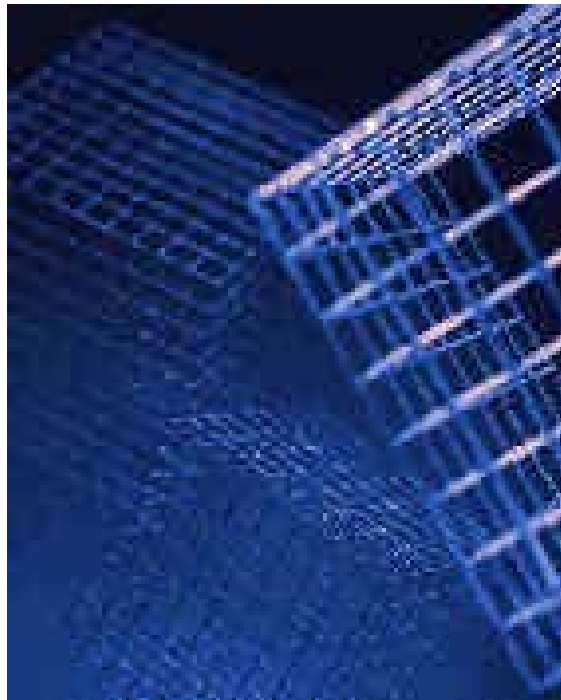








Backup and Recovery of MaxDB Using the Database Manager CLI



Version 7.5

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax
	Tip

Contents	
Icons	2
Contents	3
Introduction	4
Backup Media	4
Media Definitions	4
Examples	5
Definition of a File as Overwritable Medium for a Complete Data Backup ...	5
Definition of a Tape Device as Medium for a Complete Data Backup	5
Definition of a File as Medium for an Incremental Data Backup	5
Definition of a File as Medium for an Interactive Log Backup	5
Definition of a File as Medium for an Automatic Log Backup	6
Media Change	6
Backup	6
Backup Types	6
Commands for Creating a Backup	7
Creating a Complete Data Backup	7
Creating an Incremental Data Backup	7
Creating an Interactive Log Backup	7
Activating the Automatic Log Backup	7
Related Commands:	7
Deactivating the Automatic Log Backup	7
Checking the Status of the Automatic Log Backup	7
Example	7
Creating a Complete Data Backup with Media Change	7
Recovery	8
Examples	8
Recovering the Current Status of a Database Instance using One Complete Data, One Incremental Data, Two Log Backups and the Remaining Log in the Database	8
Recovering an Earlier Status of the Database Instance using One Complete Data and One Incremental Data without Log Backup	9

Introduction

The Database Manager is a database tool for managing database instances. You can use it to create, control, monitor, back up, and, if necessary, restore any number of database instances on the local computer or on remote computers.

The Database Manager CLI is the client program via which you establish the link to the DBM Server and exchange data. Then you can call up various functions by means of DBM options and DBM commands.

Calling up the Database Manager CLI is equivalent to opening a session. Once the commands you have entered have been processed, the session with the DBM Server is terminated.

The program is suitable for both interactive and batch operation. Script interfaces for programming database applications using Perl, Python, or Java are available.

See also: Documentation [Database Manager CLI](#)

Backup Media

For backups and recoveries, you need a backup medium. Data is backed up to this medium, from where it can then be restored again. Backup media include files, tapes, autoloaders, and pipes.

Define separate backup media for each database instance. You can combine multiple backup media in a group of parallel backup media.

Media Definitions

The path specifications and properties of the tape device, tapes, or files are grouped together in the media definition under a freely selectable, realistic name. The media for all types of backup and recovery operations can be reused under this logical name.

For example, the media definition includes information on whether this particular medium may be overwritten, the name and type of the backup device and, if tapes are used, their size. You must also specify the type of backups saved to each medium (complete data backup, incremental data backup, or log backup) and the name of the device/file with which it can be addressed through operating system functions.

