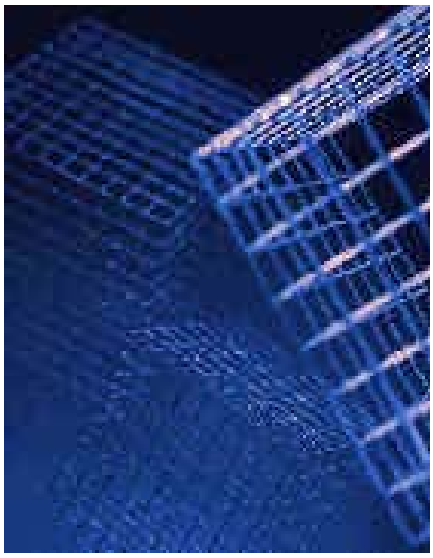
Database Manager CLI



SAP DB 7.4

Copyright

© Copyright 2003 SAP AG.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation.

For more information on the GNU Free Documentaton License see http://www.gnu.org/copyleft/fdl.html#SEC4.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Typographic Conventions

Type Style	Description
Example text	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options. Cross-references to other documentation.
Example text	Emphasized words or phrases in body text, titles of graphics and tables.
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Screen output. This includes file and directory names and their paths, messages, source code, names of variables and parameters as well as names of installation, upgrade and database tools.
`EXAMPLE TEXT`	Keys on the keyboard, for example, function keys (such as `F2`) or the `ENTER` key.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.

Database Manager CLI: SAP DB 7.4 11
Functions of the Database Manager CLI 12
Operator Management in the Database Manager 12
Operator Properties 12
Operator Authorizations 13
Displaying Status Data: DBInfoRead 14
Executing the Program LOAD: ExecLoad 15
Executing Operating System Commands: SystemCmd 15
Performing Backups: Backup 15
Installation Management: InstallMgm 16
Loading the System Tables: LoadSysTab 17
Starting the Database Instance: DBStart 17
Stopping the Database Instance: DBStop 17
Managing DBM Operators: UserMgm 18
Restoring Backups: Recovery 18
Accessing Database Files (Reading Only): DBFileRead 18
Accessing Database Parameters (Checked Writing): ParamCheckWrite 19
Accessing Database Parameters (Reading and Writing): ParamFull 19
Accessing Database Parameters (Reading Only): ParamRead 20
Access to SQL Session: AccessSQL 20
Access to Utility Session: AccessUtility 21
Opening a DBM Server Session 21
Logging On to the Database Manager 22
Command Structure 23
Input Help 23
Comment: # 24
Local System Call: ! 24
Linking Lines: / 24
Linking Lines: < 24
Options 25
Logging On to the XUSER Program: -ux 26
Displaying the Parameters for an XUSER Entry: -up 26
Display of XUSER Data: -ul 27
Opening a Service Session: -uSRV 27
Opening an SQL Session: -uSQL 27
Opening a Utility Session: -uUTL 28
Installation Directory of the Database Software: -R 29
Indicator as DBM Command: -c 29
Local Operation: -s 30

Deleting XUSER Data: -ud 30
Name of the Output File: -o 31
Name of Database Instance: -d 31
Name of the Log File: -t 32
Name of the Database Server: -n 32
Name of the Input Script: -i 33
Storing XUSER Data: -us 33
Version of the Database Manager: -V 34
XUSER Key: -uk 34
DBM Commands 35
Analyzing the Database Instance 35
Database Trace Functions 36
Overview of the Database Trace Commands 36
Displaying the Options for the Text Version of the Database Trace 37
Displaying the Logged Activities of the Database Trace 38
Deactivating the Database Trace 39
Creating the Text Version of the Database Trace 41
Accessing the Database Analyzer Program 42
Overview of the Commands for the Database Analyzer 42
Calling the Database Analyzer 42
Example of a Successful Call of the Database Analyzer 43
Example of an Error when Calling the Database Analyzer 43
Ending the Database Analyzer 44
Example of Ending the Database Analyzer Successfully 44
Examples of Errors when Ending the Database Analyzer 45
Determining the Status of the Database Analyzer 46
Example of Determining the Status Successfully 46
Example of an Error when Determining the Status 47
Accessing Database Files 47
Overview of the Commands for Accessing Database Files 48
Displaying the Diagnosis History 48
Editing Log Files of the Database Instance 49
Scrolling Through a Database File 50
List of the Database Files 51
Opening a Database File 52
Packing Database Files 53
Backing Up a Database File 54
Restoring a Database File 55
Logging On to the Operating System 55
Displaying the Version Information of the Database Manager 56

Executing the liveCache Initialization Script 57
Ending the Database Manager 58
Functions for Database Operation 58
Overview of the Commands for Database Operation 58
Updating Optimizer Statistics with the XPU Program 59
Displaying the Database Kernel Variant 60
Displaying the Operational State of the Database Instance 60
Displaying Information About the Database Instance 61
Using the Database Console 62
Scrolling in the Information About the Database Instance 63
Loading the System Tables 64
Loading SAP-Specific Tables 65
List of the Information About the Database Instance 65
Deleting Runtime Information After a Database Error 66
Restarting the Database Instance 66
Starting the Database Instance 67
Starting the LOAD Program 68
Starting the PythonLOAD Program 68
Stopping the Database Instance 69
Transferring to the ADMIN Operational State 70
Transferring to the OFFLINE Operational State 71
Transferring to the ONLINE Operational State 71
Installation and Registration Management 72
Overview of the Commands for Installation and Registration Management 72
Displaying the Version-Independent Directories 73
Changing the Software Version of the Current Database Instance 73
Defining the Version-Independent Directories 74
List of All Registered Database Instances 74
List of All Registered Versions of the Database Software 75
Deleting the Current Database Instance 76
db_drop Command: Example 76
Deleting the Registration of a Variant of the Current Database Instance 77
Deleting the Registration of a Version of the Database Software 78
Creating a Database Instance 78
Kernel Variant -a | -auto 80
Kernel Variant -f | -fast 80
Kernel Variant -q | -quick 80
Kernel Variant -s | -slow 81
db_create Command: Example 81
Registering a Variant of the Current Database Instance 82

Registering a Version of the Database Software 83
Configuring the Database Manager 83
Displaying a Configuration Parameter of the Database Manager 84
Changing a Configuration Parameter of the Database Manager 84
Configuring Database Instances 85
Parameter Properties 85
Overview of the Commands for Configuring Database Instances 87
Terminating a Parameter Session 88
Changing the Value of a Database Parameter 88
param_put Command: Example 89
Changing Volume Parameters 89
Displaying All Data for a Database Parameter 90
param_getfull Command: Example 91
Displaying All Properties for a Database Parameter 92
Displaying All Parameters of the Current Database Parameter File 92
Displaying the Data for All Database Parameters 93
Displaying the Data for All Volume Parameters 94
Displaying the Data of an Individual Volume Parameter 95
Displaying the Current Value of a Database Parameter 96
Displaying the Data Type of a Database Parameter 97
Displaying the Stored Explanatory Text 97
Displaying the Stored Help Text 98
Displaying the System Default Value 98
Displaying a Value from the Database Parameter File 99
Displaying Individual Data for a Database Parameter 99
Confirming Changes to the Database Parameter File 100
Scrolling in the Database Parameter History 101
Changing a Value Directly in the Database Parameter File 102
Opening a Parameter Session 103
Adding a Volume 104
Initializing the Database Parameters for a New Database Instance 104
Copying a Database Parameter File 105
Correcting Parameters 106
List of the Available Parameter Files 106
Deleting the Database Parameter File 107
Deleting a Database Parameter 107
Deleting Volume Parameters 108
Opening the Database Parameter History 109
param_gethistory Command: Example 110
Checking All Database Parameters 111

Resetting the Parameter File to a Previous Version 112
List of the DBM Commands 112
Backing Up and Recovering Database Instances 113
Commands for Backup Media 114
Change Date of the Media Definition File 114
Defining or Changing a Backup Medium 115
medium_put Command: Examples 116
Displaying the Media Data 117
List of All Defined Backup Media 118
Deleting a Backup Medium 119
Transferring an Existing Media Definition 120
Commands for Backing Up and Archiving 120
Canceling the Automatic Log Backup 121
Displaying the Automatic Log Backup Function 121
Displaying the Current Backup Status 122
Archiving the Version Files of the Log Area 124
Creating a Single Archive of the Version Files 124
Creating Multiple Archives of the Version Files 126
Deactivating the Automatic Log Backup 128
Terminating an Interrupted Backup 128
Activating the Automatic Log Backup 129
Continuing the Backup Without the Medium Last Reported as Full 129
Backing Up to a Succeeding Medium 130
Backing Up the Database Instance 132
Backing Up with External Backup Tools 135
Commands for the Backup History 136
Change Data of the Backup History 136
Displaying the Backup History 137
Scrolling in the Backup History 139
Fetching the Backup History 140
Closing the Backup History 140
Commands for External Backup IDs 141
Displaying External Backup IDs 141
Scrolling in the External Backup IDs 142
Releasing the Memory Occupied by the External Backup IDs 143
Fetching External Backup IDs 143
Commands for Backup Information 144
Displaying the Backup Information in the Operational State OFFLINE 144
Displaying the Backup Information in the Operational State ONLINE or ADMIN 146
Displaying the Current Status of a Backup Check 148

Checking a Backup 149
Commands for Restoring 152
Displaying Current Information About the Database Instance 153
Displaying the Current Restore Status 154
Terminating an Interrupted Restore or a Backup Check 156
Continuing the Parallel Restore or Backup Check 156
Recovering a Database Instance 158
Restoring the Parameter File from a Data Backup 161
Recreating a Damaged Index 163
Restoring with External Backup Tools 164
Restoring or Checking a Backup with a Succeeding Medium 164
Managing the DBM Operators 167
Changing DBM Operator Data 167
Creating DBM Operators 168
Displaying Operator Authorizations 168
user_getrights Command: Example 169
Displaying Operator Data 169
List of DBM Operators 170
Deleting DBM Operators 171
Accessing the Database Instance 171
Overview of Commands for Accessing the Database Instance 172
Access Using a Utility Session 172
Activating a New Database Instance 173
Displaying a Physical Database Page 173
Terminating a Utility Session 174
Opening a Utility Session 175
Writing a Physical Database Page 175
Transferring a Utility or SQL Command 176
Access Using an SQL Session 176
Updating the Optimizer Statistics 177
Displaying the Structure of a Result Set 177
Displaying Status Information for the Database Instance 178
Terminating an SQL Session 179
Scrolling in the Result Data 180
Scrolling in the Status Information of the Database Instance 180
Opening an SQL Session 181
Transferring an SQL Statement 182
Access Using a Service Session 183
Terminating a Service Session 183
Opening a Service Session 183

Access to Database Events 184
Activating a Database Event 184
Terminating a Database Event Session 185
Deactivating a Database Event 185
List of the Activated Database Events 186
Waiting for a Database Event 187
Overview of All DBM Commands 188
Reply Format 193
Operating Modes 194
Command Mode 194
Script Mode 195
Session Mode 195
Database Manager Files 195
Database Manager Log Files 196
Database Manager Configuration Files 196

Database Manager CLI: SAP DB 7.4

Purpose

The Database Manager is the administration tool for SAP DB database instances. You can use this tool to create and monitor database instances on local and remote hosts. You can back up the data from database instances, and recover the data, if necessary.

Implementation Considerations

The database management tool Database Manager can be called as a graphical user interface (Database Manager GUI [See SAP DB Library]), as a Web-based tool (Web DBM [See SAP DB Library]), or as a command-line oriented tool (Database Manager CLI). All three versions of this program offer the same range of functions.

The Database Manager CLI and the Web DBM can be installed on all operating systems supported by the SAP DB database system. The Database Manager GUI can run on Microsoft Windows only.

Each version of the Database Manager can be used for the administration of SAP DB database instances on all operating systems supported by the SAP DB database system (Operating System Platforms [See SAP DB Library]).

You can use Version 7.4 of the Database Manager for all SAP DB database systems from Version 7.1.

A sound knowledge of database administration is required to work with the Database Manager.

You can download the Database Manager CLI program free of charge from the SAP DB Homepage *http://www.sapdb.org*. It is a part of the software package for the database kernel, which means it is included in the installation profiles *All* and *Server*.

For general information on the SAP DB database system, see the documentation The SAP DB Database System [See SAP DB Library] on the SAP DB homepage under *Documentation*.

The **Database Manager CLI**, Version 7.4, is described in the following topics:

Analyzing a Database Instance [Page 35]

Logging On to the Operating System [Page 55]

Displaying the Version Information of the Database Manager [Page 56]

Executing the liveCache Initialization Script [Page 57]

Ending the Database Manager [Page 58]

Functions for Database Operation [Page 58]

Installation and Registration Management [Page 72]

Configuring the Database Manager [Page 83]

Configuring Database Instances [Page 85]

List of DBM Commands [Page 112]

Backing up and Restoring Database Instances [Page 113]

Managing DBM Operators [Page 167]

Accessing the Database Instance [Page 171]

See also:

Overview of All DBM Commands [Page 188]

Functions of the Database Manager CLI

The Database Manager CLI [Page 1] is an easy-to-use tool that can manage any number of local or remote database instances [See SAP DB Library]. These can be managed from the command line. The program is suitable for both interactive and background operation.

The Database Manager CLI is the client program of the Database Manager, which enables you to connect to the DBM Server [See SAP DB Library] and exchange data with it. The various functions of the program can be called both using options [Page 25] when calling the Database Manager CLI and using DBM commands [Page 35].

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

A request is directed to the Database Manager using a DBM command and the corresponding options. This then forwards the reply relating to this command and the specified options. As well as information on the state of the database instance, the reply contains data in edited form (Reply Format [Page 193]).

Operator Management in the Database Manager

Operators that work with the database administration tool Database Manager are called Database Manager operators (Short form: DBM Operators [See SAP DB Library]). Depending on their operator properties [Page 12], they can start and stop database instances [See SAP DB Library], perform backups, change database parameters, and so on.

DBM operators are **not** database users. You need to create database users in order to work on a database instance.

See also: *User manual: SAP DB*

Operator Properties

You can assign operator properties to DBM operators [See SAP DB Library]. (See also: Changing DBM Operator Data [Page 167].)

Properties that can be assigned to a DBM operator:

USERID, PASSWORD	Name and password for the identification of the operator.
SECONDPASSWORD	A second password used to allow other persons to work

	temporarily with an operator account, without needing to reveal or change the original password to do this (for example, for support purposes)
DISABLED	The operator is locked.
SERVERRIGHTS	Operator authorizations [Page 13] in the Database Manager
COMMENT	More detailed description of the operator

Operator Authorizations

An operator authorization is an authorization to execute certain DBM commands [Page 35] in the Database Manager.

An operator authorization may cover more than one DBM command and one DBM command may have more than one authorization assigned to it.

The operator authorizations assigned to an operator are stored in the Database Manager as `SERVERRIGHTS`.

The Database Manager recognizes the following Operator Authorizations:

Displaying Status Information [Page 14]	*DBInfoRead*
Executing the LOAD Program [Page 15]	*ExecLoad*
Executing Operating System Commands [Page 15]	*SystemCmd*
Performing Backups [Page 15]	*Backup*
Installation Management [Page 16]	*InstallMgm*
Loading the System Tables [Page 17]	*LoadSysTab*
Starting the Database Instance [Page 17]	*DBStart*
Stopping the Database Instance [Page 17]	*DBStop*
Managing DBM Operators [Page 18]	*UserMgm*
Restoring Backups [Page 18]	*Recovery*
Accessing Database Files (Reading Only) [Page 18]	*DBFileRead*
Accessing Database Parameters (Checked Writing) [Page 19]	*ParamCheckWrite*
Accessing Database Parameters (Reading and Writing) [Page 19]	*ParamFull*
Accessing Database Parameters (Reading Only) [Page 20]	*ParamRead*
Access to SQL Session [Page 20]	*AccessSQL*
Access to Utility Session [Page 21]	*AccessUtility*

Displaying Status Data: DBInfoRead

Operators who have been assigned the *DBInfoRead* operator authorization [Page 13] can execute the following DBM commands [Page 35]:

Activating a Database Event [Page 184]	`event_set`
Change Date of the Media Definition File [Page 114]	`medium_date`
Change Date of the Backup History [Page 136]	`backup_history_date`
Displaying the Automatic Log Backup Function [Page 121]	`autolog_show`
Displaying the Media Data [Page 117]	`medium_get`
Displaying the Backup History [Page 137]	`backup_history_list`
Displaying the Database Kernel Variant [Page 60]	`db_speed`
Displaying the Operational State of the Database Instance [Page 60]	`db_state`
Displaying External Backup IDs [Page 141]	`backup_ext_ids_list`
Displaying Information About the Database Instance [Page 61]	`show`
Displaying Status Information for the Database Instance [Page 178]	`info`
Terminating a Database Event Session [Page 185]	`event_release`
Terminating an SQL Session [Page 179]	`sql_release`
Scrolling in the External Backup IDs [Page 142]	`backup_ext_ids_listnext`
Scrolling in the Information About the Database Instance [Page 63]	`show_next`
Scrolling in the Status Information of the Database Instance [Page 180]	`info_next`
Scrolling in the Backup History [Page 139]	`backup_history_listnext`
Deactivating a Database Event [Page 185]	`event_delete`
Opening an SQL Session [Page 181]	`sql_connect`
Releasing the Memory Occupied by the External Backup IDs [Page 143]	`backup_ext_ids_forget`
Fetching the Backup History [Page 140]	`backup_history_open`
Fetching External Backup IDs [Page 143]	`backup_ext_ids_get`
List of All Defined Backup Media [Page 118]	`medium_getall`
List of Activated Database Events [Page 186]	`event_list`
List of the Information About the Database Instance [Page 65]	`show_list`
Closing the Backup History [Page 140]	`backup_history_close`
Copying an Existing Media Definition [Page 120]	`medium_migrate`
Waiting for a Database Event [Page 187]	`event_wait`

Executing the Program LOAD: ExecLoad

Operators who have been assigned the *ExecLoad* operator authorization [Page 13] can execute the following DBM commands [Page 35]:

Loading the System Tables [Page 17]	`load_systab`
Loading SAP-Specific Tables [Page 65]	`load_r3tab`
Starting the LOAD Program [Page 68]	`exec_xload`
Starting the PythonLOAD Program [Page 68]	`exec_load`

Executing Operating System Commands: SystemCmd

Operators who have been assigned the *SystemCmd* operator authorization [Page 13] can execute the following DBM command [Page 35]:

Executing the liveCache Initialization Script [Page 57]	`exec_lcinit`

Performing Backups: Backup

Operators who have been assigned the *Backup* operator authorization [Page 13] can execute the following DBM commands [Page 35]:

Change Date of the Media Definition File [Page 114]	`medium_date`
Change Date of the Backup History [Page 136]	`backup_history_date`
Defining or Changing a Backup Medium [Page 115]	`medium_put`
Displaying Current Information About the Database Instance [Page 153]	`db_restartinfo`
Displaying External Backup IDs [Page 141]	`backup_ext_ids_list`
Displaying the Automatic Log Backup Function [Page 121]	`autolog_show`
Displaying the Media Data [Page 117]	`medium_get`
Displaying the Backup History [Page 137]	`backup_history_list`
Displaying the Backup Information in the Operational State OFFLINE [Page 144]	`medium_labeloffline`
Displaying the Backup Information in the Operational State ONLINE or ADMIN [Page 146]	`medium_label`
Displaying the Current Status of a Backup Check [Page	`recover_state_check`

148]	
Displaying the Current Backup Status [Page 122]	`backup_state`
Deactivating the Automatic Log Backup [Page 128]	`autolog_off`
Terminating a Service Session [Page 183]	`service_release`
Terminating an Interrupted Automatic Log Backup [Page 121]	`autolog_cancel`
Terminating an Interrupted Backup [Page 128]	`backup_cancel`
Terminating an Interrupted Restore or Backup Check [Page 156]	`recover_cancel`
Terminating a Utility Session [Page 174]	`util_release`
Scrolling in the External Backup IDs [Page 142]	`backup_ext_ids_listnext`
Scrolling in the Backup History [Page 139]	`backup_history_listnext`
Activating the Automatic Log Backup [Page 129]	`autolog_on`
Opening a Service Session [Page 183]	`service_connect`
Opening a Utility Session [Page 175]	`util_connect`
Continuing a Parallel Restore or Backup Check [Page 156]	`recover_ignore`
Continuing the Backup Without the Medium Last Reported as Full [Page 129]	`backup_ignore`
Releasing the Memory Occupied by the External Backup IDs [Page 143]	`backup_ext_ids_forget`
Fetching the Backup History [Page 140]	`backup_history_open`
Fetching External Backup IDs [Page 143]	`backup_ext_ids_get`
List of All Defined Media [Page 118]	`medium_getall`
Deleting a Backup Medium [Page 119]	`medium_delete`
Checking a Backup [Page 149]	`recover_check`
Closing the Backup History [Page 140]	`backup_history_close`
Backing Up to a Succeeding Medium [Page 130]	`backup_replace`
Backing Up the Database Instance [Page 132]	`backup_start`
Copying an Existing Media Definition [Page 120]	`medium_migrate`
Restoring or Checking a Backup with Succeeding Medium [Page 164]	`recover_replace`

Installation Management: InstallMgm

Operators who have been assigned the *InstallMgm* operator authorization [Page 13] can execute the following DBM commands [See SAP DB Library]:

Changing a Configuration Parameter of the Database Manager [Page 84]	`dbm_configset`

Displaying a Configuration Parameter of the Database Manager [Page 84]	`dbm_configget`
Initializing the Database Parameters for a New Database Instance [Page 104]	`param_init`
Copying a Database Parameter File [Page 105]	`param_copy`
Deleting the Current Database Instance [Page 76]	`db_drop`
Deleting the Database Parameter File [Page 107]	`param_rmfile`
Deleting the Registration of a Variant of the Current Database Instance [Page 77]	`db_unreg`
Registering a Variant of the Current Database Instance [Page 82]	`db_reg`

Loading the System Tables: LoadSysTab

Operators who have been assigned the *LoadSysTab* operator authorization [Page 13] can execute the following DBM commands [Page 35]:

Loading the System Tables [Page 64]	`load_systab`
Loading SAP-Specific Tables [Page 65]	`load_r3tab`

Starting the Database Instance: DBStart

Operators who have been assigned the *DBStart* operator authorization [Page 13] can execute the following DBM commands [Page 35]:

Starting the Database Instance [Page 67]	`db_start`
Transferring to the ONLINE Operational State [Page 71]	`db_online`

Stopping the Database Instance: DBStop

Operators who have been assigned the *DBStop* operator authorization [Page 13] can execute the following DBM commands [Page 35]:

Restarting the Database Instance [Page 66]	`db_restart`
Stopping the Database Instance [Page 69]	`db_stop`
Transferring to the ADMIN Operational State [Page 70]	`db_admin`

Transferring to the OFFLINE Operational State [Page 71]	`db_offline`
Deleting Runtime Information After a Database Error [Page 66]	`db_clear`
Using the Database Console [Page 62]	`db_cons`

Managing DBM Operators: UserMgm

DBM operators who have been assigned the *UserMgm* operator authorization [Page 13] can execute the following DBM commands [Page 35]:

Changing DBM Operator Data [Page 167]	`user_put`
Creating DBM Operators [Page 168]	`user_create`
Displaying Operator Authorizations [Page 168]	`user_getrights`
Displaying Operator Data [Page 169]	`user_get`
List of Registered Database Users [Page 170]	`user_getall`
Deleting DBM Operators [Page 171]	`user_delete`

Restoring Backups: Recovery

Operators that have been assigned the *Recovery* operator authorization [Page 13] can execute all of the commands listed under the Backup [Page 15]. They can **additionally** execute the following DBM commands [Page 35]:

Displaying the Current Recovery Status [Page 154]	`recover_state`
Displaying Information About the Database Instance [Page 61]	`db_restart_info`
Restoring the Database Instance [Page 158]	`recover_start`
Restoring a Database Parameter File [Page 161]	`recover_config`

Accessing Database Files (Reading Only): DBFileRead

Operators who have been assigned the *DBFileRead* operator authorization [Page 13] can execute the following DBM commands [Page 35]:

Scrolling Through the Further Contents of a Database File [Page 50]	`file_getnext`
Opening a Database File [Page 52]	`file_getfirst`
Packing Database Files [Page 53]	`diag_pack`
Backing Up a Database File [Page 54]	`file_backup`
Restoring a Database File [Page 55]	`file_restore`

Accessing Database Parameters (Checked Writing): ParamCheckWrite

Operators that have been assigned the *ParamCheckWrite* operator authorization [Page 13] can execute all of the commands listed under the ParamRead [Page 20]. They can **additionally** execute the following DBM commands [Page 35]:

Changing the Value of a Database Parameter [Page 88]	**`param_put`**
Changing Volume Parameters [Page 89]	**`param_addvolume`**
Confirming Changes to the Database Parameter File [Page 100]	**`param_commitsession`**
Correcting Parameters [Page 106]	**`param_putconfirm`**
Deleting Volume Parameters [Page 108]	**`param_delvolume`**
Checking All Database Parameters [Page 111]	**`param_checkall`**
Resetting the Parameter File to a Previous Version [Page 112]	**`param_restore`**

Accessing Database Parameters (Reading and Writing): ParamFull

Operators that have been assigned the *ParamFull* operator authorization [Page 13] can execute all of the commands listed under the ParamRead [Page 20] and ParamCheckWrite [Page 19] operator authorizations. They can **additionally** execute the following DBM commands [Page 35]:

Directly Changing a Value in the Database Parameter File [Page 102]	**`param_directput`**
Adding a Volume [Page 104]	**`db_addvolume`**
Deleting a Parameter [Page 107]	**`param_directdel`**

Accessing Database Parameters (Reading Only): ParamRead

Operators who have been assigned the *ParamRead* operator authorization [Page 13] can execute the following DBM commands [Page 35]:

Terminating a Parameter Session [Page 88]	`param_abortsession`
Displaying All Data for a Database Parameter [Page 90]	`param_getfull`
Displaying All Properties of a Database Parameter [Page 92]	`param_getproperties`
Displaying All Parameters of the Current Database Parameter File [Page 92]	`param_directgetall`
Displaying the Data for All Database Parameters [Page 93]	`param_extgetall`
Displaying the Data for All Volume Parameters [Page 94]	`param_getvolsall`
Displaying the Data for Individual Volume Parameters [Page 95]	`param_getvolume`
Displaying the Current Value of a Database Parameter [Page 96]	`param_getvalue`
Displaying the Data Type of a Database Parameter [Page 97]	`param_gettype`
Displaying the Stored Explanatory Text [Page 97]	`param_getexplain`
Displaying the Stored Help Text [Page 98]	`param_gethelp`
Displaying the System Default Value [Page 98]	`param_getdefault`
Displaying a Value from the Database Parameter File [Page 99]	`param_directget`
Displaying Individual Data for a Database Parameter [Page 99]	`param_extget`
Scrolling in the Database Parameter History [Page 101]	`param_gethistorynext`
Opening a Parameter Session [Page 103]	`param_startsession`
List of Available Parameter Files [Page 106]	`param_versions`
Opening the Database Parameter History [Page 109]	`param_gethistory`

Access to SQL Session: AccessSQL

Operators who have been assigned the *AccessSQL* operator authorization [Page 13] can execute the following commands:

Displaying the Result Set Structure [Page 177]	`sql_info`
Updating the Optimizer Statistics [Page 177]	`sql_updatestat`
Updating the Optimizer Statistics with the XPU Program [Page 59]	`exec_xpu`
Scrolling in the Additional Result Data [Page 180]	`sql_fetch`
Opening an SQL Session [Page 181]	`sql_connect`

Closing an SQL Session [Page 179]	`sql_release`
Transferring an SQL Statement [Page 182]	`sql_execute`
Recreating a Damaged Index [Page 163]	`sql_recreateindex`

Access to Utility Session: AccessUtility

Operators who have been assigned the *AccessUtility* operator authorization [Page 13] can execute the following DBM commands [Page 35]:

Activating a New Database Instance [Page 173]	`util_activate`
Displaying a Physical Database Page [Page 173]	`util_getpage`
Terminating a Utility Session [Page 174]	`util_release`
Opening a Utility Session [Page 175]	`util_connect`
Adding a Volume [Page 104]	`db_addvolume`
Writing a Physical Database Page [Page 175]	`util_putpage`
Transferring a Utility or SQL Command [Page 176]	`util_execute`

Opening a DBM Server Session

Use

You open a session with the DBM Server [See SAP DB Library] and define to which database instance [See SAP DB Library] the following transferred options and commands refer. If the database instance is on a remote server, you must specify the server name **<server_node>** in addition to the name of the database instance **<database_name>**.

