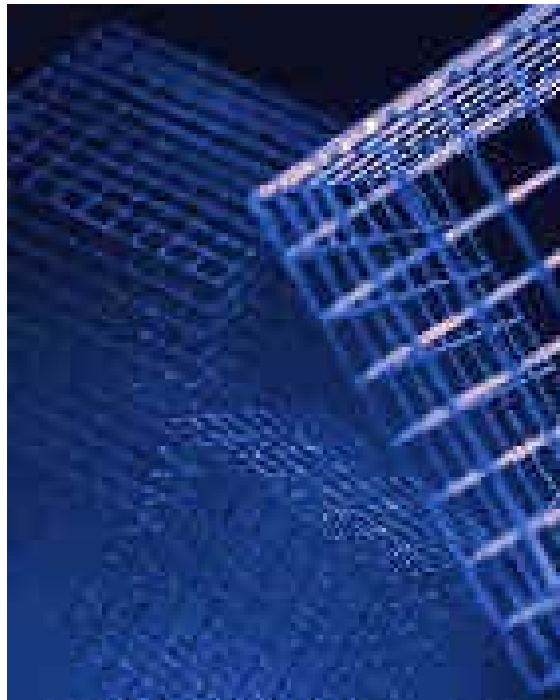# Backup and Recovery with the Database Manager CLI: SAP DB Short Description

Version 7.4

Copyright

Icons

Icon	Meaning
⚠️	Caution
🗨️	Example
➡️	Note
🧭	Recommendation
Syn	Syntax
💡	Tip

Contents

Introduction

The Database Manager CLI is a tool for managing any number of local and remote database instances. These can be managed from the command line. The program is suitable for both interactive and batch operation.

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 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.

See also: Database Manager CLI: SAP DB 7.4

Media

A backup medium is assigned to every backup you carry out. Backup media include files, tapes, and pipes.

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).

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

```
dbmcli -d <database_name> -u <user_name>,<password> medium_put
completeF /usr/sapdb/complete FILE DATA 0 8 YES
```

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

```
dbmcli -d <database_name> -u <user_name>,<password> medium_put
completeT /dev/rft0 TAPE DATA 64000 8 NO
```

Definition of a File as Medium for an Incremental Data Backup

```
dbmcli -d <database_name> -u <user_name>,<password> medium_put
incrF /usr/sapdb/incr FILE PAGES
```

Definition of a File as Medium for an Interactive Log Backup

```
dbmcli -d <database_name> -u <user_name>,<password> medium_put
logsave /usr/sapdb/log FILE LOG
```

Definition of a File as Medium for an Automatic Log Backup

```
dbmcli -d <database_name> -u <user_name>,<password> medium_put
autosave /usr/sapdb/auto 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 SAP DB 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/sapdb/log.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 or 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 Backup to a Medium

Creating a Complete Data Backup
```
dbmcli -d <database_name> -u <user_name>,<password> -uUTL -c
backup_start completeT
```

Creating an Incremental Data Backup
```
dbmcli -d <database_name> -u <user_name>,<password> -uUTL -c
backup_start incrF
```

Creating an Interactive Log Backup
```
dbmcli -d <database_name> -u <user_name>,<password> -uUTL -c
backup_start logsave
```

Activating the Automatic Log Backup
```
dbmcli -d <database_name> -u <user_name>,<password> autolog_on
```

Deactivating the Automatic Log Backup
```
dbmcli -d <database_name> -u <user_name>,<password>
autolog_off
```

Checking the Status of the Automatic Log Backup
```
dbmcli -d <database_name> -u <user_name>,<password>
autolog_show
```

Commands for Backup with Media Change
(Example of a complete data backup with media change)
Call up the Database Manager CLI in session mode
> Start complete data backup
> Media change during data backup
```
dbmcli -d <database_name> -u <user_name>,<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 data backup involved. An **older** database status can be recovered if only adata backup is 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 mode of the database.

Commands for Data Recovery

Recovering the Current Status of a Database Instance (with Data and Log Backups)

Call up Database Manager CLI in session mode
      Import complete data backup
      Import incremental data backup
      Import log backup version 001
      Media change to log backup version 002
      Complete recovery of the log backup and restart database instance implicitly

```
dbmcli -uUTL -d <database_name> -u <user_name>,<password>
    recover_start completeT
    recover_start incrF
    recover_start logsave LOG 001
    recover_replace logsave /usr/sapdb/log 002
    recover_ignore
```

Recovering an Earlier Status of the Database Instance using Data Backups without Log Backup

Call up Database Manager CLI in session mode
      Initialize volumes (delete all data and log)

Import complete data backup
Import incremental data backup
Restart database instance

```
dbmcli -uUTL -d <database_name> -u <user_name>,<password>
    util_execute INIT CONFIG
    recover_start completeT
    recover_start incrF
    db online
```