[info] Acquiring mastership information.
[info] Checking consistency in slave DS units.
[info]   Node42-DSG0 consistency: OK - row count difference 0.00%
[info]   Node42-DSG1 consistency: OK - row count difference 0.00%
[info]   Node42-DSG2 consistency: OK - row count difference 0.00%
[info] Summary: slaves checked: 3 -> PASSED: 3, FAILED: 0, UNKNOWN: 0.
```

In this output, Node42-DSG0 refers to the PLDB.

#### 2.2.4.5 Examples of Use

Check the following examples of executing the command:

- `<CUDB_node_prompt> cudbCheckConsistency`

When executed like above, the command checks the consistency manually with the default values given in the configuration file. No alarms are raised.

- `<CUDB_node_prompt> cudbCheckConsistency -m 0.00 -c 0`



When executed like above, the command checks the consistency strictly so that no difference is accepted, and all the tables are checked. No alarms are raised.

#### 2.2.4.6 **cudbCheckConsistency Configuration File**

The configuration file of the command is located in the following directory:

```
/home/cudb/monitoring/replicaConsistency/config/cudbCheckConsistency.conf
```

This file is directly included by a Bash shell script with the following prerequisites:

- Comment lines must start with a # character.
- Settings must follow the `VARIABLE=VALUE` pattern without white spaces around the = operator.

##### **MAXDIFF\_LIMIT=1.00**

Maximum tolerable row count difference in PERCENTS between tables in the master DS Unit and the slave DS Unit. The difference is counted table-by-table, and if the difference is greater than the configured tolerance limit in any of the tables, the verdict is FAILED.

The value must be a floating point number with exactly two decimal digits, in the [ 0.00 ; 100.00 ] closed interval.

Default value: 1.00

Command line option: -m | --maxdiff-limit <LIMIT>

##### **CHECK\_LIMIT=100**

If a table in the master DS Unit has fewer rows than this check limit, then the table is skipped.

The value must be an integer number in the [ 0 ; INT\_MAX ) left-closed right-open interval, where INT\_MAX is the constant defined in *POSIX.1-2008 Base Specifications, Issue 7*.

Default value: 100

Command line option: -c | --check-limit <LIMIT>

##### **CRON\_TIMESPEC='37 0 \* \* \*'**

Cron job time specification. If this value is changed, then the value must also be changed on every other CUDB node of the CUDB system, and the



new value must be installed to `cron` on both SCs of every CUDB node. On a specific CUDB node, after saving the new value, the new settings can be installed to `cron` by issuing the following command on an SC:

```
ssh OAM1 '/etc/init.d/cudbCheckConsistencySrv restart'
```

```
ssh OAM2 '/etc/init.d/cudbCheckConsistencySrv restart'
```

The value must be either “disabled” or a valid `cron` job time specification.

Default value: '37 0 \* \* \*'

Command line option: -

## 2.2.5 cudbCheckReplication

The `cudbCheckReplication` command checks if the active database cluster replication channels are functional in the CUDB system. `cudbCheckReplication` can be executed manually, but it also runs regularly as a `cron` job. When executed manually, the default configuration values can be overridden with command line options. When it finds a failure while running as a `cron` job, it raises an alarm on the node which hosts the failed slave DS Unit. For further information about alarms related to this command, refer to *Storage Engine, Replication Stopped Working in DS*, Reference [10], and *Storage Engine, Replication Stopped Working in PLDB*, Reference [11].

**Note:** If any changes are made in the configuration file on a CUDB node, then the exact same configuration changes must be performed on every other CUDB node as well. This is required because `cudbCheckReplication` operates on the CUDB system level instead of the CUDB node level. After the configuration is updated in the entire system, execute the below command on every SC of every CUDB node:

```
/etc/init.d/cudbCheckReplicationSrv restart
```

### 2.2.5.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.5.2 Syntax

- Format (1): `cudbCheckReplication [-l | --locked] [-a | --alarms] [-b | --behind-limit <LIMIT>] [-v | --verbose <LEVEL>]`
- Format (2): `cudbCheckReplication -h | --help`

### 2.2.5.3 Command Options

The following command options can be used:



- `-h` | `--help`: Displays command help message and exits.
- `-l` | `--locked`: Refuses to run if started anywhere else than on the PLDB master CUDB node, on the HA active SC. This option is useful and mandatory for running the program as a `cron` job.
- `-a` | `--alarms`: Allows to raise alarms if an open replication channel does not work. This option is useful and mandatory for running the program as a `cron` job. When started interactively, there is no need to raise alarms.
- `-b` | `--behind-limit <LIMIT>`: After injecting test data into the master DS Unit of a DSG, the command waits for the specified amount of seconds (defined with `<LIMIT>`) before starting to check if the data showed up in the slave DS Units. The value of `<LIMIT>` must be a positive integer, and its configuration variable is `BEHIND_LIMIT`.
- `-v` | `--verbose <LEVEL>`: Sets the verbosity level to the defined `<LEVEL>`. Accepted values are '1' or '2'. A value of '1' means that detailed information about the execution progress is printed to the console, but these extra details are not included in the system log. Level '2' means that the extra details are written both to the console and to the system log.

#### 2.2.5.4 Output

The command can display the following example output:

```
[info] cudbCheckReplication is running with options: '', BEHIND_LIMIT: 10
[info] Acquiring mastership information.
[info] Acquiring DSG status information.
[info] Injecting verification data to master DS units.
[info] Sleeping 10 seconds to wait for replication.
[info] Checking replication in slave DS units.
[info]   Node42-DSG0 replication: OK
[info]   Node42-DSG1 replication: OK
[info]   Node42-DSG2 replication: OK
[info] Summary: channels checked: 3 -> PASSED: 3, FAILED: 0, UNKNOWN: 0.
```

#### 2.2.5.5 Examples of Use

Use the command as shown below to check the replication channels with a wait limit of 5 seconds, and ensure that the command prints only basic progress information, and does not raise any alarm if a check fails:

```
<CUDB_node_prompt> cudbCheckReplication -b 5
```

#### 2.2.5.6 cudbCheckReplication Configuration File

The configuration file of the command is located in the following directory:



```
/home/cudb/monitoring/replicaConsistency/config/cudbCheck
Replication.conf
```

This file is directly included by a `Bash` shell script with the following prerequisites:

- Comment lines must start with a `#` character.
- Settings must follow the `VARIABLE=VALUE` pattern without white spaces around the `=` operator.

### **BEHIND\_LIMIT=10**

This setting assumes that any data injected to a master DS Unit are replicated within the defined amount of seconds to the slave DS Units. If the verification data does not arrive after the amount of seconds defined with `BEHIND_LIMIT`, the test verdict is `FAILED`. The value must be an integer number in the `[ 1 ; INT_MAX )` left-closed right-open interval, where `INT_MAX` is the constant defined in *POSIX.1-2008 Base Specifications, Issue 7*.

Default value: 10

Command line option: `-b | --behind-limit <LIMIT>`

### **CRON\_TIMESPEC='7 0 \* \* \*'**

This setting is the `cron` job time specification. If this value is changed, then the value must also be changed in every other CUDB node of the CUDB system, and the new value must be installed to `cron` on both SCs of every CUDB node. On a specific CUDB node, after saving the new value, you can install the new settings to `cron` by issuing the following command on the SCs:

```
ssh OAM1 '/etc/init.d/cudbCheckReplicationSrv restart'
```

```
ssh OAM2 '/etc/init.d/cudbCheckReplicationSrv restart'
```

The value must be either “disabled” or a valid `cron` job time specification.

Default value: `'7 0 * * *'`

Command line option : -

## **2.2.6 cudbCollectInfo**

The `cudbCollectInfo` command is used to create a tarball of CUDB logs.

### **2.2.6.1 Requisites**

`cudbOperator` users have no access to run this command.





### 2.2.6.2 Syntax

```
cudbCollectInfo [-n | --node <node id>] [-d | --ds <dsg id>] [-a | --action <action name>] [-c | --no-compress] [-e | --no-encrypt] [-h | --help] [-x | --exit-on-error]
```

### 2.2.6.3 Command Options

The following command options can be used:

- `-h | --help`: Displays command help message and exits.
- `-n | --node <node id>`: Defines the node where logs must be collected. Use the ID of a CUDB node to define a specific node, or "all" to collect logs for all nodes.
- `-d | --ds <dsg id>`: Defines the DSGs where the logs must be collected. Use the ID of the DSG to define a specific DSG, or "all" to collect logs for all DSGs. This option only applies to `ndb_error_reporter` logs, which are retrieved from local DS Units belonging to the specified DSGs.
- `-c | --no-compress`: Disables the compression of the collected logs to an archive.
- `-e | --no-encrypt`: Disables the GPG encryption of the `tgz` archive.
- `-x | --exit-on-error`: Exits the command when an error is found during collection. Default behavior is not to exit.
- `-a | --action <action name>`: `<action name>` is the name of the log collecting action to execute. The available actions are as follows:
  - `ndb_error_reporter`
  - `collect_info`
  - `sm`
  - `esa`
  - `cluster_config`
  - `measurement`
  - `ldapfe`
  - `system_status`
  - `bc_server`
  - `evip`
  - `ddci`



- `bc_client`
- `upgrade`
- `all`: This action is equivalent to specifying all other actions together.

#### 2.2.6.4 Output

An example output of the command is provided below:

```
Creating /local/tmp/cudb_collect_info_20120412-104337/1 ... OK
```

```
Waiting 1 to finish ... OK
```

```
CREATING ARCHIVE ... OK
```

```
Encrypting archive ... OK
```

```
Removing unencrypted archive ... OK
```

```
REMOVING RAW DATA ... OK
```

```
Fetch the file: /local/tmp/cudb_collect_info_20120412-104337.c
```

#### 2.2.6.5 Examples of Use

Use the command as shown below to perform all log collection actions on Node 39:

```
<CUDB_node_prompt> cudbCollectInfo -n 39 -a all
```

### 2.2.7 cudbConsistencyMgr

The `cudbConsistencyMgr` command manages consistency check tasks in the CUDB system. Refer to *CUDB Consistency Check*, Reference [12] for more information on how to use the command and perform consistency check tasks.

#### 2.2.7.1 Requisites

The user permissions of executing `cudbConsistencyMgr` are as follows:

- Root permission is required to run the command.
- `cudbadmin` can run this command through `sudo` without password.
- `cudbOperator` users have no access to run this command without password.



### 2.2.7.2

#### Syntax

- Format (1): `cudbConsistencyMgr -h | --help`
- Format (2): `cudbConsistencyMgr -v | --version`
- Format (3): `cudbConsistencyMgr -l | --list`
- Format (4): `cudbConsistencyMgr -t | --taskhistory`
- Format (5): `cudbConsistencyMgr -r | --remove <taskId> [-f | --force]`
- Format (6): `cudbConsistencyMgr -o | --order ms -n | --node <nodeid> -p | --pl [--max-replica-lag <milliseconds>] [--alarm-severity-limit <threshold>] [-D | --debug] [--debugStderr] [--task-debug] [--task-verbose]`
- Format (7): `cudbConsistencyMgr -o | --order ms -n | --node <nodeid> -d | --dsg <dsgid> [--max-replica-lag <milliseconds>] [--alarm-severity-limit <threshold>] [-D | --debug] [--debugStderr] [--task-debug] [--task-verbose]`

### 2.2.7.3

#### Command Options

The following command options can be used:

- `-h | --help`: Displays command help message and exits.
- `-v | --version`: Prints version information and exits.
- `-l | --list`: Lists elements of the Pending Task List (PTL) and the Running Task List (RTL), then exits.
- `-t | --taskhistory`: Prints the contents of the local task history log on the screen, then exits.
- `-r | --remove <taskId>`: Removes the task specified by `<taskId>` from the PTL.
- `-f | --force`: Forces the removal of the task by trying to follow the task even in the RTL.
- `-o | --order ms`: Orders master-slave consistency check between the master and slave replicas selected by the `-d | --dsg`, `-p | --pl` and `-n | --node` command options.
- `-n | --node <nodeid>`: Defines the node ID of the slave replica which is subject to the consistency check.
- `-p | --pl`: Selects the PLDB for consistency check.



- `-d | --dsg<dsgid>`: Selects the DSG with the specified `<dsgid>` for consistency check.

**Note:** Either the `-p | --pl`, or `-d | --dsg` command option must be specified when executing the command.

- `[--max-replica-lag <milliseconds>]` (default value: 10000): This parameter sets the maximum tolerable replication distance between the master and the slave. The value is provided in milliseconds, and it defines the execution conditions for the check task. The value of 10000 means that the check task is not started, and respectively a running task is stopped, if the replication lag reaches or exceeds 10000 ms.
- `[--alarm-severity-limit <threshold>]` (default: 500): This parameter is used to control the severity of the alarm that is raised when inconsistency is found. If the number of inconsistent rows is higher than the `<threshold>` value, then a major severity alarm is raised instead of the default severity.
- `[--task-verbose]`: Restricted for Ericsson personnel.
- `[--task-debug]`: Restricted for Ericsson personnel.
- `-D | --debug`: Restricted for Ericsson personnel.
- `--debugStderr`: Restricted for Ericsson personnel.

#### 2.2.7.4

#### Output

The output of the `cudbConsistencyMgr` command depends on the specified command options. Some examples are provided below.

#### Scheduling-Related Output

If the scheduling is successful, then `cudbConsistencyMgr` prints the task ID as shown below, which can be used to track the task:

```
Task UTC_2014-09-23-17-05-42_N242_U0000001955 is put into the pending task list and will be executed on node 242.
```

#### Task Listing-Related Output

The output of the command uses two lists: the PTL lists the scheduled tasks, while the RTL lists the currently running check tasks. Examples for both lists are shown below:



[Site 1]

```
RTL: UTC_2014-09-11-00-10_N121_U0000000849
      checkType=ms,source=S1-N121-D1,check=S2-N122-D1,maxReplicaLag=2000,alarmSeverityLimit=
      verboseMode=off,debugMode=off

PTL:
```

[Site 2]

```
RTL: UTC_2014-09-09-11-00-16_N122_U0000000793
      checkType=ms,source=S2-N122-D2,check=S1-N121-D2,maxReplicaLag=2000,alarmSeverityLimit=
      verboseMode=off,debugMode=off

PTL: UTC_2014-09-09-11-00-17_N122_U0000000794
      checkType=ms,source=S2-N122-D2,check=S1-N121-D2,maxReplicaLag=2000,alarmSeverityLimit=
      verboseMode=off,debugMode=off
      UTC_2014-09-09-11-00-21_N122_U0000000795
      checkType=ms,source=S2-N122-D3,check=S1-N121-D3,maxReplicaLag=2000,alarmSeverityLimit=
      verboseMode=off,debugMode=off
      UTC_2014-09-09-11-00-22_N122_U0000000796
      checkType=ms,source=S2-N122-D3,check=S1-N121-D3,maxReplicaLag=2000,alarmSeverityLimit=
      verboseMode=off,debugMode=off
      UTC_2014-09-09-11-00-23_N122_U0000000797
      checkType=ms,source=S2-N122-D3,check=S1-N121-D3,maxReplicaLag=2000,alarmSeverityLimit=
      verboseMode=off,debugMode=off
      UTC_2014-09-09-11-00-23_N122_U0000000798
      checkType=ms,source=S2-N122-D3,check=S1-N121-D3,maxReplicaLag=2000,alarmSeverityLimit=
      verboseMode=off,debugMode=off
      UTC_2014-09-09-11-00-23_N122_U0000000799
      checkType=ms,source=S2-N122-D3,check=S1-N121-D3,maxReplicaLag=2000,alarmSeverityLimit=
      verboseMode=off,debugMode=off
```

## Task History-Related Output

The output in this case consists of log lines describing the following information:

- The time when each task was added to the PTL.
- The time when each task was started and finished.
- The result of the task executions.

The results are ordered by time, and are listed from the oldest to the newest, as shown below:

```
2014-09-23 15:53:49+02:00 SC_2_1 UTC_2014-09-23-13-53-48_N243_U0000000586: task found in PTL
checkType=ms,source=S2-N243-D1,check=S1-N242-D1,maxReplicaLag=2000,alarmSeverityLimit=500,verboseMode=off,debugMode=off