A DBM Server session is the prerequisite for Logging on to the Database Manager Program [Page 22] and the execution of most DBM Commands [Page 35].

Prerequisites

You have installed the Database Manager CLI program on your server. It is part of the SAP DB database software.

Syntax

```
dbmcli – d <database_name> [-n <server_node>]
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Syntax

```
dbmcli -d <database_name>
```

Logging On to the Database Manager

Use

The Database Manager operator [See SAP DB Library] specified using **<userid>** logs on to the Database Manager program.

For local communication, the operator is already logged on to the operating system.

Prerequisites

Logging on to the Database Manager requires a session with the DBM Server [See SAP DB Library]. You can open this session before logging on to the Database Manager or together with the logon.

Two Step Logon

You can first open the session with the DBM Server and then log on to the Database Manager in a second step.

Opening a DBM Server Session [Page 21]

```
dbmcli - d <database_name> [-n <server_node>]
```

Logging On to the Database Manager

```
user_logon <userid>,<password>
```

One Step Logon

Using the option **-u | -U**, you can perform both logon steps at the same time. The Database Manager implicitly executes the DBM command [Page 35] **user_logon**.

```
dbmcli -u [<userid>,<password>] -d <database_name>
```

or logon with XUSER: **dbmcli -U [<user_key>] -d <database_name>**

- **-u**: explicit specification of the DBM operator identified with **<userid>** and **<password>**

  If you use **-u** without specifying **<userid>,<password>**, the Database Manager prompts you to enter the operator name and password. In this way you can avoid the operator password being visible in the command line, command history, and the operating system process list.

- **-U**: Logging on with a user stored in the XUSER file

  If you do not specify a **<user_key>** when using **-U**, the Database Manager uses the data under the DBMUSR XUSER key.

  You can store operator details for various tasks with the help of the program XUSER and use them for logging on.

Note the different meanings of the command with uppercase and lowercase specification.

See also: *User manual: SAP DB*

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Command Structure

Use

Options [Page 25] and a maximum of one DBM command [Page 35] with its parameters can be transferred to the program Database Manager CLI.

A DBM command can also contain line feeds.

See also: Input Help [Page 23]

If you use options with optional parameters, mark the beginning of the DBM command with the option –c [Page 29] (Indicator as DBM command: -c [Page 29]).
By doing so, you avoid the Database Manager program interpreting the beginning of the DBM command as a parameter of the previously specified option.

Prerequisites

Check under *Control Panel* → *Services* that the service XServer [See SAP DB Library] has started
(status: `started`*)*.

Syntax

```
dbmcli [<options>] [<command>]
```

Reply

See Reply Format [Page 193]

Input Help

The following input help is available for the script mode [Page 195] and the session mode [Page 195]:

Local System Call: ! [Page 24]
Comment: # [Page 24]
Linking Lines: / [Page 24]
Linking Lines: < [Page 24]

Comment:

A command which starts with a number sign is ignored.

Local System Call: !

The command to be entered is preceded by an exclamation mark.

The command is not transferred to the Database Manager [Page 1] but is executed in the local command interpreter of the operating system.

Linking Lines: /

Prerequisites

You are working in session mode [Page 195] or script mode [Page 195].

If you end a line within a multiple line DBM Command [Page 35] with a slash, the system recognizes that the command is not complete. It is not executed when a line feed is inserted.

Only the entry of a line feed without a preceding slash indicates to the system that the command is complete. It is then transferred to the Database Manager.

Linking Lines: <

Using the special character less than (<), you can transfer commands consisting of several lines to the Database Manager.

Syntax

```
'<'<stop-token> <command-begin>
```

The character string **<command-begin>** is linked to the continuation lines until a line is found which starts with **<stop-token>**.

All lines are separated from one another by a line feed.

The line with **<stop-token>** is not incorporated into the command.

Options

You can use the following options in the Database Manager CLI (Command Structure [Page 23]):

Logging On to the XUSER Program [Page 26]	**-ux**
Displaying the Parameters for an XUSER Entry [Page 26]	**-up**
Displaying XUSER Data [Page 27]	**-ul**
Opening a Service Session [Page 27]	**-uSRV**
Opening an SQL Session [Page 27]	**-uSQL | -USQL**
Opening a Utility Session [Page 28]	**-uUTL | -UUTL**
Installation Directory of the Database Instance [Page 29]	**-R <inst_path>**
Indicator as DBM Command [Page 29]	**-c**
Local Operation [Page 30]	**-s**
Deleting XUSER Data [Page 30]	**-ud**
Name of the Output File [Page 31]	**-o <file_name>**
Name of the Database Instance [Page 31]	**-d <database_name>**
Name of the Log File [Page 32]	**-t <file_name>**
Name of the Database Server [Page 32]	**-n <server_node>**
Name of the Input Script [Page 33]	**-i <file_name>**
Storing XUSER Data [Page 33]	**-us**
Version of the Database Manager [Page 34]	**-V**
XUSER Key [Page 34]	**-uk**

If you use options with optional parameters, mark the beginning of the DBM command with the option –c [Page 29] (Indicator as DBM command: -c [Page 29]).
By doing so, you avoid the Database Manager program interpreting the beginning of the DBM command as a parameter of the previously specified option.

```
dbmcli -uUTL samplename,secret -d myDB
```

This calls the Database Manager (DBMCLI) and establishes a utility session for user **samplename**, password **secret**, with registered database instance **myDB**.

Logging On to the XUSER Program: -ux

Use

Option in the Database Manager CLI (Command Structure [Page 23]).

You authorize yourself with the user under the XUSER key `DEFAULT`.

See also: *User manual: SAP DB*

Syntax

```
-ux <default_userid>,<password>
```

Displaying the Parameters for an XUSER Entry: -up

Use

Option [Page 25] in the Database Manager CLI.

You store the connection parameters for an entry in the XUSER program.

See also: *User manual: SAP DB*

Prerequisites

You are also using the option -us (Storing XUSER data) at the same time.

Syntax

```
-up <param>=<value>; ...;
```

<param>	**SQLMODE | TIMEOUT | CACHELIMIT | ISOLATION | DBLOCALE**
<value>	for **SQLMODE:** **INTERNAL, ANSI, DB2, ORACLE,SAPR3** for **ISOLATION:** **0,1,2,3,10,15,20,30**

Display of XUSER Data: -ul

Use

Option in the Database Manager CLI (Command Structure [Page 23]).

You request the operator data stored in the XUSER program.

The system displays a table of the existing XUSER keys with the assigned operator names.

See also: *User manual: SAP DB*

Prerequisites

You are also using option -ux [Page 26] (Logon to XUSER) at the same time.

Syntax

```
-ul
```

Opening a Service Session: -uSRV

Use

Option [Page 25] in the Database Manager CLI

Certain DBM commands [Page 35] require a service session before they can be executed.

If you specify this option, the DBM command to open a service session [Page 183] is executed implicitly.

When you end the Database Manager CLI, the service session is also ended.

Prerequisites

You are also using the option -d [Page 31] at the same time to specify a database instance and the option -u [Page 22] to log on to the Database Manager.

Syntax

```
-uSRV
```

Opening an SQL Session: -uSQL

Use

Option [Page 25] in the Database Manager CLI.

Certain DBM commands [Page 35] require an SQL session before they can be executed.

If you specify this option, the DBM command to open an SQL session [Page 181] is executed implicitly. The SQL session is established with the transferred operator data.

If no operator is specified with this option, the Database Manager CLI uses the data of the current DBM operator [See SAP DB Library].

Operator details can be stored for various tasks with the help of the program XUSER and used for logging on.

Note the different meanings of the command with uppercase and lowercase specification.
-uSQL: Explicit specification of the operator with the **<userid>** and **<password>**
-USQL: Logon with an operator stored in the XUSER file.

See also: *User manual: SAP DB*

Prerequisites

You are also using the option -d [Page 31] at the same time to specify a database instance and the option -u [Page 22] to log on to the Database Manager.

Syntax

-uSQL [<userid>,<password>]

or logon with XUSER: -USQL <user_key>

Opening a Utility Session: -uUTL

Use

Option [Page 25] in the Database Manager CLI.

A utility session is a prerequisite for the execution of certain DBM commands [Page 35] .

If you specify this option, the DBM command to open a utility session [Page 175] is executed implicitly. A utility session is established with the transferred operator data.

If no operator is specified with **-uUTL**, the Database Manager CLI uses the data of the first DBM operator [See SAP DB Library].

If you do not specify a **<user_key>** when using **-UUTL**, the Database Manager CLI uses the data under the XUSER key **c**.

Operator details can be stored for various tasks with the help of the program XUSER and used for logging on.

Note the different meanings of the command with uppercase and lowercase specification.
-uUTL: Explicit specification of the operator with the **<userid>** and **<password>**
-UUTL: Logon with an operator stored in the XUSER file.

See also: *User manual: SAP DB*

Prerequisites

You are also using the option -d [Page 31] at the same time to specify a database instance and the option -u [Page 22] to log on to the Database Manager.

Syntax

-uUTL [<userid>,<password>]

or logon with XUSER: -UUTL [<user_key>]

Installation Directory of the Database Software: -R

Use

Option [Page 25] in the Database Manager CLI

On servers that contain several versions of the database system software it is necessary to address the correct version of the database. Enter the path of the desired version under **<inst_path>**.

Syntax

-R <inst_path>

dbmcli -R "C:\Program Files\SAP DBTech\V72"

When you set up a session there is a check whether the specified version on the server is recorded in the list of registered versions. If it is not, the session connection is refused.

When calling the Database Manager CLI using option **-R**, a specified database version can only be connected if the database installation has been correctly registered [Page 82].

If you specify the option -d [Page 31] and the name of a database instance in addition to the option **-R**, option **-R** is ignored. In this case, the Database Manager CLI uses the version of the database software assigned to the specified database instance.

Indicator as DBM Command: -c

Use

Option [Page 25] in the Database Manager CLI

Use the option **-c** to logically delimit a Database Manager CLI option with optional parameters and a DBM command [See SAP DB Library] from each other. The Database Manager CLI program interprets all specifications following **-c** as a DBM command.

Syntax

-c <DBM_command>

If you do not specify **-c** between the option with optional parameters and the DBM command, the beginning of the DBM command is interpreted as an operator name/password combination for the option –uSQL [Page 27]:

```
d:\v74>dbmcli -d a73 -u dbm,dbmp -uSQL sql_execute select *
from tables

ERR

-24988,ERR_SQL: sql error

-4008,Unknown user name/password combination
```

You mark the beginning of the DBM command with the option **–c.** The default value is used for the **–uSQL** option.

```
d:\v74>dbmcli -d a73 -u dbm,dbmp -uSQL –c sql_execute select
* from tables

OK

...
```

Local Operation: -s

Use

Option [Page 25] in the Database Manager CLI

If you call the Database Manager CLI with this option, no communication takes places with a DBM Server [See SAP DB Library]. In this case, you use the internal DBM Server functions of the Database Manager CLI. This means that you are also logged on to the operating system and can execute all of the commands that would require a logon to the operating system without the use of the **-s** option.

Syntax

-s

Deleting XUSER Data: -ud

Use

Option [Page 25] in the Database Manager CLI

You delete user data from the program XUSER.

You can delete the following user data:

- User data for a particular combination of the name of a database instance and the server on which this database instance is running
- User data for an XUSER key

See also: *User manual: SAP DB*

Deleting User Data for a Particular Combination of Database Instance and Database Server

Syntax

-ud

You authorize yourself with the user currently stored in the XUSER program.

```
-d <database_instance> [-n <server_node>] -u
<userid>,<password> -ud
```

or

You authorize yourself with the user under the XUSER key `DEFAULT`.

```
-d <database_name> [-n <server_node>] -ux
<default_userid>,<password> -ud
```

Deleting User Data Using an XUSER Key

Syntax

-ud

You authorize yourself with the user currently stored in the XUSER program.

```
-uk <userkey> -u <userid>,<password> -ud
```

or

You authorize yourself with the user under the XUSER key `DEFAULT`.

```
-uk <userkey> -ux <default_userid>,<password> -ud
```

Name of the Output File: -o

Use

Option [Page 25] in the Database Manager CLI.

All details are written to the file specified as **<file_name>**.

Syntax

```
-o <file_name>
```

Name of Database Instance: -d

Use

Option [Page 25] in the Database Manager CLI.

The name specified for the database instance [See SAP DB Library] in **<database_name>** applies for the whole session. All database-specific DBM commands [Page 35] relate to this database instance.

When you set up a session there is a check whether the specified database instance on the server is recorded in the list of registered database instances. If it is not, the session connection is refused.

Prerequisites

The database instance specified in **<database_name>** exists (Creating a Database Instance [Page 78]).

Syntax

```
-d <database_name>
```

Name of the Log File: -t

Use

Option [Page 25] in the Database Manager CLI

All commands transferred to the Database Manager and the respective replies are logged in the specified file **<file_name>**. The existing content of the file is retained.

Syntax

```
-t <file_name>
```

Name of the Database Server: -n

Use

Option when opening a session with the Database Manager CLI.

The DBM Server [See SAP DB Library] program on the database server specified with **<server_node>** is addressed.

Prerequisites

Program X-Server is active on node **<server_node>** (*Status*: `Started`).

See also: *User manual: SAP DB*

Syntax

```
-n <server_node>
```

Name of the Input Script: -i

Use

Option [Page 25] in the Database Manager CLI

If you specify the option **-i**, the DBM command(s) [Page 35] listed in the file `<file_name>` are processed.

Syntax

```
-i <file_name>
```

Storing XUSER Data: -us

Use

Option [Page 25] in the Database Manager CLI.

You store user data in the XUSER program.

You can store this data as

- User data for a particular combination of the name of a database instance and the server on which this database instance is running
- User data for an XUSER key

See also: *User manual: SAP DB*

Storing user data for a particular combination of the name of the database instance and the database server

Prerequisites

You use the options -d [Page 31] (Name of the Database Instance) and, optionally, -n [Page 32] (Name of the Database Server) at the same time.

You authorize yourself for the XUSER program with the currently stored user or the user under the XUSER key `DEFAULT`.

Syntax

```
-us <new_userid>,<new_password>
```

You authorize yourself with the user currently stored in the XUSER program

```
-d <database_instance> [-n <server_node>] [-u
<userid>,<password>] -us <new_userid>,<new_password>
```

or

You authorize yourself with the operator under the XUSER key `DEFAULT`

```
-d <database_instance> [-n <server_node>] [-ux
<default_userid>,<password>] -us <new_userid>,<new_password>
```

Storing user data using an XUSER key

Prerequisites

You use the option -uk [Page 34] (XUSER key) at the same time.

You authorize yourself for the XUSER program with the currently stored user or the user under the XUSER key `DEFAULT`.

Syntax

```
-us <new_userid>,<new_password>
```

You authorize yourself with the user currently stored in the XUSER program.

```
-uk <userkey> [-u <userid>,<password>] -us
<new_userid>,<new_password>
```

or

You authorize yourself with the user under the XUSER key `DEFAULT`.

```
-uk <user_key> [-ux <default_userid>,<password>] -us
<new_userid>,<new_password>
```

Version of the Database Manager: -V

Use

Option [Page 25] in the Database Manager CLI.

You request the version of the Database Manager.

Syntax

```
-V
```

XUSER Key: -uk

Use

Option [Page 25] in the Database Manager CLI.

You specify the XUSER key under which the user data is to be stored in the XUSER program.

See also: *User manual: SAP DB*

Syntax

```
-uk <user_key>
```

DBM Commands

Use

DBM commands are commands in the Database Manager CLI [Page 1] for managing database instances [See SAP DB Library].

You can use these to perform the following tasks:

Analyzing a Database Instance [Page 35]

Logging On to the Operating System [Page 55]

Displaying the Version Information of the Database Manager [Page 56]

Executing the liveCache Initialization Script [Page 57]

Ending the Database Manager [Page 58]

Functions for Database Operation [Page 58]

Installation and Registration Management [Page 72]

Configuring the Database Manager [Page 83]

Configuring Database Instances [Page 85]

List of DBM Commands [Page 112]

Backing up and Restoring Database Instances [Page 113]

Managing DBM Operators [Page 167]

Accessing the Database Instance [Page 171]

All commands during a session with the Database Manager relate to the database instance whose name was specified with the option -d [Page 31] when logging on to the Database Manager CLI.

See also: Command Structure [Page 23]

Prerequisites

To right to execute a DBM command [Page 35] can be linked to three prerequisites:

- Logging On to the Database Manager [Page 22]
- Operator Authorization [Page 13]
- Logging On to the Operating System [Page 55]

You can find out which of these prerequisites must be fulfilled for the execution of a particular DBM command from the descriptions of the individual DBM commands.

Syntax

```
<command_name> [<parameters>]
```

Analyzing the Database Instance

You have the option of performing various analyses of the database instance.

The Database Manager gives you the following options:

- Accessing the Database Analyzer Program [Page 42]
  You can use this program to monitor the performance of the database system for a definable period of time.
- Database Trace Functions [Page 36]
  Use the database trace to log selected activities of the database kernel in detail. Note that the activated database trace can have a significant influence on the performance of the database system.
- Accessing Database Files [Page 47]
  The system creates a range of files for each database instance, depending on which actions have been performed. These files are saved in the run directory of the database instance. Before a run directory is defined when the database instance is installed, the files are written directly to the directory `wrk` under the installation directory.

Database Trace Functions

Use

Use these functions only when told to do so by SAP DB Support [See SAP DB Library]. The database trace is not required for normal database operation.

Using the database trace [See SAP DB Library], you log selected activities of the database kernel in detail. You can activate, deactivate, and evaluate the database trace using the commands described below.

Prerequisites

Note the operator authorizations required for the particular commands DBM commands [Page 35] (Operator Authorizations [Page 13]).

Overview of the Database Trace Commands [Page 36]

Overview of the Database Trace Commands

Displaying Options for the Text Version of the Database Trace [Page 37]	**trace_protopt**
Displaying the Logged Activities of the Database Trace [Page 38]	**trace_show**
Deactivating the Database Trace [Page 39]	**trace_off**
Creating the Text Version of the Database Trace [Page 41]	**trace_prot**

Displaying the Options for the Text Version of the Database Trace

Use

You can specify various options when creating the text version of the database trace [Page 41].

You display which options are permissible for the text version of the current database trace using this command. You receive a list with the names and description of the valid options.

By specifying options, you restrict which trace output for the internal database components is copied to the text file. Use these options in consultation with SAP DB Support [See SAP DB Library].

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14] or DBFileRead [Page 18].

Syntax

trace_protopt

dbmcli -d DB -u dbm,dbm trace_protopt

```
OK
name,option
a       Order Interface (AK)
b       Record Interface (BD)
k       Show Message Block (KB)
m       Message Block
e       No Entrypos Output
s       Strategy
t       Time
x       Switch Output (Slow Kernel)
```

a	The trace output of the internal database component order interface is copied to the text file
b	The trace output of the internal database component record interface is copied to the text file
k	
m	The internal data structure with the name message block is transferred to the text file
e	
s	Information about the processing strategy is copied to the text file
T	
X	Additional trace output is copied to the text file

Reply

```
OK<NL>
name,option<NL>
<name>,<option><NL>
```

```
<name>,<option><NL>
...
```

Displaying the Logged Activities of the Database Trace

Use

You display which activities of the database kernel are currently logged.

All activities of the database kernel that can be logged are displayed under the column name `Description`. The word `ON` appears in the `State` column for activities that are actually currently being logged.

If you specify the option **`verbose`**, the system displays an extended description of the database trace [See SAP DB Library].

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14] or DBFileRead [Page 18].

Syntax

`trace_show [verbose]`

`dbmcli -d DB -u dbm,dbm trace_show`

```
OK
Name             State   Level   Description
DEFAULT          OFF     -       Default
INDEX            OFF     -       Index
LOCK             ON      -       Lock
LONG             OFF     -       Long
PAGES            OFF     -       Pages
STANDARD         OFF     -       Standard
TABLE            OFF     -       Table
OBJECT           OFF     -       Objects
OBJECTADD        OFF     -       Objects add
OBJECTGET        OFF     -       Objects get
OBJECTALTER      OFF     -       Objects alter
OBJECTFREE       OFF     -       Objects free
SELECT           OFF     -       Select
INSERT           OFF     -       Insert
UPDATE           OFF     -       Update
DELETE           OFF     -       Delete
ORDER            OFF     -       Order
OPTIMIZER        OFF     -       Optimize
TIME             OFF     -       Time
CHECK            OFF     -       Check
COMMANDS         ON      -       Commands
TOPICCONVERTER   OFF     0       Topic Converter
TOPICDATA        OFF     0       Topic Data
TOPICFRAMECTRL   OFF     0       Topic Framecontrol
TOPICLOG         OFF     0       Topic Log
```

```
TOPICIOMAN      ON      7       Topic I/O manager
TOPICMEMORY     OFF     0       Topic Memory
TOPICDEVSPACE   OFF     0       Topic Devspace
TOPICRUNTIME    OFF     0       Topic Runtime
TOPICALLOCATOR  OFF     0       Topic Allocator
CHECKCONVERTER  OFF     0       Check Converter
CHECKDATA       OFF     0       Check Data
CHECKFRAMECTRL  OFF     0       Check Framecontrol
CHECKLOG        OFF     0       Check Log
CHECKIOMAN      OFF     0       Check I/O manager
CHECKMEMORY     OFF     0       Check Memory
CHECKDEVSPACE   ON      3       Check Devspace
CHECKRUNTIME    OFF     0       Check Runtime
CHECKALLOCATOR  OFF     0       Check Allocator
```

Reply

```
OK<NL>
name    State     Level           Description<NL>
<name>  (ON|OFF)  (-|<level>)     <short description>[,<long
description>]<NL>
<name>  (ON|OFF)  (-|<level>)     <short description>[,<long
description>]<NL>
...
```

Deactivating the Database Trace

Use

You determine which activities of the database kernel should no longer be logged or completely deactivate the database trace [See SAP DB Library].

You specify which database kernel activities should no longer be logged with the **<activity_list>** option. Separate multiple activities by a space.

By specifying **ALL**, you deactivate the log function for all activities, and therefore completely deactivate the database trace.

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14] or DBFileRead [Page 18].

Syntax

```
trace_off <activity_list> | ALL
<activity_list> :: = <activity> | <activity> <activity_list>
<activity> :: see table
```

Deactivate the database trace for the areas LOCK and TOPICIOMAN

```
dbmcli -d a74 -u dbm,dbmp trace_off LOCK TOPICIOMAN
OK
```

Option <activity>

Name
ALL
CHECK
CHECKALLOCATOR
CHECKCONVERTER
CHECKDATA
CHECKDEVSPACE
CHECKFRAMECTRL
CHECKIOMAN
CHECKLOG
CHECKMEMORY
CHECKRUNTIME
COMMANDS
DEFAULT
DELETE
INDEX
INSERT
LOCK
LONG
OBJECT
OBJECTADD
OBJECTALTER
OBJECTFREE
OBJECTGET
OPTIMIZER
ORDER
PAGES
SELECT
STANDARD
TABLE
TIME
TOPICALLOCATOR
TOPICCONVERTER
TOPICDATA
TOPICDEVSPACE
TOPICFRAMECTRL
TOPICIOMAN
TOPICLOG

TOPICMEMORY
TOPICRUNTIME
UPDATE

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Creating the Text Version of the Database Trace

Use

Use this command only when told to do so by SAP DB Support [See SAP DB Library]. The database trace is not required for normal database operation.

You transfer the binary version of the database trace to the text version. You can use all values and combinations of values for **<option>** options that you received as a reply when Displaying the Options for the Text Version of the Database Trace [Page 37].

The binary version of the database trace has the file name `knltrace` and is stored in the run directory [See SAP DB Library] of the database instance (see also: Database Files [See SAP DB Library]).

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14] or DBFileRead [Page 18].

You have created the database trace (Activating the Database Trace [See SAP DB Library]).

You have determined the file ID of the database trace file (List of Database Files [Page 51]).

Syntax

```
trace_prot <option> | <option>...
```

Creating the text version of the database trace for the areas `a - Order Interface` and `b - Record Interface`:

```
dbmcli -d DB -u dbm,dbmp trace_prot ab

OK
```

Reply

An OK message.

In the event of errors, see Reply Format [Page 193].

Accessing the Database Analyzer Program

The Database Analyzer [See SAP DB Library] program is a tool for analyzing the database instance [See SAP DB Library]. It replaces the previous tools `x_wizard` and `X_wiztrc`, and makes it easier to find out the causes of performance problems.

You can use the Database Analyzer irrespective of database instance type [See SAP DB Library] and the version of the database software. It is possible to access the database instance from a remote server with this program.

Overview of the Commands for the Database Analyzer [Page 42]

Overview of the Commands for the Database Analyzer

Calling the Database Analyzer [Page 42]	**dban_start**
Ending the Database Analyzer [Page 44]	**dban_stop**
Determining the Status of the Database Analyzer [Page 46]	**dban_state**

Calling the Database Analyzer

Use

You call the Database Analyzer program in the Database Manager. The Database Manager then transfers the database names, the user data for the SYSDBA user and the options specified with the command to the Database Analyzer.

If the Database Analyzer is already active, only its status is determined.

Syntax

dban_start [<options>]

```
d:\v74>dbmcli -d a74 -u dbm,dbm dban_start
```

Reply

You receive the output of the Database Analyzer.

Example of a Successful Call of the Database Analyzer [Page 43]

Example of an Error when Calling the Database Analyzer [Page 43]

Example of a Successful Call of the Database Analyzer

Name of the database instance: `a74`

Name and operator password of the DBM operator: `dbm,dbm`

`d:\v74>dbmcli -d a74 -u dbm,dbm dban_start`

Reply:

```
OK
0,OK: everything works fine
0, Database Analyzer -d A74 -u *,* -state

SAP DB Database Analyzer, The Performance Analysis Tool, Version 7.4
Copyright (c) 1998-2002 by SAP AG


Used protocol directory: D:\V74\USR\wrk\A74\analyzer
Used configuration file: D:\V74\USR\env\dbanalyzer74.cfg

INFO 13: Database Analyzer active in directory
"D:\V74\USR\wrk\A74\analyzer".