The general syntax is as follows:

```
medium_put <name> [<location> [<type> [<backup_type> [<size>
[<block_size> [<overwrite> [<autoloader> [<os_command>
[<tool_type>]]]]]]]]]]
```

For a detailed description of the variables see the section *Creating or Changing a Backup Medium* in the documentation *Database Manager CLI*.

Examples

Definition of a File as Overwritable Medium for a Complete Data Backup

Log on as the user `DBM` with password `DBM` to the Database Manager CLI, specify the database `HOTELDB` to be backed up, specify the DBM command `medium_put` for creating the backup medium, specify the name of the medium to be created (`completeF`), its location consisting of path and file name (`/usr/maxDB/completetefile`), medium type (`FILE`), backup type (`DATA`), size (`0`) and block size (`0`), and whether it may be overwritten (`NO`):

```
dbmcli -d HOTELDB -u DBM,DBM medium_put completeF
/usr/maxdb/completetefile FILE DATA 0 0 NO
```

Definition of a Tape Device as Medium for a Complete Data Backup

Log on as the user `DBM` with password `DBM` to the Database Manager CLI, specify the database `HOTELDB` to be backed up, specify the DBM command `medium_put` for creating the backup medium, specify the name of the medium to be created (`completeT`), its location (`/dev/rft0`), medium type (`TAPE`), backup type (`DATA`), size (`64000` KB) and block size (`8`), and whether it may be overwritten (`NO`):

```
dbmcli -d HOTELDB -u DBM,DBM medium_put completeT /dev/rft0
TAPE DATA 64000 8 NO
```

Definition of a File as Medium for an Incremental Data Backup

Log on as the user `DBM` with password `DBM` to the Database Manager CLI, specify the database `HOTELDB` to be backed up, specify the DBM command `medium_put` for creating the backup medium, specify the name of the medium to be created (`incrF`), its location consisting of path and file name (`/usr/maxDB/incrfile`), medium type (`FILE`), backup type (`PAGES`), size (`0`) and block size (`0`), and whether it may be overwritten (`NO`):

```
dbmcli -d HOTELDB -u DBM,DBM medium_put incrF
/usr/maxdb/incrfile FILE PAGES 0 0 NO
```

Definition of a File as Medium for an Interactive Log Backup

Log on as the user `DBM` with password `DBM` to the Database Manager CLI, specify the database `HOTELDB` to be backed up, specify the DBM command `medium_put` for creating the backup medium, specify the name of the medium to be created (`logF`), its location consisting of path and file name (`/usr/maxDB/logfile`), medium type (`FILE`), backup type (`LOG`):

```
dbmcli -d HOTELDB -u DBM,DBM medium_put logF
/usr/maxdb/logfile FILE LOG
```

Definition of a File as Medium for an Automatic Log Backup

Log on as the user `DBM` with password `DBM` to the Database Manager CLI, specify the database `HOTELDB` to be backed up, specify the DBM command `medium_put` for creating the backup medium, specify the name of the medium to be created (`autologF`), its location consisting of path and file name

`/usr/maxDB/autologfile`, medium type (`FILE`), backup type (`AUTO`):

```
dbmcli -d HOTELDB -u DBM,DBM medium_put autologF
/usr/maxdb/autologfile FILE AUTO
```

Media Change

If the capacity of the medium is insufficient for the entire backup, you must carry out a media change. In other words, the backup is written to the first medium until this is full, and then a succeeding medium is used. If you foresee this occurring, call up the Database Manager CLI in session mode, because you must not interrupt the session during the backup operation.

Backup

Backup Types

The MaxDB database system supports:

- Data backups (complete and incremental) and
- Log backups (automatic and interactive)

With complete data backups, all the data is saved (but only the areas of the database instance that are actually occupied).

With an incremental data backup, all data that has been changed since the last complete data backup, is saved.