2014-09-23 15:53:49+02:00 SC_2_1 UTC_2014-09-23-13-53-48_N243_U0000000586: task execution started

2014-09-23 15:54:00+02:00 SC_2_1 UTC_2014-09-23-13-53-48_N243_U0000000586: task execution finished with result
Successful completion, no difference found. Exit code: 0.
```

### 2.2.7.5 Examples of Use

Check the following examples of executing the command:

- Execute the command as shown below to order a slave consistency check of DSG 33 on node 100:



```
<CUDB_node_prompt> cudbConsistencyMgr --order ms --node  
100 --dsg 33
```

Alternatively, use the shortened syntax of the command options as follows:

```
<CUDB_node_prompt> cudbConsistencyMgr -o ms -n 100 -d 33
```

- Execute the command as shown below to order a slave consistency check of the PLDB on node 200:

```
<CUDB_node_prompt> cudbConsistencyMgr --order ms --node  
200 --pl
```

Alternatively, use the shortened syntax of the command options as follows:

```
<CUDB_node_prompt> cudbConsistencyMgr -o ms -n 200 -p
```

- Execute the command as shown below to list the running and pending tasks:

```
<CUDB_node_prompt> cudbConsistencyMgr --list
```

- Execute the command as shown below to remove a known task from the PTL:

```
<CUDB_node_prompt> cudbConsistencyMgr --remove  
UTC_2014-09-02-11-32-46_N47_U0000000007
```

Alternatively, use the shortened syntax of the command option as follows:

```
<CUDB_node_prompt> cudbConsistencyMgr -r UTC_2014-09-02  
-11-32-46_N47_U0000000007
```

- Execute the command with the `--force` option as shown below to remove a known task from the RTL, or even abort its execution:

```
<CUDB_node_prompt> cudbConsistencyMgr --remove  
UTC_2014-09-02-11-32-46_N47_U0000000008 --force
```

Alternatively, use the shortened syntax of the command option as follows:

```
<CUDB_node_prompt> cudbConsistencyMgr -r UTC_2014-09-02-1  
1-32-46_N47_U0000000008 -f
```

- Execute the command as shown below to check the history of the scheduled tasks:

```
<CUDB_node_prompt> cudbConsistencyMgr --taskhistory
```

Alternatively, use the shortened syntax of the command option as follows:

```
<CUDB_node_prompt> cudbConsistencyMgr -t
```



## 2.2.8 cudbDataBackup

The `cudbDataBackup` command executes a data backup in the complete CUDB system. Refer to *CUDB Backup and Restore Procedures*, Reference [13] for further details about when to use it.

### 2.2.8.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.8.2 Syntax

- Format (1): `cudbDataBackup [-T | --Timeout <seconds>] [-R | --Retries-number <number_of_retries>] [-t | --time <seconds>] [-S | --Slack-backup <minutes>] [-F | --Force] [-q | --DisablePG] [-L | --Parallel-mode] [-u | --user-facility <facility>] [-p | --print] [-D | --Debug]`
- Format (2): `cudbDataBackup -s | --schedule <cron_expression> [-T | --Timeout <seconds>] [-R | --Retries-number <number_of_retries>] [-t | --time <seconds>] [-S | --Slack-backup <minutes>] [-F | --Force] [-q | --DisablePG] [-L | --Parallel-mode] [-u | --user-facility <facility>] [-p | --print] [-D | --Debug]`
- Format (3): `cudbDataBackup -U | --Unschedule <cron_expression> [-T | --Timeout <seconds>] [-R | --Retries-number <number_of_retries>] [-t | --time <seconds>] [-S | --Slack-backup <minutes>] [-F | --Force] [-q | --DisablePG] [-L | --Parallel-mode] [-u | --user-facility <facility>] [-p | --print] [-D | --Debug]`
- Format (4): `cudbDataBackup -h | --help`

### 2.2.8.3 Command Options

The following command options can be used:

- `-T | --Timeout <seconds>`: Restricted for Ericsson personnel.
- `-R | --Retries-number <number_of_retries>`: Defines the number of retry attempts to establish the connection to the CUDB node where a specific piece of the data backup program is taken, if it fails the first time. The default value is 2.
- `-t | --time <seconds>`: Defines the time (in seconds) between retries. The default value is 5 seconds.



- `-S | --Slack-backup <minutes>`: Defines the time (in minutes) available to finish the provisioning operations before starting the backup. The default value is 2 minutes.
- `-F | --Force`: Forces the backup, even if the notification to the Provisioning Gateway (PG) fails.
- `-q | --DisablePG`: Disables PG notification.
- `-L | --Parallel-mode`: Orders a backup for each CUDB node element (that is for the PLDB and all DS Units like DS1, DS2, and so on) at the same time instead of one after another (that is, not when the previous one is finished).
- `-u | --user-facility <facility>`: Restricted for Ericsson personnel.
- `-s | --schedule`: Creates a periodic system data backup according to the defined `<cron_expression>`. The cron expression follows the below UNIX format:

`<minute> <hour> <day_of_month> <month_of_year> <day_of_week>`

The accepted values and value ranges are as follows:

- `<minute>`: `<0 - 59>`
- `<hour>`: `<0 - 23>`
- `<day_of_month>`: `<1 - 31>`
- `<month_of_year>`: `<1 - 12>`
- `<day_of_week>`: `<0 - 7>`

**Note:** Values 0 and 7 for `<day_of_week>` both mean the same day, Sunday.

Every value can be replaced with an asterisk (“\*”), meaning all possible values for the field. See the below execution for an example:

```
cudbDataBackup -s "5 0 * * *"
```

**Note:** In case the `-s | --schedule` option is used, then any additional `cudbDataBackup` options listed in this section must be stated after this option.

- `-U | --Unschedule`: Removes a periodic system data backup according to the defined `<cron_expression>`. The cron expression follows the same format as described for the `-s | --schedule` command option. This command option must be executed with the same optional arguments that were used to set the schedule.





- `-p | --print`: Prints the information of the scheduled periodic system data backup.
- `-D | --Debug`: Restricted for Ericsson personnel.
- `-h | --help`: Displays command help message and exits.

#### 2.2.8.4 Output

The command output is a log of the performed changes. The command result is provided at the end of the log.

#### 2.2.8.5 Examples of Use

The following example procedure shows how to execute `cudbDataBackup` with the `-S | --Slack-backup` option:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the `cudbDataBackup` command as follows:

```
<CUDB_node_prompt> cudbDataBackup -S 5
```

The expected output must be similar to the below example:

```
/home/cudb/systemDataBackup
Listening for current PLDB and DSGs status reports (may
take up to 2 minutes)
Starting backup...
Before calling pg_notification
cudb_backup ver(1.3.4)
BEGIN MANAGEMENT FOR LAST BACKUP Backup
2013-10-31_15-09 finished successfully in :
PLDB in CUDB node 27
NDB node PL0
NDB node PL1
DS#2 in CUDB node 27
NDB node DS2_0
NDB node DS2_1
DS#1 in CUDB node 29
NDB node DS1_0
NDB node DS1_1
Attempting to de-block Provisioning Gateway. This may
take up to a couple of minutes.
```



## 2.2.9 cudbDbDiskManage

The `cudbDbDiskManage` command manages the space of the Binary Large Object (BLOB) attributes stored on the disk storage system for certain CUDB object classes.

### 2.2.9.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.9.2 Syntax

- Format (1): `cudbDbDiskManage -a | --allocate <objectClass> <size>`
- Format (2): `cudbDbDiskManage -l | --list <objectClass>`
- Format (3): `cudbDbDiskManage -h | --help`

### 2.2.9.3 Command Options

The following command options can be used:

- `-a | --allocate <objectClass> <size>`: Allocates extra space for the BLOB attributes stored on the disk storage system belonging to the object class defined with `<objectClass>`. The `<size>` is in megabytes and is rounded up to the following multiple of 128 as long as the provided value is greater than 128; otherwise, the exact size is added.
- `-l | --list <objectClass>`: Lists the occupied and free space for storing attribute values on the disk storage system for the object class defined with `<objectClass>`. Refer to *CUDB Node Schema Update*, Reference [14] for more information on the available object classes.
- `-h | --help`: Displays command help message and exits.

### 2.2.9.4 Output

The command reports `OK` in case of success, and `error` in case of failure.

### 2.2.9.5 Examples of Use

The following example procedure shows how to allocate a set amount of space on the disk storage system for a BLOB object with `cudbDbDiskManage`:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:



```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. `<CUDB_node_prompt> cudbDbDiskManage -a <BLOB_object_name_stored_on_disk> <Allocated_disk_space_MB>`

The expected output of the above command must look similar to the below example:

```
Obtaining SQL Servers Information ...
NODE <DS_ID or PL_ID> : <type>
Connection strings : <mysql ip:port> <mysql ip:port>
Disk space assigned to the object class <BLOB_object_name_stored_on_disk> before increase : <x> Megabytes
Disk space assigned to the object class <BLOB_object_name_stored_on_disk> after increase : <x> Megabytes
```

**Note:** The value of `<x>` in the above output must be greater than, or equal to the value defined with `<Allocated_disk_space_MB>`

## 2.2.10

### cudbDsgMastershipChange

The `cudbDsgMastershipChange` command is used to move the master of a DSG to the selected node. It must be launched on the CUDB node that will hold the new master replica.

When executed, the command first checks the PLDB replication status on the node where the command is invoked. If the PLDB is unable to synchronize or replication status info is not available, the mastership change will be rejected. After this, the command checks if the new master replica has a longer delay than the time specified by the `--time` parameter: if it does, it rejects the mastership change. Then, for a period of four seconds, all write operations toward the current master are rejected to allow the new master replica to catch up with the current master. After this four-second period passes, the command checks if the new master is completely synchronized with the current master to avoid any potential data loss. If the check succeeds, the master of the specified DSG or PLDB is changed to the node.

In case Automatic Mastership Change (AMC) is active, the system periodically tries to move the master of each DSG to a DSG replica with higher priority. To avoid undesired master movements, disable the AMC feature. Refer to the “Automatic Mastership Change” section of *CUDB High Availability*, Reference [15] for more information.

#### 2.2.10.1

#### Requisites

`cudbOperator` users have no access to run this command.



### 2.2.10.2

#### Syntax

- Format (1): `cudbDsgMastershipChange -d | --dsg <dsgId> [-t | --time <seconds>] [-D | --Debug] [-f | --force]`
- Format (2): `cudbDsgMastershipChange -p | --pl [-t | --time <seconds>] [-D | --Debug] [-f | --force]`
- Format (3): `cudbDsgMastershipChange -h | --help`

### 2.2.10.3

#### Command Options

The following command options can be used:

- `-d | --dsg <dsgId>`: Identifies the DSG where the master change is executed. The value of `<dsgId>` must be between 1 and 255.
- `-p | --pl`: Refers to the local PLDB cluster.
- `-t | --time <seconds>`: Defines the time (in seconds) of the allowed synchronization delay between the current master replica and the future master replica to perform the mastership change. In case the current master replica is not close enough, the command exits and does not perform the mastership change. The default value is 3 seconds.
- `-D | --Debug`: Restricted for Ericsson personnel.
- `-f | --force`: Forces to complete the mastership change, even in the following cases:
  - The PLDB is unable to synchronize on the node where the command is invoked.
  - The PLDB replication status information is unavailable on the node where the command is invoked.
  - Replication channels are unavailable.
  - The replication in the new target master does not align with the current master.

This option is not exclusive with `--time`. To force a mastership change with great distance between the new master and the current master, use a great value for the `--time` parameter.

- `-h | --help`: Displays command help message and exits.

### 2.2.10.4

#### Output

The possible outputs for the command are as follows:

- In case of successful mastership change:



- Processing DSG mastership change...  
cudbDsgMastershipChange: Success.  
DSG 2 master replica moved from node N2 to N3 successfully.
- Processing PL mastership change...  
cudbDsgMastershipChange: Success.  
PL master replica moved from node N2 to N3 successfully.
- In case of failed mastership change:
  - Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
DSG 2 master replica can not be moved to node N3 because replication with node N2 is not aligned.
  - Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
Replication channel is not up.

**Note:** In the two cases above, an error message is also provided describing the situation. In these cases, the `--force` option can be used to proceed with the mastership change. However, consider that in this case, the mastership change most probably results in data loss because the candidate master cannot be synchronized with the current master. In case of data loss, replication fails, and the *Storage Engine, Unable to Synchronize Cluster in DS, Major*, Reference [16] or *Storage Engine, Unable to Synchronize Cluster in PLDB, Major*, Reference [17] alarms are raised. Refer to the appropriate alarm OPI for more information on how to clear the alarm.

- Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
Replication is delayed more than 3 secs.

**Note:** In the latter example, the `--time` option can be used to increase the allowed synchronization time delay between replicas. If the mastership still cannot be changed because the candidate master cannot align with the current master, the `--force` option can be used in addition to the `--time` option to force the change.

- Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
DSG 2 master replica can not be moved. Check site SM leader logs.
- Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
The node already has the master replica of the specified DSG.



- Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
The DSG specified is masterless.
- Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
The DS hosting the specified DSG in the node where the command is invoked is disabled.
- Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
The node hosting the specified DSG has PL down or no SM available.
- Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
The specified DSG is not alive.
- Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
There is no replica distance info in BC cluster.
- Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
The PLDB is unable to synchronize on the node where the command is invoked.
- Processing DSG mastership change...  
cudbDsgMastershipChange: Failed.  
Replication status info for PLDB on the node where the command is invoked is not available.

### 2.2.10.5 Examples of Use

The following example procedure shows how to change the master of DSG 2 with `cudbDsgMastershipChange`:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>  
ss>
```

3. Execute the command as follows:

```
<CUDB_node_prompt> cudbDsgMastershipChange -d 2
```

The expected output must be similar to the below example:

```
Processing DSG mastership change...
```



```
cudbDsgMastershipChange: Success.
DSG 2 master replica moved from node 80 to 81
successfully.
```

## 2.2.11 **cudbDsgProvisioningManage**

The `cudbDsgProvisioningManage` command is used to disable or enable a specific DSG during the provisioning of Distribution Entries (DEs). When using the command, consider the following:

- When the provisioning lock is set on a DSG, no more DEs are assigned automatically to that DSG. However, this provisioning lock can be overridden (and therefore new DEs can be created on a provisioning-locked DSG) by specifying the affected DSG in the provisioning operation as the place where the new DEs must be stored.
- The provisioning lock does not affect every provisioning-related operation, but only the creation of new DEs. Existing DEs can still be deleted on a provisioning-locked DSG, and Lightweight Directory Access Protocol (LDAP) operations (Add, Add/Modify, Delete, and so on) can also be performed on data stored under the existing DEs.

### 2.2.11.1 **Requisites**

`cudbOperator` users have no access to run this command.

### 2.2.11.2 **Syntax**

- Format (1): `cudbDsgProvisioningManage -e | --enable <DSGId> [-D | --Debug]`
- Format (2): `cudbDsgProvisioningManage -d | --disable <DSGId> [-D | --Debug]`
- Format (3): `cudbDsgProvisioningManage -s | --status <DSGId> [-D | --Debug]`
- Format (4): `cudbDsgProvisioningManage -h | --help`

### 2.2.11.3 **Command Options**

The following command options can be used:

- `-e | --enable <DSGId>`: Enables provisioning in the specified DSG. `<DSGId>` must be a valid DSG identifier present in the system.
- `-d | --disable <DSGId>`: Disables provisioning in the specified DSG. `<DSGId>` must be a valid DSG identifier present in the system.



- `-s | --status <DSGId>`: Shows the provisioning status of the specified DSG. `<DSGId>` must be a valid DSG identifier present in the system. The possible return values are `enable` or `disable` for provisioning.
- `-D | --Debug`: Restricted for Ericsson personnel.
- `-h | --help`: Displays command help message and exits.

#### 2.2.11.4 Output

An example output of the command is shown below:

```
cudbDsgProvisioningManage: Failed.  
Invalid syntax  
  
cudbDsgProvisioningManage: Failed.  
Unrecognized command line argument or arguments  
  
cudbDsgProvisioningManage: Failed.  
DSG %d is not valid  
  
cudbDsgProvisioningManage: Failed.  
Unexpected error.  
  
cudbDsgProvisioningManage: Success.  
DSG %d is already enabled.  
  
cudbDsgProvisioningManage: Success.  
DSG %d is already disabled.  
  
cudbDsgProvisioningManage: Success.  
DSG %d is enabled for provisioning.  
  
cudbDsgProvisioningManage: Success.  
DSG %d is disabled for provisioning.
```

#### 2.2.11.5 Examples of Use

The following example procedure shows how to enable provisioning on DSG 1 with `cudbDsgProvisioningManage`:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>  
ss>
```

3. Execute the command as follows:





```
<CUDB_node_prompt> cudbDsgProvisioningManage --enable 1
```

The expected output must be similar to the below example:

```
cudbDsgProvisioningManage: Success
```

## 2.2.12 cudbGetLogs

The `cudbGetLogs` command collects preventive maintenance logs for log analysis. For more information about this command, refer to *CUDB Logchecker*, Reference [6].