Database          = A74
Node              =
Rundirectory      = D:\V74\USR\wrk\A74
Configfile        = D:\V74\USR\env\dbanalyzer74.cfg
Protocoldirectory = D:\V74\USR\wrk\A74\analyzer
Interval         = 900
Reconnect         = false
ProcessID         = 1884
SessionID         = 657
Started           = 2002-09-11 09:16:29
```

The Database Analyzer was successfully started or was already active.

Example of an Error when Calling the Database Analyzer

Name of the database instance: `a74`

Name and operator password of the DBM operator: `dbm,dbm`

`d:\v74>dbmcli -d a74 -u dbm,dbm dban_start`

Reply:

```
ERR
-24964,ERR_EXECUTE: error in program execution
```

```
1,Database Analyzer -d A74 -u *,* -state

SAP DB Database Analyzer, The Performance Analysis Tool, Version 7.4
Copyright (c) 1998-2002 by SAP AG

ERROR 4: Cannot connect to database.
-709 [SAP AG][SQLOD32 DLL][SAP DB]Unable to connect to data source;-
709 CONNECT:(database not running).
```

The Database Analyzer could not be started.
The system displays a description of the cause of the error.

Ending the Database Analyzer

Use

You end the Database Analyzer program in the Database Manager.

If the Database Analyzer is already active, only its status is determined.

Syntax

dban_stop

```
d:\v74>dbmcli -d a74 -u dbm,dbm dban_stop
```

Reply

You receive the output of the Database Analyzer.

Example of Ending the Database Analyzer Successfully [Page 44]

Example of Errors when Ending the Database Analyzer [Page 45]

Example of Ending the Database Analyzer Successfully

Name of the database instance: `a74`

Name and operator password of the DBM operator: `dbm,dbm`

```
d:\v74>dbmcli -d a74 -u dbm,dbm dban_stop
```

Reply:

```
OK
0,OK: everything works fine
```

```
0, Database Analyzer -d A74 -u *,* -stop

SAP DB Database Analyzer, The Performance Analysis Tool, Version 7.4
Copyright (c) 1998-2002 by SAP AG

Used protocol directory: D:\V74\USR\wrk\A74\analyzer
Used configuration file: D:\V74\USR\env\dbanalyzer74.cfg
```

The Database Analyzer was successfully ended.

Examples of Errors when Ending the Database Analyzer

Name of the database instance: `a74`

Name and operator password of the DBM operator: `dbm,dbm`

`d:\v74>dbmcli -d a74 -u dbm,dbm dban_stop`

Response Example 1:

```
ERR
-24964,ERR_EXECUTE: error in program execution
1, Database Analyzer -d A74 -u *,* -stop

SAP DB Database Analyzer, The Performance Analysis, Version 7.4
Copyright (c) 1998-2002 by SAP AG

Used protocol directory: D:\V74\USR\wrk\A74\analyzer
Used configuration file: D:\V74\USR\env\dbanalyzer74.cfg

ERROR 20: Database Analyzer not active in directory
"D:\V74\USR\wrk\A74\analyzer".
```

The Database Analyzer could not be ended.
`ERROR 20` means that the Database Analyzer is not active.

Response Example 2:

```
ERR
-24964,ERR_EXECUTE: error in program execution
1,dbanalyzer -d A74 -u *,* -stop

SAP DB Database Analyzer, The Performance Analysis Tool, Version 7.4
Copyright (c) 1998-2002 by SAP AG

ERROR 4: Cannot connect to database.

-709 [SAP AG][SQLOD32 DLL][SAP DB]Unable to connect to data source;-
709 CONNECT: (database not running).
```

The Database Analyzer could not be ended.
The system displays a description of the cause of the error.

Determining the Status of the Database Analyzer

Use

You determine whether the Database Analyzer is active.

Syntax

dban_state

d:\v74>dbmcli -d a74 -u dbm,dbm dban_state

Reply

You receive the output of the Database Analyzer.

Example of Determining the Status Successfully [Page 46]

Example of an Error when Determining the Status [Page 47]

Example of Determining the Status Successfully

Name of the database instance: `a74`

Name and operator password of the DBM operator: `dbm,dbm`

d:\v74>dbmcli -d a74 -u dbm,dbm dban_state

Response Example 1:

```
OK
0,OK: everything works fine
0, Database Analyzer -d A74 -u *,* -state

SAP DB Database Analyzer, The Performance Analysis Tool, Version 7.4
Copyright (c) 1998-2002 by SAP AG

Used protocol directory: D:\V74\USR\wrk\A74\analyzer
Used configuration file: D:\V74\USR\env\dbanalyzer74.cfg

INFO 20: Database Analyzer not active in directory
"D:\V74\USR\wrk\A74\analyzer".
```

The status was successfully determined. `INFO 20` means that the Database Analyzer is not active.

Response Example 2:

```
OK
0,OK: everything works fine
0,dbanalyzer -d A74 -u *,* -state

SAP DB Database Analyzer, The Performance Analysis Tool, Version 7.4
Copyright (c) 1998-2002 by SAP AG
```

```
Used protocol directory: D:\V74\USR\wrk\A74\analyzer
Used configuration file: D:\V74\USR\env\dbanalyzer74.cfg

INFO 13: Database Analyzer active in directory
"D:\V74\USR\wrk\A74\analyzer".

Database          = A74
Node              =
Rundirectory      = D:\V74\USR\wrk\A74
Configfile        = D:\V74\USR\env\dbanalyzer74.cfg
Protocoldirectory = D:\V74\USR\wrk\A74\analyzer
Interval          = 900
Reconnect         = false
ProcessID         = 1896
SessionID         = 639
Started           = 2002-09-10 16:51:40
```

The status was successfully determined. `INFO 13` means that the Database Analyzer is active. The system displays additional information.

Example of an Error when Determining the Status

Name of the database instance: `a74`

Name and operator password of the DBM operator: `dbm,dbm`

`d:\v74>dbmcli -d a74 -u dbm,dbm dban_stop`

Reply

```
ERR
-24964,ERR_EXECUTE: error in program execution
1, Database Analyzer -d A74 -u *,* -state

SAP DB Database Analyzer, The Performance Analysis, Version 7.4
Copyright (c) 1998-2002 by SAP AG

ERROR 4: Cannot connect to database.

-709 [SAP AG][SQLOD32 DLL][SAP DB]Unable to connect to data source;-
709 CONNECT:(database not running).
```

The state could not be determined.
The system displays a description of the cause of the error.

Accessing Database Files

Use

Use this command to access database files [See SAP DB Library] in the current database instance.

Prerequisites

Note the operator authorizations [Page 13] required for the particular DBM commands [Page 35].

Overview of the Commands for Accessing Database Files [Page 48]

Overview of the Commands for Accessing Database Files

Displaying the Diagnosis History [Page 48]	**diag_histlist**
Editing Log Files of the Database Instance [Page 49]	**file_operation**
Scrolling Through a Database File [Page 50]	**file_getnext**
List of the Database Files [Page 51]	**file_getlist**
Opening a Database File [Page 52]	**file_getfirst**
Packing Database Files [Page 53]	**diag_pack**
Backing Up a Database File [Page 54]	**file_backup**
Restoring a Database File [Page 55]	**file_restore**

Displaying the Diagnosis History

Use

You display the available diagnosis history.

If a database instance [See SAP DB Library] was not properly stopped, the system backs up certain database files during the restart of the database instance (Starting the Database Instance [Page 67]). These files are required for the diagnosis of problems that occurred previously. The files are backed up by the system in a diagnosis backup and stored in the directory `DIAGHISTORY` in the run directory [See SAP DB Library] of the database instance (Database Manager default value). You can define this path using the parameter `DIAG_HISTORY_PATH`.

If you run the command without specifying the option **<YYYYMMDDHHMMSS>**, the system displays a list of all available diagnosis backups. These are listed with their time stamps and the paths under which the backups [See SAP DB Library] are stored. This means that you can also access the diagnosis backup using operating system resources.

If you enter the command again with the option **<YYYYMMDDHHMMSS>** (timestamp), the system displays the file IDs of the files belonging to the diagnosis backup. Using the file ID, you can display the contents of a database file (Opening a Database File [Page 52]).

Prerequisites

You have the DBM operator authorization DBFileRead [Page 18].

Syntax

diag_histlist [<YYYYMMDDHHMMSS>]

Command without option specified

```
dbmcli -d DB -u dbm,dbmp diag_histlist
```

```
OK
20010706140709
d:\sapdb\usr\wrk\DB\DIAGHISTORY\A74_20010706_14-07-09
20010706162223
d:\sapdb\usr\wrk\DB\DIAGHISTORY\A74_20010706_16-22-23
```

Command with option specified

```
dbmcli -d DB -u dbm,dbmp diag_histlist 20010706162223
```

```
OK
DIAGHIST#20010706_16-22-23\knldiag
DIAGHIST#20010706_16-22-23\knldump
DIAGHIST#20010706_16-22-23\knltrace
DIAGHIST#20010706_16-22-23\rtedump
```

Reply

The result of the request depends on whether you specify the option **<YYYYMMDDHHMMSS>** (time stamp).

After Entering a Command Without an Option

If you do not specify a time stamp, the system displays a list of all available diagnosis backups.

```
OK<NL>
<YYYYMMDDHHMMSS> <directory><NL>
<YYYYMMDDHHMMSS> <directory><NL>
...
```

After Entering a Command with an Option

If you specify a time stamp, the system displays a list of the files that belong to the diagnosis backup.

```
OK<NL>
<file_id><NL>
<file_id><NL>
...
```

In the event of errors, see Reply Format [Page 193].

Editing Log Files of the Database Instance

Use

You are deleting part of the content of the log files of the database instance [See SAP DB Library] or complete files.

This command can be used for log files only and not for other database files [See SAP DB Library]. If you use the command on a file for which it is not valid, you generate the error
`-24996 ERR_PARAM - wrong parameters.`

You specify the operation that you want to perform with the parameter **OP**.

Prerequisites

You have the DBM operator authorization DBFileRead [Page 18].

You have used the command for displaying the list of database files [Page 51] to find out the file ID of the file or group of files that you want to edit.

Syntax

file_operation <file_id> OP=DELETE [DATE=<yyyymmddhhmmss>] | OP=SHRINK DATE=<yyyymmddhhmmss>|LINE=<n>

<file_id>	Log file(s) of the database instance
OP=DELETE	The file(s) is/are deleted. You can specify that all files that have not been changed since the specified date, **DATE=<yyyymmddhhmmss>**, are to be deleted.
OP=SHRINK	Part of the content of the file is deleted. If you also specify the parameter **DATE=<yyyymmddhhmmss>**, all lines that were written before the specified date are deleted. If, instead, you specify the parameter **LINE=<n>**, the contents of the file are reduced to **<n>** lines. The **<n>** lines that were written most recently are retained.

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Scrolling Through a Database File

Use

You are requesting the rest of the content of an opened database file. Use the **<file_handle>** displayed by the system when you open the database file [Page 52].

Prerequisites

You have opened a database file. The keyword `CONTINUE` in the reply shows that you have not yet read the whole file.

You have the DBM operator authorization DBFileRead [Page 18].

You have determined the file ID of the file (List of Database Files [Page 51]).

Syntax

`file_getnext <file_id> <file_handle>`

Reply

Output in ASCII Mode

```
OK<NL>
[CONTINUE|END]<NL>
<file_length>:20<data_length>:20<NL>
<data><NL>
<data><NL>
...
```

Output in Binary Mode

```
OK<NL>
[CONTINUE|END]<NL>
<file_length><data_length><NL>
<data><NL>
```

Values for the individual fields of the reply

END	Contents of the file have been completely transferred, file is automatically closed
CONTINUE	File has additional entries that have not been transferred due to the limited size of the reply memory. Call these by entering the command used above again
<file_length>	File length
<data_length>	Data length
<data>	Data value

The total length of file is not determined again and is, therefore, set to 0.

List of the Database Files

Use

You are requesting a list of the database files of the current database instance [See SAP DB Library].

You determine the scope of the displayed file list by using the parameter **<list_level>**.

Possible values for **<list_level>**:

0	Default value of the Database Manager CLI, display the most important files
1	Display all files

Prerequisites

You have the DBM operator authorization DBFileRead [Page 18].

Syntax

file_getlist <list_level>

Only those files are listed that are actually available. The list may therefore vary with the state of the database instance.

Reply

An OK message. Then it lists the database files currently available.

```
OK
key_name,mode,size,date,time,comment,file_name
<file_id> (ASCII|BINARY) <file_length> <date> <time> <comment>
<file_name>
<file_id> (ASCII|BINARY) <file_length> <date> <time> <comment>
<file_name>
...
```

Values for the individual fields of the reply

`file_id`	Key identifying name of a database file
`file_length`	Size of database file
`date`	Date last changed
`time`	Time last changed
`comment`	Additional information
`file_name`	File name

Opening a Database File

Use

You open the database file specified under **<file_id>** and transfer the first block. Using the **<file_id>**, the Database Manager determines the actual name of the file and whether it is a binary or text file.

File access using this command is limited to the database files [Page 196]. For this reason this command does not necessitate logging on to the operating system.

You can request the files that are currently available and their `<file_id>` with the command **file_getlist** (List of the Database Files [Page 51]).

Prerequisites

You have the DBM operator authorization DBFileRead [Page 18].

Syntax

file_getfirst <file_id>

Options for <file_id>

See: Database Files [Page 196]

Reply

Output in ASCII Mode

```
OK<NL>
<file_handle><NL>
[CONTINUE|END]<NL>
<file_length>:20<data_length>:20<NL>
<data><NL>
<data><NL>
...
```

Output in Binary Mode

```
OK<NL>
file_handle><NL>
[CONTINUE|END]<NL>
<file_length><data_length><NL>
<data><NL>
```

Values for the individual fields of the reply

<file_handle>	Numeric value that can be used for subsequent access to this file.
END	The contents of the file have been transferred in full. The file is closed automatically.
CONTINUE	The file contains further entries that were not transferred due to the limited storage available for replies. Interrogate this data by entering the above command or close the file.
<file_length>	File length
<data_length>	Data length
<data>	Data

In ASCII mode the data is read from the file line by line. Each line is extended by a line feed (0x0D,0x0A) in the output area.

On a UNIX server, the line feed only consists of one character. This is why the value initially output for the file length may be smaller than the data length.

In the event of errors, see Reply Format [Page 193].

Packing Database Files

Use

To be able to comprehensively diagnose a database problem, SAP DB-Support [See SAP DB Library] requires a significant number of database instance [See SAP DB Library] files.

Use this command to collect all of the required database files [See SAP DB Library] in a compressed archive file.

Default values of the Database Manager for the archive file:

File name	diagpack.tgz

Directory	Run directory [See SAP DB Library] for the database instance
`<file_id>`	`DIAGTGZ`

When opening the file, note that it is stored in binary format (Opening a Database File [Page 52]).

Prerequisites

You have the DBM operator authorization DBFileRead [Page 18].

Syntax

```
diag_pack [file=<archive>] [class=<class_spec> | <class_spec>,...]
[date=<time_stamp>]
```

<archive>	Name and path of the archive file that you want to create
<class_spec>	Class of files that you want to include in the package The default value of the Database Manager is the class `protocol` Separate multiple classes with commas.
<time_stamp>	Time stamp of the diagnosis backup Additional parameter when class `hist` specified Determine this by Displaying the Diagnosis History [Page 48]

Options for <class_spec>

Class	Description
`protocol`	General log files
`backup`	Log files from backups and restores
`config`	Configuration files
`lvc`	Additional log files for a database of instance type [See SAP DB Library] liveCache

Backing Up a Database File

Use

Use this command to create a backup copy of the specified database file. Specify the name of the database file in **<**`file_id`**>**.

Normally, you do not have to create backup copies of internal Database Manager files because the program creates backups automatically every time it writes to the configuration files.

Prerequisites

You have the DBM operator authorization DBFileRead [Page 18].

You have determined the file ID of the relevant database file (List of the Database Files [Page 51]).

Syntax

```
file_backup <file_id>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Restoring a Database File

Use

Use this command to copy the backup copy of the database file specified using **<file_id>** back to its original position. If a file already exists there, it will be overwritten.

Normally, you do not have to copy backups of internal Database Manager [Page 1] files to their original position manually because the Database Manager will automatically use the backups if it is unable to access the original file.

Prerequisites

You have the DBM operator authorization DBFileRead [Page 18].

You have determined the file ID of the file that you want to copy (List of Database Files [Page 51]).

Syntax

```
file_restore <file_id>
```

Reply

There is an OK message after the command has been executed successfully.

In the event of errors, see Reply Format [Page 193].

Logging On to the Operating System

Use

So that the Database Manager can access a remote server once with operating system rights, log on to the operating system with a DBM command [Page 35] in session mode [Page 195] or script mode [Page 195].

This logon is needed for executing the following DBM Server commands [Page 35].

Creating a Database Instance [Page 78]	**db_create**
Deleting the Registration of a Version of the Database Software [Page 78]	**inst_unreg**

Registering a Version of the Database Software [Page 83]	**inst_reg**

Prerequisites

No special operator authorization [Page 13] is required for this command.

Syntax

user_system <dbm_user>,<password>

Reply

An OK message.

In the event of errors, see Reply Format [Page 193].

Displaying the Version Information of the Database Manager

Use

You display the version information of the Database Manager.

Prerequisites

You do not need to log on to the Database Manager to execute this DBM command [Page 35].

Nor do you require any operator authorization [Page 13].

Syntax

dbm_version

Reply

```
OK
VERSION     = <version><NL>
BUILD       = <build_number><NL>
OS          = <os><NL>
INSTROOT    = <dependent_path><NL>
LOGON       = <logon_state><NL>
CODE        = <code><NL>
SWAP        = <swap><NL>
UNICODE     = (YES|NO)<NL>
INSTANCE    = (OLTP|LVC|CS|BW|unknown)<NL>
SYSNAME     = <os><NL>
```

Values of the reply lines

<version>	Version number of the Database Manager
<build_number>	Identification number of the Database Manager
<os>	Name of the operating system
<rundir>	Installation directory of the version-dependent components
<logon_state>	Displays whether a logon to the operating system has been performed:

	`True` logged on `False` not logged on
`<code>`	Display of the character set used in the Database Manager: ASCII or EBCDIC
`<swap>`	Displays of the swap type (internal representation of numerical values) used in the Database Manager : `no` hihi-hilo-lohi-lolo `full` lolo-lohi-hilo-hihi `half` lohi-lolo-hihi-hilo
`UNICODE`	Configuration of the current database instance as a UNICODE database
`INSTANCE`	Database Instance Type [See SAP DB Library]
`SYSNAME`	Name of the operating system

In the event of errors, see Reply Format [Page 193].

Executing the liveCache Initialization Script

This command is only relevant in connection with SAP applications.

Use

You are logged on to the Database Manager CLI [Page 1] and start the liveCache initialization script stored on the database server from this program.

You can specify parameters with this DBM command [Page 35] that are passed to the liveCache initialization script by the Database Manager. The output from the liveCache initialization script is supplied by the Database Manager CLI.

Prerequisites

You have the operator authorization SystemCmd [Page 15].

Syntax

```
exec_lcinit [<lcinitparams>]
```

Successful Reply

```
OK<NL>
0,<err_description><NL>
<pgmcode>,<command><NL>
<command_output_lines><NL>
```

Error Message

```
ERR<NL>
<err_code>,<err_description><NL>
<pgm_code>,<command><NL>
<command_output_lines><NL>
```

Values for the individual fields of the reply

<err_code>	Error number of the Database Manager, if successful the error number is 0.
<err_description>	Description of the error
<pgm_code>	Return value of the executed program
<command>	Command line executed by the Database Manager
<command_output_lines>	Output text of the program
<NL>	Line feed

Ending the Database Manager

Use

This command ends the session with the Database Manager.

Syntax

```
release | bye | exit | quit
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Functions for Database Operation

Use

For the day-to-day operation of the database instance [See SAP DB Library], there are DBM commands [Page 35] provided. You can use these commands to request general information on the status of the database instance and transfer the instance to the required operational state [See SAP DB Library].

Prerequisites

Note the operator authorizations required for the particular commands DBM commands [Page 35] (Operator Authorizations [Page 13]).

Overview of the Commands for Database Operation [Page 58]

Overview of the Commands for Database

Operation

Displaying the Database Kernel Variant [Page 60]	`db_speed`
Displaying the Operational State of the Database Instance [Page 60]	`db_state`
Displaying Information About the Database Instance [Page 61]	`show`
Using the Database Console [Page 62]	`db_cons`
Scrolling in the Information About the Database Instance [Page 63]	`show_next`
Loading the System Tables [Page 64]	`load_systab`
Loading SAP-Specific Tables [Page 65]	`load_r3tab`
List of the Information About the Database Instance [Page 65]	`show_list`
Deleting Runtime Information After a Database Error [Page 66]	`db_clear`
Restart the Database Instance [Page 66]	`db_restart`
Starting the Database Instance [Page 67]	`db_start`
Starting the LOAD Program [Page 68]	`exec_xload`
Starting the PythonLOAD Program [Page 68]	`exec_load`
Stopping the Database Instance [Page 69]	`db_stop`
Transferring to the ADMIN Operational State [Page 70]	`db_admin`
Transferring to the OFFLINE Operational State [Page 71]	`db_offline`
Transferring to the ONLINE Operational State [Page 71]	`db_online`

Updating Optimizer Statistics with the XPU Program

Use

The optimizer is part of the database instance [See SAP DB Library] and ensures the fastest possible processing of database accesses. To do this, it accesses the statistics about the contents of database tables. **See also:** *Optimizer: SAP DB 7.4*

You update the statistics about the contents of the tables of the database instance using the XPU program.

You use the parameters for the XPU program as parameters.

Prerequisites

You have the DBM operator authorization AccessSQL [Page 20].

The database instance is in ONLINE operational state.

Syntax

exec_xpu <xpu_params>

Reply

You receive the reply of the XPU program.

```
OK<NL>
<xpu_output>
```

In the event of errors, see Reply Format [Page 193].

Displaying the Database Kernel Variant

Use

You display the variant of the database kernel of the current database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

The database instance is in operational state [See SAP DB Library] ONLINE or ADMIN.

Syntax

db_speed

Reply

```
OK<NL>
Speed<NL>
(FAST|QUICK|SLOW|UNKNOWN|)
```

FAST	Fastest database kernel variant, with minimum logging and runtime checks.
QUICK	Variant of the database kernel with various logs and runtime checks.
SLOW	Variant of the database kernel with extensive logs and runtime checks.
UNKNOWN	The kernel variant cannot be determined.

If the database is in the OFFLINE operational state, the system answers this command with an error message.

Displaying the Operational State of the Database

Instance

Use

You display the operational state [See SAP DB Library] of the database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

Syntax

db_state

Reply

OK<NL>
State<NL>
(ONLINE|ADMIN|OFFLINE|UNKNOWN|STOPPED INCORRECTLY)

State	Description
ONLINE	The database instance is ready.
ADMIN	The database instance has been shutdown to the ADMIN operational state. Only special administrative tasks can be performed.
OFFLINE	The database instance has not been started.
STOPPED INCORRECTLY	Only with a UNIX operating system: The database stopped after an error.
UNKNOWN	The operational state cannot be determined.

Displaying Information About the Database Instance

Use

You display information about the database instance [See SAP DB Library]; about statistics, the operational state [See SAP DB Library], tasks, and so on. To do this, the **show** command obtains and outputs information from program x_cons.

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

Syntax

show <command>

Possible values for <command>

IO	Input/output activities in regular database operation

AIO	Input/output activities for data backups [See SAP DB Library]
STORAGE	Configuration and current status of the various storage areas
TASKS	List of available tasks
ACTIVE [DW|SV|US]	Active tasks for **[Datawriter-Task|Server-Task|User-Task]**
RUNNABLE [DW|SV|US]	Runnable tasks (waiting for CPU)
T_C [DW|SV|US|Tx]	Task-specific information on **[Datawriter_Task | Server Task | User Task | Task No.x]**
VERSIONS	Current variant of the database kernel and the runtime environment
REGIONS	Information about the critical regions used in the kernel
STATE	Operational state of the database instance
RTE	Runtime environment
QUEUES	Queues of runnable tasks (waiting for CPU)
SUSPENDS	Information about suspend states anywhere in the overall system
SLEEP	The CPU load from user kernel threads measured by the database kernel.
THRD_TIMES	Information from the system about the CPU load from user kernel threads (Microsoft Windows NT operating system only)
ALL	All of the information provided by the database console

Reply

```
OK<NL>
[CONTINUE]<NL>
<info_record><NL>
<info_record><NL>
```

`Continue`	More result data ready
`<info_record>`	Information

Using the Database Console

Use

You use the database console to analyze and influence the runtime behavior of the database instance [See SAP DB Library].

As parameters, you use a command for the database console, **<cons_cmd>**, and additional parameters for the database console **<cons_param>**.

Use the database console only when told to do so by SAP DB Support [See SAP DB Library].

Prerequisites

You have the DBM operator authorization DBStop [Page 17].

The database instance is in the operational state `ADMIN` or `ONLINE`.

Syntax

```
db_cons <cons_cmd> [<cons_param>]
```

Reply

You receive the database console reply.