With log backups, all the pages that have been written to the log volume since the last log backup are saved. The log is saved in versioned files. This means that when the log is saved, the system appends a counter to the file name specified in the media definition (for example, `/usr/maxdb/logfile.001`). Please note that the versioned files generated take up storage space on the hard disk. To prevent the hard disk from being completely filled, you must use the tools in your operating system to save these files to a different location (for example, to tapes) and delete them from your hard disk. You can also use the DBM commands for archiving the version files with the help of external backup tools.

You have the option of backing up the log automatically or interactively. If automatic log backup is activated, a log segment is saved as soon as it has been filled. This log segment is then released again. This procedure makes it virtually impossible for the log to overflow and stop the database instance, and also saves you from having to constantly monitor the fill level of the log.



A prerequisite for creating a log backup is that you have already created a complete data backup for the current database instance.

Commands for Creating a Backup

Creating a Complete Data Backup

```
dbmcli -d <database_name> -u
<dbm_user_name>,<dbm_user_password> -uUTL -c backup_start
completeT
```

Creating an Incremental Data Backup

```
dbmcli -d <database_name> -u
<dbm_user_name>,<dbm_user_password> -uUTL -c backup_start
incrF
```

Creating an Interactive Log Backup

```
dbmcli -d <database_name> -u
<dbm_user_name>,<dbm_user_password> -uUTL -c backup_start logF
```

Activating the Automatic Log Backup

```
dbmcli -d <database_name> -u
<dbm_user_name>,<dbm_user_password> autolog_on
```

Related Commands:

Deactivating the Automatic Log Backup

```
dbmcli -d <database_name> -u
<dbm_user_name>,<dbm_user_password> autolog_off
```

Checking the Status of the Automatic Log Backup

```
dbmcli -d <database_name> -u
<dbm_user_name>,<dbm_user_password> autolog_show
```

Example

Creating a Complete Data Backup with Media Change

1. Call up the Database Manager CLI in session mode
2. Start complete data backup
3. Change media during data backup

```
dbmcli -d <database_name> -u
<dbm_user_name>,<dbm_user_password> -uUTL
backup_start completeT
backup_replace completeT
```

Recovery

Before you recover the database instance after a database error, you should decide which earlier version of the database instance you want to recover. To recover the **current** database status after a database error (in other words, the status immediately prior to the error), you must reload all the log backups to the system that were carried out after the start of the last data backup involved. An **older** database status can be recovered if only data backups are used.

If necessary, you must import the existing backups in the following sequence:

- A complete data backup
- An incremental data backup that may exist
- Then the existing log backups
(Files from log backups that were saved by means of the operating system to other locations - for example, tape devices - must be made available again in file form prior to the start of the recovery operation.)



When you recover data, you must use the Database Manager CLI in session mode, since recovery operations do not permit you to interrupt the session.

You can only recover the database instance in the `ADMIN` operational state of the database.

Examples

Recovering the Current Status of a Database Instance using One Complete Data, One Incremental Data, Two Log Backups and the Remaining Log in the Database

1. Call up Database Manager CLI in session mode
2. Import complete data backup
3. Import incremental data backup
4. Import log backup version 001
5. Change medium inserting log backup version 002

```
dbmcli -uUTL -d <database_name> -u
<dbm_user_name>,<dbm_user_password>
recover_start completeT
recover_start incrF
recover_start logF LOG 001
recover_replace logF /usr/maxdb/logfile 002
```

The Database then continues recovering the log from the remaining log in the log volume and restarts the database instance implicitly.

Recovering an Earlier Status of the Database Instance using One Complete Data and One Incremental Data without Log Backup

1. Call up Database Manager CLI in session mode
2. Initialize volumes (delete all data and log)
3. Import complete data backup
4. Import incremental data backup
5. Restart the database instance

```
dbmcli -uUTL -d <database_name> -u  
<dbm_user_name>,<dbm_user_password>  
util_execute INIT CONFIG  
recover_start completeT  
recover_start incrF  
db_online
```