### 2.2.12.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.12.2 Syntax

```
cudbGetLogs [-d | --debug] [-b | --bash-debug] [-c |  
--consistency-threshold <threshold>] [-f | --force-c-check]  
[-h | --help]
```

### 2.2.12.3 Command Options

The following command options can be used:

- `-d | --debug`: Restricted for Ericsson personnel.
- `-b | --bash-debug`: Restricted for Ericsson personnel.
- `-c | --consistency-threshold <threshold>`: Sets the lightweight consistency check error threshold (the default value is 500).
- `-f | --force-c-check`: Forces the lightweight consistency check to run.
- `-h | --help`: Displays command help message and exits.

### 2.2.12.4 Output

An example output for the command is provided below:

```
Starting /opt/ericsson/cudb/OAM/bin/cudbGetLogs ...  
Grepping logs and creating /home/cudb/monitoring/preventiv  
eMaintenance/CUDB_107_201201241136.log ...  
The log file is saved as : /home/cudb/monitoring/preventi  
veMaintenance/CUDB_107_201201241136.log
```



### 2.2.12.5 Examples of Use

- Execute the command as shown below to run it in normal mode:

```
<CUDB_node_prompt> cudbGetLogs
```

- Execute the command as shown below to run it in debug mode:

```
<CUDB_node_prompt> cudbGetLogs --debug
```

- Execute the command as shown below to run it in debug mode and also force a lightweight consistency check to run with a threshold of 1:

```
<CUDB_node_prompt> cudbGetLogs --debug --force-c-check  
--consistency-threshold 1
```

## 2.2.13 cudbHaState

The `cudbHaState` commands automatically prints the cluster status.

### 2.2.13.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.13.2 Syntax

**cudbHaState**

### 2.2.13.3 Command Options

Not applicable.

### 2.2.13.4 Output

The output shows the LDE and AMF cluster state, the CoreMW, COM state and the SI and SU states. The output must be similar to the below example:

```
LOTC cluster uptime:
-----
Sat Apr 20 18:33:30 2013
LOTC cluster state:
-----
Node safNode=SC_2_1 joined cluster | Sat Apr 20 18:33:40
2013
Node safNode=SC_2_2 joined cluster | Sat Apr 20 18:33:30
2013
Node safNode=PL_2_3 joined cluster | Sat Apr 20 18:35:39
2013
```



```

Node safNode=PL_2_4 joined cluster | Sat Apr 20 18:35:43
2013
Node safNode=PL_2_5 joined cluster | Sat Apr 20 18:35:43
2013
Node safNode=PL_2_6 joined cluster | Sat Apr 20 18:35:43
2013
Node safNode=PL_2_7 joined cluster | Sat Apr 20 18:35:48
2013
Node safNode=PL_2_8 joined cluster | Sat Apr 20 18:35:48
2013
Node safNode=PL_2_9 joined cluster | Sat Apr 20 18:35:53
2013
Node safNode=PL_2_10 joined cluster | Sat Apr 20 18:35:48
2013
AMF cluster state:
-----
saAmfNodeAdminState."safAmfNode=SC-1,safAmfCluster=myAm
fCluster": Unlocked
saAmfNodeOperState."safAmfNode=SC-1,safAmfCluster=myAmfCl
uster": Enabled
saAmfNodeAdminState."safAmfNode=SC-2,safAmfCluster=myAm
fCluster": Unlocked
saAmfNodeOperState."safAmfNode=SC-2,safAmfCluster=myAmfCl
uster": Enabled
saAmfNodeAdminState."safAmfNode=PL-3,safAmfCluster=myAm
fCluster": Unlocked
saAmfNodeOperState."safAmfNode=PL-3,safAmfCluster=myAmfCl
uster": Enabled
saAmfNodeAdminState."safAmfNode=PL-4,safAmfCluster=myAm
fCluster": Unlocked
saAmfNodeOperState."safAmfNode=PL-4,safAmfCluster=myAmfCl
uster": Enabled
saAmfNodeAdminState."safAmfNode=PL-5,safAmfCluster=myAm
fCluster": Unlocked
saAmfNodeOperState."safAmfNode=PL-5,safAmfCluster=myAmfCl
uster": Enabled
saAmfNodeAdminState."safAmfNode=PL-6,safAmfCluster=myAm
fCluster": Unlocked
saAmfNodeOperState."safAmfNode=PL-6,safAmfCluster=myAmfCl
uster": Enabled
saAmfNodeAdminState."safAmfNode=PL-7,safAmfCluster=myAm
fCluster": Unlocked
saAmfNodeOperState."safAmfNode=PL-7,safAmfCluster=myAmfCl
uster": Enabled
saAmfNodeAdminState."safAmfNode=PL-8,safAmfCluster=myAm
fCluster": Unlocked
saAmfNodeOperState."safAmfNode=PL-8,safAmfCluster=myAmfCl
uster": Enabled
saAmfNodeAdminState."safAmfNode=PL-9,safAmfCluster=myAm
fCluster": Unlocked

```



```

saAmfNodeOperState."safAmfNode=PL-9,safAmfCluster=myAmfCl
uster": Enabled
saAmfNodeAdminState."safAmfNode=PL-10,safAmfCluster=myAm
fCluster": Unlocked
saAmfNodeOperState."safAmfNode=PL-10,safAmfCluster=myAmfCl
uster": Enabled
CoreMW HA state:
-----
CoreMW is assigned as ACTIVE in controller SC-2
CoreMW is assigned as STANDBY in controller SC-1
COM state:
-----
COM is assigned as ACTIVE in controller SC-2
COM is assigned as STANDBY in controller SC-1
SI HA state:
-----
saAmfSISUHASState."safSu=SC-2,safSg=2N,safApp=ERIC-CUDB_CUD
BOI"."safSi=2N-1": active(1)
saAmfSISUHASState."safSu=SC-2,safSg=DS1_2N,safApp=ERIC-CUDB
_CS"."safSi=DS1_2N-1": active(1)
saAmfSISUHASState."safSu=SC-2,safSg=DS2_2N,safApp=ERIC-CUDB
_CS"."safSi=DS2_2N-1": active(1)
saAmfSISUHASState."safSu=SC-2,safSg=PLDB_2N,safApp=ERIC-CU
DB_CS"."safSi=PLDB_2N-1": active(1)
saAmfSISUHASState."safSu=SC-2,safSg=2N,safApp=ERIC-CUDB_LDA
PFE_MONITOR"."safSi=2N-1": active(1)
saAmfSISUHASState."safSu=SC-1,safSg=2N,safApp=ERIC-CUDB_CUD
BOI"."safSi=2N-1": standby(2)
saAmfSISUHASState."safSu=SC-1,safSg=2N,safApp=ERIC-CUDB_LDA
PFE_MONITOR"."safSi=2N-1": standby(2)
saAmfSISUHASState."safSu=SC-1,safSg=DS2_2N,safApp=ERIC-CU
DB_CS"."safSi=DS2_2N-1": standby(2)
saAmfSISUHASState."safSu=SC-1,safSg=DS1_2N,safApp=ERIC-CU
DB_CS"."safSi=DS1_2N-1": standby(2)
saAmfSISUHASState."safSu=SC-1,safSg=PLDB_2N,safApp=ERIC-CUD
B_CS"."safSi=PLDB_2N-1": standby(2)
saAmfSISUHASState."safSu=PL-3,safSg=2N,safApp=ERIC-CUDB_SOA
P_NOTIFIER"."safSi=2N-1": active(1)
saAmfSISUHASState."safSu=PL-4,safSg=2N,safApp=ERIC-CUDB_SOA
P_NOTIFIER"."safSi=2N-1": standby(2)
SU States:
-----
Status OK

```

### 2.2.13.5

### Examples of Use

```
<CUDB_node_prompt> cudbHaState
```



## 2.2.14 cudbLdapFeRestart

The `cudbLdapFeRestart` command is used to perform a rolling restart on the LDAP Front Ends (FEs).

By default, the LDAP FE processes in the node are restarted one by one. An LDAP FE process is not restarted until the previous one is not fully operative (waiting up to a maximum of 90 seconds). This ensures that the LDAP connections are equally balanced between all LDAP FEs after executing the command.

**Note:** The current connections of the LDAP FEs are dropped when the processes are restarted.

### 2.2.14.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.14.2 Syntax

```
cudbLdapFeRestart [-p | --parallel] [--no-prompt] [-h | --help]
```

### 2.2.14.3 Command Options

The following command options can be used:

- `-p | --parallel`: Restricted for Ericsson personnel.

---

### Caution!

Never use the command option above in a live system. Use it only during node installation, otherwise traffic loss and degradation can occur.

---

- `--no-prompt`: Restricted for Ericsson personnel.
- `-h | --help`: Displays command help message and exits.

### 2.2.14.4 Output

The command output is a log of the command progress. The result of the command is provided at the end of the log.

### 2.2.14.5 Examples of Use

The following example procedure shows how to execute `cudbLdapFeRestart`:



1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the command as follows:

```
<CUDB_node_prompt> cudbLdapFeRestart
```

The expected output must be similar to the below example:

```
Reading necessary data from /home/cudb/common/config
/cudbSystem.xml
Restarting slapd process in host 10.22.89.3
Stopping slapd...OK
Making sure cudbLdapFeMonitor is running (to start
slapd)...OK
Waiting for slapd to be ready...OK
slapd ready after 24 seconds
-----
Restarting slapd process in host 10.22.89.4
Stopping slapd...OK
Making sure cudbLdapFeMonitor is running (to start
slapd)...OK
Waiting for slapd to be ready...OK
slapd ready after 56 seconds
-----
Restarting slapd process in host 10.22.89.5
Stopping slapd...OK
Making sure cudbLdapFeMonitor is running (to start
slapd)...OK
Waiting for slapd to be ready...OK
slapd ready after 10 seconds
-----
Restarting slapd process in host 10.22.89.6
Stopping slapd...OK
Making sure cudbLdapFeMonitor is running (to start
slapd)...OK
Waiting for slapd to be ready...OK
slapd ready after 12 seconds
-----
Restarting slapd process in host 10.22.89.7
Stopping slapd...OK
Making sure cudbLdapFeMonitor is running (to start
slapd)...OK
Waiting for slapd to be ready...OK
slapd ready after 12 seconds
-----
```



```
Restarting slapd process in host 10.22.89.8
Stopping slapd...OK
Making sure cudbLdapFeMonitor is running (to start
slapd)...OK
Waiting for slapd to be ready...OK
ready after 11 seconds
-----
Restarting slapd process in host 10.22.89.9
Stopping slapd...OK
Making sure cudbLdapFeMonitor is running (to start
slapd)...OK
Waiting for slapd to be ready...OK
slapd ready after 7 seconds
-----
Restarting slapd process in host 10.22.89.10
Stopping slapd...OK
Making sure cudbLdapFeMonitor is running (to start
slapd)...OK
Waiting for slapd to be ready...OK
slapd ready after 15 seconds
-----
Restarting of slapd processes is finished
```

## 2.2.15 **cudbManageBCServer**

The `cudbManageBCServer` command manages the Blackboard Coordination (BC) servers in the CUDB node. Refer to *CUDB High Availability*, Reference [15] for further information about BC servers and the BC cluster.

### 2.2.15.1 **Requisites**

This command acts on the BC server running on the blade or Virtual Machine (VM) where it is executed.

### 2.2.15.2 **Syntax**

- Format (1): `cudbManageBCServer -check`
- Format (2): `cudbManageBCServer -check4`
- Format (3): `cudbManageBCServer -checkMode`
- Format (4): `cudbManageBCServer -start`
- Format (5): `cudbManageBCServer -stop`
- Format (6): `cudbManageBCServer -restart`
- Format (7): `cudbManageBCServer -h | --help`



### 2.2.15.3 Command Options

The following command options can be used:

- `-check`: Checks if a BC server is running on the blade or VM.
- `-check4`: By using a four-letter command, this option checks if the process is running and if it is connected to a BC cluster with quorum.
- `-checkMode`: Checks if the specific BC server is the leader of the cluster, or just a follower.
- `-start`: Starts a BC Server in the blade or VM.
- `-stop`: Stops a BC Server in the blade or VM.
- `-restart`: Restarts a BC Server in the blade or VM.
- `-h | --help`: Displays command help message and exits.

### 2.2.15.4 Output

When executed with the `-checkMode` option, the command returns an output similar to the below example:

```
cudbManageBCServer -checkMode
Actual Host=SC_2_1
Mode: follower
```

### 2.2.15.5 Examples of Use

```
<CUDB_node_prompt> cudbManageBCServer -checkMode
```

## 2.2.16 cudbManageLibDataDist

The `cudbManageLibDataDist` command checks if a specific CUDB node is using the default distribution policy or a custom one. Refer to *CUDB System Administrator Guide*, Reference [5] for further details about using the command.

### 2.2.16.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.16.2 Syntax

```
cudbManageLibDataDist [-l | --load] [-o | --unload] [-s | --status] [-h | --help]
```





### 2.2.16.3 Command Options

The following command options can be used:

- `-l | --load`: Loads a new library stored in the following temporary folder: `/home/cudb/oam/configMgmt/commands/tmp`.
- `-u | --unload`: Unloads the library stored in `/usr/lib/`.
- `-s | --status`: Checks the status of the distribution policy.
- `-h | --help`: Displays command help message and exits.