```
OK<NL>
<console_output>
```

In the event of errors, see Reply Format [Page 193].

Scrolling in the Information About the Database Instance

Use

You display additional information about the database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

You have entered a command to display information about the database instance [Page 61]. The keyword `CONTINUE` in the reply indicates that further information is available; this information has not yet been output because of the limited size of the reply memory.

Syntax

```
show_next
```

Reply

```
OK<NL>
[CONTINUE | END]<NL>
<info_record><NL>
<info_record><NL>
```

Values in the individual fields of the reply

`END`	The requested information on the database instance has been transferred in full. The file is closed automatically.
`Continue`	More result data is available but was not transferred due to the limited size of the reply memory. You can call up this data by entering the **`show_next`** command again.
`<info_record>`	Information

Loading the System Tables

Use

System tables are internal tables for the system that users do not define or change.

You have to reload the system tables after a change of database software version.

You do not need to reload system tables when restoring a database instance [Page 158].

In order to load the system tables, the DBM Server [See SAP DB Library] must know the name and password of the SYSDBA [See SAP DB Library] and the password of the DOMAIN user [See SAP DB Library]. If this is not already the case, enter the name and password of the SYSDBA under **-u <sysdba>,<password>**, and the DOMAIN user's password under **-ud <domain_password>**.

See also: *User manual: SAP DB*

Prerequisites

You have the DBM operator authorization ExecLoad [Page 15] or LoadSysTab [Page 17].

Syntax

```
load_systab [-u <sysdba>,<password>] [-ud <domain_password>]
```

Reply

```
OK<NL>
<errcode>,<err_description><NL>
<pgmcode>,<command><NL>
<xload_output_lines><NL>
<xload_protocol_lines><NL>
```

Values for the individual fields of the reply

<errcode>	Error number of the DBM Server program. If successful the error number is 0
<err_description>	Description of the error
<pgmcode>	Return value of the executed program
<command>	Command line executed by the DBM Server program
<xload_output_lines>	Output text of the program
<xload_protocol_lines>	Lines of the LOAD log
<NL>	Line feed

Loading SAP-Specific Tables

Use

The SAP-specific tables contain information for the SAP application about the status of the database instance [See SAP DB Library].

You maintain the contents of the SAP-specific tables. Normally, these tables are maintained by the SAP application on a regular basis.

Prerequisites

You have the DBM operator authorization ExecLoad [Page 15] or LoadSysTab [Page 17].

Syntax

```
load_r3tab
```

Reply

```
OK<NL>
<errcode>,<err_description><NL>
<pgmcode>,<command><NL>
<xload_output_lines><NL>
<xload_protocol_lines><NL>
```

Values for the individual fields of the reply

<errcode>	Error number of DBM Server. If successful the error number is 0
<err_description>	Description of the error
<pgmcode>	Return value of the executed program
<command>	Command line executed by the DBM Server
<xload_output_lines>	Output text of the program
<xload_protocol_lines>	Lines of the LOAD log
<NL>	Line feed

List of the Information About the Database Instance

Use

You display the list of all keywords <keyword> for information about the database instance. You then use the command **show <keyword>** (Displaying Information About the Database Instance [Page 61]) to display the desired information.

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

Syntax

```
show_list
```

Deleting Runtime Information After a Database Error

Use

If runtime information is still present in the operating system after a database error, it is not possible to restart the database instance [See SAP DB Library].

Using this command, you remove this runtime information from the operating system and therefore allow the restart of the database instance.

Prerequisites

You have the DBM operator authorization DBStop [Page 17].

Syntax

```
db_clear
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Restarting the Database Instance

Use

You transfer the database instance from the operational state [See SAP DB Library] `ADMIN` or `ONLINE` first into the `OFFLINE` operational state and then into the operational state `ONLINE`.

Prerequisites

You have the DBM operator authorization DBStop [Page 17].

Syntax

```
db_restart [<option>]
```

Options

You can use the following options **(<option>)** with the command:

-f | -fast [Page 80]	The *fast* variant of the database kernel is started.
-q | -quick [Page 80]	The *quick* variant of the database kernel is started.
-s | -slow [Page 81]	The *slow* variant of the database kernel is started.

-d | -dump	A dump is generated on transferring from ADMIN or ONLINE to OFFLINE.

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Starting the Database Instance

Use

You are switching the database state from the operational state [See SAP DB Library] OFFLINE to the state ADMIN (Transferring to the Operational State ADMIN [Page 70]).

Prerequisites

You have the DBM operator authorization DBStart [Page 17].

On the Microsoft Windows operating system, this database instance must be registered as a service under the name SAP DB: <database_name>. This is done implicitly when you create a database instance [Page 78] with the Database Manager.

Syntax

db_start [<option>]

Options

Options **(<options>)** that determine the kernel variant that starts the database instance:

-f | -fast [Page 80]

-q | –quick [Page 80]

-s | –slow [Page 81]

If you do not specify an option, the system makes a service entry for the fastest available variant of the database kernel.

You should only use the options **-q | -quick** and **-s | -slow** if errors occur and in consultation with SAP DB Support [See SAP DB Library].

Reply

An OK message.

In the event of errors, see Reply Format [Page 193].

Starting the LOAD Program

Use

With this command, the LOAD program is executed on the current database instance with the operator specified in **<userid> <password>**. Specify a LOAD script file **<file>**. LOAD options in double quotation marks are transferred uninterpreted to the LOAD program. When specifying a relative file name, this is used relative to the directory `<dependent_path>\env`.

(**See also:** Database Manual LOAD)

Prerequisites

You have the DBM operator authorization ExecLoad [Page 15].

Syntax

```
exec_xload <userid> <password> <file> ["<option>"]
```

Reply

```
OK<NL>
<errcode>,<errdescription><NL>
<pgmcode>,<command><NL>
<xload_output_lines><NL>
<xload_protocol_lines><NL>
```

Values for the individual fields of the reply

<errcode>	Error number of the DBM Server program. If successful the error number is 0
<errdescription>	Description of the error
<pgmcode>	Return value of the executed program
<command>	Command line executed by the DBM Server program
<xload_output_lines>	Output text of the program
<xload_protocol_lines>	Lines of the LOAD log
<NL>	Line feed

Starting the PythonLOAD Program

Use

PythonLOAD is an environment programmed in the script language Python that allows load operations to be executed. As a response to the execution of this script, the system displays information about the execution status of the program as a reply.

The PythonLOAD program is executed on the current database instance [See SAP DB Library] with the operator specified in **<userid> <password>**. Specify a PythonLOAD script

file **<file>**. If you specify a relative file name, it is used relative to the directory <dependent_path>\env.

PythonLOAD options in double quotation marks are transferred uninterpreted to the PythonLOAD program.

Prerequisites

You have the DBM operator authorization ExecLoad [Page 15].

Syntax

```
exec_load <userid> <password> <file> ["<option>"]
```

Reply

```
OK<NL>
<errcode>,<errdescription><NL>
<pgmcode>,<command><NL>
<pythonload output lines><NL>
```

Values for the individual fields of the reply

<errcode>	Error number of the DBM Server program. If successful the error number is 0
<errdescription>	Description of the error
<pgmcode>	Return value of the executed program
<command>	Command executed by the DBM Server program
<pythonload output lines>	Output text of the program
<NL>	Line feed

Stopping the Database Instance

Use

With this command, you transfer the database instance [See SAP DB Library] from the operational state [See SAP DB Library] ADMIN or ONLINE to the operational state OFFLINE.

Prerequisites

You have the DBM operator authorization DBStop [Page 17].

Syntax

```
db_stop [<option>]
```

Option

If you specify the value **-d | -dump** as an **<option>**, a dump is created when the database instance transfers from the ADMIN to the OFFLINE operational state.

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Transferring to the ADMIN Operational State

Use

The database instance [See SAP DB Library] is transferred to the `ADMIN` operational state [See SAP DB Library].

If the database instance is in the operational state `ONLINE`, it is first transferred to the `OFFLINE` operational state and only then to the `ADMIN` operational state.

Prerequisites

You have the DBM operator authorization DBStop [Page 17].

Syntax

```
db_admin [<option>]
```

Options

Options **<option>** you can use with the command:

-f | -fast [Page 80]	When transferring from the `OFFLINE` operational state to `ADMIN` under Microsoft Windows NT/Windows 2000, the *fast* variant of the database kernel is started.
-q | -quick [Page 80]	When transferring from the `OFFLINE` operational state to `ADMIN` under Microsoft Windows NT/Windows 2000, the *quick* variant of the database kernel is started.
-s | -slow [Page 81]	When transferring from the `OFFLINE` operational state to `ADMIN` under Microsoft Windows NT/Windows 2000, the *slow* variant of the database kernel is started.

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Transferring to the OFFLINE Operational State

Use

The database instance is transferred to the operational state [See SAP DB Library] `OFFLINE`.

Prerequisites

You have the DBM operator authorization DBStop [Page 17].

Syntax

```
db_offline [<option>]
```

Options

You can enter the command with the following option **<option>** :

-d | -dump	A dump is generated when transferring from the operational states `ADMIN` or `ONLINE` to `OFFLINE`.

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Transferring to the ONLINE Operational State

Use

You transfer the database instance [See SAP DB Library] to the `ONLINE` operational state [See SAP DB Library].

Prerequisites

You have the operator authorization DBStart [Page 17].

Syntax

```
db_online [<option>]
```

Options

Options **<option>** you can use with the command:

-f | -fast [Page 80]	The *fast* variant of the database kernel is started.
-q | -quick [Page 80]	The *quick* variant of the database kernel is started.
-s | -slow [Page 81]	The *slow* variant of the database kernel is started.

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Installation and Registration Management

Use

You manage information on the versions of the database software and the generated database instances [See SAP DB Library] in Registration Management.

DBM Commands [Page 35] are provided for maintaining the installation(s) of the database software and for registering and deleting database instances. Many of these commands, however, do not need to be called explicitly during normal database operation, but are called by installation programs of the database software or by the database application.

Overview of the Commands for Installation and Registration Management [Page 72]

Prerequisites

Note the operator authorizations required for the particular DBM commands [Page 35] (Operator Authorizations [Page 13]).

Overview of the Commands for Installation and Registration Management

Changing the Software Version of the Current Database Instance [Page 73]	`db_reg -R`
Creating a Database Instance [Page 78]	`db_create`
Displaying the Version-Independent Directories [Page 73]	`dbm_getpath`
Defining the Version-Independent Directories [Page 74]	`dbm_setpath`
List of All Registered Database Instances [Page 74]	`db_enum`
List of All Registered Versions of the Database Software [Page 75]	`inst_enum`
Deleting the Current Database Instance [Page 76]	`db_drop`
Deleting the Registration of a Variant of the Current Database Instance [Page 77]	`db_unreg`
Deleting the Registration of a Version of the Database Software [Page 78]	`inst_unreg`
Registering a Variant of the Current Database Instance [Page 82]	`db_reg`
Registering a Version of the Database Software [Page 83]	`inst_reg`

Displaying the Version-Independent Directories

Use

You display the version-independent directories configured on the database server.
If more than one version of the database software is installed on your computer, version-independent data or programs are stored and found in these directories.

Prerequisites

You do not need to log on to the Database Manager to execute this DBM command [Page 35]. Nor do you require any operator authorization [Page 13].

Syntax

```
dbm_getpath <independent_data_path> | <independent_program_path>
```

See also: Variables [See SAP DB Library]

Changing the Software Version of the Current Database Instance

Use

You change the version of the database software for the current database instance [See SAP DB Library].

You can only switch between versions of the SAP DB software that **do not** require migration of the database instance.

Prerequisites

The database instance is in operational state `OFFLINE` (Taking the Database Instance OFFLINE [Page 71]).

You have called the Database Manager CLI with the -d [Page 31] option.

Syntax

```
db_reg -R
```

Procedure

1. End all sessions with the DBM Server [See SAP DB Library] program, particularly all sessions in the Database Manager GUI [Page 1].
2. Run the command **`db_register -R`**
3. If you are working with the Database Manager CLI in session mode [Page 195], terminate the Database Manager CLI now.

Result

For every subsequent access to this database instance, the Database Manager program uses the new version of the database software.

Defining the Version-Independent Directories

Use

You define the version-independent directories.
If more than one version of the database software is installed on your computer, version-independent data or programs are stored and found in these directories.

Prerequisites

You do not need to log on to the Database Manager to execute this DBM command [Page 35].

Nor do you require any operator authorization.

Syntax

```
dbm_setpath <independent_data_path> | <independent_program_path>
<path>
```

See also: Variables [See SAP DB Library]

Reply

An OK message.

In the event of errors, see Reply Format [Page 193].

List of All Registered Database Instances

Use

You can call up the list of all database instances [See SAP DB Library] registered on the server.

If you specify the `[-s]` option, the service database instances for each version of the database software are also listed (Access Using a Service Session [Page 183]).

Prerequisites

You have called the Database Manager using the -n [Page 32] option.

You do not need to log on to the Database Manager to execute this DBM command [Page 35].

Nor do you require an operator authorization.

Syntax

```
db_enum [-s]
```

Reply

The system outputs an OK message. Then a table is displayed, listing all of the registered database instances in tabular format.

```
OK<NL>
<database_name><TAB><dependent_path><TAB><version>[<TAB><description>
]<NL>
```

```
<database_name><TAB><dependent_path
><TAB><version>[<TAB><description>]<NL>
...
```

Reply values

<database_name>	Name of database instance
<dependent_path>	Directory that contains the programs dependent on the database version.
<version>	Version of the database software
<description>	More information about the database instance

```
myDB1 C:\Program Files\SAP DBTech\V72" 7.2.1.0  fast
   offline
myDB2 C:\Program Files\SAP DBTech\V73" 7.3.0.14 slow
   running
```

List of All Registered Versions of the Database Software

Use

You can call up the list of all registered versions of the database software on the database server.

Prerequisites

You do not need to log on to the Database Manager to execute this DBM command [Page 35].

Nor do you require an operator authorization.

Syntax

inst_enum

Reply

The system outputs an OK message. The system then displays a table of the registered versions <version> and their installation paths <dependent_path>.

```
OK<NL>
<version><TAB><dependent_path><NL>
<version><TAB><dependent_path><NL>
...
```

```
OK
7.2.1.0  "C:\Program Files\SAP DBTech\V72"
7,30,00,14  "C:\Program Files\SAP DBTech\V73"
```

Deleting the Current Database Instance

Use

Use this command to delete a database instance [See SAP DB Library].

If you specify the option **WITHOUTFILES** for this command, the database files on the DBM Server [See SAP DB Library] are retained. If you do not specify this option, the files for the database instance are also deleted.

Prerequisites

The database instance is in the operational state `OFFLINE` (Taking the Database Instance OFFLINE [Page 71]).

You have the DBM operator authorization InstallMgm [Page 16].

Syntax

```
db_drop [WITHOUTFILES]
```

Example [Page 76]

Reply

An OK message.

In the event of errors, see Reply Format [Page 193].

db_drop Command: Example

1. Stop the database instance [See SAP DB Library]:

   dbmcli –d [Page 31] **<database_name> <dbm_user>,< password>** db_stop [Page 69]

   or

   dbmcli -d <database_ name> -u [Page 22] **<dbm_user>,< password>** db_offline [Page 71]

2. Delete the current database instance from the database server:

   dbmcli -d <database_ name> -u <dbm_user>,< password> db_drop [Page 76]

   This deletes the database instance's files.

Deleting the Registration of a Variant of the Current Database Instance

Use

You can register several variants of one database instance [See SAP DB Library] (Registering a Variant of the Current Database Instance [Page 82]). The difference between these variants is that they use different variants of the database kernel.

With this command, you remove the specified variant of the current database instance from the server. Once you have done this, you will not be able to start this variant of the database instance again. The deletion procedure has no effect on the current operational state [See SAP DB Library] of the database instance. A database instance in `ONLINE` or `ADMIN` operational state remains active and is therefore recognized by the system.

All parameters listed here apply only for the Microsoft Windows NT/Windows 2000 operating systems. The command is ignored under the UNIX operating system.

Prerequisites

The database instance is in the `OFFLINE` operational state.

You have the DBM operator authorization InstallMgm [Page 16].

Syntax

```
db_unreg <option>
```

Options

You use the following options (**<option>**) to specify which database instance entry variant is to be deleted:

-f | -fast [Page 80]

-q | -quick [Page 80]

-s | -slow [Page 81]

You should only use the options **-q | -quick** and **-s | -slow** if errors occur and in consultation with SAP DB Support [See SAP DB Library].

```
db_unreg myDB -f
```

The registration of the **fast** variant of database instance **myDB** is deleted.

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Deleting the Registration of a Version of the Database Software

Use

You remove an entry from the list of registered versions of the database software. Components of a version that require this entry are then no longer operable.

Specify the installation path **<dependent_path>** as a parameter.

Prerequisites

You have called the Database Manager CLI with -R [Page 29] option.

You are logged on to the operating system (Operating System Logon [Page 55]).

You do not require any DBM operator authorization.

Syntax

```
inst_unreg <dependent_path>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Creating a Database Instance

Use

You are the operating system user and are creating a new database instance [See SAP DB Library].

Prerequisites

The following prerequisites must be met on the server on which you want to create the database instance:

Communication	Microsoft Windows	UNIX
local	Operating system user has administration rights for database server * (Implicit logon to the operating system)	Logon on to the operating system using option –s [Page 30] (local operation)
remote	Operating system user has administration rights for database server and the *log on as batch job* * right Log on to the operating system using **db_create** and option **<os_user>,<password>**	Log on to the operating system using **db_create** and the option **<os_user>,<password>**

If several versions of the database software are available on the database server, you must specify the path of the version to which this database instance is to be assigned when

registering the database instance. To do this, use the option -R [Page 29] (installation directory of the database software) when calling the Database Manager CLI.

*(You can find information about maintaining operating system users in your operating system documentation.)

Syntax

```
db_create [<option>] <database_name> <dbm_user>,<password>
[<os_user>,<password>]
```

<database_name>	Name of database instance, maximum length eight characters
<userid>,<password>	DBM operator [See SAP DB Library] This operator is stored on the database server when you start to create the database. Subsequent access to the database instance is possible only with this operator.
[<os_user>,<password>]	Operating system user If you want to create the database instance on a remote server, you must also specify the operating system user for this server. If the logon to the operating system fails, the database instance cannot be installed.

From now on, to access this database instance you need the **<userid>** and **<password>** specified at registration.

Options

Options **(<option>)** that determine which database instance kernel variant is registered:

-f | -fast [Page 80]

-a | –auto [Page 80]

You should only use the options -q | –quick [Page 80] and -s | –slow [Page 81] if errors occur and in consultation with SAP DB Support [See SAP DB Library].

If you do not specify an option, the system makes a service entry for the fastest available variant of the database kernel.

All parameters listed here apply to the Microsoft Windows operating system only.

These parameters are ignored on the UNIX operating system.

Creating a local database instance on Microsoft Windows:
```
db_create myDB1 samplename,secret
```

Creating a remote database instance on Microsoft Windows:
```
db_create myDB2 samplename,secret winuser,win
```

Reply

There is an OK message after the command has been executed successfully.

In the event of errors, see Reply Format [Page 193].

Example: How to Create a Database Instance [Page 81]

Kernel Variant -a | -auto

-a | -auto	If you use this option, under Microsoft Windows NT the service entry *Start-up-type* in the Control Panel is set to `automatic`. Under the UNIX operating system this parameter is ignored.

Kernel Variant -f | -fast

-f | -fast	Default value of the Database Manager CLI Fastest database kernel variant, with minimum logging and runtime checks. Specifying this option has the following effect: **Microsoft Windows** When you create a database instance [Page 78], the service entry of the *fast* variant of the database kernel is used. The kernel variant *fast* is started when you enter the command that starts the database instance [Page 67]. **UNIX** The command that registers the database instance [See SAP DB Library] does not work. Specifying the option **-f | -fast** therefore only makes sense when starting the database instance.

Kernel Variant -q | -quick

-q | -quick	Variant of the database kernel with various logs and runtime checks. Specifying this option has the following effect: **Microsoft Windows** When you create a database instance [Page 78], the service entry of the *quick* variant of the database kernel is used. When the command to start the database instance [Page 67] is run, the kernel variant *quick* of the database instance [See SAP DB Library] is started. **UNIX** The command for registering the database instance does not work. Specifying the option **-q | -quick**, therefore, only makes sense when starting the database instance.

However, these options should only be used in case of error and after consulting SAP DB Support [See SAP DB Library].

Kernel Variant -s | -slow

`-s	-slow`	Variant of the database kernel with extensive logs and runtime checks. Specifying this option has the following effect: **Microsoft Windows** When you create a database instance [Page 78], the service entry of the *slow* variant of the database kernel is used. When you execute the command for starting the database instance [Page 67], the kernel variant *slow* of the database instance [See SAP DB Library] is started. **UNIX** The command for registering the database instance does not work. Specifying the option `-s	-slow`, therefore, only makes sense when starting the database instance.

However, these options should only be used in case of error and after consulting SAP DB Support [See SAP DB Library].

db_create Command: Example

To create a new database instance [Page 78] with the Database Manager CLI [Page 1], you only need a few DBMCLI commands.

First execute the command `db_create` and define the **first** Database Manager operator [See SAP DB Library].

```
dbmcli db_create <database_name> <dbm_user>,<password>
```

- If you can choose between multiple versions of the database software, make your choice by specifying the option:
  `dbmcli db_create <database_name> <dbm_user>,<password> -R <dependent_path>`.
- If you want to install a new database on a remote server, also specify the option **`-n <server_node>`**. You also need authorization for the operating system:

  ```
  dbmcli -n <server_node> db_create <database_name>
  <dbm_user>,<password> <os_user>,<password>
  ```

- On UNIX platforms there is no difference between remote and local communication. Therefore authorization is always required. When you are creating a new database instance locally, you can avoid communication and, therefore, authorization by using the option **`-s`**:

  ```
  dbmcli -s db_create <database_name> <dbm_user>,< password>
  ```

For the other steps need to create the database instance, use a DBMCLI script:

```
dbmcli -d <database_name> -u <dbm_user>,<password> -i <script_file>
```

Example of a script:

param_startsession [Page 103]
param_init [Page 104]
param_put [Page 88] `MAXUSERTASKS 5`
param_checkall [Page 111]
param_commitsession [Page 100]
param_addvolume [Page 89] `1 SYS sys_001 F`
`param_addvolume 1 LOG LOG_001 F 2000`
`param_addvolume 1 DATA DAT_001 F 10000`
db_admin [Page 70]
util_connect [Page 175]
util_execute [Page 176] `INIT CONFIG`
util_activate [Page 173] `<sysdba>,<password>`
load_systab [Page 64] `-ud <domain_password>`

Modify the values of the database parameters according to your requirements.

Registering a Variant of the Current Database Instance

Use

You register a variant of a database instance [See SAP DB Library]. If you do not specify any options, the system makes a service entry for the fastest available variant of the database kernel.

The **-f | -fast**, **-q | -quick**, and **-s | -slow** options should only be used in case of error and in consultation with SAP DB Support [See SAP DB Library].

The options **-f | -fast**, **-q | -quick**, and **-s | -slow** allow you to register a database instance that has already been registered in an additional variant. If the same variant is registered more than once, the existing entry is overwritten with the same data.

These options are only valid for the Microsoft Windows NT/Windows 2000 operating system.

Under the UNIX operating system these parameters are ignored.

Prerequisites

You have the DBM operator authorization InstallMgm [Page 16].

The database instance is in the `OFFLINE` operational state.

You have called the Database Manager CLI with the -d [Page 31] option.

Syntax

```
db_reg [<option>]
```

Options

-f | -fast [Page 80]

-q | –quick [Page 80]

-s | –slow [Page 81]

See also: Deleting the Registration of a Variant of the Current Database Instance [Page 77]

Registering a Version of the Database Software

Use

You enter the software version of the Database Manager [Page 1] from the current database installation in the list of registered versions of the database software.

The version is registered once only, upon installation of the relevant database software. The installation must be registered for this software to function correctly.

Prerequisites

You have called the Database Manager CLI with the -R [Page 29] option.

You are logged on to the operating system (Logging On to the Operating System [Page 55]).

Syntax

```
inst_reg [-k <key>][-c]
```

Options

You must specify a unique key **-k <key>** if you have installed the same version of the database software more than once.

If only the client programs of the database software have been installed on a computer, register the client with the **-c** option.

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Configuring the Database Manager

Use

You can use the following DBM commands [Page 35] to configure the Database Manager:

Displaying a Configuration Parameter of the Database Manager [Page 84]	**dbm_configget**
Changing a Configuration Parameter of the Database Manager [Page 84]	**dbm_configset**

Prerequisites

You have the operator authorization InstallMgm [Page 16].

Displaying a Configuration Parameter of the Database Manager

Use

You display the value of the configuration parameter of the Database Manager specified in **<parameter_name>**.

The parameter values are stored in the encrypted configuration file. Specify the option **-raw** to obtain the encrypted values. If you do not specify the option, the unencrypted value is displayed.

Note that parameter names are case sensitive when working with the Database Manager.

Prerequisites

You have the DBM operator authorization InstallMgm [Page 16].

Syntax

```
dbm_configget [-raw] <parameter_name>
```

Displaying the encrypted value **VALUE** with the **-raw** option.

```
dbmcli -d DB -u dbm,dbm dbm_configget -raw VALUE
OK
c92ee241db0bf5632bc029bcf98f186b440bb548c0f056eb
```

Reply

```
OK<NL>
<parameter_value><NL>
```

In the event of errors, see Reply Format [Page 193].

Changing a Configuration Parameter of the Database Manager

Use

You set the value specified for the parameter specified as **<parameter_name>** in the configuration of the Database Manager. The new parameter value is registered by the system, and is transferred to the configuration file as the valid value at the next restart [See SAP DB Library] of the database instance.

You can specify that the value is to be stored unencrypted with the option **-raw**.

Maximum number of characters for encrypted values: 18
A longer value is shortened to 18 characters by the Database Manager CLI.

Maximum number of characters for unencrypted values: 512

Prerequisites

You have the DBM operator authorization InstallMgm [Page 16].

Syntax

```
dbm_configset <parameter_name> <value>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Configuring Database Instances

Use

The database parameters [See SAP DB Library] are stored in the **description file** and the **parameter file**.

The default values, value areas, calculation rules, and parameter properties the Database Manager needs to process parameters correctly are all stored in the description file.

The description file and a parameter file that has default values assigned to it are installed with the database instance [See SAP DB Library]. With the commands for the configuration of database instances you change the entries in the parameter file of the database instance.

Prerequisites

For some of the commands for configuring database instances you require a parameter session [Page 103].

Note the operator authorizations required for the particular commands DBM commands [Page 35] (Operator Authorizations [Page 13]).

Procedure

When logging on to the Database Manager [Page 22], use the -d option to specify the name of the database instance [Page 31] for which you want to edit the parameters. On entering the **`param_*`** commands, a search is made in the parameter file belonging to this database instance for the value of the respective parameters.

If you do not specify a name or you specify the name of a non-existent database, the required parameter cannot be found.

Overview of the Commands for Configuring Database Instances [Page 87]

Parameter Properties

Property	Explanation	Possible values
`CASESENSITIVE`	Upper/lower case distinctions are relevant (for example, file	`YES` | `NO`

	names under UNIX)	
CHANGE	Parameter can be changed	**YES | NO**
CLEAR	In a new installation, the parameter must not be copied from another database instance [See SAP DB Library] (examples: database instance name, volumes [See SAP DB Library])	**YES | NO**
VOLUME	Type of parameter (**Yes**: volume parameter, **NO**: kernel parameter)	**YES | NO**
DISPLAYNAME	Character string (may include spaces) displayed in place of the parameter name in the Database Manager GUI [Page 1]	
DYNAMIC	Parameter has a position indicator in the description file, such as `DATAVOL_?` In the parameter file, it becomes `DATAVOL_00, DATAVOL_01` and so on.	**YES | NO**
GROUP	Assignment to a group for display in the Database Manager GUI **GENERAL**: General Database Parameters [See SAP DB Library], visible **EXTENDED**: Special Database Parameters [See SAP DB Library], visible **SUPPORT**: Support Database Parameters [See SAP DB Library], visible **NO:** The parameter is not accessible using the Database Manager GUI **INFO:** The parameter contains special information for users	**GENERAL** **| EXTENDED** **| SUPPORT** **| NO** **| INFO**
`INSTANCES`	`Instance type for which the parameter is relevant` `No entry: Parameter is generally valid`	**LVC | OLTP | BW | CS**
INTERN	The value is only in the description file, not in the parameter file (Configuring Database Instances [Page 85])	**YES | NO**
LASTKNOWNGOOD	`Valid value of the parameter with which the database instance was last started`	
MANDATORY	Mandatory parameter	**YES | NO**
MAX	`Maximum parameter value`	
MIN	`Minimum parameter value`	
MODIFY	You can still change the parameter after the database instance has been generated (for example: RUNDIRECTORY: **NO**)	**YES | NO**
OVERRIDE	You can overwrite the DBM Server [See SAP DB Library]'s default value	**YES | NO | HIGHER**
VALUESET	Permitted value set for a parameter The individual values must be separated by spaces or tabs. Permitted string constants in character string parameters must be set off in quotation marks.	