### 2.2.16.4 Output

The command must return an output similar to the below example:

```
Customer distribution policy active
```

### 2.2.16.5 Examples of Use

The following example procedure shows how to check the status of the distribution policy with `cudbManageLibDataDist -s`:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
ss>
```

3. Execute the command as follows:

```
<CUDB_node_prompt> cudbManageLibDataDist -s
```

The expected output must be similar to the below example:

```
CUDB default distribution policy active
```

## 2.2.17 cudbManageStore

The `cudbManageStore` command executes administrative orders over clusters installed in the CUDB node. Refer to *CUDB System Administrator Guide*, Reference [5] for more information about using the command.

### 2.2.17.1 Requisites

The requisites of executing this command are as follows:

- `cudbOperator` users have no access to run this command.



- The command requires execution confirmation for some options.

### 2.2.17.2

#### Syntax

- Format (1A): `cudbManageStore -p | --pl -o | --order <orderName> [--no-prompt] [-l | --location <backupPath>] [--Script-path | -S <sqlPath>] [--no-start-replication | -n] [ --no-ldapfe-restart | -x ] [--restore-stored-procedures | -s] [--fresh-backup | -f]`
- Format (1B): `cudbManageStore -p | --pl -o | --order initndb -i | --ndbid <ndbid>`
- Format (2A): `cudbManageStore -d | --ds {<dsId>,<dsId>,...} -o | --order} <orderName> [-l | --location <backupPath>] [--Script-path | -S <sqlPath>] [--no-start-replication | -n] [ --no-ldapfe-restart | -x ] [--restore-stored-procedures | -s] [ --fresh-backup | -f ]`
- Format (2B): `cudbManageStore -d | --ds <dsId> -o | --order initndb -i | --ndbid <ndbid>`
- Format (3): `cudbManageStore -a | --all -o | --order <orderName> [--no-prompt] [-l | --location <backupPath>] [--Script-path | -S <sqlPath>] [--no-start-replication | -n] [--restore-stored-procedures | -s] [ --fresh-backup | -f ]`
- Format (4): `cudbManageStore -h | --help`

### 2.2.17.3

#### Command Options

The following command options can be used:

- `-p | --pl`: Specifies the local PLDB cluster.
- `-d | --ds <dsId>`: Specifies the local DS clusters. `<dsId>` can take the following values:
  - 1 to 17: The `dsId` corresponding to each DSG cluster configured in the CUDB node. Applies to the specified DSG storage clusters.
  - `all`: Applies to all DSG clusters configured in the CUDB node.
- `-a | --all`: Refers to all the local clusters (both PLDB and DSG).
- `-o | --order <ordernam>`: Executes the specified administrative order. The allowed orders in a CUDB storage engine are as follows:
  - **backup**: Performs a cluster data backup.



- **initialize:** Starts the cluster from scratch and rules out all its previous data.
- **maintenance:** Takes the cluster out of service. Not involved in master movement decisions.
- **initndb:** Restricted for Ericsson personnel.
- **kickstart:** This order starts replication in the cluster. The order can be used only when a CUDB node is initially installed, when a system data restore or a group data restore is executed, or when a new DSG or PLDB replica is added into the CUDB system.
- **mysqldslvrestart:** Restricted for Ericsson personnel.
- **mysqldch1mstrestart:** Restricted for Ericsson personnel.
- **mysqldch2mstrestart:** Restricted for Ericsson personnel.
- **mysqldacsrestart:** Restricted for Ericsson personnel.
- **ndbrestart:** Performs a rolling restart on the database nodes in the cluster.
- **masterreplicationrestart:** Restarts master processes for a specific storage or DSG
- **ready:** Joins back the cluster to the CUDB system from maintenance mode.
- **refresh:** Forces the cluster to read configuration.
- **restart:** Restarts the cluster taking into account its previous data.
- **restore:** Loads a previous data backup in the cluster.
- **start:** Starts the cluster, taking into account its previous data when it was stopped.
- **status:** Shows the cluster status. The possible return values are as follows:
  - is being initialized
  - is being restored
  - is being performing a backup
  - is being stopped
  - is Stopped
  - is Maintenance Mode
  - is being (re)Started



- is being refreshed
- executing kickstart order
- is in ready mode
- **stop**: Stops all processes belonging to the cluster.
- **toggleddebug**: Toggles debug log for the Cluster Supervisor (CS) processes.
- **mgmnoderestart**: Forces to restart the NDB `mgm` in the node.
- **restorestoredprocedures**: Restores stored procedures for application counters.
- `--no-prompt`: Restricted for Ericsson personnel.
- `-l | --location <backupPath>`: Specifies the backup path from which backup files can be obtained from each NDB node belonging to the involved database cluster. If this path is not provided when the **restore** order is requested, then the default path of the cluster (specified with the `BackupDataDir` entry of the **config.ini** file in the `[ndbd default]` section) is applied.
- `-S | --Script-path`: Specifies a Structured Query Language (SQL) script to be run after the **restore** process is successfully finished. If `<sqlPath>` is not provided, the default script applies. This option is used only to restore the process over a data partition belonging to the PLDB.
- `-i | --ndbid`: Restricted for Ericsson personnel.
- `-n | --no-start-replication`: Used to disable automatic replication start after a successful restore operation. Use this option only after a Group Data Restore. Refer to *CUDB Backup and Restore Procedures*, Reference [13] for further information.
- `-x | --no-ldapfe-restart`: Used to disable reestablishing connections to the LDAP FE.
- `-f | --fresh-backup`: Restricted for Ericsson personnel.
- `-h | --help`: Displays command help message and exits.
- `-s | --restore-stored-procedures`: Stored procedures of application counters are also restored after a successful **restore** order.

**Note:** The `-l | --location`, `-S | --Script-path`, `-n | --no-start-replication` and `-s | --restore-stored-procedures` options are valid only for the **restore** order.




---



---

## Warning!

Some DSG cluster manipulation orders, such as starting a stopped cluster, or a cluster with `in maintenance` status, may raise or clear several SNMP alarms for a short period of time. However, once the order is executed, the situation regarding alarms reaches a stable state, coherent with the system state after executing the orders.

---



---

### 2.2.17.4 Output

An example of command output is shown below:

```
cudbManageStore stores to process: pl.
Store pl is in ready mode.
cudbManageStore command successful.
```

### 2.2.17.5 Examples of Use

The following examples show how to execute `cudbManageStore`. Before all command executions, the following procedure must be performed:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
ss>
```

Some examples of using `cudbManageStore` are provided below:

- Execute the `cudbManageStore` command as follows to stop the PLDB cluster in the current CUDB node:

```
<CUDB_node_prompt> cudbManageStore --pl -o stop
```

The expected output must be similar to the below example:

```
WARNING: Stopping a PLDB replica will take PLDB and all
DSG replicas on this node out of service. Are you sure
you want to continue (y/n)?
y
Action will be executed

cudbManageStore stores to process: pl.

Launching order stop to pl in dsgroup 0.
```



```
Waiting for stop order(s) to be completed in CUDB Node
44 for stores : pl.
stop order finished succesfully in CUDB Node 44 for
store pl.
stop order(s) completed in CUDB Node 44 for stores :
pl.
Stores where order stop was succesfully completed: pl.
cudbManageStore command successful.
```

- Execute the `cudbManageStore` command as follows to restart the PLDB and all DSG clusters in the current CUDB node:

```
<CUDB_node_prompt> cudbManageStore -a --order restart
```

The expected output must be similar to the below example:

```
WARNING: Restarting the PLDB replica will take PLDB
and all DSG replicas on this node out of service for a
while. Are you sure you want to continue (y/n)?
Y
Action will be executed
```

```
cudbManageStore stores to process: pl ds1 (in
dsgroup2) ds2 (in dsgroup3).
Launching order restart to pl in dsgroup 0.
Launching order restart to ds1 in dsgroup 2.
Launching order restart to ds2 in dsgroup 3.
Waiting for restart order(s) to be completed in CUDB
Node 44 for stores : pl ds1 ds2.
restart order finished succesfully in CUDB Node 44 for
store ds1.
restart order finished succesfully in CUDB Node 44 for
store ds2.
restart order finished succesfully in CUDB Node 44 for
store pl.
restart order(s) completed in CUDB Node 44 for stores :
pl ds1 ds2.
Stores where order restart was succesfully completed:
ds1 ds2 pl.
cudbManageStore command successful.
```

- Execute the `cudbManageStore` command as follows to create a backup for the local DS cluster 1:

```
<CUDB_node_prompt> cudbManageStore --ds 1 -o backup
```

The expected output must be similar to the below example:

```
cudbManageStore stores to process: ds1 (in dsgroup1).

Starting Backup ...
Launching order Backup for ds1 in dsgroup 1.
```



```
Waiting for backup order(s) to be completed in CUDB
Node 121 for stores : ds1.
backup order finished successfully in CUDB Node 121 for
store ds1.
BACKUP-999 renamed in PL_2_7 to /local/cudb/mysql/ndbd/
backup/BACKUP/BACKUP-2016-05-12_12-13
BACKUP-999 renamed in PL_2_8 to /local/cudb/mysql/ndbd/
backup/BACKUP/BACKUP-2016-05-12_12-13
backup order(s) completed in CUDB Node 121 for stores
: ds1.
Stores where order backup was succesfully completed:
ds1.
cudbManageStore command successful.
```

- Execute the `cudbManageStore` command as follows to move all local clusters in this CUDB node to maintenance mode:

```
<CUDB_node_prompt> cudbManageStore -d all -o maintenance
```

The expected output must be similar to the below example:

```
WARNING: Getting a DS cluster into maintenance mode
will take it out of service. Are you sure you want to
continue (y/n)?
y
Action will be executed
```

```
cudbManageStore stores to process: ds1 (in dsgroup1)
ds2 (in dsgroup2).
Launching order maintenance to ds1 in dsgroup 1.
Launching order maintenance to ds2 in dsgroup 2.
Waiting for maintenance order(s) to be completed in
CUDB Node 29 for stores : ds1 ds2.
maintenance order finished succesfully in CUDB Node 29
for store ds1.
maintenance order finished succesfully in CUDB Node 29
for store ds2.
maintenance order(s) completed in CUDB Node 29 for
stores : ds1 ds2.
Stores where order maintenance was succesfully
completed: ds1 ds2.
cudbManageStore command successful.
```

- Execute the `cudbManageStore` command as follows to restore backup and application counter procedures on local DS cluster 1:

```
<CUDB_node_prompt>cudbManageStore --ds 1 -o restore
--location <backupPath> -s
```

The expected output must be similar to the below example:



```

WARNING: Restoring a DS cluster will replace previous
contents. Are you sure you want to continue (y/n)?
Y
Action will be executed

Starting restore in CUDB Node 42 for store ds1,
backup path /local/cudb/mysql/ndbd/backup/BACKUP/BACKU
P-2016-04-11_15-46,
sql scripts path /home/cudb/storageEngine/config/schema
/ds/internal/restoreTempSql.
Waiting for restore order(s) to be completed in CUDB
Node 42 for stores : ds1.
restore order finished successfully in CUDB Node 42 for
store ds1.
restore order(s) completed in CUDB Node 42 for stores
: ds1.
Stores where order restore was successfully completed:
ds1.
Closing connections for all blades of DSUnitGroup 1.
Successfully restored application counter procedures
for ds1.
cudbManageStore command successful.

```

## 2.2.18 cudbMpstat

The `cudbMpstat` command monitors the CPU load on one or more blades or VMs in a CUDB node, with adjustable periodicity.

### 2.2.18.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.18.2 Syntax

- Format (1): `cudbMpstat [ (-i | --include-blade) <blade> | (-e | --exclude-blade) <blade> ] [ -d | --disable-clean ] [ (-p | --periodicity) <sec> ] [ (-I | --interval) <n> ]`
- Format (2): `cudbMpstat [ (-i | --include-blade) <blade> | (-e | --exclude-blade) <blade> ] [ -d | --disable-clean ] [ (-p | --periodicity) <sec> ] [ (-I | --interval) <n> ] (-f | --file) <filename> [(-r | --rotation) <m>]`
- Format (3): `cudbMpstat -h | --help`

### 2.2.18.3 Command Options

The following command options can be used:





- `-h | --help`: Displays command help message and exits.
- `-i | --include-blade <blade>`: Includes the blade or VM defined with `<blade>` in the group of monitored blades or VMs. Use “SC” or “PL” as values to include all SCs or payload blades or VMs, respectively.
- `-e | --exclude-blade <blade>`: Excludes the blade or VM defined with `<blade>` from the group of monitored blades or VMs. By default, all SCs and payload blades or VMs are monitored. Use “SC” or “PL” as values to exclude all SCs or payload blades or VMs.

**Note:** If neither the `--include-blade`, nor the `--exclude-blade` options are used, then the command monitors all SCs and payload blades or VMs.

- `-d | --disable-clean`: Disables clear screen so that the output can be logged into a file.
- `-p | --periodicity <sec>`: Sets the periodicity to the configured amount of seconds.
- `-I | --interval <n>`: Prints output for every `<n>` samples. If the value of `<n>` is larger than “1”, then the command prints the average/maximum for each entry.
- `-f | --file <filename>`: Prints the output to a file instead of the terminal. No coloring or interactive control is performed.
- `-r | --rotation <m>`: Sets the number of entries after which the log file is rotated. By default, the log file is not rotated.

The default values of the above command options are as follows:

- If the `--file` option is not used, then the monitoring information is written to the terminal. In this case, the default value of `-p | --periodicity` is “2”, while `-I | --interval` is “1”.
- If the `--file` option is used, then the monitoring information is written to a file. In this case, the default values of the options are as follows:

- `-p | --periodicity`: 10
- `-I | --interval`: 6
- `-r | --rotation`: 720

This means that when the command logs to a file, then it makes a sample every 10 seconds, outputs the statistics every minute, and rotates the log files every 12 hours by default.

## 2.2.18.4 Output

An example output for `cudbMpsstat` is shown below:



cpu/blade	SC_2_2	SC_2_1	PL_2_9	PL_2_8	PL_2_7	PL_2_6	PL_2_5	PL_2_4	PL_2_3	PL_2_10
all	0.37	20.29	1.06	1.5	1.06	0.62	0.87	0.31	0.37	0.68
0	1.51	100	2.48	0	0.5	0	0.5	0	0	0
1	0	11.5	5.03	3.48	3.5	0	6	0	3.48	3.98
2	0.5	5.46	0	1.48	2.48	0	0	0.5	0	0.5
3	0	10.95	0	1.5	0.5	0	0	0	0	1.48
4	0	7.95	0	1.5	1.48	0	0	0	0	0
5	0	5.93	0	1.98	0	4.95	0	2.48	0	0
6	0	10.45	0	1	0	0.98	0.5	0	0	0
7	0.51	11.93	0	1.5	0.5	0.5	0	0	0	0.5

### 2.2.18.5 Examples of Use

- Execute the command as shown below to disable the clear screen, so that the output can be logged to a file:

```
<CUDB_node_prompt> cudbMpstat --disable-clean
```

- Execute the command as shown below to run it with specific periodicity and interval settings:

```
<CUDB_node_prompt> cudbMpstat --periodicity 1 --interval 4
```

## 2.2.19 cudbPmJobReload

The `cudbPmJobReload` command restarts the Performance Management (PM) agent to allow it to get new configuration. Refer to *CUDB System Administrator Guide*, Reference [5] for more information on when to use it.

### 2.2.19.1 Requisites

The requisites of using the command are as follows:

- This command is executed in all CUDB nodes to ensure that the same PM configuration is used in the whole CUDB system.



- `cudbOperator` users have no access to run this command.

### 2.2.19.2 Syntax

```
cudbPmJobReload [-t | --time <timeout>] [-D | --Debug] [-u  
| --user-facility <facility_level>] [-h | --help]
```

### 2.2.19.3 Command Options

The following command options can be used:

- `-t | --time-out <timeout>`: Sets the timeout, in seconds, to try the operations. The default value is 10 seconds.
- `-D | --Debug`: Restricted for Ericsson personnel.
- `-u | --user-facility <facility-level>`: Restricted for Ericsson personnel.
- `-h | --help`: Displays command help message and exits.

### 2.2.19.4 Output

When executed, the command must return an output similar to the below example:

```
Stopping PmAgent in node 10.22.27.10 ... OK
Waiting for ESA PmAgent to go off line.
ESA PmAgent has been successfully stopped.
Starting PmAgent in node 10.22.27.10 ... OK
```

### 2.2.19.5 Examples of Use

The following example shows how to execute `cudbPmJobReload`:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>  
ss>
```

3. Execute the command as follows:

```
<CUDB_node_prompt> cudbPmJobReload
```

The expected output must be similar to the below example:

```
Stopping PmAgent in node 10.22.28.1 ... OK
```



```
Waiting for ESA PmAgent to go off line.
ESA PmAgent has been successfully stopped.
Starting PmAgent in node 10.22.28.1 ... OK
```

## 2.2.20 cudbPrepareStore

The `cudbPrepareStore` command creates databases and tables for the PLDB or the specified DS Unit. Refer to *CUDB System Administrator Guide*, Reference [5] for further details on when to use it.

### 2.2.20.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.20.2 Syntax

- Format 1 : `cudbPrepareStore -p | --pl [ -s | --scope <scopeValue>]`
- Format 2: `cudbPrepareStore -d | --ds <dsId> [ -s | --scope <scopeValue>]`
- Format 3: `cudbPrepareStore -a | --all [ -s | --scope <scopeValue>]`

### 2.2.20.3 Command Options

The following command options can be used:

- `-p | --pl`: Runs the command for the local PLDB cluster.
- `-d | --ds <dsId>`: Runs the command for the local DS cluster defined with `<dsId>`. The value of `<dsId>` is the ID of the specific DS Unit, indicated either with an integer between 1-17 (for example, 1 stands for DS1, while 17 means DS17), or with "all" if the command must be executed for all local DS Units.
- `-a | --all`: Runs the command for all local clusters (both PLDB and DSG) configured in the specific CUDB node.
- `-s | --scope <scopeValue>`: Optional command option defining the scope of the command. The value of `<scopeValue>` can be either `appsrv` (referring to application scope) or `all` (referring to all scope).

**Note:** If `--scope <scopeValue>` is not used, the command is executed with the `--scope all` setting by default.



#### 2.2.20.4 Output

An example output of the command is shown below:

```
Starting
Obtaining SQL Servers Information ...
Done
Skipping Step 1 ...
Executing Step 2 ...
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
cudb-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
identities-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
msc-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
cpsps-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
nph-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
auth-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
ftest-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
avg-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
eps-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
ims-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
sm-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
ldap-indexes-ds.sql on 10.22.23.5:15010
..ok
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
cudb-ds.sql on 10.22.23.6:15010
..ok
```



```

Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
identities-ds.sql on 10.22.23.6:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
msc-ds.sql on 10.22.23.6:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
csps-ds.sql on 10.22.23.6:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
nph-ds.sql on 10.22.23.6:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
auth-ds.sql on 10.22.23.6:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
ftest-ds.sql on 10.22.23.6:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
avg-ds.sql on 10.22.23.6:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
eps-ds.sql on 10.22.23.6:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
ims-ds.sql on 10.22.23.6:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
sm-ds.sql on 10.22.23.6:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
ldap-indexes-ds.sql on 10.22.23.6:15010
..ok
Done
Finished

```

### 2.2.20.5

#### Examples of Use

The following example shows how to prepare the databases and tables on DS1 with `cudbPrepareStore` in application scope:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the command as follows:



```
<CUDB_node_prompt> cudbPrepareStore --ds 1 -s appsrv
```

The expected output must be similar to the below example:

```
Starting
Obtaining SQL Servers Information ...
Done
Skipping Step 1 ...
Executing Step 2 ...
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
cudb-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
identities-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
msc-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
cpsps-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
nph-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
auth-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
ftest-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
avg-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
eps-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
ims-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
sm-ds.sql on 10.22.23.5:15010
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
ldap-indexes-ds.sql on 10.22.23.5:15010
..ok
..ok
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/
cudb-ds.sql on 10.22.23.6:15010
..ok
```



```
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
identities-ds.sql on 10.22.23.6:15010  
..ok  
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
msc-ds.sql on 10.22.23.6:15010  
..ok  
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
csps-ds.sql on 10.22.23.6:15010  
..ok  
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
nph-ds.sql on 10.22.23.6:15010  
..ok  
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
auth-ds.sql on 10.22.23.6:15010  
..ok  
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
ftest-ds.sql on 10.22.23.6:15010  
..ok  
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
avg-ds.sql on 10.22.23.6:15010  
..ok  
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
eps-ds.sql on 10.22.23.6:15010  
..ok  
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
ims-ds.sql on 10.22.23.6:15010  
..ok  
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
sm-ds.sql on 10.22.23.6:15010  
..ok  
Applying /home/cudb/storageEngine/config/schema/ds/appSrv/  
ldap-indexes-ds.sql on 10.22.23.6:15010  
..ok  
Done  
Finished
```

## 2.2.21 cudbReallocate

The `cudbReallocate` command performs the Subscription Reallocation (also known as "reallocation") of the Distribution Entries (DEs) between DSGs. When executing the command, the complete DE subtree is moved. Refer to *CUDB Subscription Reallocation*, Reference [18] for more information on when to use the command.





### 2.2.21.1 Requisites

The requisites of using the command are as follows:

- Execute the command in the node where the PLDB master replica is located. For more information, refer to *CUDB System Administrator Guide*, Reference [5].
- `cudbOperator` users have no access to run this command.

### 2.2.21.2 Syntax

- Format (1): Reallocate by moving DS entries from the source DSG:  
`cudbReallocate -s | --source <dsgId> [-d | --destination <dsgId>] -p | --entriespercentage <number 1-100> [-o | --output <filepath>] [-D | --Debug] [-n | --nthreads <number>]`
- Format (2): Reallocate a list of elements to a specified destination:  
`cudbReallocate -d | --destination <dsgId> -l | --list <filepath> [-o | --output <filepath>] [-D | --Debug] [-n | --nthreads <number>]`
- Format (3): Request the status of DSGs relevant to the reallocate command: `cudbReallocate -u | --status [-D | --Debug]`
- Format (4): Show command help message: `cudbReallocate [-h | --help]`

### 2.2.21.3 Command Options

The following command options can be used:

- `-u | --status`: Prints the occupancy percentage and the memory level of all DSGs in the system. The memory occupation of the DSG is determined by the replica with the highest occupation. The possible status of the output are the following:
  - Below the `eligible` level.
  - Below the `stop` level.
  - Below the `warning` level.
  - Above the `warning` level.
  - Above the `full` level.
- `-s | --source <dsgId>`: Defines the source DSG with `<dsgId>`. The value of `<dsgId>` must be an integer between 1-255. Refer to *CUDB Node Configuration Data Model Description*, Reference [1] for more information.



- `-d | --destination <dsgId>`: Defines the destination DSG on which entries are reallocated. The value of `<dsgId>` must be an integer between 1-255. If this command option is not used, then the command selects one or more suitable DSGs as reallocation targets automatically.
- `-p | --entriespercentage <number 1-100>`: Defines the percentage of the DS entries in the specific source DSG that is required to be moved away. The value must be an integer between 1-100.
- `-l | --list`: Defines the file containing the list of primary network identities which identify the DEs to reallocate. The file contains lines of strings with the following format: `netIDType:= netIDValue`. The file is case sensitive.

Typical values of `netIDType` for common applications are as follows:

- IMPI
- IMPU
- IMSI
- IPADDRESS
- MSISDN

An example for these values is provided below:

```
IMSI:=99999
MSISDN:=100003
```

- `-o | --output`: Defines the output file where information on DE reallocation failures is stored. The default name of the output file is `cudbReallocationResult <YYYY-MM-DD-HH24-MI>.txt`, and is stored in the current directory. The file is deleted if the `cudbReallocate` command finishes without errors. The contents of the output file depend on whether the `--list` option was used when executing the command:
  - If the `--list` option was used when executing the command, the output file contains one line per failed network identity, with the following format:

```
netIDType:= netIDValue; result <result>
```

In the above example, the value of `<result>` can be as follows:

- `already moved`, if the entry was already in the destination DSG.
- `invalid entry`, if the specified network identity cannot be translated into a valid DE.
- `not moved`, in case of any other faulty situation.



- If the `--list` option was not used when executing the command, then each line in the output file states the DN of one of the DEs which could not be moved. If the DN was moved, but there was an error deleting it from the source DSG, an extra "moved"-suffix is added to the DN.
- `-n | --nthreads`: Defines the number of threads assigned to perform the reallocation procedure. The greater the number is, the faster the reallocation process is performed (up to a certain limit).

**Note:** Greater values have greater potential impact on traffic, and can put more stress on the affected source and target DSGs.

Consider the following when configuring `--nthreads`:

- The default value (and therefore the default number of threads used for reallocation) is 1. This results in the slowest replication process, but also in the lowest impact on traffic.
- For faster reallocation, set `--nthreads` to any value ranging from 2 to 5.
- Use the reallocation command during low-traffic hours.
- Use it repeatedly but either with small chunks of subscribers (`--list`) or small percentages (`--entriespercentage`) at a time, to make sure that reallocating those small chunks of data fits into the low-traffic time window.
- `-D | --Debug`: Restricted for Ericsson personnel.
- `-h | --help`: Displays command help message and exits.

#### 2.2.21.4 Output

The following list contains some example output messages:

- Reallocating from DSG `<id>` to DSG `<id>`.  
...  
cudbReallocate: Success.
- Reallocating from DSG `<id>` to DSG `<id>`.  
cudbReallocate: Partial Success.  
Reallocate DSG `<n>` partially succeeded - unable to move some entries. Consult output file for entries that could not be moved.
- Reallocating from DSG `<id>` to DSG `<id>`.  
cudbReallocate: Partial Success.  
Reallocate DSG `<n>` partially succeeded - All entries were not moved to destination DSG `<y>`  
Refer to output file for details of failure entries where moved.



An entry of "already moved" indicates the entry is already in the destination DSG.  
An entry of "invalid entry" indicates that the entry was not found or is not a primary network identity.  
An entry of "not moved" indicates that could not be moved due to a temporary failure.

- Reallocating from DSG <id> to DSG <id>.  
cudbReallocate: Partial Success.  
Reallocate DSG <n> partially succeeded - All entries were not moved to destination DSG <y>  
No entries were moved to destination DSG <y>.  
An entry of "already moved" indicates the entry is already in the destination DSG.  
An entry of "invalid entry" indicates that the entry was not found or is not a primary network identity.  
An entry of "not moved" indicates that could not be moved due to a temporary failure.
- cudbReallocate: Failed.  
Unexpected error.
- cudbReallocate: Failed.  
Invalid format: unrecognised option '<option>'  
...
- cudbReallocate: Failed.  
Reconciliation is ongoing.
- cudbReallocate: Failed.  
Source DSG <n> is not valid.
- cudbReallocate: Failed.  
Invalid format: Invalid DSG value, should be in range 1 - 255
- cudbReallocate: Failed.  
The list <inputFile> could not be found or could not be opened.
- cudbReallocate: Failed.  
The output <outputFile> already exists.
- cudbReallocate: Failed.  
<outputFile>  
Reallocate DSG <n> failed - DS master change detected in source DSG <y>.
- cudbReallocate: Failed.  
<outputFile>  
Reallocate DSG <x> failed - DS master change detected in destination DSG <y>.



- cudbReallocate: Failed.  
`<outputFile>`  
 Reallocate DSG `<n>` failed - PLDB master change detected in destination.
- cudbReallocate: Failed.  
 Invalid format: Unexpected format
- cudbReallocate: Failed.  
 Invalid format: Parameter `[list]` requires parameter `[destination]`  
 ...
- cudbReallocate: Failed.  
 Invalid format: Parameter `[source]` and parameter `[list]` are mutually exclusive  
 ...
- cudbReallocate: Failed.  
 Invalid format: Source and destination DSG IDs should not be identical  
 ...
- cudbReallocate: Failed.  
 PLDB master is not in the local node.
- cudbReallocate: Failed.  
 PLDB is not present in the local node.
- cudbReallocate: Failed.  
 CudbReallocate is already running.
- cudbReallocate: Failed.  
 The output `<outputFile>` file opening error.
- cudbReallocate: Failed.  
 Destination DSG `<n>` is not valid.
- cudbReallocate: Failed.  
 CudbReallocate is already running.

**Note:** This output is displayed after an unexpected error in the command execution. This output is also displayed if reallocation is in process. If no other reallocation process is running, this output is displayed due to a previous reallocation process finished with an unexpected error. To unlock CudbReallocate, use the following command:

```
rm -rf /cluster/home/cudb/.lock.cudbReallocate
```

Do not delete this file if the command is up and running, to prevent unexpected behaviors.



### 2.2.21.5 Examples of Use

The following example shows how to check the occupancy percentage and memory level of all DSGs in the system with the `cudbReallocate` command:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the command as follows:

```
<CUDB_node_prompt> cudbReallocate --status
```

The expected output must be similar to the below example:

```
cudbReallocate: Success
DSG1 occupancy: 40%, Above the warning level
DSG2 occupancy: 40%, Below the stop level
```

**Note:** In case of error, the output file is generated with no moved entries.

## 2.2.22 cudbReconciliationMgr

The `cudbReconciliationMgr` command allows to check, print, and modify the pending reconciliation task list. Refer to *CUDB Data Storage Handling*, Reference [19] for further details about when to use it.

### 2.2.22.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.22.2 Syntax

- Format (1): `cudbReconciliationMgr -c | --check <dsId>`
- Format (2): `cudbReconciliationMgr -a | --add <dsId>`
- Format (3): `cudbReconciliationMgr -l | --list`
- Format (4): `cudbReconciliationMgr [-h | --help]`

### 2.2.22.3 Command Options

The following command options can be used:

- `-c | --check <dsId>`: Checks if the DSG identified with `<dsId>` is in the Pending Task List.



- `-a` | `--add`: Adds a new pending reconciliation task for the DSG if the DS master is on the current CUDB node.
- `-l` | `--list`: Lists all pending tasks, that is, the contents of the Pending Task List.
- `-h` | `--help`: Displays command help message and exits. If the command is executed without options, the command use and help information is shown.

#### 2.2.22.4 Output

The command output is a log of the execution progress. The command result is provided at the end of the log.

#### 2.2.22.5 Examples of Use

The following examples show how to execute `cudbReconciliationMgr`. Before all command executions, the following procedure must be performed:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

Some examples of using `cudbReconciliationMgr` are provided below:

- Execute `cudbReconciliationMgr` as shown below to check if DS1 is in the Pending Task List:

```
<CUDB_node_prompt> cudbReconciliationMgr -c 1
```

If DS1 is on the Pending Task List, the command has no output. Otherwise, the output provides the error(s) fetched from the database.

- Execute `cudbReconciliationMgr` as shown below to add a new pending reconciliation task for DS2:

```
<CUDB_node_prompt> cudbReconciliationMgr -a 2
```

If the task for DS2 is added, the command has no output. Otherwise, the output provides the error(s) fetched from the database.



## 2.2.23 cudbRemoteTrust

The `cudbRemoteTrust` command is used in scalability procedures during installation to establish new SSH trust relationships between the installed CUDB node and the rest of the CUDB nodes.

It is also used to disable legal warning banner to be displayed for internal CUDB logins. Refer to Ericsson personnel for more information.

### 2.2.23.1 Requisites

The requisites of using the command are as follows:

- This command must be executed in all CUDB nodes.
- `cudbOperator` users have no access to run this command.

### 2.2.23.2 Syntax

```
cudbRemoteTrust [-l | --local] [-a | --all [-p |  
--persistent]] [-b | --banner] [-h | --help]
```

### 2.2.23.3 Command Options

The following command options can be used:

- `-l | --local`: Establishes the trust relationships in the current blade or VM of the CUDB node where the command is executed.
- `-a | --all`: Establishes the trust relationships in every blade or VM of the CUDB node where the command is executed.
- `-p | --persistent`: Stores the trust relationship settings in the following location:

```
/home/cudb/common/config/sshTrust
```

**Note:** This command option can be executed only if the `--all` option is also executed.

This command option implies the execution of `--banner` option, too.

- `-b | --banner`: Disables legal warning banner for logins, targeting the local CUDB node over SSH, in case the source address of the SSH connection is a CUDB VIP from the CUDB network. For example, when an internal CUDB login is initiated from a remote node towards the local node.

**Note:** This command option will also be executed when the `--all` `--persistent` combined options are executed.





- `-h | --help`: Displays command help message and exits.

#### 2.2.23.4 Output

Successful execution provides no output.

#### 2.2.23.5 Examples of Use

The following example shows how to execute `cudbRemoteTrust` on every blade or VM of the current CUDB node:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the command as follows:

```
<CUDB_node_prompt> cudbRemoteTrust -a
```

### 2.2.24 cudbResumeProvisioningNotification

The `cudbResumeProvisioningNotification` command is used to send a notification to the Provisioning Gateway (PG) to resume provisioning. Refer to *Storage Engine, Backup Notification Failure To Provisioning Gateway*, Reference [20] for further details about when to use it.

#### 2.2.24.1 Requisites

`cudbOperator` users have no access to run this command.

#### 2.2.24.2 Syntax

```
cudbResumeProvisioningNotification [-u | --user-facility <facility>] [-D | --debug] [-h | --help]
```

#### 2.2.24.3 Command Options

The following command options can be used:

- `-u | --user-facility <facility>`: Restricted for Ericsson personnel.
- `-D | --debug`: Restricted for Ericsson personnel.
- `-h | --help`: Displays command help message and exits.



#### 2.2.24.4 Output

The command output is a log of the changes performed. The command result is provided at the end of the log.

#### 2.2.24.5 Examples of Use

The following example shows how to execute `cudbResumeProvisioningNotification`:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the command as follows:

```
<CUDB_node_prompt> cudbResumeProvisioningNotification
```

### 2.2.25 cudbServiceContinuity

The `cudbServiceContinuity` command is used to change DSG mastership manually to a working site that has been left in Minority after a CUDB system split situation. The `cudbServiceContinuity` command maximizes service, allowing traffic operation with the resources available in the minority partition. New masters will be elected for all DSGs configured in that site.

**Note:** PLDB mastership is not affected by the command.

Refer to *CUDB High Availability*, Reference [15] for further details about using the command.

#### 2.2.25.1 Requisites

The requisites of using the command are as follows:

- The site must be in a system split situation of Minority to accept this command.
- This command is not supported if the CUDB system is deployed in more than 3 sites.
- The command requires execution confirmation.

#### 2.2.25.2 Syntax

```
cudbServiceContinuity [-h | --help]
```



### 2.2.25.3 Command Options

The following command option can be used:

- `-h` | `--help`: Displays command help message and exits.

### 2.2.25.4 Output

The command can return the following output messages:

- In case of a minority system split situation, the command returns the following success message:

```
cudbServiceContinuity
WARNING: This command will take all masters excluding PLDB in this site! Are you sure you want to continue (y/n)
The node /cudb/actions already existed
Set in path: /cudb/actions DATA: serviceContinuity
Service Continuity successfully executed
```

- In case the execution of the command is not confirmed (`n` is selected instead of `y`), the command returns the following message:

```
cudbServiceContinuity
WARNING: This command will take all masters excluding PLDB in this site! Are you sure you want to continue (y/n)
No action executed
```

- When executed with the `-h` or `--help` option, the command returns following output:

```
cudbServiceContinuity -h
=====
USE of cudbserviceContinuity
=====
Format : cudbserviceContinuity [-h | --help]
Where:
      -h | --help           Shows command usage and help
```

- In case of a majority system split situation, the command returns the following failure message:

```
cudbServiceContinuity
WARNING: This command will take all masters excluding PLDB in this site! Are you sure you want to continue (y/n)
SC can not be set in a majority (majority[1, 2, 3] AR[] NS[])
Command failed
```

- In case of a symmetrical split situation, the command returns the following failure message:

```
cudbServiceContinuity
WARNING: This command will take all masters excluding PLDB in this site! Are you sure you want to continue (y/n)
SC can not be set in a evenSplit (evenSplit[2] AR[] NS[])
Command failed
```

**Note:** If the command fails for other reasons than the ones above, contact the next level of maintenance support.



### 2.2.25.5 Examples of Use

```
<CUDB_node_prompt> cudbServiceContinuity
```

## 2.2.26 cudbSwBackup

The `cudbSwBackup` command performs a software and configuration backup in the CUDB node. The command rotates software and configuration backup files automatically, and asks for confirmation when the oldest backup is about to be deleted. Refer to *CUDB System Administrator Guide, Reference* [5] for more information on using the command.

### 2.2.26.1 Requisites

The requisites of using the command are as follows:

- `cudbOperator` users have no access to run this command.
- The command requires execution confirmation for some options.

### 2.2.26.2 Syntax

- Format (1): `cudbSwBackup [-c | --create <backup name>] [[-f | --force] -r | --restore <backup name>] [-d | --delete <backup name>] [-i | --info] [-l | --list] [-s | --schedule '<cron expression>'] [-p | --print] [-h | --help]`
- Format (2): `cudbSwBackup -o | --delete-oldest -c | --create <backup name>`
- Format (3): `cudbSwBackup [--no-prompt] --restore <backup name>`

### 2.2.26.3 Command Options

The following command options can be used:

- `-c | --create <backup name>`: Creates a new CUDB software and configuration backup.
- `-r | --restore <backup name>`: Restores an already created CUDB software and configuration backup.
- `--no-prompt`: Restricted for Ericsson personnel.
- `-d | --delete <backup name>`: Deletes an already created CUDB software and configuration backup.



- `-i | --info`: Prints information about the CUDB and LDE backup location and file format.
- `-l | --list`: Lists the created CUDB software and configuration backups.
- `-s | --schedule <cron expression>`: Creates a periodic software and configuration backup according to the defined *<cron expression>*.
- `-p | --print`: Prints the information on the scheduled periodic software and configuration backup.
- `-o | --delete-oldest`: Removes the oldest backup without confirmation message, if more than 5 backups exist.

**Note:** This option must be followed by the `-c | --create` option.

- `-f | --force`: Allows SW restore when the `mysqlds` of the PLDB are down (sql backup is not restored) or the CUDB node does not have one defined.

**Note:** If the PLDB cluster is defined, is up, and replication is working after the LDE cluster reboot, restore the system monitor tables by executing the `cudbApplyConfig -s init` command. Otherwise, refer to the procedure in *Storage Engine, Unable to Synchronize Cluster in PLDB, Major*, Reference [17] to get replication up and running and to restore the sql backup.

- `-h | --help`: Displays command help message and exits.

#### 2.2.26.4

#### Output

The output of the command is a log of the performed actions. An example output is provided below:

```
Pack and compress all directories and files under
/home/cudb/* except swbackup, systemDataBackup and
automatedBackupStorage directories.
```

```
IMM data persisted
Obtaining SQL Servers Information ...
Creating tables backup into file /cluster/home/cudb/swback
up/newbackup-cudbSmpConfig.sql
Connecting to host 10.22.49.5:15000 , database:
cudb_system_monitor
let's tar the directory /cluster/home/cudb excluding
backup directory itself home/cudb/swbackup/
...
```

```
Let's execute the CMW command cmw-partial-backup-create
newbackup
```



The resulting file is saved in /cluster/storage/no-backup/newbackup.tar  
 Backup successfully created.  
 The backup files: newbackup, are located in directories:  
 /cluster/home/cudb/swbackup and  
 /cluster/storage/no-backup

### 2.2.26.5 Examples of Use

The following example shows how to create a new software and configuration backup (named newbackup) with cudbSwBackup:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the command as follows:

```
<CUDB_node_prompt> cudbSwBackup -c newbackup
```

The expected output must be similar to the below example:

```
Let's execute the CMW command cmw-partial-backup-create
newbackup
The resulting files will be saved in /cluster/storage/no-
backup/newbackup.tar
CMW backup completed
tar: Removing leading `/' from member names
/cluster/storage/no-backup/coremw/cmw-backup//newbackup.
tar.gz
/cluster/storage/no-backup/comsa_for_coremw-apr9010555/bac
kup/newbackup.tar.gz
/cluster/storage/no-backup/com-apr9010443/backup/newback
up.tar.gz
/cluster/snapshot//cmwea-snapshot-newbackup.tar.gz
Backup successfully created.
The backup files: newbackup, are located in directories:
/cluster/home/cudb/swbackup and
/cluster/storage/no-backup
```

```
<CUDB_node_prompt> cudbSwBackup -i
```

CUDB SW Backups are stored in two directories:  
 /cluster/home/cudb/swbackup and  
 /cluster/storage/no-backup  
 The backup files are:  
 - One tar backup file



- One gz backup file
- One sql backup file (Only in nodes with PLDB)

The following example shows how to restore a previously created software and configuration backup (named newbackup) with `cudbSwBackup`:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the command as follows:

```
<CUDB_node_prompt> cudbSwBackup --restore newbackup
```

The expected output must be similar to the below example:

```
WARNING: Restoring the CUDB software will replace previous
software and configuration for the CUDB node. Are you
sure you want to continue (y/n)?y
Check for ongoing or pending applyConfig...
Waiting for OIWorker to write to file and release lock.
Obtaining SQL Servers Information ...
Checking SQL connections...
```

## 2.2.27

### cudbSwVersionCheck

The `cudbSwVersionCheck` command gathers information on the installed software packages from both the object model and LDE, and then prints the current installation state for each node in the LDE cluster. It also compares the current state to the official package list of the installed CUDB release, and shows the differences between the reference file and the currently installed packages.

The package information listing and the comparison of the installed packages can be requested separately with the use of the `--packages` and `--compare` options, respectively. If none of these options are used when executing the command, both actions are performed.

#### 2.2.27.1

#### Requisites

`cudbOperator` users have no access to run this command.



### 2.2.27.2 Syntax

```
cudbSwVersionCheck [-p | --packages] [-c | --compare] [(-f
| --reffile <REFFILE>) | (-d | --refdir <REFDIR>)] [-h |
--help]
```

### 2.2.27.3 Command Options

The following command options can be used:

- `-h | --help`: Displays command help message and exits.
- `-p | --packages`: Prints package installation status.
- `-c | --compare`: Compares the package installation state to the `cudbReference` file and prints the differences, if there is any.

**Note:** Omitting the `-c` and `-p` options has the same effect as using both of them.

- `-f | --reffile <REFFILE>`: Restricted for Ericsson personnel.
- `-d | --refdir <REFDIR>`: Restricted for Ericsson personnel.

### 2.2.27.4 Output

An example output is provided below:

```
CUDB_67 SC_2_1# cudbSwVersionCheck
```

```
Checking SW on blades.....
```

SW num	2_1	2_2	2_3	2_4	2_5	2_6	2_7	2_8	2_9	2_10
1	XO	XO								
2			XO	XO	XO	XO	XO	XO	XO	XO
3	XO	XO								
4	XO	XO	XO	XO	XO	XO	XO	XO	XO	XO
5	XO	XO								
6	XO	XO								
7	XO	XO	XO	XO	XO	XO	XO	XO	XO	XO
8	XO	XO	XO	XO	XO	XO	XO	XO	XO	XO
9	XO	XO								
10	XO	XO								
11	XO	XO	XO	XO	XO	XO	XO	XO	XO	XO
12	XO	XO								
13	XO	XO								
14	XO	XO								
15	XO	XO								
16	XO	XO								
17	XO	XO								
18	XO	XO	XO	XO	XO	XO	XO	XO	XO	XO





19				XO	XO	XO	XO	XO	XO	XO	XO
20	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
21	XO	XO									
22				XO	XO	XO	XO	XO	XO	XO	XO
23	XO	XO									
24	XO	XO									
25	XO	XO									
26	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
27	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
28	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
29	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
30	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
31	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
32	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
33	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
34				XO	XO						
35	XO	XO									
36	XO	XO									
37				XO	XO	XO	XO	XO	XO	XO	XO
38	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
39	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
40	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
41	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
42	XO	XO									
43				XO	XO	XO	XO	XO	XO	XO	XO
44	XO	XO									
45	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
46	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
47	XO	XO									
48	XO	XO									
49				XO	XO	XO	XO	XO	XO	XO	XO
50	XO	XO									
51											
52	XO	XO		XO	XO	XO	XO	XO	XO	XO	XO
53	XO	XO									
54	XO	XO									
55	XO	XO									
56	XO	XO									
57	XO	XO									
58	XO	XO									

#### LEGEND:

X: queried by the "cmw-repository-list --node <blade> <sw\_string>" command  
O: queried by the "immlist <sw\_string>" command

SDPs imported to the LOTC cluster:

-----

SW num	SW string	Used
1	ERIC-COREMW_SC-CXP9017565_1-R5K01	XO



2	ERIC-LINUX_PAYLOAD-CXP9013152_3-R4F04	XO
3	ERIC-CUDB_SMBC_PROCESS-CXP9030366_6-R1A02	XO
4	ERIC-CUDB_BC_SERVER-CXP9030364_6-R1A01	XO
5	ERIC-CUDB_BC_CLIENT-CXP9030367_6-R1A02	XO
6	ERIC-CUDB_LDAPFE_MONITOR-CXP9015809_6-R1A10	XO
7	ERIC-CUDB_CBATMPPATCHES-CXP9030357_6-R1A10	XO
8	ERIC-COREMW_OPENSAT-CXP9017656_1-R5K01	XO
9	ERIC-CUDB_APPCOUNTERS-CXP9015318_6-R1A10	XO
10	ERIC-CUDB_BC_SERVER_MONITOR-CXP9030379_6-R1A01	XO
11	ERIC-COREMW_COMMON-CXP9017566_1-R5K01	XO
12	ERIC-COM-CXP9017585_2-R6K01	XO
13	ERIC-ComSa-CXP9017697_3-R2K01	XO
14	ERIC-CUDB_ESA-CXP9015816_6-R1A09	XO
15	ERIC-CUDB_CURATOR-CXP9030365_6-R1A02	XO
16	ERIC-CUDB_BCCLUSTER_CMD-CXP9030380_6-R1A01	XO
17	ERIC-CUDB_BACKUP_RESTORE-CXP9015818_6-R1A10	XO
18	ERIC-CUDB_BACKUP_CMD-CXP9016194_6-R1A05	XO
19	ERIC-CUDB_LDAPFE-CXP9015319_6-R1A10	XO
20	ERIC-CUDB_KEEPAIVE-CXP9015322_6-R1A10	XO
21	ERIC-CUDB_CS_SC_CMD-CXP9030358_6-R1A10	XO
22	ERIC-CUDB_CS_PL_CMD-CXP9019599_6-R1A10	XO
23	ERIC-CUDB_CS-CXP9015812_6-R1A10	XO
24	ERIC-CUDB_COUNTCOLLECTOR-CXP9015814_6-R1A07	XO
25	ERIC-CUDB_LIB_REALLOC-CXP9030372_6-R1A10	XO
26	ERIC-CUDB_LIB_BCCLI-CXP9030363_6-R1A01	XO
27	ERIC-CUDB_JAVA-CXP9015823_6-R1A04	XO
28	ERIC-CUDB_MANAGE_BCSERVER_CMD-CXP9030381_6-R1A01	XO
29	ERIC-CUDB_LIB_BOOST-CXP9030227_6-R1A04	XO
30	ERIC-CUDB_LIB_ALARMS-CXP9030217_6-R1A10	XO
31	ERIC-CUDB_LIB_SMMCP-CXP9015821_6-R1A10	XO
32	ERIC-CUDB_LIB_LDAP-CXP9030233_6-R1A10	XO
33	ERIC-CUDB_LIB_AMF-CXP9017480_6-R1A10	XO
34	ERIC-CUDB_LDAP_SOAP_NOTIF-CXP9016181_6-R1A10	XO
35	ERIC-CUDB_LDAP_CMD-CXP9030234_6-R1A03	XO
36	ERIC-CUDB_REALLOC_CMD-CXP9030373_6-R1A10	XO
37	ERIC-CUDB_MYSQL_NDBD-CXP9030225_6-R1A02	XO
38	ERIC-CUDB_MYSQL_COMMON-CXP9030223_6-R1A10	XO
39	ERIC-CUDB_LIB_LOGGER-CXP9030219_6-R1A10	XO
40	ERIC-CUDB_LIB_CDXQ-CXP9030216_6-R1A10	XO
41	ERIC-CUDB_LIB_CONF_MGMT-CXP9030218_6-R1A10	XO
42	ERIC-CUDB_MYSQL_MGMD-CXP9030224_6-R1A02	XO
43	ERIC-CUDB_PL_PLATFORM-CXP9015820_6-R1A10	XO
44	ERIC-CUDB_LOADMONITOR-CXP9017650_6-R1A10	XO
45	ERIC-CUDB_LIB_XERCES-CXP9030220_6-R1A01	XO
46	ERIC-CUDB_LIB_ZKCLI-CXP9030362_6-R1A01	XO
47	ERIC-CUDB_PDSH_CMD-CXP9030391_6-R1A01	XO
48	ERIC-CUDB_OI-CXP9015813_6-R1A10	XO
49	ERIC-CUDB_MYSQL_SERV-CXP9030226_6-R1A02	XO
50	ERIC-CUDB_RECONCILIATION-CXP9015808_6-R1A10	XO
51	ERIC-CUDB_NODE_CONFIG-CXP9015320_6-R1A10	XO
52	ERIC-CUDB_SECURITY-CXP9016002_6-R1A10	XO