See also: *Database Manager GUI: SAP DB 7.4*

Overview of the Commands for Configuring Database Instances

Terminating a Parameter Session [Page 88]	`param_abortsession`
Changing the Value of a Database Parameter [Page 88]	`param_put`
Changing Volume Parameters [Page 89]	`param_addvolume`
Displaying All Data for a Database Parameter [Page 90]	`param_getfull`
Displaying All Properties of a Database Parameter [Page 92]	`param_getproperties`
Displaying All Parameters of the Current Database Parameter File [Page 92]	`param_directgetall`
Displaying the Data for All Database Parameters [Page 93]	`param_extgetall`
Displaying the Data for All Volume Parameters [Page 94]	`param_getvolsall`
Displaying the Data for Individual Volume Parameters [Page 95]	`para_getvolume`
Displaying the Current Value of a Database Parameter [Page 96]	`param_getvalue`
Displaying the Data Type of a Database Parameter [Page 97]	`param_gettype`
Displaying the Stored Explanatory Text [Page 97]	`param_getexplain`
Displaying the Stored Help Text [Page 98]	`param_gethelp`
Displaying the System Default Value [Page 98]	`param_getdefault`
Displaying a Value from the Database Parameter File [Page 99]	`param_directget`
Displaying Individual Data for a Database Parameter [Page 99]	`param_extget`
Confirming Changes to the Database Parameter File [Page 100]	`param_commitsession`
Scrolling in the Database Parameter History [Page 101]	`param_gethistorynext`
Directly Changing a Value in the Database Parameter File [Page 102]	`param_directput`
Opening a Parameter Session [Page 103]	`param_startsession`
Adding a Volume [Page 104]	`db_addvolume`
Initializing the Database Parameters for a New Database Instance [Page 104]	`param_init`
Copying a Database Parameter File [Page 105]	`param_copy`
Correcting Database Parameters [Page 106]	`param_putconfirm`
List of Available Parameter Files [Page 106]	`param_versions`
Deleting the Database Parameter File [Page 107]	`param_rmfile`
Deleting a Database Parameter [Page 107]	`param_directdel`
Deleting Volume Parameters [Page 108]	`param_delvolume`
Opening the Database Parameter History [Page 109]	`param_gethistory`
Checking All Database Parameters [Page 111]	`param_checkall`
Resetting the Parameter File to a Previous Version [Page 112]	`param_restore`

The commands of the class **param_direct*** access the parameter file of the database instance directly. These commands do not require a parameter session and they **cannot** be rolled back (Rollback [See SAP DB Library]).

Terminating a Parameter Session

Use

You terminate the parameter session. All of the changes [Page 88] and corrections [Page 106] made to database parameters in this session are rejected. The internal data structures in which the contents of the description file and the parameter file were stored are removed (see also: Configuring Database Instances [Page 85]).

If you change values without a parameter session, that is, directly in the parameter file (Make direct changes with param_directput [Page 102] and delete directly with param_directdel [Page 107]), the changes are immediately written to the parameter file, that is, they cannot be rejected.

Prerequisites

You have opened a parameter session [Page 103].

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_abortsession
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Changing the Value of a Database Parameter

Use

You can use this command to change the value of a database parameter in the internal data structures and to store it in the Database Manager.

In contrast to the function for changing a value directly in the database parameter file [Page 102], the parameter is not written straight to the parameter file and you can still reject it by not confirming the parameter change or by terminating the parameter session [Page 88].

The change of this parameter is rejected if the value entered or the change is impermissible due to the parameter properties.

After changing the database parameter, you must also execute the command to check all database parameters [Page 111].

Prerequisites

You have opened a parameter session [Page 103].

You have the DBM operator authorization ParamCheckWrite [Page 19].

Syntax

```
param_put <parameter_name> <user_value>
```

Example [Page 89]

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

param_put Command: Example

You change the value of the database parameter MAXUSERTASKS to the value 5.

param_startsession [Page 103]
param_init [Page 104]
param_put [Page 88] `MAXUSERTASKS 5`
param_checkall [Page 111]
param_commitsession [Page 100]

Changing Volume Parameters

Use

Extend an existing database instance [See SAP DB Library] using the command **db_addvolume** (Adding a Volume [Page 104]). Use the command **param_addvolume** described here only for Creating a New Database Instance [Page 78] or in consultation with SAP DB Support [See SAP DB Library].

Inconsistencies in the volume settings can lead to damage to the database instance.

Use this command to enter the volume parameters in the parameter file of the database instance. The new parameter value is registered by the system, and is transferred to the parameter file as the valid value at the next restart [See SAP DB Library] of the database instance.

You change volume parameters in two steps:

1. Change the entries for the parameters **in the parameter file**

2. Transfer the corresponding information **to the database kernel** using a utility command (Transferring a Utility or SQL Command [Page 176])

If this command fails, you must also remove the entries in the parameter file of the database instance.

No check is made of whether there is sufficient disk capacity available to add the volume.

No check is made of whether the corresponding volume of the database instance is mirrored (Log Mode [See SAP DB Library] `DUAL`).

Prerequisites

You have the DBM operator authorization ParamFull [Page 19].

Syntax

```
param_addvolume <volno> <volmode> <volname> <voltype> <volsize>
```

Options

<volno>	Volume [See SAP DB Library] number
<volmode>	Type of the volume
<volname>	Name of volume/file of the volume
<voltype>	Type of volume (such as file, raw device)
<volsize>	Size of the volume

Option <volmode>

DATA	Data Volume [See SAP DB Library]; the number of the data volume is assigned by the system as a four-digit number with leading zeros when creating the volume
LOG	Log Volume [See SAP DB Library]; the number of the log volume is assigned by the system as a three-digit number with leading zeros when creating the volume
MLOG	Mirrored log volume (Log Mode [See SAP DB Library] `DUAL`)

Reply

An OK message.

In the event of errors, see Reply Format [Page 193].

Displaying All Data for a Database Parameter

Use

You request all of the data stored for the database parameter identified as **<parameter_name>**.

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_getfull <parameter_name>
```

Example [Page 91]

Reply

The system outputs an OK message followed by all of the data for the parameter concerned.

```
OK<NL>
<type><NL>
<default><NL>
<value><NL>
<property> <value><NL>
<property> <value><NL>
...
HELP
<helpline><NL>
<helpline><NL>
...
EXPLAIN
<explainline><NL>
<explainline><NL>
...
```

Reply values

<type>	Data type of the parameter
<default>	System default value
<value>	Current value, taken from the parameter file at the start of the session
<property>	Property of the Parameter [Page 85]
<helpline>	Help text
<explainline>	Explanatory text

In the event of errors, see Reply Format [Page 193].

param_getfull Command: Example

```
Displaying the data for the parameter RUNDIRECTORY
```

```
dbmcli -d DB -u dbm,dbm param_getfull RUNDIRECTORY
```

```
OK
c64
d:\sapdb\usr\wrk\DB
CHANGE         YES
INTERN         NO
MANDATORY      YES
CLEAR          YES
DYNAMIC        NO
CASESENSITIVE  YES
OVERRIDE       NO
DEVSPACE       NO
MODIFY         YES
GROUP          GENERAL
DISPLAYNAME
VALUESET
MAX
MIN
INSTANCES
```

```
LASTKNOWNGOOD  d:\sapdb\usr\wrk\DB
HELP
Path where context and diagnosis information is stored
EXPLAIN
Path where context and diagnosis information is stored for this
database instance.
(char(64))
```

Displaying All Properties for a Database Parameter

Use

You request all properties [Page 85] of the database parameter identified as **<parameter_name>**. All properties <property> are listed with their values <value>.

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

param_getproperties <parameter_name>

Reply

```
OK<NL>
<property> <value><NL>
<property> <value><NL>
...
```

In the event of errors, see Reply Format [Page 193].

Displaying All Parameters of the Current Database Parameter File

Use

You display all database parameters [See SAP DB Library] of the current database instance [See SAP DB Library] with their values. One parameter with name <parameter_name> and value <value> is displayed per line.

The **param_directgetall** command accesses the database instance parameter file directly. It does not require a parameter session and so **cannot** be rolled back (Rollback [See SAP DB Library]).

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

param_directgetall

Reply

```
OK<NL>
<parameter_name> <value><NL>
<parameter_name> <value><NL>
...
```

In the event of errors, see Reply Format [Page 193].

Displaying the Data for All Database Parameters

Use

You request all database parameters [See SAP DB Library] from the description file and the parameter file of the database instance (see also: Configuring Database Instances [Page 85]).

If the reply is successful, the name of each database parameter `<parameter_name>`, its data type `<type>` and its current value `<value>` are displayed. This is the value that is copied from the parameter file at the start of the session. If the parameter is not in it, the default value from the description file is displayed.

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

param_extgetall [<property>=<value>]

<property> Option

By specifying properties [Page 85] you can restrict what is displayed. However, each property can only be specified once. If the same property is entered more than once, the value of the last entry is used.

A parameter must correspond to the specified value in all specified properties (AND link). Several values for one property separated by commas are treated as OR-linked.

Property	Possible values
CASESENSITIVE	**YES | NO**
CHANGE	**YES | NO**
CLEAR	**YES | NO**
VOLUME	**YES | NO**
DISPLAYNAME	Character string (may include spaces) displayed in place of the parameter name in the Database Manager GUI
DYNAMIC	**YES | NO**
GROUP	**GENERAL | EXTENDED | SUPPORT | NO**
INFO	**YES | NO**

INSTANCES	Database instance types [See SAP DB Library] for which the parameter is relevant No entry: Parameter is generally valid
INTERN	**YES | NO**
MANDATORY	**YES | NO**
MAX	Maximum parameter value
MIN	Minimum parameter value
MODIFY	**YES | NO**
OVERRIDE	**YES | NO | HIGHER**
VALUESET	Permitted parameter value set

param_getextall CHANGE=YES GROUP=GENERAL

This command returns all parameters that are changeable **and** assigned to the *General - General database parameters* group.

param_getextall GROUP=GENERAL,EXTENDED

This command returns all parameters that are assigned to the *General - General database parameters* group **or** the *Extended - Special database parameters* group.

Reply

```
OK<NL>
<parameter_name> <type> <value><NL>
<parameter_name> <type> <value><NL>
...
```

In the event of errors, see Reply Format [Page 193].

Displaying the Data for All Volume Parameters

Use

You display the data for all volumes [See SAP DB Library] of a database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

param_getvolsall [<volmode>]

Options

<volmode>	Type of the volume

Option <volmode>

DATA	Data Volume [See SAP DB Library]; the number of the data volume is

	assigned by the system as a four-digit number with leading zeros when creating the volume
LOG	Log Volume [See SAP DB Library]; the number of the log volume is assigned by the system as a three-digit number with leading zeros when creating the volume
MLOG	Mirrored log volume (Log Mode [See SAP DB Library] **DUAL**)

Displaying the data for all volumes of the type `DATA`

```
dbmcli -d DB -u dbm,dbm param_getvolsall DATA
```

```
OK
MAXDATAVOLUMES    3
DATAVOL_0001        10000       F   DAT_001
DATAVOL_0002        11000       F   DAT_002
```

Reply

The system displays a list the data for the volumes.

```
OK<NL>
<parameter_name> <value>
<parameter_name> <value>
...
<volume> <volsize> <voltype> <volname>
<volume> <volsize> <voltype> <volname>
...
```

Values for the Fields of the Reply

`<parameter_name>`	Name of a database parameter relevant for volumes
`<value>`	Value of the parameter
`<volume>`	Volume ID
`<volname>`	Name of volume/file of the volume
`<voltype>`	Type of volume (such as file, raw device)
`<volsize>`	Size of the volume

In the event of errors, see Reply Format [Page 193].

Displaying the Data of an Individual Volume Parameter

Use

You display the data of an individual volume [See SAP DB Library] parameter of the current database instance.

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_getvolume <volno> <volmode>
```

Options

<devno>	Number of the volume
<devmode>	Type of the volume

Option <devmode>

DATA	Data Volume [See SAP DB Library]; the number of the data volume is assigned by the system as a four-digit number with leading zeros when creating the volume
LOG	Log Volume [See SAP DB Library]; the number of the log volume is assigned by the system as a three-digit number with leading zeros when creating the volume
MLOG	Mirrored log volume (Log Mode [See SAP DB Library] **DUAL**)

```
dbmcli -d DB -u dbm,dbm param_getvolume 1 DATA

OK
DAT_001
F
10000
```

Reply

```
OK<NL>
<volname>
<voltype>
<volsize>
```

Values for the Fields of the Reply

<volname>	Name of volume/file of the volume
<voltype>	Type of volume (such as file, raw device)
<volsize>	Size of the volume

In the event of errors, see Reply Format [Page 193].

Displaying the Current Value of a Database Parameter

Use

You display the current value of the database parameter [See SAP DB Library] specified in **<parameter_name>**.

The system displays the value of the parameter as saved in the database parameter file the last time the database instance [See SAP DB Library] was started.

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_getvalue <parameter_name>
```

Reply

```
OK<NL>
<value>
```

In the event of errors, see Reply Format [Page 193].

Displaying the Data Type of a Database Parameter

Use

You display the current value of the parameter specified as **<parameter_name>**.

There is an OK message after the command has been executed successfully. This is followed by the data type <type> of the specified parameter. The following output values for <type> are possible:

```
int2 | int4 | c8 | c18 | c24 | c40 | c64
```

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_gettype <parameter_name>
```

Reply

```
OK<NL>
<type>
```

In the event of errors, see Reply Format [Page 193].

Displaying the Stored Explanatory Text

Use

You display the stored explanatory text for the database parameter [See SAP DB Library] **<parameter_name>**.

The system outputs an OK message followed by the explanatory text <explanation_line>. The line is empty, if no explanatory text is stored.

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_getexplain <parameter_name>
```

Reply

```
OK<NL>
<explanation_line><NL>
<explanation_line><NL>
...
```

In the event of errors, see Reply Format [Page 193].

Displaying the Stored Help Text

Use

You display the stored help text for the database parameter **<parameter_name>**.

The system outputs an OK message followed by the help text <helpline>. The line is empty, if no help text is stored.

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_gethelp <parameter_name>
```

Reply

```
OK<NL>
<helpline><NL>
<helpline><NL>
...
```

In the event of errors, see Reply Format [Page 193].

Displaying the System Default Value

Use

You display the system default value of the database parameter [See SAP DB Library] specified in **<parameter_name>**.

The system outputs an OK message and a line showing the system <default> value. The line is blank if the database parameter does not exist.

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

param_getdefault <parameter_name>

Reply

```
OK<NL>
<default>
```

In the event of errors, see Reply Format [Page 193].

Displaying a Value from the Database Parameter File

Use

The parameter file of the database instance [See SAP DB Library] is searched for the value of the database parameter [See SAP DB Library] specified in **<parameter_name>**.

After a successful search an OK message is displayed. The name of the database parameter `<parameter_name>` and its value `<value>` are displayed in the following line. The message output if the specified parameter is not in the parameter file is `not found`.

The **param_directget** command accesses the database instance parameter file directly. It does not require a parameter session and so **cannot** be rolled back (Rollback [See SAP DB Library]).

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

param_directget <parameter_name>

Reply

```
OK<NL>
<parameter_name> <value><NL>
```

In the event of errors, see Reply Format [Page 193].

Displaying Individual Data for a Database Parameter

Use

You display the name, data type, and current value of the database parameter [See SAP DB Library] specified in **<parameter_name>**. The description file and the parameter file of the current database instance [See SAP DB Library] are searched (see also: Configuring Database Instances [Page 85]).

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_extget <parameter_name>
```

Reply

```
OK<NL>
<parameter_name> <type> <value><NL>
```

<parameter_name>	Name of the database parameter
<type>	Data type of the database parameter
<value>	Current value This is the value that is copied from the parameter file at the start of the session. If the parameter is not in it, the default value from the description file is displayed here.

In the event of errors, see Reply Format [Page 193].

Confirming Changes to the Database Parameter File

Use

You confirm all changes to the database parameters [See SAP DB Library] and transfer these to the parameter file of the database instance [See SAP DB Library]. Before storing them the check status of all parameters is verified.

If you specify the **NOCLOSE** option, the parameter session remains open after the database parameters have been stored. If you do not specify this option, the parameter session is closed after the database parameters have been stored.

The changed values in the parameter file are only effective after a restart [See SAP DB Library] of the database instance.

Prerequisites

You have changed parameter values in a parameter session [Page 103].

You have the DBM operator authorization ParamCheckWrite [Page 19].

Syntax

```
param_commitsession [NOCLOSE]
```

Successful Reply

The system outputs an OK message.

Error Message

Execution of the command is refused:

```
ERR
14,ERR_XPCHECK_CN00 : param check failure/request
<identifier>      <checkstatus>
```

```
<user_value>
<computed_value>
```

Values for the individual fields of the reply

<checkstatus>	The following check status may occur: Mandatory: An obligatory database parameter has been assigned an empty value. Constraint: An condition defined in the description file could not be fulfilled (Configuring Database Instances [Page 85]). Request: The user input deviates both from the value transferred to the system and confirmed by it earlier, as well as from the system default value.
<user_value>	Value that the operator entered with **param_put**.
<computed_value>	Default value calculated by the system

Check status request is displayed only if the user entry varies from the value of the database parameter in the parameter file.

Continuation of Procedure

If the check status is a request, you can correct the error as follows:

Specify the valid value with **param_putconfirm** (Correcting Database Parameters [Page 106]). Otherwise the system default <computed_value> is used this for parameter for all further calculations of other parameters.

In the following cases, the parameter change is refused:

- The parameter has the value NOBODY for the CHANGE property [Page 85]
- The parameter has the value NO for the property MODIFY and the parameter file was already checked by the kernel (parameter __PARAM_CHANGED___ exists)
- You are dealing with a volume parameter (VOLUME property = YES) and there are changed kernel parameters in the parameter file (VOLUME property= NO) that have not yet been checked by the kernel
  (Exception: You are creating a new database instance)
- The parameter is a kernel parameter and volume parameters have already been changed, but not yet checked by the kernel.

Scrolling in the Database Parameter History

Use

You call up the rest of the contents of the database parameter history.

If you receive a reply of only:

```
OK
<header_line><NL>
```

then all specified entries have already been displayed.

If you have set selection criteria in the options on opening the database parameter history, then these also apply while you are scrolling through the parameter history.

Prerequisites

You have opened the database parameter history (Opening the Database Parameter History [Page 109]).

You have the DBM operator authorization ParamRead [Page 20].

Syntax

param_gethistorynext

Reply

```
OK
<header_line><NL>
<param_line><NL>
<param_line><NL>
...
```

Values for the individual fields of the reply

<header_line>	Contains the names of the fields displayed S stands for STATE and describes the database parameter's change state
<param_line>	Contains the values for the fields described in the <header_line>, the values are in the same column position as the field description in the <header_line> Field S for STATE can contain any of the following entries: c: changed d: deleted a: active

Changing a Value Directly in the Database Parameter File

Use

With this command you change the parameters of the database instance [See SAP DB Library]. Unlike the function changing the value of a database parameter [Page 88], the value **<value>** for the prameter identified as **<parameter_name>** is written straight to the parameter file. An entry is only made if the parameter is already contained in the parameter file.

In the following cases the parameter change is refused:

- The parameter has the value NOBODY for the CHANGE property [Page 85]

- The parameter has the value NO for the property MODIFY and the parameter file was already checked by the kernel (parameter __PARAM_CHANGED___ exists)
- You are dealing with a volume [See SAP DB Library] parameter (property VOLUME = YES) and there are changed kernel parameters in the parameter file (property VOLUME = NO) that have not yet been checked by the kernel
  (Exception: You are creating a new database instance)
- The parameter is a kernel parameter and volume parameters have already been changed, but not yet checked by the kernel.

The entered value for the parameter is not checked for consistency. Therefore unqualified changes to parameters are also possible using the command **param_directput**. These can then **not** be rejected by param_abortsession [Page 88] because the changes are written directly to the parameter file.

Prerequisites

You have the DBM operator authorization ParamFull [Page 19].

Syntax

```
param_directput <parameter_name> <value>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Opening a Parameter Session

Use

You open a parameter session to check or change the parameters of a database instance [See SAP DB Library].

The DBM Server [See SAP DB Library] loads the description file and parameter file of the database instance (see also: Configuring Database Instances [Page 85]) and stores the contents in internal data structures. All access operations to this internal data take place within a parameter session.

In case of syntax error an error message is displayed which contains the file name, line number and line text.

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_startsession
```

Reply

The system outputs an OK message.

In case of syntax error an error message is displayed which contains the file name, line number and line text.

Adding a Volume

Use

You define additional data volumes [See SAP DB Library] or log volumes [See SAP DB Library] to extend the disk space for the database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization ParamFull [Page 19] or AccessUtility [Page 21].

Syntax

```
db_addvolume <volmode> <volname> <voltype> <size>
```

<volmode>	Type of volume that is to be added. Possible values are: **DATA** (Data volume) **LOG** (Log volume)
<volname>	Name of device/file
<voltype>	Type of device or fill that is to be added. Possible values are: **F** (File) **L** (Link) **R** (Raw Device)
<size>	Size of the volume to be added in pages [See SAP DB Library]

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Initializing the Database Parameters for a New Database Instance

Use

You initialize all of the database parameters needed to generate a new database instance [See SAP DB Library] (Database Parameters [See SAP DB Library]).

This procedure uses the name of the database instance [Page 31] from the session with the Database Manager. There must not be database parameter file for this database instance.

All database parameters are first assigned their default values from the description file (see also: Configurating Database Instances [Page 85]). Then a complete calculation run is carried out. The values from the calculation run overwrite the default values.

The new parameter values in the database parameter file are checked for correct syntax. The system does not display a message about conditions that are not fulfilled or missing obligatory values.

The initialization is performed only within the Database Manager. The parameters are saved to the database parameter file only when you confirm changes to the database parameter file [Page 100].

Prerequisites

You have opened a parameter session [Page 103].

You have the DBM operator authorization InstallMgm [Page 16].

Syntax

```
param_init
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Copying a Database Parameter File

Use

You copy the parameter file of an existing database instance [See SAP DB Library] as the initial configuration for a new database instance. You can only copy a source parameter file if the new database instance does not yet have one.

All parameter changes made during the current parameter session up to this point are overwritten by the copied parameter file. Parameters that cannot be changed because of their properties [Page 85] retain their system default values copied from the description file. This applies to the following parameters:

- **RUNDIRECTORY** contains a directory name **<independent_data_path>/wrk/<database_name>**
- **KERNELVERSION** is initialized with the version number stored in the DBM Server

Specify the name of the source database in **<source_db>**.

(**See also:** Configuring Database Instances [Page 85])

Prerequisites

You have opened a parameter session [Page 103].

You have the DBM operator authorization InstallMgm [Page 16].

Syntax

```
param_copy <source_db>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Correcting Parameters

Use

You correct the value of a parameter. In doing so, you have the choice of assigning either the user input **<user_value>** or the system default value **<computed_value>** to the parameter **<parameter_name>** registered with `request`.

The value is stored in the parameter manager and becomes effective after a Restart of the Database Instance [Page 66]. Parameters that you correct with this command are only queried if both the current user input and the system default value do not match the confirmed value.

You can only use this command for SAP DB database systems up to version 7.3.

Prerequisites

You have the DBM operator authorization ParamCheckWrite [Page 19].

You have opened a parameter session [Page 103] and checked parameters.

The check status `Request` for a parameter was indicated.

Syntax

```
param_putconfirm <parameter_name> <user_value>|<computed_value>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

List of the Available Parameter Files

Use

You request a list of all available parameter files.

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_versions
```

Reply

```
OK<NL>
<file><NL>
<file><NL>
...
```

`<file>`	Name of the parameter file

In the event of errors, see Reply Format [Page 193].

Deleting the Database Parameter File

Use

You can delete the parameter file.

Use this command with great care.

With the command **param_restore** (Resetting the Parameter File to a Previous Version [Page 112]), you can restore the version of the parameter file that was valid before the deletion.

Prerequisites

You have the DBM operator authorization InstallMgm [Page 16].

Syntax

```
param_rmfile
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Deleting a Database Parameter

Use

You delete the parameter specified as **<parameter_name>** from the database parameter file.

The changed values are transferred to the parameter file and are effective after the next restart [See SAP DB Library] of the database instance.

Use this command with great care.

The **param_directdel** command accesses the database instance parameter file directly. It does not require a parameter session and so **cannot** be rolled back (Rollback [See SAP DB Library]).

Prerequisites

You have the DBM operator authorization ParamFull [Page 19].

Syntax

```
param_directdel <parameter_name>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Deleting Volume Parameters

Use

You delete the data for the specified volume [See SAP DB Library] from the database parameter file.

Use this command with great care.

Inconsistencies in the volume settings can lead to damage to the database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization ParamCheckWrite [Page 19].

Syntax

```
param_delvolume <volno> <volmode>
```

Options

<volno>	Number of the volume
<volmode>	Type of the volume

Option <volmode>

	Data Volume [See SAP DB Library]; the number of the data volume is assigned by the system as a four-digit number with leading zeros when creating the volume
LOG	Log Volume [See SAP DB Library]; the number of the log volume is assigned by the system as a three-digit number with leading zeros when creating the volume
MLOG	Mirrored log volume (Log Mode [See SAP DB Library] `DUAL`)

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Opening the Database Parameter History

Use

You open the database parameter history [See SAP DB Library] and transfer the first block of the contents. The entries are output in reverse chronological sequence, that is, starting with the most recent and ending with the oldest.

You can limit the output by specifying options. If you enter several selection criteria, then only those parameter changes that correspond to all the criteria are output.

If the parameter history entries have been incompletely transferred, you can call up the rest of the contents using the command for scrolling through the database parameter history [Page 101].

Prerequisites

You have the DBM operator authorization ParamRead [Page 20].

Syntax

```
param_gethistory [fields=<field_list>] [name=<parameter_name>]
[date=<yyyymmdd>] [group=<group_list>] [state=<state_list>]
```

Example [Page 110]

Options

<field_list>	You can restrict the quantity of history fields supplied by the Database Manager by specifying a field list. Enter the fields that you want output here using a comma as a separator. The fields you select will always be output in the following sequence: Possible entries: **DATE** Date when the parameter was changed (**YYYYMMDD**) **TIME** Time when the parameter was changed (**00HHMMSS**) **NAME** Parameter name **NEWVALUE** New value of the parameter **OLDVALUE** Old value of the parameter **STATE** The change status of the parameter **GROUP** Group to which the parameter is assigned If you do not specify a field list, then all fields will be output.
<parameter_name>	You can display the history of one specific parameter by specifying the parameter name. If you do not specify a parameter name, the entries for all parameters in the history are displayed.
<yyyymmdd>	If you specify a date, then only those parameter changes that have taken place since this date will be output. If you do not specify a date, all entries in the history are supplied.
<group_list>	You can restrict the database parameters for which the changes are displayed by specifying a parameter group. Use a comma to separate entries. Possible specifications are: **GENERAL:** General Database Parameters [See SAP DB Library] **EXTENDED:** Special Database Parameters [See SAP DB Library] **SUPPORT:** Support Database Parameters [See SAP DB Library] If you do not specify a parameter group, then the parameter changes for all groups will be output.

<state_list>	You can specify the change state to determine whether active, changed or deleted database parameters should be output. Use a comma to separate entries. Possible entries are: **A** (active), **C** (changed), **D** (deleted) if you do not specify a state, then all parameter changes will be output.

Reply

```
OK
<header_line><NL>
<param_line><NL>
<param_line><NL>
...
```

Values for the individual fields of the reply

<header_line>	Contains the names of the fields displayed S stands for STATE and describes the parameter's status
<param_line>	Contains the values for the fields described in the <header_line>, the values are in the same column position as the field description in the <header_line> Field S for STATE (change status of the database parameters) can contain any of the following entries: c: changed d: deleted a: active

param_gethistory Command: Example

Opening the parameter history for the parameter _MAXDATAVOLUMES

```
dbmcli -d myDB -u dbm,dbmp param_gethistory name=_MAXDATAVOLUMES
```

```
OK
DATE     TIME     NAME             NEWVALUE     OLDVALUE        S GROUP
20010613 00152700 _MAXDATAVOLUMES 5            4               A
SUPPORT
20010613 00152700 _MAXDATAVOLUMES 4            3               C
SUPPORT
20010613 00152659 _MAXDATAVOLUMES 3                            C
SUPPORT
```

Checking All Database Parameters

Use

You check the entirety of all database parameters using the properties, calculation formulas, and conditions stored in the description file (Configuring Database Instances [Page 85]).

Prerequisites

You have created a parameter session [Page 103].

You have the DBM operator authorization ParamCheckWrite [Page 19].

Syntax

param_checkall [<mode>]

Option <mode>

Using the parameter **<mode>**, you can control the behavior of the Database Manager for parameters with the check status `request`.

Possible values of **<mode>**:

No entry	The system default values are used for all parameters (necessary for background processing).
GENERAL	Only the parameters from the group *General - General Database Parameters* are requested during the check status `request`.
EXTENDED	Only the parameters from the groups *General* – General Database Parameters [See SAP DB Library] and *Extended* - Special Database Parameters [See SAP DB Library] are checked.
SUPPORT	This option is only relevant for SAP DB Support [See SAP DB Library]. All parameters (*General*, *Extended*, and *Support* groups) are checked (Support Database Parameters [See SAP DB Library]).

Reply

If an error occurs the database parameter check is canceled with an error message.

```
ERR
14,ERR_XPCHECK_CN00 : param check failure/request
<parameter_name>      <check_status>
<user_value>
<computed_value>
```

`<parameter_name>`	Name of database parameter
`<check_status>`	The following are possible values for the check status: `Mandatory` – An obligatory parameter has been assigned a blank value. `Constraint` – a constraint could not be fulfilled. `Request` – the operator input deviates both from the value transferred to the system and confirmed by it earlier, as well as from the system default value.
`<user_value>`	Value that the operator entered with **param_put**.
`<computed_value>`	Default value calculated by the system

For open `requests` the parameter changes cannot be confirmed (Confirming Changes to the Database Parameter File [Page 100]).

Continuation of Procedure

Correct the parameter in question using **`param_putconfirm`** (parameter correction [Page 106]) or **`param_put`** (change a parameter value [Page 88])and carry out a new check with the command **`param_checkall [<mode>]`**.

The new parameter value is registered by the system, and is transferred to the parameter file as the valid value at the next restart [See SAP DB Library] of the database instance.

Resetting the Parameter File to a Previous Version

Use

The parameter file is reset to the version specified by **`<version_number>`**.

This parameter file will be used when the database instance [See SAP DB Library] is next started.

Prerequisites

You have the DBM operator authorization ParamCheckWrite [Page 19].

Syntax

```
param_restore <version_number>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

List of the DBM Commands

Use

You request the list of all DBM commands [Page 35]. In addition, you receive a short syntax description for all commands.

The list of displayed commands can be restricted via the parameter **`<command>`**. Only those commands are displayed which start with the specified character string.

Prerequisites

You do not need to log on to the Database Manager to execute this DBM command [Page 35].

Syntax

```
help [-obsolete] <command>
```

```
d:\v73>dbmcli help autosave

OK
autosave_cancel          (obsolete version of
autolog_cancel)
autosave_off             (obsolete version of autolog_off)
autosave_on              (obsolete version of autolog_on)
autosave_show            (obsolete version of autolog_show)
```

Option

Specify the option **-obsolete**, to display exclusively the obsolete commands. The system displays a reference to the current name for each of these commands.

```
d:\v73>dbmcli help -obsolete

OK
autosave_cancel          (obsolete version of
autolog_cancel)
autosave_off             (obsolete version of autolog_off)
autosave_on              (obsolete version of autolog_on)
autosave_show            (obsolete version of autolog_show)
backup_save_cancel       (obsolete version of autolog_on)
backup_save_ignore       (obsolete version of backup_ignore)
.....
```

Reply

```
OK<NL>
<command_name> <description><NL>
<command_name> <description><NL>
...
```

In the event of errors, see Reply Format [Page 193].

Backing Up and Recovering Database Instances

Use

You can use the Database Manager CLI to carry out complete data backups [See SAP DB Library], incremental data backups [See SAP DB Library], and log backups [See SAP DB Library] and to import these backups if hardware errors occur, thereby restoring the database to a consistent state.

Prerequisites

Note the operator authorizations [Page 13] required for the particular DBM commands [Page 35].

Commands for Backup Media [Page 114]

Commands for Backups [Page 120]

Commands for the Backup History [Page 136]

Commands for External Backup IDs [Page 141]

Commands for Backup Information [Page 144]

Commands for Restoring [Page 152]

Commands for Backup Media

Use

A backup medium [See SAP DB Library] is assigned to every backup action. Backup media can be files, tapes, and pipes.

The path specifications and properties of the tape device, tapes, or files are grouped together in the media definition under a freely selectable, practical name. Under these names, the backup media can be re-used for all possible backup operations.

Prerequisites

Note the operator authorizations [Page 13] required for the particular commands DBM commands [Page 35].

Syntax

Change Date of the Media Definition File [Page 114]	**medium_date**
Defining or Changing a Backup Medium [Page 115]	**medium_put**
Displaying the Media Data [Page 117]	**medium_get**
List of All Defined Backup Media [Page 118]	**medium_getall**
Deleting a Backup Medium [Page 119]	**medium_delete**
Transferring an Existing Media Definition [Page 120]	**medium_migrate**

Change Date of the Media Definition File

Use

You request the date on which the current media definition file was last changed (Database Manager Files [Page 196]).

Prerequisites

You have the DBM operator authorizations Backup [Page 15] and DBInfoRead [Page 14].

Syntax

medium_date

Reply

```
OK<NL>
<YYYYMMDDHHMMSS><NL>
```

In the event of errors, see Reply Format [Page 193].

Defining or Changing a Backup Medium

Use

You create a new backup medium [See SAP DB Library] or change the data for a backup medium that already exists. To do this, you must make the following specifications:

<name>	The backup medium specified in **<name>** is created or updated. The name of the medium may comprise a group name and a member name. These are separated by an oblique. **<name> ::= [<group_name>/]<member_name>** Groups of Parallel Backup Media [See SAP DB Library] of this type are used for parallel backups [See SAP DB Library] and restores. If you want to perform a backup or a restore using one of the external backup tools [See SAP DB Library] ADSM/TSM, NetWorker, Backint for Oracle, or Backint for SAP DB, ensure the name of the medium starts with ADSM, NSR, BACK, or BACK. In this case, you can only enter **PIPE** under **<type>** (Backing Up with External Backup Tools [Page 135]).
<location>	The name of the device/file with which it can be addressed through operating system functions.
<type>	Type of the backup medium. Possible values are: **AUTO** **TAPE** **FILE** **NOREWIND** **PIPE** **UNKNOWN** The media type **AUTO** is the default value for the command to Activate the Automatic Log Backup [Page 129]. There can only be one backup medium of this type. If there is already a backup medium of type **AUTO**, and you create a backup medium of type **AUTO** again, the system changes the media type of the existing backup medium to **LOG**.
<backup_type>	The type of backup for which the medium is to be used: **DATA** (complete data backup [See SAP DB Library]), **PAGES** (incremental data backup), or **LOG** (log backup [See SAP DB Library])
<size>	Maximum number of pages [See SAP DB Library] that can be written to the backup medium. This is necessary to change the tape correctly, for example. If the backup media is of sufficient size, a zero can be specified.
<block_size>	Number of pages that are transferred when accessing the medium (default: 8).
<overwrite>	This option is only relevant for the medium type **FILE** and describes the behavior if the file is already present. Permitted values are: **NO** **YES**
<autoldr>	Specifies whether the device changes the medium automatically: **YES** **NO**

<oscmd>	Operating system command to be executed before backing up to a succeeding medium [Page 130] or restoring a succeeding medium [Page 164].

Prerequisites

You have the DBM operator authorization Backup [Page 15].

Syntax

```
medium_put <name> <location> <type> <backup_type> <size> <block_size>
<overwrite> <autoldr> <oscmd>
```

Examples: Creating Backup Media [Page 116]

Reply

```
OK
```

In the event of errors, see Reply Format [Page 193].

medium_put Command: Examples

Defining a File as a backup medium [See SAP DB Library] for an interactive log backup [See SAP DB Library]:

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

Defining a file as a backup medium for an automatic log backup [See SAP DB Library]:

```
dbmcli -d <database_instance> -u <userid>,<password> medium_put
autosave /usr/sapdb/auto FILE AUTO
```

Defining a file as an overwritable backup medium for a complete data backup [See SAP DB Library]:

```
dbmcli -d <database_instance> -u <userid>,<password> medium_put
completeF /usr/sapdb/complete FILE DATA 0 0 YES
```

Defining a tape device as a backup medium for a complete data backup

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

Defining a File as a backup medium for an incremental data backup:

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

Displaying the Media Data

Use

You display the media data for the backup medium [See SAP DB Library] specified in **<name>**.

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

Syntax

medium_get <name>

Reply

```
OK<NL>
<name> <location> <type> <backup_type> <size> <block_size>
<overwrite> <autoldr> <oscmd> <date_created> <date_modified> <NL>
```

The fields of the one-line output are separated by tabs.

Values for the individual fields of the reply

<name>	Name to identify the backup medium [See SAP DB Library]. The name of the backup medium may comprise a group name and a member name. These are separated by a slash. <name> ::= [<group_name>/]<member_name> This type of group is used with parallel backup or restore operations (Parallel Backup [See SAP DB Library]).
<location>	The name of the device/file with which it can be addressed through operating system functions.
<type>	Type of the backup medium; possible values are: TAPE FILE NO-REWIND PIPE AUTOLOADER UNKNOWN
<backup_type>	Type of backup for which the backup medium is to be used: Possible values are DATA for a complete data backup [See SAP DB Library] PAGES for an incremental data backup [See SAP DB Library] LOG for a log backup [See SAP DB Library]
<size>	Maximum number of pages which can be written to the backup medium. This is necessary to change the tape correctly, for example. If the media are of sufficient size, a zero can be specified.
<block_size>	Number of pages [See SAP DB Library] that are transferred when accessing the medium (default: 8).
<overwrite>	This option is only relevant for the medium type FILE and describes the behavior if the file is already present. Permitted values are: NO YES VERSION

<autoldr>	Specifies whether the device changes the medium automatically: YES NO
<oscmd>	Operating system command to be executed before backing up to a succeeding medium [Page 130] or restoring a succeeding medium [Page 164].
<date_created>	Date on which the backup medium was created
<date_modified>	Date on which the backup medium was changed

In the event of errors, see Reply Format [Page 193].

List of All Defined Backup Media

Use

You can call up a list of all defined backup media [See SAP DB Library].

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

Syntax

medium_getall

Reply

```
OK<NL>
<name> <location> <type> <backup_type> <size> <block_size>
<overwrite> <autoldr> <oscmd> <date_created> <date_modified> <NL>

<name> <location> <type> <backup_type> <size> <block_size>
<overwrite> <autoldr> <oscmd> <date_created> <date_modified> <NL>

...
```

Values for the individual fields of the reply

<name>	Name to identify the backup medium [See SAP DB Library]. The name of the medium may comprise a group name and a member name. These are separated by an oblique. <name> ::= [<grpname>/]<membername> This type of group is used with parallel backup operations [See SAP DB Library] and parallel restore operations.
<location>	The name of the device/file Used for operating system functions.
<type>	Type of the backup medium Possible values are: TAPE FILE NO-REWIND PIPE AUTOLOADER UNKNOWN

<backup_type>	The type of backup for which the medium is to be used: DATA (complete data backup [See SAP DB Library]) PAGES (incremental data backup [See SAP DB Library]) LOG (log backup [See SAP DB Library])
<size>	Maximum number of pages [See SAP DB Library] which can be written to the medium. This is necessary to change the tape correctly, for example. If the backup media is of sufficient size, a zero can be specified.
<block_size>	Number of pages that are transferred when accessing the medium (default value: 8).
<overwrite>	This option is only relevant for the media type FILE and describes the behavior if the file is already present. Permissible values are: NO YES VERSION
<autoldr>	Specifies whether the device changes the medium automatically: YES NO
<oscmd>	Operating system command to be executed before backing up to a succeeding medium [Page 130] or restoring a succeeding medium [Page 164].

In the event of errors, see Reply Format [Page 193].

Deleting a Backup Medium

Use

You delete the backup medium [See SAP DB Library] specified as **<name>**.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

Syntax

```
medium_delete <name>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Transferring an Existing Media Definition

Use

You can transfer existing definitions for backup media [See SAP DB Library] from the media definition file `control.med` (database version 6.2) to the new media definition file `dbm.mmm` (as of database version 7.2.4.2).

(See also: Database Files [Page 196])

The media definition from the old file is not transferred if there is an identical media name in the old and new media definition files.

As the old definitions for backup media do not contain a backup type (incremental data backup, complete data backup, or log backup), you should define this afterwards using the command **`medium_put`** (Defining or Changing a Backup Medium [Page 115]). Otherwise you must specify this when starting the backup process.

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

Syntax

`medium_migrate`

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Commands for Backing Up and Archiving

Use

With the Database Manager, you can perform complete data backups [See SAP DB Library], incremental data backups [See SAP DB Library], and log backups [See SAP DB Library].

For log backups, we recommend that you enable automatic log backup [See SAP DB Library]. The option of interactive log backups [See SAP DB Library] is also available to you, however.

Prerequisites

Note the operator authorizations [Page 13] required for the particular commands DBM commands [Page 35].

The prerequisite for executing all other backup commands is to execute the command to backup the database instance [Page 132] first.

Syntax

Canceling the Automatic Log Backup [Page 121]	**`autolog_cancel`**
Displaying the Automatic Log Backup Function [Page 121]	**`autolog_show`**

Displaying the Current Backup Status [Page 122]	`backup_state`
Deactivating the Automatic Log Backup [Page 128]	`autolog_off`
Terminating an Interrupted Backup [Page 128]	`backup_cancel`
Creating a Single Backup of the Version Files [Page 124]	`archive_stage`
Activating the Automatic Log Backup [Page 129]	`autolog_on`
Continuing the Backup Without the Medium Last Reported as Full [Page 129]	`backup_ignore`
Creating Multiple Archives of the Version Files [Page 126]	`archive_stage_repeat`
Backing Up to a Succeeding Medium [Page 130]	`backup_replace`
Backing Up the Database Instance [Page 132]	`backup_start`
Backing Up with External Backup Tools [Page 135]	`---`

Canceling the Automatic Log Backup

Use

You can cancel an automatic log backup [See SAP DB Library] while it is running. This deactivates the *automatic log backup* function. Use this command to prevent technical problems with the backup medium [See SAP DB Library], for example.

If you want the log to be backed up automatically again later, then you have to explicitly enable the log backup again (Activating the Automatic Log Backup [Page 129]).

Prerequisites

You have the DBM operator authorization Backup [Page 15].

Syntax

```
autolog_cancel
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Displaying the Automatic Log Backup Function

Use

You determine whether the automatic log backup [See SAP DB Library] function is enabled or disabled.

Prerequisites

You have the DBM operator authorization DBFileRead [Page 18] or DBInfoRead. [Page 14].

Syntax

autolog_show

```
dbmcli -d <database_name> -u <userid>,<password>
autolog_show
```

```
OK<NL>
AUTOLOG IS OFF <NL>
```

Reply

```
OK<NL>
AUTOLOG IS (OFF | ON)<NL>
```

AUTOLOG IS ON	Automatic log backup is enabled
AUTOLOG IS OFF	Automatic log backup is disabled

In the event of errors, see Reply Format [Page 193].

Displaying the Current Backup Status

Use

You display the current status of the backup operation.

You can use this request during a backup of the database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have started a backup (Backing Up the Database Instance [Page 132]).

Syntax

backup_state

Reply

```
OK<NL>
Returncode              <value><NL>
Date                    [<value>]<NL>
Time                    [<value>]<NL>
Server                  [<value>]<NL>
Database                [<value>]<NL>
Kernel Version          [<value>]<NL>
Pages Transferred       [<value>]<NL>
Pages Left              [<value>]<NL>
Volume Count            [<value>]<NL>
Medianame               [<value>]<NL>
Location                [<value>]<NL>
Errortext               [<value>]<NL>
Label                   [<value>]<NL>
```

```
Is Consistent           [<value>]<NL>
First LOG Page          [<value>]<NL>
Last LOG Page           [<value>]<NL>
DB Stamp 1 Date         [<value>]<NL>
DB Stamp 1 Time         [<value>]<NL>
DB Stamp 2 Date         [<value>]<NL>
DB Stamp 2 Time         [<value>]<NL>
Page Count              [<value>]<NL>
Devices Used            [<value>]<NL>
Database ID             [<value>]<NL>
Max Used Data Page      [<value>]<NL>
```

Values for the individual fields of the reply

`Date`	Date
`Time`	Time
`Server`	Name of the database server
`Database`	Name of database instance
`Kernel Version`	Version of the database software
`Pages Transferred`	Number of pages transferred
`Pages Left`	Number of pages still to be transferred
`Volumes`	Number of backup media [See SAP DB Library] used
`Medianame`	Name of backup medium
`Location`	File or device name
`Errortext`	Error message text
`Label`	Backup Id [See SAP DB Library]
`Is Consistent`	For data backup [See SAP DB Library] only: backup is internally consistent
`First LOG Page`	For data backup: first page of log backup [See SAP DB Library] to be read For log backup [See SAP DB Library]: first page saved in the log
`Last LOG Page`	For log backup only: last page saved in log
`DB Stamp 1 Date` `DB Stamp 1 Time`	Time stamp for first page of log backup
`DB Stamp 2 Date` `DB Stamp 2 Time`	Time stamp for last page of log backup
`Page Count`	Total number of pages saved
`Devices Used`	Number of backup devices used
`Database ID`	Database ID used to identify data and log backups that belong together
`Max Used Data Page`	Maximum number of pages used (indication of minimum database size when backup is imported)

In the event of errors, see Reply Format [Page 193].

Archiving the Version Files of the Log Area

Use

To increase the security of your database system, we recommend that you archive the version files created during Log Backup [See SAP DB Library] and stored on the hard disk of your system at regular intervals (Version Files [See SAP DB Library]). To do this, copy the version files to another backup medium. (See also: Backup Strategy [See SAP DB Library])

You can perform this process manually or use external backup tools [See SAP DB Library]. For information about which external backup tools are supported by the Database Manager CLI, see the Backing Up with External Backup Tools [Page 135] section.

There are DBM commands available for archiving with external backup tools, with which you can copy the version files once or several times to external backup media.

Archiving the Version Files Once [Page 124]

Archiving the Version Files More Than Once [Page 126]

Creating a Single Archive of the Version Files

Use

You archive the original version files [See SAP DB Library] for the log area. To do this, you copy the version files using an external backup tool [See SAP DB Library] to an external backup medium [See SAP DB Library].

If you want to backup the same version files more than once to different external backup media, first use the `archive_stage` command described here and then the command to Archive the Version Files of the Log Area More Than Once [Page 126].

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have backed up the log (Log Backup [See SAP DB Library]).

The external backup medium is defined for the backup type `LOG` (Defining or Changing a Backup Medium [Page 115]).

Syntax

```
archive_stage <ext_backup_medium> <log_medium> [VERIFY | NOVERIFY]
[REMOVE | KEEP] [FNL <list>]
```

<ext_backup_medium>	Name of the external backup medium
<log_medium>	Name of the backup medium [See SAP DB Library] that was used for the log backup

VERIFY	System default value The archived files are compared with the original files.
NOVERIFY	Only the usual checks as to whether the backup was successful are performed by the system. The system displays error messages and the output of the backup tools.
REMOVE	System default value The original files are deleted after they have been successfully archived to the external backup medium.
KEEP	The original files are archived, but not then deleted.
FNL	File Number List: Keyword of the list of files to be archived
<list>	File name or area or file names, you can leave out leading zeros. Separate multiple specifications using commas. If you do not specify a file list, all version files on the backup medium that are also in the backup history [See SAP DB Library] are copied.

In most cases, it is not necessary to specify a file number list, as all completed version files are archived with the command `archive_stage`. The system obtains the information about which files are complete from the backup history. Only specify the keyword `FNL` and the file number list if version files that you are sure were created are not in the backup history due to an error. This would be the case, for example, if the version files with the number 10 and 12 are in the backup history, but number 11 is missing.

Archiving the originals of the version files `1` and `3-10`
These are on the backup medium `LOGMEDIUM` and are archived to the external backup medium `EXTMEDIUM`. Only the usual system checks are to be performed. The version files should not then be deleted.

```
archive_stage EXTMEDIUM LOGMEDIUM NOVERIFY KEEP FNL 1,3-10
```

Reply:

```
OK
Processed Stage Files    9
Bytes Processed       12845056
Deleted Stage Files      0
Remaining Stage Files    2
```

Result

The system displays an OK message and information about the archived version files.

The archived version files are assigned their own external backup IDs [See SAP DB Library] and are entered in the backup history [See SAP DB Library].

`Processed Stage Files`	Processed version files

`Bytes Processed`	Processed bytes
`Deleted Stage Files`	Deleted version files
`Remaining Stage Files`	Remaining version files

Creating Multiple Archives of the Version Files

Use

You archive the original version files [See SAP DB Library] of the log area more than once on to different external backup media [See SAP DB Library]. You can also use different external backup tools [See SAP DB Library] to do this.

Exactly the same version files are processed as were copied to the first external backup medium using the command to archive the version files once [Page 124], which must be executed once.

The command to archive version files more than once can only be executed within the same session with the DBM Server as the command to archive the version files once, which you must execute first (Opening a DBM Server Session [Page 21]).

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have first executed the command to archive the version files once [Page 124], specifying the `KEEP` option.

The external backup media are defined for the backup type `LOG` (Defining or Changing a Backup Medium [Page 115]).

Syntax

`archive_stage_repeat <ext_backup_medium> [VERIFY | NOVERIFY] [REMOVE | KEEP]`

`<ext_backup_medium>`	Name of the external backup medium
`VERIFY`	System default value The archived files are compared with the original files.
`NOVERIFY`	Only the usual checks as to whether the backup was successful are performed by the system. If appropriate, the system displays error messages and the output of the backup tools.
`REMOVE`	System default value The original files are deleted after they have been successfully archived to the external backup medium.

KEEP	The original files are archived, but not then deleted.

Multiple archiving of the original version files `1` and `3-10`
These are on the backup medium `LOGMEDIUM`, and are first archived to the external backup medium `EXTMEDIUM_1` and then to the medium `EXTMEDIUM_2`. Only the usual system checks are to be performed. The version files are not to be deleted.

archive_stage [Page 124] **EXTMEDIUM_1 LOGMEDIUM NOVERIFY KEEP FNL 1,3-10**

Reply:

```
OK
Processed Stage Files    9
Bytes Processed       12845056
Deleted Stage Files      0
Remaining Stage Files    2
```

archive_stage_repeat EXTMEDIUM_2

Reply:

```
OK
Processed Stage Files    9
Bytes Processed       12845056
Deleted Stage Files      9
Remaining Stage Files    0
```

Result

The system displays an OK message and information about the archived version files.

The archived version files are assigned their own external backup IDs [See SAP DB Library] and are entered in the backup history [See SAP DB Library].

`Processed Stage Files`	Processed version files
`Bytes Processed`	Processed bytes
`Deleted Stage Files`	Deleted version files
`Remaining Stage Files`	Remaining version files

Deactivating the Automatic Log Backup

Use

You deactivate the automatic log backup [See SAP DB Library].

Prerequisites

You have the DBM operator authorization Backup [Page 15].

Syntax

autolog_off

```
dbmcli -d <database_name> -u <userid>,<password>
autolog_off
```

```
OK
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Terminating an Interrupted Backup

Use

You can exit an interrupted backup process completely.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have begun backing up the database instance [Page 132]. The backup operation was interrupted.

Syntax

backup_cancel

Reply

In the reply to this command only the field `Return code` is assigned a value.

```
OK<NL>
Return code                <value>
```

In the event of errors, see Reply Format [Page 193].

Activating the Automatic Log Backup

Use

You enable automatic log backup [See SAP DB Library]. For **<medium>**, specify the backup medium [See SAP DB Library] on which the log is to be automatically backed up.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

Syntax

autolog_on <medium>

```
dbmcli -d <database_name> -u <dbm_user>,<password>
autolog_on
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Continuing the Backup Without the Medium Last Reported as Full

Use

Use this command to continue a backup [See SAP DB Library] without the last medium that was reported full.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have started backing up the database instance [Page 132]. The backup operation was interrupted because the backup medium [See SAP DB Library] was reported full.

Syntax

backup_ignore

Reply

The reply to this command provides information about the backup supplied by the kernel.

This information is only given when the backup has been ended or interrupted. This command may therefore take a long time to execute.

```
OK<NL>
Returncode              <value><NL>
Date                    [<value>]<NL>
Time                    [<value>]<NL>
Server                  [<value>]<NL>
Database                [<value>]<NL>
Kernel Version          [<value>]<NL>
Pages Transferred       [<value>]<NL>
```

```
Pages Left                [<value>]<NL>
Volume Count              [<value>]<NL>
Medianame                 [<value>]<NL>
Location                  [<value>]<NL>
Errortext                 [<value>]<NL>
Label                     [<value>]<NL>
Is Consistent             [<value>]<NL>
First LOG Page            [<value>]<NL>
Last LOG Page             [<value>]<NL>
DB Stamp 1 Date           [<value>]<NL>
DB Stamp 1 Time           [<value>]<NL>
DB Stamp 2 Date           [<value>]<NL>
DB Stamp 2 Time           [<value>]<NL>
Page Count                [<value>]<NL>
Devices Used              [<value>]<NL>
```

In particular, analyze the `Returncode` reply field, which contains a numeric value supplied by the kernel.