```

53 |          ERIC-CUDB_SYSHEALTH-CXP9030087_6-R1A10 | XO |
54 |          ERIC-CUDB_SC_PLATFORM-CXP9015817_6-R1A10 | XO |
55 | ERIC-CUDB_SLES_SCREEN_CMD-CXP9030371_6-R1A01 | XO |
56 |          ERIC-LINUX_CONTROL-CXP9013151_3-R4F04 | XO |
57 |          ERIC-CUDB_SW_MGMT-CXP9015815_6-R1A10 | XO |
58 |          ERIC-CUDB_SM_CLIENT-CXP9015810_6-R1A10 | XO |

```

SW Version of the CUDB Node:

-----

Reference: /home/coremw\_appdata/incoming/cudb-install-temp/cudbReference

Version: CUDB13B CXP9020214/6 R1A10

Differences:

reference <|> current sw vers

-----

ERIC-CUDB\_SM\_PROCESS-CXP9015811\_6-R1A10

<

### 2.2.27.5 Examples of Use

Execute the command as follows to print the package installation status and compare the package installation state to the contents of the `cudbReference` file:

```
<CUDB_node_prompt> cudbSwVersionCheck --packages --compare
```

### 2.2.28 cudbSystemDataBackupAndRestore

The `cudbSystemDataBackupAndRestore` command automates the system data backup and restore processes. In case of `kickstart` command failure, the execution is not stopped. Furthermore, a retry mechanism exists, which executes the `kickstart` command a maximum of 2 more times with a 5-second delay in-between them. The encountered errors are stored and displayed in a summary at the script exit. In case of error, the exit code of `cudbSystemDataBackupAndRestore` is the number of errors encountered during execution. The `cudbSystemDataBackupAndRestore` script exits when running `kickstart_replication` and leaves most of the slave DS replica without the replication started. Refer to *CUDB Backup and Restore Procedures*, Reference [13] for further details on when to use the command.

**Note:** Do not run the command `cudbSystemDataBackupAndRestore` while configuration changes are being applied.

**Note:** To avoid collision with the Self-Ordered Backup and Restore function, do not perform system data backup or restore if a Self-Ordered Backup and Restore process is running on any node of the system.

See Section 2.2.29 on page 84 for further information about the status of the Self-Ordered Backup and Restore process.



### 2.2.28.1 Requisites

`cudbOperator` users have no access to run this command.

### 2.2.28.2 Syntax

- Format (1): `cudbSystemDataBackupAndRestore -c | --create [-F | --fast-copy] [-f | --forced] [-b | --cudbdatabackup-option <option_string>]`
- Format (2): `cudbSystemDataBackupAndRestore -c | --create -k | --keep-local [ -f | --forced ] [ -b | --cudbdatabackup-option <option_string>]`
- Format (3): `cudbSystemDataBackupAndRestore -r | --restore <backup_path> [-F | --fast-copy] [-s | --cudbdatarestore-option <option_string>]`
- Format (4): `cudbSystemDataBackupAndRestore -R | --restore-date <backup_date> [ -F | --fast-copy ] [ -f | --forced ] [ -s | --cudbdatarestore-option <option_string>]`
- Format (5): `cudbSystemDataBackupAndRestore -h | --help`

### 2.2.28.3 Command Options

The following command options can be used:

- `-h | --help`: Displays command help message and exits.
- `-c | --create`: Creates a system data backup.
- `-r | --restore <backup_path>`: Restores a system data backup from the specified directory path.
- `-R | --restore-date <backup_date>`: Restores a system data backup created on the defined date. The format of `<backup_date>` is `YYYY-MM-DD_HH24-MI`.

**Note:** The `--create`, `--restore` and `--restore-date` options are mutually exclusive.

- `-k | --keep-local`: Forces to keep the backup files on the CUDB nodes where they are generated. If this option is used, the created backups will not be collected to the CUDB node where the command was ordered.



**Note:** If `-k` | `--keep-local` is omitted, the generated backup files are collected to the CUDB node where the command was ordered, provided that there is enough storage space. The files are copied to the cluster area in the following directory:

`/home/cudb/automatedBackupStorage`

- `-F` | `--fast-copy`: Disables the default SCP speed limit when copying backup files from one node to another.

**Note:** Using this option yields a much higher network use.

- `-f` | `--forced`: Disables the checking of `/home/cudb/systemDataBackup` in case of backup creation or backup restore with the `-R` option. Furthermore, in case of backup creation, the checking of `/home/cudb/automatedBackupStorage` is disabled.
- `-b` | `--cudbdatabackup-option <option_string>`: Option arguments to pass to the `cudbDataBackup` command. The default value is `-L`.
- `-s` | `--cudbdatarestore-option <option_string>`: By default, `-L` is included, but it can be discarded. Additional options to use when doing a system data restore (`cudbDataRestore` parameters):
  - `-T` | `--Timeout <seconds>`: Restricted for Ericsson personnel.
  - `-R` | `--Retries-number <number of retries>`: Defines the number of retry attempts to establish the connection to the CUDB nodes where the specific DS Units or the PLDB must be restored. The default value is 2.
  - `-S` | `--Script-path <script>`: Defines the path where the post-restore commands are located. Do not use this option in a production environment.
  - `-u` | `--user-facility <facility>`: Enables Syslog facility for logging. Do not use this option in a production environment.
  - `-L` | `--parallel-mode`: Orders a restore for each CUDB node element (that is for the PLDB and all DS Units like DS1, DS2, and so on) at the same time instead of one after another (that is, not when the previous one is finished). The option does not guarantee that the restore takes place on the PLDB first, then on the DS Units in parallel.
  - `-D` | `--Debug`: Restricted for Ericsson personnel.

## 2.2.28.4 Output

The following output examples are provided below:

- The below example output indicates normal operation:



```

CUDB_44 SC_2_2# cudbSystemDataBackupAndRestore -c

Checking /home/cudb/automatedBackupStorage on local node
-----

Local node: ... EMPTY

Checking /home/cudb/systemDataBackup on nodes
-----

44: ... EMPTY
43: ... EMPTY

CREATE PART
-----

/home/cudb/systemDataBackup
Listening for current PLDB and DSGs status reports (may take upto 2 minutes)
Starting backup...
Before calling pg notification
BackupStart [INFO] - PS Notification trying stop notification was successfully sent.
PS Notification successfully sent to : 172.31.233.139
PS Notification failed to : NONE
PS not notified : NONE

cudb_backup      ver(1.3.4)

BEGIN MANAGEMENT FOR LAST BACKUP
Backup 2014-05-14_20-11 finished successfully in :
PLDB in CUDB node 44
    NDB node PL0
    NDB node PL1
    NDB node PL2
    NDB node PL3
DSG#1 in CUDB node 43
    NDB node DS1_0
    NDB node DS1_1
DSG#255 in CUDB node 43
    NDB node DS2_0
    NDB node DS2_1
Attempting to de-block Provisioning Gateway. This may take up to a couple of minutes.

COLLECT PART
-----

Copying backup piece from node 44 for dsg 0 for ndb node id 3 ... OK
Copying backup piece from node 44 for dsg 0 for ndb node id 4 ... OK
Copying backup piece from node 44 for dsg 0 for ndb node id 5 ... OK
Copying backup piece from node 44 for dsg 0 for ndb node id 6 ... OK
Copying backup piece from node 43 for dsg 1 for ndb node id 3 ... OK
Copying backup piece from node 43 for dsg 1 for ndb node id 4 ... OK
Copying backup piece from node 43 for dsg 255 for ndb node id 3 ... OK
Copying backup piece from node 43 for dsg 255 for ndb node id 4 ... OK

Please copy /home/cudb/automatedBackupStorage to a safe location to be able to restore the backup pieces when need.

ERROR SUMMARY
-----

None.

```

- The below example output shows an error caused by executing `cudbSystemDataBackupAndRestore` with the `--restore` option, while configuration changes are being applied:



```
CUDB_210 SC_2_2# cudbSystemDataBackupAndRestore --restore /home/cudb/automatedBackupStorage
```

```
VALIDATE PART
```

```
-----
```

```
NODE: 210
```

```
  DSG: 0
        NDB: 3
        NDB: 4
        NDB: 5
        NDB: 6
  DSG: 1
        NDB: 3
        NDB: 4
  DSG: 255
        NDB: 3
        NDB: 4
```

```
DISTRIBUTE PART
```

```
-----
```

```
Creating /home/cudb/systemDataBackup on node 210
```

```
Node 210 has DSG 0      -> copying ... id3-OK id4-OK id5-OK id6-OK
Node 210 has DSG 1      -> copying ... id3-OK id4-OK
Node 210 has DSG 255    -> copying ... id3-OK id4-OK
```

```
Creating /home/cudb/systemDataBackup on node 244
```

```
Node 244 has DSG 0      -> copying ... id3-OK id4-OK id5-OK id6-OK
Node 244 has DSG 1      -> copying ... id3-OK id4-OK
Node 244 has DSG 255    -> copying ... id3-OK id4-OK
```

```
EXTRACT PART
```

```
-----
```

```
Node 210 has DSG 0      -> extracting ... id3-OK id4-OK id5-OK id6-OK
Node 210 has DSG 1      -> extracting ... id3-OK id4-OK
Node 210 has DSG 255    -> extracting ... id3-OK id4-OK
Node 244 has DSG 0      -> extracting ... id3-OK id4-OK id5-OK id6-OK
Node 244 has DSG 1      -> extracting ... id3-OK id4-OK
Node 244 has DSG 255    -> extracting ... id3-OK id4-OK
```

```
RESTORE PART
```

```
-----
```

```
Starting System Data Backup with command cudbDataRestore -B 2015-07-27_12-40 -L 2>&1 ...
cudbDataRestore      ver(1.3.6)
```

```
Trying for restore the backup from [2015-07-27_12-40] data files.
```

```
Error executing the restore command in PL/DS node in CUDBNode#244
```

```
Error executing the restore command in PL/DS node in CUDBNode210. Failed due parallel execution of cudbApplyCo
```

```
Check logs and cudbApplyConfig.lock file in /home/cudb/oam/configMgmt/commands/config directory!
```

```
Restores process finished with error.
```

```
ERROR: unable to restore backup
```

```
ERROR SUMMARY
```

```
-----
```

```
ERROR: unable to restore backup
```

- The below example output shows another error in another erroneous scenario:



## ERROR SUMMARY

```
-----  
Unable to kickstart pl on node 24  
Unable to kickstart dsd 1 on node 24  
Unable to kickstart dsd 2 on node 24
```

### 2.2.28.5 Examples of Use

- Execute the command as follows to create a system data backup with the SCP limit disabled:

```
<CUDB_node_prompt> cudbSystemDataBackupAndRestore  
--create --fast-copy
```

- Execute the command as follows to create a system data backup with parallel mode enabled and PG notifications disabled:

```
<CUDB_node_prompt> cudbSystemDataBackupAndRestore -c -b  
"-q -L"
```

- Execute the command as follows to restore a system data backup from a specified location:

```
<CUDB_node_prompt> cudbSystemDataBackupAndRestore  
--restore /home/cudb/cudbSystemDataBackup
```

- Execute the command as follows to restore a system data backup created on the defined date:

```
<CUDB_node_prompt> cudbSystemDataBackupAndRestore  
--restore-date 2014-01-01_08-30
```

### 2.2.29 cudbSystemStatus

The `cudbSystemStatus` command automatically prints the general system status and the local node information. The output contains information on the following components:

- CUDB software version
- Blackboard Coordination (BC) server process status and BC cluster information.
- System Monitor (SM) status
- Master and Slave cluster status
- Network DataBase (NDB) status
- Replication status and active channel





- Alarms
- Database cluster server connection
- CUDB processes

**Note:** If the `cudbSystemStatus` command is executed on a system with different hardware types, the command output will also contain information on the hardware types used in the system. This hardware information is listed below the CUDB software version output.

### 2.2.29.1 Requisites

This command can be accessed by `cudbOperator` user using `sudo`.

### 2.2.29.2 Syntax

```
cudbSystemStatus [-a | --alarms] [-b | --bc-status]
[-c | --cluster-status] [-m | --check-mysql] [-p |
--check-cudbprocess] [-r | --replication-status] [-s |
--sm-status] [-t | --hardware-type] [-v | --version] [-C
| --new-cluster-status] [-R | --new-replication-status]
[-h | --help]
```

### 2.2.29.3 Command Options

The following command options can be used:

- `-a | --alarms`: Prints alarms.
- `-b | --bc-status`: Prints the status of the BC processes.
- `-s | --sm-status`: Prints the SM status.
- `-c | --cluster-status`: Restricted for Ericsson personnel.
- `-C | --new-cluster-status`: Prints cluster status.
- `-m | --check-mysql`: Checks database cluster server connections.
- `-p | --check-cudbprocess`: Checks CUDB processes relevant to the correct system behavior.
- `-r | --replication-status`: Restricted for Ericsson personnel.
- `-R | --new-replication-status`: Prints replication status. The possible output of the option for the DSGs are as follows:
  - M: Master
  - Sn: Slave - replication channel #n is active.
  - S?: Slave - Replication status is unknown.



- [S]: Slave - replication channels are down.
- Xm: Wrong state - masterless.
- Xu: Wrong state - unreachable.
- -t | --hardware-type: Prints the value of the hwType attribute of each CUDB node and whether each DSG in the CUDB system is running on hybrid hardware (that is, whether the DS Units of a certain DSG are hosted on nodes with different hardware types). Refer to *CUDB Node Configuration Data Model Description*, Reference [1] for more information about the hwType attribute.
- -v | --version: Prints the CUDB version.
- -h | --help: Displays command help message and exits.

**Note:** If no options are used when executing the command, `cudbSystemStatus` runs as if the `-b`, `-v`, `-s`, `-C`, `-R`, `-a`, `-m`, and `-p` options have been used. If at least one command option is executed, the command collects information related only to that command option.

Also, if no options are used when executing the command on a system with different hardware types, `cudbSystemStatus` runs the `-t | --hardware-type` option in addition to the options listed above.

**Note:** In case the CUDB system is deployed on native BSP 8100 hardware and in Hybrid Systems where some DSGs consist of different hardware platforms (GEP3 and GEP5), memory usage values are recalculated for DSGs located on GEP5 nodes. The memory levels for GEP5 DSGs shown here are different from those reported by the database cluster Management Client (`ndb_mgm`).

#### 2.2.29.4

#### Output

The command shows information on the relevant system status. BC cluster information is showing status of the BC clusters in all the sites, while SM information shows the status of the SM leader elected in every site.

If, for any reason, the BC cluster has no quorum enough to work as a cluster, it is indicated in the output, and no SM leader appears on the affected site.

#### 2.2.29.5

#### Examples of Use

Perform the following steps to execute the command:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:



```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
ss>
```

### 3. Execute the command as follows:

```
<CUDB_node_prompt> cudbSystemStatus
```

- The expected output must be similar to the below example:

```
<CUDB_node_prompt> cudbSystemStatus

Execution date: Sat Aug 10 11:41:23 CEST 2013

CUDB Software Version:
!- CUDB DESIGN DISTRIBUTION: CUDB13B CXP9020214/6 R1B549

Checking BC clusters:

[Site 1]

SM leader: Node 10 OAM1

Node 172.80.123.10
  BC server in SC_2_1 ..... running
  BC server in SC_2_2 ..... running (Leader)
  BC server in PL_2_5 ..... running

[Site 2]

SM leader: Node 11 OAM1

Node 172.80.123.11
  BC server in SC_2_1 ..... running
  BC server in SC_2_2 ..... running (Leader)
  BC server in PL_2_5 ..... running

[Site 3]

SM leader: Node 9 OAM2

Node 172.80.123.9
  BC server in SC_2_1 ..... running
  BC server in SC_2_2 ..... running (Leader)
  BC server in PL_2_5 ..... running

Checking System Monitor BC status in local node:

SM-BC in OAM1 ..... running
SM-BC in OAM2 ..... running

Checking Clusters status:
Node 9:
  PL Cluster (29%) .....OK
  DSG1 Cluster (23%) .....OK
  DSG255 Cluster (23%) .....OK
Node 10:
  PL Cluster (29%) .....OK
  DSG1 Cluster (23%) .....OK
  DSG255 Cluster (23%) .....OK
Node 11:
  PL Cluster (29%) .....OK
  DSG1 Cluster (22%) .....OK
  DSG255 Cluster (22%) .....OK

Checking NDB status:
  PL NDB's (2/2) .....OK
  DS1 NDB's (2/2) .....OK
  DS2 NDB's (2/2) .....OK

Checking Replication Channels in the System:
Node   | 9 | 10 | 11
=====
```



PLDB	---	S1	S1	M
DSG 1	---	S1	S1	M
DSG 255	---	S1	S1	M

Printing Detailed Replication Status for the Slave Replicas:

Node 9:

```
Replication in DSG0(Chan=1) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
Replication in DSG1(Chan=1) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
Replication in DSG255(Chan=1) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
```

Node 10:

```
Replication in DSG0(Chan=1) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
Replication in DSG1(Chan=1) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
Replication in DSG255(Chan=1) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
```

Node 11:

There are no Slave clusters

Printing Alarms...