See also: *Messages: SAP DB 7.4*

If it is clear from the message number that the backup was only interrupted, the session must be continued with the continuation command: (backup_replace [Page 130], backup_ignore [Page 129]), or terminated with backup_cancel [Page 128].

In the event of errors, see Reply Format [Page 193].

Backing Up to a Succeeding Medium

Use

You have started a backup [See SAP DB Library]. This was interrupted because the system reported that the backup medium [See SAP DB Library] is full.

You define a new backup medium. The backup continues to the medium you defined.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

Syntax

backup_replace <medium>

Reply

The reply to this command provides information about the backup supplied by the kernel.

This information is only given when the backup has been ended or interrupted. This command may therefore take a long time to execute.

```
OK<NL>
Returncode              <value><NL>
Date                    [<value>]<NL>
Time                    [<value>]<NL>
Server                  [<value>]<NL>
```

```
Database                [<value>]<NL>
Kernel Version          [<value>]<NL>
Pages Transferred       [<value>]<NL>
Pages Left              [<value>]<NL>
Volumn Count            [<value>]<NL>
Medianame               [<value>]<NL>
Location                [<value>]<NL>
Errortext               [<value>]<NL>
Label                   [<value>]<NL>
Is Consistent           [<value>]<NL>
First LOG Page          [<value>]<NL>
Last LOG Page           [<value>]<NL>
DB Stamp 1 Date         [<value>]<NL>
DB Stamp 1 Time         [<value>]<NL>
DB Stamp 2 Date         [<value>]<NL>
DB Stamp 2 Time         [<value>]<NL>
Page Count              [<value>]<NL>
Devices Used            [<value>]<NL>
Database ID             [<value>]<NL>
Max Used Data Page      [<value>]<NL>
```

In particular, analyze the `Returncode` reply field, which contains a numeric value supplied by the kernel.

See also: *Messages: SAP DB 7.4*

If it is clear from the message number that the backup was only interrupted, the session must be continued with the continuation command: (backup_replace [Page 130], backup_ignore [Page 129]), or terminated with backup_cancel [Page 128].

Values for the individual fields of the reply

`Date`	Date
`Time`	Time
`Server`	Name of the database server
`Database`	Name of database instance
`Kernel Version`	Database kernel version
`Pages Transferred`	Number of pages [See SAP DB Library] transferred
`Pages Left`	Number of pages still to be transferred
`Volumes`	Number of backup volumes used
`Medianame`	Name of backup medium
`Location`	File or device name
`Errortext`	Error message text
`Label`	Backup ID [See SAP DB Library]
`Is Consistent`	For data backup [See SAP DB Library] only: backup is internally consistent
`First LOG Page`	For data backup: first page of log backup to be read For log backup [See SAP DB Library]: first page saved in the log
`Last LOG Page`	For log backup only: last page saved in log

DB Stamp 1 Date DB Stamp 1 Time	Time stamp for first page of log backup
DB Stamp 2 Date DB Stamp 2 Time	Time stamp for last page of log backup
Page Count	Total number of pages saved
Devices Used	Number of backup devices used
Database ID	Database ID used to identify data and log backups that belong together
Max Used Data Page	Maximum number of pages used (indication of minimum database size when backup is imported)

In the event of errors, see Reply Format [Page 193].

Backing Up the Database Instance

Use

You back up the database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have started a utility session [Page 175].

Syntax

backup_start <medium> [<type>] [AUTOIGNORE]

<medium>	Name of the backup medium [See SAP DB Library]; for a backup to multiple parallel backup media (parallel backup [See SAP DB Library]), specify the name of the media group here. This must first be defined (Defining or Changing a Backup Medium [Page 115]).
<type>	Type of Backup: **DATA** (Complete data backup [See SAP DB Library]), **PAGES** (Incremental data backup), or **LOG** (Log Backup [See SAP DB Library])
AUTOIGNORE	The backup is to be automatically continued without the last backup medium reported as full (Continuing the Backup Without the Medium Last Reported as Full [Page 129])

If you are using backup media that is already defined, it is vital you follow the current naming conventions for backup media (see Defining or Changing a Backup Medium [Page 115]) to avoid error messages.

Reply

The reply to this command provides information about the backup supplied by the kernel.

This information is only given when the backup has been ended or interrupted. This command may therefore take a long time to execute.

```
OK<NL>
Returncode                <value><NL>
Date                      [<value>]<NL>
Time                      [<value>]<NL>
Server                    [<value>]<NL>
Database                  [<value>]<NL>
Kernel Version            [<value>]<NL>
Pages Transferred         [<value>]<NL>
Pages Left                [<value>]<NL>
Volumes                   [<value>]<NL>
Medianame                 [<value>]<NL>
Location                  [<value>]<NL>
Error text                [<value>]<NL>
Label                     [<value>]<NL>
Is Consistent             [<value>]<NL>
First LOG Page            [<value>]<NL>
Last LOG Page             [<value>]<NL>
DB Stamp 1 Date           [<value>]<NL>
DB Stamp 1 Time           [<value>]<NL>
DB Stamp 2 Date           [<value>]<NL>
DB Stamp 2 Time           [<value>]<NL>
Page Count                [<value>]<NL>
Devices Used              [<value>]<NL>
Database ID               [<value>]<NL>
Max Used Data Page        [<value>]<NL>
```

Values for the individual fields of the reply

`Date`	Date
`Time`	Time
`Server`	Name of the database server
`Database`	Name of database instance
`Kernel Version`	Database kernel version
`Pages Transferred`	Number of pages [See SAP DB Library] transferred
`Pages Left`	Number of pages still to be transferred
`Volumes`	Number of backup volumes used
`Medianame`	Name of backup medium
`Location`	File or device name
`Errortext`	Error message text
`Label`	Backup ID
`Is Consistent`	For data backup only: backup is internally consistent
`First LOG Page`	For data backup: first page of log backup to be read For log backup: first page saved in log
`Last LOG Page`	For log backup only: last page saved in log
`DB Stamp 1 Date` `DB Stamp 1 Time`	Time stamp for first page of log backup
`DB Stamp 2 Date` `DB Stamp 2 Time`	Time stamp for last page of log backup
`Page Count`	Total number of pages saved

Devices Used	Number of backup devices used
Database ID	Database ID used to identify data and log backups that belong together
Max Used Data Page	Maximum number of pages used (indication of minimum database size when backup is imported)

Reply in the Event of an Error

If an error occurs while you are using an external backup tool [See SAP DB Library], you will receive a reply in the following format:

```
ERR<NL>
<errcode>, <err_description><NL>
[<extended_description><NL>]
```

<errcode>	Error message number **See also:** *Messages: SAP DB 7.4*
<err_description>	Description of the error
<extended_description>	Cause of error

The following errors may occur:

<errcode>	**<err_description>**	**Explanation**
-24927	ERR_TOOLCHK: the external backup tool was not found	The external backup tool could not be found or has been installed incorrectly.
-24926	ERR_MEDIUMCHK: the medium cannot be used with an external backup tool	The specified backup medium cannot be used with the backup tool to which the medium name refers (Defining and Changing a Backup Medium [Page 115]).
-24925	ERR_PREPARE: prepare of the backup operation failed	The preparations necessary to use the backup tool were not made correctly.
-24924	ERR_DBREQ: cannot start database kernel request	The database instance was unable to start the backup.
-24923	ERR_TOOLREQ: cannot start external backup tool correctly	The backup tool could not be started correctly.
-24922	ERR_OPCHK: cannot check state of backup operation	Unable to check the status of the database instance or the backup tool.
-24921	ERR_POSTOP: cannot finish backup operation correctly	Although the backup was successful, the post-processing steps required could not be performed.
-24920	ERR_BACKUPOP: backup operation was unsuccessful	The backup failed due to a problem with the database or the backup tool.
-24919	ERR CLEANUP: cannot clean up correctly after backup	Although the backup was successful, the temporary

	operation	system resources that were used could not be freed up again.

If it is clear from the message number that the backup was only interrupted, the session must be continued with the continuation command: (backup_replace [Page 130], backup_ignore [Page 129]), or terminated with backup_cancel [Page 128].

Backing Up with External Backup Tools

The Database Manager CLI currently supports the use of the following external backup tools:

- ADSM/TSM (IBM(Tivoli)
- Backint for Oracle
- Backint for SAP DB
- NetWorker (Legato)

If you want to use an external backup tool that is not included on this list, please contact Support.

You can also use external backup tools to archive log backups [See SAP DB Library] (Archiving the Version Files of the Log Area [Page 124]).

Using the Backup Tools

Start the backup operation directly from the Database Manager CLI. The naming conventions for backup media [See SAP DB Library] enable the program to recognize the external backup tool and start it. The pipes for transferring the backup data are implicitly created by the Database Manager CLI during the backup process. They must not exist beforehand.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

The database instance is in the operational state `ADMIN` or `ONLINE`. You have defined the backup media in accordance with the naming conventions for external backup media (Defining or Changing a Backup Medium [Page 115]).

Procedure

Proceed as described in Backing Up the Database Instance [Page 132].

Backing up with ADSM:

```
dbmcli -u dbm,dbm -uUTL -d mydb backup_start ADSMData
recovery data
```

Commands for the Backup History

Use

The system writes the backup history [See SAP DB Library] to the file `dbm.knl`. This file is stored in the run directory [See SAP DB Library] of the database instance (see also: Database Manager Log Files [Page 196]).

Information on all the backup and restore actions that have been performed is recorded chronologically in the backup history.

The Database Manager CLI provides you with a number of DBM commands [Page 35], with which you can work with the backup history.

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

Syntax

Change Date of the Backup History [Page 136]	**backup_history_date**
Displaying the Backup History [Page 137]	**backup_history_list**
Scrolling in the Backup History [Page 139]	**backup_history_listnext**
Fetching the Backup History [Page 140]	**backup_history_open**
Closing a Backup History [Page 140]	**backup_history_close**

Change Data of the Backup History

Use

You can request the date on which the current backup history [See SAP DB Library] was last changed.

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

Syntax

backup_history_date

Reply

```
OK<NL>
<YYYYMMDDHHMMSS><NL>
```

In the event of errors, see Reply Format [Page 193].

Displaying the Backup History

Use

The system displays the content of the backup history [See SAP DB Library]. You can restrict or extend what is displayed using options.

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

You have fetched the backup history (Fetching the Backup History [Page 140]).

Syntax

```
backup_history_list <options>
```

Options

-c <columns>	With option **-c** you can restrict the number of columns to be displayed. The keywords for the individual columns are: **KEY** **LABEL** **ACTION** **STAMP1** **STAMP2** **START** **STOP** **FIRSTLOG** **LASTLOG** **LOG** **MEDIA** **PAGES** **VOLUMES** **RC** **ERROR** If several columns are specified, you separate them by commas.
-k <key>	Only the line of the backup history that contains the keyword specified under **<key>** is displayed. This option cannot be used with the **-r** option.
-l <label>	Only the lines of the backup history that contain the specified backup ID [See SAP DB Library] are displayed. This option cannot be used with the **-r** option.
-a <action>	Only the lines of the backup history that correspond to the specified backup type are output. This option cannot be used with the **-r** option.
-r	All data backups [See SAP DB Library] that are required to recover the database instance are displayed. If the log volume [See SAP DB Library] is intact, only the data backups that match the available log are displayed.
-r LAST	Beginning with the last complete data backup, all backups that are required to recover the database instance are displayed.
-r <key>	Beginning with the complete data backup specified in

	<key>, all backups that are required to recover the database instance are displayed.
-u <yyyymmddhhmmss>	Only those lines in the backup history that contain data that was saved in the database instance before the specified time are displayed. This option cannot be used with the **-r** option.
-m	For each line that relates to a backup, information is supplied on the backup medium [See SAP DB Library] used.
-e	For each line that relates to a backup, information is supplied on the relevant external backup ID [See SAP DB Library].

Reply

```
OK<NL>
(END|CONTINUE)<NL>
<history_line><NL>
[<media_line><NL>]
[<external_backup_id-line><NL>]
<history_line><NL>
[<media_line><NL>]
[<external_backup_id-line><NL>]
....
```

Values for the individual fields of the reply

END	The contents of the backup history have been transferred in full. The file is closed automatically.
CONTINUE	The backup history contains further entries that were not transferred due to the limited storage available for replies. You can interrogate this data by entering the above command or close the backup history [Page 140].
<history_line>	Information on the completed backups. The individual columns are separated by |. A separate line is output for each backup.
<media_line>	Information on the medium used for the backup The line is prefixed with an M: and is followed by columns separated by |.
<external_backup_id_line>	If the backup was created using an external backup tool, information on the external backup ID is provided here. You have to specify this information when restoring a backup [Page 158] from an external backup tool. The line is prefixed with E: which, in turn, is followed by columns separated with |.

In the event of errors, see Reply Format [Page 193].

Scrolling in the Backup History

Use

You read more of the backup history [See SAP DB Library]. Options you specified for displaying the backup history remain active.

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

You have fetched the backup history (Fetching the Backup History [Page 140]) and executed the command for displaying the backup history [Page 137]. The keyword `CONTINUE` in the reply shows that you have not yet read the whole file.

Syntax

`backup_history_listnext`

Reply

```
OK<NL>
[END|CONTINUE]<NL>
<history_line><NL>
[<media_line><NL>]
[<external_backup_id-line><NL>]
<history_line><NL>
[<media_line><NL>]
[<external_backup_id-line><NL>]
....
```

Values for the individual fields of the reply

`END`	The contents of the backup history have been transferred in full. The file is closed automatically.
`CONTINUE`	The backup history contains further entries that were not transferred due to the limited storage available for replies. You can display this data by entering the above command or close the backup history [Page 140].
`<history_line>`	Information on the completed backups [See SAP DB Library]. The individual columns are separated by |. A separate line is displayed for each backup.
`<media_line>`	Information on the backup medium [See SAP DB Library] used for the backup The line is prefixed with an `M:` and is followed by columns separated by |.
`<external_backup_id_line>`	If the backup was created using an external backup tool, information on the external backup ID is provided here. You have to specify this information when restoring a database instance [Page 158] from an external backup tool. The line is prefixed with `E:` which, in turn, is followed by columns separated with `|`.

In the event of errors, see Reply Format [Page 193].

Fetching the Backup History

Use

With this command you fetch the current backup history [See SAP DB Library]. The backup history is also updated with the current information about the utilized backup tools.

To view the content, choose **backup_history_list** (Displaying the Backup History [Page 137]) or **backup_history_listnext** (Scrolling Through the in the Backup History [Page 139]).

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

Syntax

```
backup_history_open <option>
```

Option

-e	If you specify the option **-e**, in addition to the backup history information, the system displays the status of the availability of the respective backups in the external backup tools [See SAP DB Library].

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Closing the Backup History

Use

You close a previously displayed backup history (Displaying the Backup History [Page 137]).

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

Syntax

```
backup_history_close
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Commands for External Backup IDs

Use

Backups created with external backup tools [See SAP DB Library] can be identified uniquely by means of an external backup ID [See SAP DB Library]. These IDs are logged in the backup history [See SAP DB Library]. They can be called up using the commands for the backup history [Page 136] and must be entered when restoring using external backup tools [Page 164].

To enable restores without a backup history, the external backup IDs of the backups currently in the backup tool can, however, also be determined using the DBM commands [See SAP DB Library].

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

Syntax

Displaying External Backup IDs [Page 141]	**backup_ext_ids_list**
Scrolling in the External Backup IDs [Page 142]	**backup_ext_ids_listnext**
Releasing the Memory Occupied by the External Backup IDs [Page 143]	**backup_ext_ids_forget**
Fetching External Backup IDs [Page 143]	**backup_ext_ids_get**

Displaying External Backup IDs

Use

You display the information requested by a backup tool. This information comprises, for each backup [See SAP DB Library], the availability status, the external backup ID [See SAP DB Library], the backup type, and the creation date/time.

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

You have fetched a backup tool's external backup IDs (Fetching External Backup IDs [Page 143]).

Syntax

backup_ext_ids_list

Reply

```
OK<NL>
(END|CONTINUE)<NL>
[<external_backup_id_line><NL>]
[<external_backup_id_line><NL>]
....
```

Values for the individual fields of the reply

END	All of the information requested by the backup tool was transferred in full.
CONTINUE	Information on further backups is available. This was not transferred because of the limited storage available for replies. You can scroll [Page 142] through this information or release the memory occupied by it [Page 143].
<external_backup_id_line>	A line containing information on one of the backups registered in the backup tool. The individual columns are separated by a |.

In the event of errors, see Reply Format [Page 193].

Scrolling in the External Backup IDs

Use

You display the external backup IDs [See SAP DB Library] of the individual backup media [See SAP DB Library] of a backup [See SAP DB Library]. In addition, the system displays the availability status, the backup type, and the creation time of the backup for each backup medium.

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

You have fetched the external backup IDs of a backup tool (Fetching External Backup IDs [Page 143]) and executed the command for displaying the external backup IDs [Page 141]. The keyword CONTINUE in the reply shows that you have not yet read the whole file.

Syntax

```
backup_ext_ids_listnext
```

Reply

```
OK<NL>
(END|CONTINUE)<NL>
[<external_backup_id_line><NL>]
[<external_backup_id_line><NL>]
....
```

Values for the individual fields of the reply

END	All of the requested information has been transferred in full.
CONTINUE	Information on further backups is available. This was not transferred because of the limited storage available for replies. You can call up this information by entering the above command again or release the occupied memory [Page 143].
<external_backup_id_line>	A line containing information on one of the backups still registered in the backup tool. The individual columns are separated by a line (|).

In the event of errors, see Reply Format [Page 193].

Releasing the Memory Occupied by the External Backup IDs

Use

You release the memory in which the external backup IDs [See SAP DB Library] requested by a backup tool are stored.

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

Syntax

```
backup_ext_ids_forget
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Fetching External Backup IDs

Use

You display the external backup IDs [See SAP DB Library] of backups [See SAP DB Library] for a database instance [See SAP DB Library] that are currently registered in the backup tool specified by **<medium>**.

Prerequisites

You have the DBM operator authorization Backup [Page 15] or DBInfoRead [Page 14].

Syntax

```
backup_ext_ids_get <medium> [<database_name>] [<server_node>]
```

<medium>	Name of the backup medium [See SAP DB Library]; this must be defined beforehand (Defining or Changing a Backup Medium [Page 115])
<database_name>	Name of the database instance from which the backups were created
<server_node>	Name of the server on which the backup was created

Reply

The system outputs an OK message.

Use the command **backup_ext_ids_list** (see Reading External Backup IDs [Page 141]) to display the external backup IDs.

In the event of errors, see Reply Format [Page 193].

Commands for Backup Information

Use

Whatever operational state [See SAP DB Library] the database instance is in, you can request the information held on the backup media [See SAP DB Library] about backups [See SAP DB Library]. You can use various DBM commands [Page 35] to do this.

Prerequisites

You have the operator authorization Backup [Page 15].

Syntax

Displaying the Backup Information in the Operational State OFFLINE [Page 144]	**Medium_labeloffline**
Displaying the Backup Information in the Operational State ONLINE or ADMIN [Page 146]	**Medium_label**
Displaying the Current Status of a Backup Check [Page 148]	**recover_state_check**
Checking a Backup [Page 149]	**recover_check**

Displaying the Backup Information in the Operational State OFFLINE

Use

You display the information about the backup stored on a backup medium [See SAP DB Library]. The system accesses the backup medium directly so that this function can be executed, if the database instance [See SAP DB Library] is in the operational state [See SAP DB Library] `OFFLINE`. If the database instance is in operational state `ONLINE` or `ADMIN`, use the command for displaying the backup information in the operational state ONLINE or ADMIN [Page 146].

Prerequisites

You have the DBM operator authorization Backup [Page 15].

Syntax

```
medium_labeloffline <medium>
```

Reply

```
OK<NL>
Returncode                <value><NL>
Date                      [<value>]<NL>
Time                      [<value>]<NL>
Server                    [<value>]<NL>
Database                  [<value>]<NL>
Kernel Version            [<value>]<NL>
Pages Transferred         [<value>]<NL>
Pages Left                [<value>]<NL>
Volume Count              [<value>]<NL>
Medianame                 [<value>]<NL>
Location                  [<value>]<NL>
Errortext                 [<value>]<NL>
Label                     [<value>]<NL>
Is Consistent             [<value>]<NL>
First LOG Page            [<value>]<NL>
Last LOG Page             [<value>]<NL>
DB Stamp 1 Date           [<value>]<NL>
DB Stamp 1 Time           [<value>]<NL>
DB Stamp 2 Date           [<value>]<NL>
DB Stamp 2 Time           [<value>]<NL>
Page Count                [<value>]<NL>
Devices Used              [<value>]<NL>
Database ID               [<value>]<NL>
Max Used Data Page        [<value>]<NL>
```

In particular, analyze the `Returncode` reply field, which contains a numeric value supplied by the kernel.

See also: *Messages: SAP DB 7.4*

Values for the individual fields of the reply

`Date`	Date
`Time`	Time
`Server`	Name of the database server
`Database`	Name of database instance
`Kernel Version`	Database kernel version
`Pages Transferred`	Number of pages transferred
`Pages Left`	Number of pages still to be transferred
`Volumes`	Number of backup volumes used
`Medianame`	Name of backup medium
`Location`	File or device name
`Errortext`	Error message text
`Label`	Backup ID
`Is Consistent`	For data backup [See SAP DB Library] only: backup is internally consistent
`First LOG Page`	For data backup: first page of the log backup [See SAP DB Library] to be read

	For log backup: first page saved in the log
`Last LOG Page`	For log backup only: last page saved in log
`DB Stamp 1 Date` `DB Stamp 1 Time`	Time stamp for first page of log backup
`DB Stamp 2 Date` `DB Stamp 2 Time`	Time stamp for last page of log backup
`Page Count`	Total number of pages saved
`Devices Used`	Number of backup devices used
`Database ID`	Database ID used to identify data and log backups that belong together
`Max Used Data Page`	Maximum number of pages used (indication of minimum database size when backup is imported)

In the event of errors, see Reply Format [Page 193].

Displaying the Backup Information in the Operational State ONLINE or ADMIN

Use

You display the information stored on a backup medium [See SAP DB Library] about the backup [See SAP DB Library].

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have started a utility session (Open a utility session [Page 175]). The database instance [See SAP DB Library] is in the operational state [See SAP DB Library] `ONLINE` or `ADMIN`.

Syntax

`medium_label <medium>`

Specify the backup medium from which the information is to be read under **`<medium>`**.

For parallel backup media, specify only the name of the group of parallel backup media [See SAP DB Library]. You must first define this group (Defining or Changing a Backup Medium [Page 115]).

Reply

```
OK<NL>
Returncode                <value><NL>
Date                      [<value>]<NL>
Time                      [<value>]<NL>
Server                    [<value>]<NL>
Database                  [<value>]<NL>
Kernel Version            [<value>]<NL>
Pages Transferred         [<value>]<NL>
Pages Left                [<value>]<NL>
Volume Count              [<value>]<NL>
```

```
Medianame               [<value>]<NL>
Location                [<value>]<NL>
Errortext               [<value>]<NL>
Label                   [<value>]<NL>
Is Consistent           [<value>]<NL>
First LOG Page          [<value>]<NL>
Last LOG Page           [<value>]<NL>
DB Stamp 1 Date         [<value>]<NL>
DB Stamp 1 Time         [<value>]<NL>
DB Stamp 2 Date         [<value>]<NL>
DB Stamp 2 Time         [<value>]<NL>
Page Count              [<value>]<NL>
Devices Used            [<value>]<NL>
Database ID             [<value>]<NL>
Max Used Data Page      [<value>]<NL>
```

In particular, analyze the `Returncode` reply field, which contains a numeric value supplied by the kernel.

See also: *Messages: SAP DB 7.4*

Values for the individual fields of the reply

`Date`	Date
`Time`	Time
`Server`	Name of the database server
`Database`	Name of database instance
`Kernel Version`	Database kernel version
`Pages Transferred`	Number of pages transferred
`Pages Left`	Number of pages still to be transferred
`Volumes`	Number of backup volumes used
`Medianame`	Name of backup medium
`Location`	File or device name
`Errortext`	Error message text
`Label`	Backup ID [See SAP DB Library]
`Is Consistent`	For data backup [See SAP DB Library] only: backup is internally consistent
`First LOG Page`	For data backup: first page of log backup to be read For log backup [See SAP DB Library]: first page saved in the log
`Last LOG Page`	For log backup only: last page saved in log
`DB Stamp 1 Date` `DB Stamp 1 Time`	Time stamp for first page of log backup
`DB Stamp 2 Date` `DB Stamp 2 Time`	Time stamp for last page of log backup
`Page Count`	Total number of pages saved
`Devices Used`	Number of backup devices used
`Database ID`	Database ID used to identify data and log backups that belong together

Max Used Data Page	Maximum number of pages used (indication of minimum database size when backup is imported)

In the event of errors, see Reply Format [Page 193].

Displaying the Current Status of a Backup Check

Use

You display information about the current status of a backup check [Page 149]. You can run this command while a backup is being checked.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have opened a service session (Opening a Service Session [Page 183]).

Syntax

recover_state_check

Reply

```
OK<NL>
Return code               <value><NL>
Date                     [<value>]<NL>
Time                     [<value>]<NL>
Server                   [<value>]<NL>
Database                 [<value>]<NL>
Kernel Version           [<value>]<NL>
Pages Transferred        [<value>]<NL>
Pages Left               [<value>]<NL>
Volumes                  [<value>]<NL>
Medianame                [<value>]<NL>
Location                 [<value>]<NL>
Error text               [<value>]<NL>
Label                    [<value>]<NL>
Is Consistent            [<value>]<NL>
First LOG Page           [<value>]<NL>
Last LOG Page            [<value>]<NL>
DB Stamp 1 Date          [<value>]<NL>
DB Stamp 1 Time          [<value>]<NL>
DB Stamp 2 Date          [<value>]<NL>
DB Stamp 2 Time          [<value>]<NL>
Page Count               [<value>]<NL>
Devices Used             [<value>]<NL>
Database ID              [<value>]<NL>
Max Used Data Page       [<value>]<NL>
```

Values for the individual fields of the reply

Date	Date
Time	Time
Server	Name of the database server

`Database`	Name of database instance
`Kernel Version`	Version of the database software
`Pages Transferred`	Number of pages [See SAP DB Library] transferred
`Pages Left`	Number of pages still to be transferred
`Volumes`	Number of backup volumes used
`Medianame`	Name of the backup medium [See SAP DB Library]
`Location`	File or device name
`Errortext`	Error message text
`Label`	Backup ID [See SAP DB Library]
`Is Consistent`	For data backup [See SAP DB Library] only: backup is internally consistent
`First LOG Page`	For data backup: first page of log backup to be read For log backup [See SAP DB Library]: first page saved in the log
`Last LOG Page`	Only for log backup: last page saved in log
`DB Stamp 1 Date` `DB Stamp 1 Time`	Time stamp for first page of log backup
`DB Stamp 2 Date` `DB Stamp 2 Time`	Time stamp for last page of log backup
`Page Count`	Total number of pages saved
`Devices Used`	Number of backup devices used
`Database ID`	Database ID used to identify data and log backups that belong together
`Max Used Data Page`	Highest page number assigned (indication of minimum database size when backup is imported)

In the event of errors, see Reply Format [Page 193].