```
[Aug 09 16:12:17] ( Root Login Failed @172.80.122.10 )
[Aug 09 16:16:00] ( Root Login Failed @10.22.10.199 )
```

Checking MySQL server connection:

```
MySQL Master Servers connection .....OK
MySQL Slave Servers connection .....OK
MySQL Access Servers connection .....OK
```

Checking Process:

```
OAMs.....
Cluster Supervisor.....Running
System Monitor BC.....Running
Reconciliation process.....Running in: OAM1
Smp-client.....Running
Management Server Process (ndb_mgmd).....Running
KeepAlive process.....Running
Alarm Correlator.....Running
ESA.....Running
LDAP counter.....Running in: OAM1
Log Handler process.....Running
PLs.....
Storage Engine process (ndbd).....Running
LDAP FE.....Running
KeepAlive process.....Running
MySQL server process (Master).....Running
MySQL server process (Slave).....Running
MySQL server process (Access).....Running
CudbNotifications process.....Running
LDAP FE Monitor process.....Running
DSs.....
Storage Engine process (ndbd).....Running
LDAP FE.....Running
KeepAlive process.....Running
MySQL server process (Master).....Running
MySQL server process (Slave).....Running
MySQL server process (Access).....Running
LDAP FE Monitor process.....Running
```

- In case of systems with different hardware types, the beginning of the expected output must be similar to the below example:

```
<CUDB_node_prompt> cudbSystemStatus
Execution date: Sat Mar 12 14:27:48 CET 2016

CUDB Software Version:
!- CUDB DESIGN DISTRIBUTION: CUDB16A CXP9020214/9 R2A99

Checking Hardware Type:
This system is working on following hardware types: EBS_GEP3, EBS_GEP5.
...
```

- In case the `-t | --hardware-type` option is used on hybrid systems, the expected output must be similar to the below example:



```
<CUDB_node_prompt> cudbSystemStatus -t
Execution date: Sat Mar 12 14:43:28 CET 2016
```

```
NodeId hwType
49 EBS_GEP3
82 EBS_GEP3
154 EBS_GEP5
157 EBS_GEP5

DsgId isHybrid
0 true
11 false
12 false
13 false
14 false
15 false
16 false
21 false
22 false
23 false
24 false
25 false
26 false
27 false
28 false
```

- In case replication is stopped in one or more DSGs and Self-Ordered Backup and Restore is enabled, the expected output is similar to the below example:

Checking Replication Channels in the System:

```
Node      | 9 | 10 | 11 |
=====
PLDB      | S1 | S1 | M  |
DSG 1     | S1 | [S] | M  |
DSG 255   | Xu | S1 | M  |
```

Printing Detailed Replication Status for the Slave Replicas:

```
Node 9:
Replication in DSG0(Chan=1) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
Replication in DSG1(Chan=1) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
Replication in DSG255(Chan=1) .... Dsg is Unreachable
Self-Ordered Backup and Restore is running: restoring backup

Node 10:
Replication in DSG0(Chan=1) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
Replication in DSG1(Chan=1) .... Stopped - From Thu Apr 14 08:54:02 CEST 2016
Self-Ordered Backup and Restore is running: transferring backup
Replication in DSG255(Chan=1) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0

Node 11:
There are no Slave clusters
```

## 2.2.30 cudbTakeAllMasters

The `cudbTakeAllMasters` command is used to change mastership manually to a working site in case of emergency (such as a CUDB system split situation), when a surviving site is available in the CUDB system with reduced service. `cudbTakeAllMasters` maximizes the service in the surviving site, allowing normal operation with the resources available in this site.

Refer to *CUDB High Availability*, Reference [15] for further details about the use of this command.



---

---

## Warning!

Execute this command only as a manual intervention in the special case that some sites are failing, and the system is in a split situation with reduced service.

---

---

### 2.2.30.1

#### Requisites

The requisites of using this command are as follows:

- The site must be in a system split situation of `Minority` to accept this command.
- This command is not supported if the CUDB system is deployed in more than 3 sites.
- The command requires execution confirmation.

### 2.2.30.2

#### Syntax

```
cudbTakeAllMasters [-h | --help]
```

### 2.2.30.3

#### Command Options

The following command option can be used:

- `-h | --help`: Displays command help message and exits.

### 2.2.30.4

#### Output

The command can return the following output messages:

- When executed with the `-h` or `--help` option, the command returns the following output:

```
cudbTakeAllMasters -h
=====
USE of cudbTakeAllmasters
=====
Format : cudbTakeAllmasters [-h | --help]
```

Where:

```
-h | --help           Displays command help message and exits
```

- In case of a `majority` system split situation, the command may return the following failure message:



```
cudbTakeAllMasters
WARNING: This command will take all masters in this site! \
Are you sure you want to continue (y/n)? y
TAM can not be set in a majority (majority[1, 2] AR[] NS[])
Command failed
```

- When executed on a CUDB deployment with more than 3 sites, the command returns the following message:

```
cudbTakeAllMasters
WARNING: This command will take all masters in this site! \
Are you sure you want to continue (y/n)? y
TAM can not be set in deployment with more than 3 sites
Command failed
```

**Note:** If the command fails for other reasons than the ones above, contact the next level of maintenance support.

## 2.2.30.5 Examples of Use

```
<CUDB_node_prompt> cudbTakeAllMasters
```

## 2.2.31 cudbUnitDataBackupAndRestore

The `cudbUnitDataBackupAndRestore` command resolves data inconsistencies between replicas by creating a data backup on the master replica and restoring this data backup on the inconsistent slave replica. Refer to *CUDB Backup and Restore Procedures*, Reference [13] for further details about when to use it

### 2.2.31.1 Requisites

The requisites of using the command are as follows:

- The command location must be included in the `PATH` system environment variable.
- `cudbOperator` users have no access to run this command.
- The command requires execution confirmation for some options.

### 2.2.31.2 Syntax

- Format (1): `cudbUnitDataBackupAndRestore (-p | --p1) (-n | --node) <nodeId> [(-f | --fast-copy)] [-q | --queued-copy] [--no-prompt]`



- Format (2): `cudbUnitDataBackupAndRestore (-d | --dsg) <dsgId> (-n | --node) <nodeId> [(-f | --fast-copy)] [-q | --queued-copy] [--no-prompt]`
- Format (3): `cudbUnitDataBackupAndRestore -h | --help`

### 2.2.31.3

#### Command Options

The following command options can be used:

- `-p | --pl`: Sets the PLDB cluster as the cluster to operate on. The command does not work, just exits with an error message, if any of the DS clusters in the node is the master replica of its data partition.
- `-d | --dsg <dsgId>`: Sets the DSG cluster defined with `<dsgId>` as the DSG cluster to operate on.
- `-n | --node <nodeId>`: Sets the slave node defined with the `<nodeId>` as the slave node on which the backup will be restored,
- `-h | --help`: Displays command help message and exits.
- `-f | --fast-copy`: Disables the default SCP speed limit when copying backup files from one node to another. Consider that this yields a much higher network use.
- `-q | --queued-copy`: Restricted for Ericsson personnel.
- `--no-prompt`: Restricted for Ericsson personnel.

### 2.2.31.4

#### Output

An example output of the command is provided below:





```

CUDB 41 SC 2_1# cudbUnitDataBackupAndRestore -d 1 -n 42
WARNING: Restoring the DS cluster will replace previous contents. Are you sure you want to continue (y/n)? y
Action will be executed

CREATE PART
-----

creating backup on node 41

cudbManageStore stores to process:  ds1 (in dsgroup1).

Starting Backup ...
Launching order Backup for ds1 in dsgroup 1.
Obtaining Mgm Information.
Trying backup on mgmt access 1, wait a moment ...
ndb mgm 10.22.41.1 2372 -e "START BACKUP 999 WAIT COMPLETED"
..ok
BACKUP-999 renamed in PL_2_7 to /local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46
BACKUP-999 renamed in PL_2_8 to /local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46

Backup finished successfully for store ds1.
Stores where order backup was successfully completed:  ds1.
cudbManageStore command successful.

DIR CREATE PART
-----

creating backup directory /local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46 on node 42 on blade 10.22.42.7
creating backup directory /local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46 on node 42 on blade 10.22.42.8

COPY PART
-----

copying backup files from node 41 blade 10.22.41.7:/local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46
to node 42 blade 10.22.42.7:/local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46 =>
copying backup files from node 41 blade 10.22.41.8:/local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46
to node 42 blade 10.22.42.8:/local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46

RESTORE PART
-----

Restoring backup on node 42

cudbManageStore stores to process:  ds1 (in dsgroup1).

Launching restore order in CUDB Node 42 to store ds1 in dsgroup 1.
Starting restore in CUDB Node 42 for store ds1, =>
backup path /local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46 =>
sql scripts path /home/cudb/storageEngine/config/schema/ds/internal/restoreTempSql.
Waiting for restore order(s) to be completed in CUDB Node 42 for stores :  ds1.
restore order finished successfully in CUDB Node 42 for store ds1.
restore order(s) completed in CUDB Node 42 for stores :  ds1.
Stores where order restore was successfully completed:  ds1.
Closing connections for all blades of DSUnitGroup 1.
No stored procedures were found to restore.
cudbManageStore command successful.

Performing cleanup
delete backup directory /local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46 on node 42 on blade 10.22.42.7
delete backup directory /local/cudb/mysql/ndbd/backup/BACKUP/BACKUP-2016-04-11_15-46 on node 42 on blade 10.22.42.8
cudbUnitDataBackupAndRestore successfully ended

```

## 2.2.31.5 Examples of Use

- Execute the command as follows to create a backup of DSG2 on the master node, and restore it on node 39:



```
<CUDB_node_prompt> cudbUnitDataBackupAndRestore -dsg 2  
--node 39
```

- Execute the command as follows to create a backup of the PLDB on the master node, and restore it on node 40:

```
<CUDB_node_prompt> cudbUnitDataBackupAndRestore -p -n 40
```

## 2.2.32 cudbUpdateUserInfo

---

---

### Attention!

This command is deprecated. To update the local node configuration with the latest changes of LDAP users, use the administrative operation `updateUserInfo`. Refer to *CUDB Node Configuration Data Model Description*, Reference [1] for more information on `updateUserInfo`.

---

---

The `cudbUpdateUserInfo` command updates the local node configuration with the latest changes of LDAP users in the CUDB node where the command is executed.

### 2.2.32.1 Requisites

The requisites of using the command are as follows:

- All previous configuration changes performed through the CUDB configuration CLI must be successfully applied on the node before executing the command.
- `cudbOperator` users have no access to run this command.

### 2.2.32.2 Syntax

```
cudbUpdateUserInfo [-D | --Debug] [-h | --help] [-i |  
--internal]
```

### 2.2.32.3 Command Options

The following command options can be used:

- `-D | --Debug`: Restricted for Ericsson personnel.
- `-h | --help`: Displays command help message and exits.
- `-i | --internal`: Restricted for Ericsson personnel.



#### 2.2.32.4 Output

The output informs about the success or failure of executing the command. An example output for each scenario is shown below:

- The output message of a successful execution must be similar to the below example:

```
Success:  User information successfully updated in node
<nodeId>
```

- The output message of a failed execution must be similar to the below example:

```
Failed.  Unexpected error
```

#### 2.2.32.5 Examples of Use

Perform the following steps to execute the command:

1. Connect to the CUDB node with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_IP_Address>
```

2. Connect to the SC with the following command:

```
<CUDB_node_prompt> ssh <admin_user>@<CUDB_Node_OAM_IP_Address>
```

3. Execute the command as follows:

```
<CUDB_node_prompt> cudbUpdateUserInfo
```

The expected output must be similar to the below example:

```
Success:  User information successfully updated in node
<nodeId>
```

## 2.3 Internal Commands

This section lists the internal commands of the CUDB system, available for Ericsson personnel only.

### 2.3.1 cudbBCServersRestart

This command can be executed by Ericsson personnel only.

### 2.3.2 cudbCopyAclsConfFile

This command can be executed by Ericsson personnel only.



### **2.3.3      cudbEvipConfigExtension**

This command can be executed by Ericsson personnel only.

### **2.3.4      cudbEvipEncapsulator**

This command can be executed by Ericsson personnel only.

### **2.3.5      cudbExecuteAllBlades**

This command can be executed by Ericsson personnel only.

### **2.3.6      cudbFollowLdapfeLogs**

This command can be executed by Ericsson personnel only.

### **2.3.7      cudbInstall**

This command can be executed by Ericsson personnel only.

### **2.3.8      cudbLicensingTool**

This command can be executed by Ericsson personnel only.

### **2.3.9      cudbManageSite**

This command can be executed by Ericsson personnel only.

### **2.3.10     cudbOomConfigurator**

The `cudbOomConfigurator` command is used only by a restricted process.

### **2.3.11     cudbParallelCommandRun**

This command can be executed by Ericsson personnel only.

### **2.3.12     cudbSdplInfo**

This command can be executed by Ericsson personnel only.

**2.3.13      cudbSdpInstall**

This command can be executed by Ericsson personnel only.

**2.3.14      cudbSdpReplace**

This command can be executed by Ericsson personnel only.

**2.3.15      cudbSdpUninstall**

This command can be executed by Ericsson personnel only.

**2.3.16      cudbSetDsgMaster**

This command can be executed by Ericsson personnel only.

**2.3.17      cudbSetPartitionStatus**

This command can be executed by Ericsson personnel only.

**2.3.18      cudbTpsStat**

This command can be executed by Ericsson personnel only.





## Glossary

For the terms, definitions, acronyms and abbreviations used in this document, refer to *CUDB Glossary of Terms And Acronyms*, Reference [21].







## Reference List

### CUDB Documents

- [1] *CUDB Node Configuration Data Model Description*
- [2] *CUDB Node Logging Events*
- [3] *CUDB Security and Privacy Management*
- [4] *CUDB Users and Passwords*, 3/00651-HDA 104 03/10
- [5] *CUDB System Administrator Guide*
- [6] *CUDB Logchecker*
- [7] *CUDB Application Counters*
- [8] *Storage Engine, Potential Data Inconsistency between Replicas Found in DS*
- [9] *Storage Engine, Potential Data Inconsistency between Replicas Found in PLDB*
- [10] *Storage Engine, Replication Stopped Working in DS*
- [11] *Storage Engine, Replication Stopped Working in PLDB*
- [12] *CUDB Consistency Check*
- [13] *CUDB Backup and Restore Procedures*
- [14] *CUDB Node Schema Update*
- [15] *CUDB High Availability*
- [16] *Storage Engine, Unable to Synchronize Cluster in DS, Major*
- [17] *Storage Engine, Unable to Synchronize Cluster in PLDB, Major*
- [18] *CUDB Subscription Reallocation*
- [19] *CUDB Data Storage Handling*
- [20] *Storage Engine, Backup Notification Failure To Provisioning Gateway*
- [21] *CUDB Glossary of Terms And Acronyms*

### Other Ericsson Documents



[22] *LDE Management Guide*