Checking a Backup

Use

You check that the backup [See SAP DB Library] is consistent and complete. This command is executed independently of the running functions of the current database instance [See SAP DB Library] but can temporarily adversely impact database instance performance.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have opened a service session [Page 183].

Syntax

```
recover_check <medium> <type> [ExternalBackupID <external_backup_ID>]
[<nnn>] [UNTIL <date> <time>]
```

<medium>	Name of the backup medium [See SAP DB Library]; this must be defined beforehand (Defining or Changing a Backup Medium [Page 115])
<type>	Type of Backup: **DATA** (Complete data backup [See SAP DB Library]), **PAGES** (Incremental data backup) or **LOG** (Log Backup [See SAP DB Library])

Options

ExternalBackupID <external_backup_ID>	To check a backup created with an external backup tool [See SAP DB Library], enter a backup ID name that the external tool will recognize.
<nnn>	Actual backup version on the backup media that is being checked; relevant only for media of type **FILE**
UNTIL <date> <time>	For log backups you can enter an exact time up to which the log backups are to be checked.

The backup specified is just checked, not restored.

Reply

```
OK<NL>
Returncode                <value><NL>
Date                      [<value>]<NL>
Time                      [<value>]<NL>
Server                    [<value>]<NL>
Database                  [<value>]<NL>
Kernel Version            [<value>]<NL>
Pages Transferred         [<value>]<NL>
Pages Left                [<value>]<NL>
Volume Count              [<value>]<NL>
Mediumname                [<value>]<NL>
Location                  [<value>]<NL>
Errortext                 [<value>]<NL>
Label                     [<value>]<NL>
Is Consistent             [<value>]<NL>
First LOG Page            [<value>]<NL>
Last LOG Page             [<value>]<NL>
DB Stamp 1 Date           [<value>]<NL>
DB Stamp 1 Time           [<value>]<NL>
DB Stamp 2 Date           [<value>]<NL>
DB Stamp 2 Time           [<value>]<NL>
Page Count                [<value>]<NL>
Devices Used              [<value>]<NL>
Database ID               [<value>]<NL>
Max Used Data Page        [<value>]<NL>
```

Values for the individual fields of the reply

Date	Date
Time	Time
Server	Name of the database server
Database	Name of database instance

Kernel Version	Database kernel version
Pages Transferred	Number of pages [See SAP DB Library] transferred
Pages Left	Number of pages still to be transferred
Volumes	Number of backup volumes used
Medianame	Name of backup medium
Location	File or device name
Errortext	Error message text
Label	Backup ID [See SAP DB Library]
Is Consistent	For data backup only: backup is internally consistent
First LOG Page	For data backup: first page of log backup to be read For log backup: first page saved in log
Last LOG Page	For log backup only: last page saved in log
DB Stamp 1 Date DB Stamp 1 Time	Time stamp for first page of log backup
DB Stamp 2 Date DB Stamp 2 Time	Time stamp for last page of log backup
Page Count	Total number of pages saved
Devices Used	Number of backup devices used
Database ID	Database ID used to identify data and log backups that belong together
Max Used Data Page	Maximum number of pages used (indication of minimum database size when backup is imported)

Reply in the Event of an Error

If an error occurs during the process, you will receive a reply in the following format:

```
ERR<NL>
<errcode>, <err_description><NL>
[<extended_description><NL>]
```

<errcode>	Error message number **See also**: *Messages: SAP DB 7.2 and 7.3*
<err_description>	Description of the error
<extended_description>	Cause of error

The following errors may occur:

<errcode>	**<err_description>**	**Explanation**
-24927	ERR_TOOLCHK: the external backup tool was not found	The external backup tool could not be found or has been installed incorrectly.
-24926	ERR_MEDIUMCHK: the medium cannot be used with an external backup tool	The medium specified cannot be used with the backup tool the medium name refers to (Defining and Changing a Backup Medium [Page 115]) .

-24925	ERR_PREPARE: prepare of the backup operation failed	The preparations necessary to use the backup tool were not made correctly.
-24924	ERR_DBREQ: cannot start database kernel request	The database instance was unable to start the check.
-24923	ERR_TOOLREQ: cannot start external backup tool correctly	The backup tool could not be started correctly.
-24922	ERR_OPCHK: cannot check state of backup operation	Unable to check the status of the database instance or the backup tool.
-24921	ERR_POSTOP: cannot finish backup operation correctly	Although the backup was successful, the post-processing steps required could not be performed.
-24920	ERR_BACKUPOP: backup operation was unsuccessful	The check failed due to a problem with the database or the backup tool.
-24919	ERR CLEANUP: cannot clean up correctly after backup operation	Although the check was successful, the temporary system resources that were used could not be freed up again.

Commands for Restoring

Use

The Database Manager supports the restoring of the database instance [Page 158] after hardware errors. The last state of the database is restored. A prerequisite for this is that all the data specified in the backup history [See SAP DB Library] is available.

To obtain the highest possible throughput of data, complete data backups [See SAP DB Library] and incremental data backups [See SAP DB Library] can be imported from a number of backup media [See SAP DB Library] simultaneously. The number of parallel backup media used for the restore does not depend on the number of parallel backup media used to make the backup originally. Even a backup made to single medium plus continuation media can be imported in parallel.

You define the maximum number of backup media that can be imported concurrently using the parameter MAX_BACKUP_DEVS. The use of up to 32 tape devices allows you to reduce backup and restore times considerably.

Prerequisites

Note the operator authorizations [Page 13] required for the particular commands DBM commands [Page 35].

The prerequisite for executing all other restore commands is to execute the command to restore the database instance [Page 158] first.

Syntax

Displaying Current Information About the Database Instance [Page 153]	**db_restartinfo**
Displaying the Current Recovery Status [Page 154]	**recover_state**
Terminating an Interrupted Restore or Backup Check [Page 156]	**recover_cancel**
Continuing a Parallel Restore or Backup Check [Page 156]	**recover_ignore**
Restoring the Database Instance [Page 158]	**recover_start**
Restoring the Parameter File from a Data Backup [Page 161]	**recover_config**
Recreating a Damaged Index [Page 163]	**sql_recreateindex**
Restoring with External Backup Tools [Page 164]	**---**
Restoring or Checking a Backup with Succeeding Medium [Page 164]	**recover_replace**

Displaying Current Information About the Database Instance

Use

You can display the current status of the database instance [See SAP DB Library] after importing a backup [See SAP DB Library]. Information is displayed on the state of the log and the ability of the database instance to Restart (Restartability [See SAP DB Library]).

Prerequisites

You have the DBM operator authorization Recovery [Page 18].

You have executed the command for restoring a database instance [Page 158].

Syntax

db_restartinfo

dbmcli -d myDB -u dbm,dbm db_restartinfo

```
OK
Used LOG Page           10636
First LOG Page          8640
Restartable             1
Id Restart Record       P46643:DB_20010629_115558
Id LOG Info             P46643:DB_20010629_115558
Consistent              1
```

Reply

```
OK<NL>
Used LOG Page            <number><NL>
First LOG Page           <number><NL>
Restartable              (1|0)<NL>
Id Restart Record        <id><NL>
Id LOG Info              <id><NL>
Consistent               (1|0)<NL>
```

Values for the individual fields of the reply

`Used LOG Page`	Next log page to be used after restarting
`First LOG Page`	Oldest available log page
`Restartable`	Restartable (`1` = yes, `0` = no)
`Id Restart Record`	Identifier of the restart record for the database instance
`Id LOG Info`	Identifier for the database instance log
`Consistent`	Consistency of the database instance, that is, can be restarted without log (`1` = yes, `0` = no)

In the event of errors, see Reply Format [Page 193].

Displaying the Current Restore Status

Use

You display information about the current status of the restore. You can execute this command while a backup of a database instance is being restored.

Prerequisites

You have the DBM operator authorization Recovery [Page 18].

You have executed the command for restoring a database instance [Page 158]. The operation was interrupted.

Syntax

```
recover_state
```

Reply

```
OK<NL>
Return code               <value><NL>
Date                     [<value>]<NL>
Time                     [<value>]<NL>
Server                   [<value>]<NL>
Database                 [<value>]<NL>
Kernel Version           [<value>]<NL>
Pages Transferred        [<value>]<NL>
Pages Left               [<value>]<NL>
Volumes                  [<value>]<NL>
Medianame                [<value>]<NL>
Location                 [<value>]<NL>
```

```
Error text              [<value>]<NL>
Label                   [<value>]<NL>
Is Consistent           [<value>]<NL>
First LOG Page          [<value>]<NL>
Last LOG Page           [<value>]<NL>
DB Stamp 1 Date         [<value>]<NL>
DB Stamp 1 Time         [<value>]<NL>
DB Stamp 2 Date         [<value>]<NL>
DB Stamp 2 Time         [<value>]<NL>
Page Count              [<value>]<NL>
Devices Used            [<value>]<NL>
Database ID             [<value>]<NL>
Max Used Data Page      [<value>]<NL>
```

Values for the individual fields of the reply

Date	Date
Time	Time
Server	Name of the database server
Database	Name of the database
Kernel Version	Database kernel version
Pages Transferred	Number of pages transferred
Pages Left	Number of pages still to be transferred
Volumes	Number of backup volumes used
Medianame	Name of the backup medium [See SAP DB Library]
Location	File or device name
Errortext	Error message text
Label	Backup ID [See SAP DB Library]
Is Consistent	For data backup [See SAP DB Library] only: backup is internally consistent
First LOG Page	For data backup: first page of log backup [See SAP DB Library] to be read For log backup [See SAP DB Library]: first page saved in the log
Last LOG Page	Only for log backup: last page saved in log
DB Stamp 1 Date DB Stamp 1 Time	Time stamp for first page of log backup
DB Stamp 2 Date DB Stamp 2 Time	Time stamp for last page of log backup
Page Count	Total number of pages saved
Devices Used	Number of backup devices used
Database ID	Database ID used to identify data and log backups that belong together
Max Used Data Page	Highest page number assigned (indication of minimum database size when backup is imported)

In the event of errors, see Reply Format [Page 193].

Terminating an Interrupted Restore or a Backup Check

Use

You definitively terminate an interrupted restore or a backup check.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have executed the command for Restoring the Database Instance [Page 158] or for Checking a Backup [Page 149]. The operation was interrupted.

Syntax

```
recover_cancel
```

Reply

In the reply to this command only the field `Returncode` is assigned a value.

```
OK<NL>
Returncode <value>
```

See also: *Messages: SAP DB 7.4*

In the event of errors, see Reply Format [Page 193].

Continuing the Parallel Restore or Backup Check

Use

You continue reading in from the other backup media [See SAP DB Library]. As a reply to this command, you receive the information about the import of the backup [See SAP DB Library].

The output is, however, only given when the backup has been completely read in or is interrupted. This command may therefore take a long time to execute.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have started restoring a database instance [Page 158] by reading data from several media simultaneously. The procedure was interrupted because at least one medium no longer has a succeeding backup medium.

Syntax

```
recover_ignore
```

Reply

```
OK<NL>
Returncode                <value><NL>
Date                      [<value>]<NL>
Time                      [<value>]<NL>
Server                    [<value>]<NL>
Database                  [<value>]<NL>
Kernel Version            [<value>]<NL>
Pages Transferred         [<value>]<NL>
Pages Left                [<value>]<NL>
Volume Count              [<value>]<NL>
Medianame                 [<value>]<NL>
Location                  [<value>]<NL>
Errortext                 [<value>]<NL>
Label                     [<value>]<NL>
Is Consistent             [<value>]<NL>
First LOG Page            [<value>]<NL>
Last LOG Page             [<value>]<NL>
DB Stamp 1 Date           [<value>]<NL>
DB Stamp 1 Time           [<value>]<NL>
DB Stamp 2 Date           [<value>]<NL>
DB Stamp 2 Time           [<value>]<NL>
Page Count                [<value>]<NL>
Devices Used              [<value>]<NL>
Database ID               [<value>]<NL>
Max Used Data Page        [<value>]<NL>
```

In particular, analyze the `Returncode` reply field, which contains a numeric value supplied by the kernel.

See also: *Messages: SAP DB 7.4*

If it is clear from the message number that the recovery was only interrupted, the session must be continued with the continuation command (**`recover_ignore`**) or terminated (recover_cancel [Page 156]).

Values for the individual fields of the reply

`Date`	Date
`Time`	Time
`Server`	Name of the database server
`Database`	Name of the database instance [See SAP DB Library]
`Kernel Version`	Database kernel version
`Pages Transferred`	Number of pages [See SAP DB Library] transferred
`Pages Left`	Number of pages still to be transferred
`Volumes`	Number of backup volumes used
`Medianame`	Name of backup medium
`Location`	File or device name
`Errortext`	Error message text
`Label`	Backup ID [See SAP DB Library]
`Is Consistent`	For data backup [See SAP DB Library] only: backup is internally consistent

`First LOG Page`	For data backup: first page of log backup [See SAP DB Library] to be read For log backup: first page saved in log
`Last LOG Page`	Only for log backup: last page saved in log
`DB Stamp 1 Date` `DB Stamp 1 Time`	Time stamp for first page of log backup
`DB Stamp 2 Date` `DB Stamp 2 Time`	Time stamp for last page of log backup
`Page Count`	Total number of pages saved
`Devices Used`	Number of backup devices used
`Database ID`	Database ID used to identify data and log backups that belong together
`Max Used Data Page`	Maximum number of pages used (indication of minimum database size when backup is imported)

In the event of errors, see Reply Format [Page 193].

Recovering a Database Instance

Use

Use this command to import a backup [See SAP DB Library] and recover the state of the database that it contains. The reply to this command gives you information about the import of the backup. However, you see this information only if the backup has been imported completely, or if the backup was interrupted. This command may, therefore, take a long time to execute.

If automatic log backups were activated before the recovery was started, they are **not** reactivated automatically after the recovery.

To reactivate the log backups, execute the command for activating automatic log backups [Page 129].

Prerequisites

You have the DBM operator authorization Recovery [Page 18].

Syntax

```
recover_start <medium> <type> [ExternalBackupID <external_backup_ID>]
[<nnn>] [UNTIL <date> <time>] [AUTOIGNORE]
```

`<medium>`	Backup medium [See SAP DB Library] from which you want to import the backup. When you restore a backup made on multiple parallel backup

	media (a parallel backup), enter the name of the media group here.
<type>	Type of the backup that you want to import: **DATA** (complete data backup [See SAP DB Library]), **PAGES** (incremental data backup [See SAP DB Library]), or **LOG** (log backup [See SAP DB Library])

Options

ExternalBackupID <external_backup_ID>	To restore a backup created with an external backup tool [See SAP DB Library], enter a backup ID name here that the external backup tool can recognize as the backup.
<nnn>	Actual backup version on the backup medium that you want to read; relevant only for media of type **FILE**.
UNTIL <date> <time>	You can specify a time up to which you want to import log backups.
AUTOIGNORE	In the case of a parallel restore, the process is automatically continued by the system (Continuing a Parallel Restore [Page 156]).

It is vital that you follow the current media naming conventions (see Defining or Changing a Backup Medium [Page 115]) to avoid error messages.

Reply

```
OK<NL>
Return code               <value><NL>
Date                     [<value>]<NL>
Time                     [<value>]<NL>
Server                   [<value>]<NL>
Database                 [<value>]<NL>
Kernel Version           [<value>]<NL>
Pages Transferred        [<value>]<NL>
Pages Left               [<value>]<NL>
Volumes                  [<value>]<NL>
Medianame                [<value>]<NL>
Location                 [<value>]<NL>
Error text               [<value>]<NL>
Label                    [<value>]<NL>
Is Consistent            [<value>]<NL>
First LOG Page           [<value>]<NL>
Last LOG Page            [<value>]<NL>
DB Stamp 1 Date          [<value>]<NL>
DB Stamp 1 Time          [<value>]<NL>
DB Stamp 2 Date          [<value>]<NL>
DB Stamp 2 Time          [<value>]<NL>
Page Count               [<value>]<NL>
Devices Used             [<value>]<NL>
Database ID              [<value>]<NL>
Max Used Data Page       [<value>]<NL>
```

Values for the individual fields of the reply

Date	Date
Time	Time
Server	Name of the database server

Database	Name of the database instance
Kernel Version	Database kernel version
Pages Transferred	Number of pages transferred
Pages Left	Number of pages still to be transferred
Volumes	Number of backup media used
Medianame	Name of the backup medium
Location	File or device name
Errortext	Error message text
Label	Backup ID [See SAP DB Library]
Is Consistent	Only for data backup: backup is internally consistent
First LOG Page	For data backup: First page [See SAP DB Library] of the log backup to be read For log backup: first page saved in log
Last LOG Page	Only for log backup: last page saved in log
DB Stamp 1 Date DB Stamp 1 Time	Time stamp for first page of log backup
DB Stamp 2 Date DB Stamp 2 Time	Time stamp for last page of log backup
Page Count	Total number of pages saved
Devices Used	Number of backup devices used
Database ID	Database ID used to identify data and log backups that belong together
Max Used Data Page	Maximum page number (indication of minimum database size when backup is imported)

Reply in the Event of an Error

If an error occurs while you are using an external backup tool, you receive a reply in the following format:

```
ERR<NL>
<errcode>, <err_description><NL>
[<extended_description><NL>]
```

<errcode>	Error message number **See also:** *Messages: SAP DB 7.4*
<err_description>	Description of the error
<extended_description>	Cause of the error

The following errors may occur:

<errcode>	**<err_description>**	Explanation
-24927	ERR_TOOLCHK: the external backup tool was not found	The external backup tool could not be found or has been installed incorrectly.
-24926	ERR_MEDIUMCHK: the medium cannot be used with an	The specified backup medium cannot be used with the

	external backup tool	backup tool to which the medium name refers (Defining or Changing a Backup Medium [Page 115]).
-24925	ERR_PREPARE: prepare of the backup operation failed	The preparations necessary to use the backup tool were not made correctly.
-24924	ERR_DBREQ: cannot start database kernel request	The database instance was unable to start the restore operation.
-24923	ERR_TOOLREQ: cannot start external backup tool correctly	The backup tool could not be started correctly.
-24922	ERR_OPCHK: cannot check state of backup operation	Unable to check the status of the database instance or the backup tool.
-24921	ERR_POSTOP: cannot finish backup operation correctly	Although the recovery was successful, the post-processing steps required could not be performed.
-24920	ERR_BACKUPOP: backup operation was unsuccessful	The recovery failed due to a problem with the database or the backup tool.
-24919	ERR CLEANUP: cannot clean up correctly after backup operation	Although the recovery was successful, the temporary system resources used could not be freed up again.

Restoring the Parameter File from a Data Backup

Use

You recover the parameter file from the data backup [See SAP DB Library] stored on a backup medium [See SAP DB Library].

All parameters except for **SERVERDB** are taken over. The parameter **SERVERDB** is converted to the value of the current database instance [See SAP DB Library].

The parameter file is stored under the name of the current database instance.

Prerequisites

You have the DBM operator authorization Recovery [Page 18].

You have opened a service session [Page 183].

You have executed the command for restoring a database instance [Page 158].

Syntax

```
recover_config <medium>
```

Reply

```
OK<NL>
Return code               <value><NL>
Date                     [<value>]<NL>
Time                     [<value>]<NL>
Server                   [<value>]<NL>
Database                 [<value>]<NL>
Kernel Version           [<value>]<NL>
Pages Transferred        [<value>]<NL>
Pages Left               [<value>]<NL>
Volumn Count             [<value>]<NL>
Medianame                [<value>]<NL>
Location                 [<value>]<NL>
Error text                [<value>]<NL>
Label                    [<value>]<NL>
Is Consistent            [<value>]<NL>
First LOG Page           [<value>]<NL>
Last LOG Page            [<value>]<NL>
DB Stamp 1 Date          [<value>]<NL>
DB Stamp 1 Time          [<value>]<NL>
DB Stamp 2 Date          [<value>]<NL>
DB Stamp 2 Time          [<value>]<NL>
Page Count               [<value>]<NL>
Devices Used             [<value>]<NL>
Database ID              [<value>]<NL>
Max Used Data Page       [<value>]<NL>
```

Values for the individual fields of the reply

`Date`	Date
`Time`	Time
`Server`	Name of the database server
`Database`	Name of database instance
`Kernel Version`	Database kernel version
`Pages Transferred`	Number of pages [See SAP DB Library] transferred
`Pages Left`	Number of pages still to be transferred
`Volumes`	Number of backup media [See SAP DB Library] used
`Medianame`	Name of the backup medium
`Location`	File or device name
`Errortext`	Error message text
`Label`	Backup ID [See SAP DB Library]
`Is Consistent`	For data backup only: backup is internally consistent
`First LOG Page`	For data backup: first page of log backup to be read For log backup [See SAP DB Library]: first page saved in the log
`Last LOG Page`	For log backup only: last page saved in log
`DB Stamp 1 Date` `DB Stamp 1 Time`	Time stamp for first page of log backup
`DB Stamp 2 Date` `DB Stamp 2 Time`	Time stamp for last page of log backup

`Page Count`	Total number of pages saved
`Devices Used`	Number of backup devices used
`Database ID`	Database ID used to identify data and log backups that belong together
`Max Used Data Page`	Maximum number of pages used (indication of minimum database size when backup is imported)

In the event of errors, see Reply Format [Page 193].

Recreating a Damaged Index

Use

If you do not specify an options, you recover **all** damaged indexes with this command. If you specify data for a certain index, only the selected index is recovered:

To save time, indexes are not maintained when a backup is imported. These must be recovered explicitly. You recover damaged indexes or indexes that are no longer up to date.

The database software ignores damaged indexes. For this reason, they cannot cause errors, but do impair the performance of the database instance.

Prerequisites

You have the DBM operator authorization AccessSQL [Page 20].

You have opened an SQL session [Page 181].

You have executed the command for restoring a database instance [Page 158].

Syntax

```
sql_recreateindex [<scheme>.<table>.<index>]
```

Options

`<scheme>`	Scheme
`<table>`	Table name
`<index>`	Index name or field name

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Restoring with External Backup Tools

If you want to use an external backup tool [See SAP DB Library] that is not included on this list, contact SAP DB Support [See SAP DB Library].

The Database Manager CLI supports the use of the following external backup tools:

- ADSM/TSM (IBM/Tivoli)
- Backint for Oracle
- Backint SAP DB
- NetWorker (Legato)

Using the Backup Tools

Start the operation for restoring the database instance [Page 158] from the Database Manager CLI. The naming conventions for external backup media [See SAP DB Library] enable the program to recognize the external backup tool and start it.

Procedure

First determine the external backup ID [See SAP DB Library] for the required backup (Fetching External Backup IDs [Page 143], Displaying External Backup IDs [Page 141]). The Database Manager uses this ID to identify the backup that is to be restored.

The number of media in a group of parallel backup media [See SAP DB Library] must correspond to the number of media used to create the backup.

Proceed as described in Restoring the Database Instance [Page 158].

Restoring using NetWorker:

```
dbmcli -u dbm,dbm -uUTL -d myDB recover_start NSRPages pages
ExternalBackupID 9025
```

Restoring or Checking a Backup with a Succeeding Medium

Use

You specify the next medium from which reading in the backup [See SAP DB Library] is to continue. As a reply to this command, you receive the information about the import of the backup. The output is, however, only given when the backup has been completely read in or is interrupted. This command may therefore take a long time to execute.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have executed the command for restoring a database instance [Page 158]. The procedure was interrupted, because all backup data has been read from one backup medium [See SAP DB Library].

Syntax

```
recover_replace <medium> [ExternalBackupID <ext_backup_ID>] [<loc>
[<nnn>]]
```

<medium>	Medium from which the backup is to be read. When restoring a backup made on several media at the same time, you must enter the name of the group of parallel backup media [See SAP DB Library] here.

Options

ExternalBackupID <external_backup_ID>	To restore a backup created with an external backup tool [See SAP DB Library], enter a backup ID name here with which the external backup tool will recognize the backup.
<loc>	Name of the device/the file in which the backup to be restored is stored. Operating system functions must be able to address the device/file using **<loc>.**
<nnn>	Actual backup version on the backup medium that is to be read; relevant only for a backup medium of the type **FILE**.

It is vital you follow the current media naming conventions (see Defining or Changing a Backup Medium [Page 115]) to avoid error messages.

Reply

```
OK<NL>
Returncode                <value><NL>
Date                      [<value>]<NL>
Time                      [<value>]<NL>
Server                    [<value>]<NL>
Database                  [<value>]<NL>
Kernel Version            [<value>]<NL>
Pages Transferred         [<value>]<NL>
Pages Left                [<value>]<NL>
Volume Count              [<value>]<NL>
Medianame                 [<value>]<NL>
Location                  [<value>]<NL>
Errortext                 [<value>]<NL>
Label                     [<value>]<NL>
Is Consistent             [<value>]<NL>
First LOG Page            [<value>]<NL>
Last LOG Page             [<value>]<NL>
DB Stamp 1 Date           [<value>]<NL>
DB Stamp 1 Time           [<value>]<NL>
DB Stamp 2 Date           [<value>]<NL>
DB Stamp 2 Time           [<value>]<NL>
Page Count                [<value>]<NL>
Devices Used              [<value>]<NL>
Database ID               [<value>]<NL>
Max Used Data Page        [<value>]<NL>
```

In particular, analyze the `Returncode` reply field, which contains a numeric value supplied by the kernel.

See also: *Messages: SAP DB 7.2 and 7.3*

If it is clear from the message number that the recovery was only interrupted, the session must be continued with the continuation command (**recover_replace**) or terminated (recover_cancel [Page 156]).

Values for the individual fields of the reply

Date	Date
Time	Time
Server	Name of the database server
Database	Name of the database
Kernel Version	Database kernel version
Pages Transferred	Number of pages transferred
Pages Left	Number of pages still to be transferred
Volumes	Number of backup media used
Medianame	Name of the backup medium
Location	File or device name
Errortext	Error message text
Label	Backup ID [See SAP DB Library]
Is Consistent	For data backup [See SAP DB Library] only: backup is internally consistent
First LOG Page	For data backup: first page of log backup [See SAP DB Library] to be read For log backup: first page saved in log
Last LOG Page	For log backup only: last page saved in log
DB Stamp 1 Date DB Stamp 1 Time	Time stamp for first page of log backup
DB Stamp 2 Date DB Stamp 2 Time	Time stamp for last page of log backup
Page Count	Total number of pages saved
Devices Used	Number of backup devices used
Database ID	Database ID used to identify data and log backups that belong together
Max Used Data Page	Maximum number of pages used (indication of minimum database size when backup is imported)

In the event of errors, see Reply Format [Page 193].

Managing the DBM Operators

Use

The Database Manager CLI provides a number of DBM commands [Page 35] with which you can display data for DBM operators or transfer operator data to the Database Manager.

Changing DBM Operator Data [Page 167]	`user_put`
Creating DBM Operators [Page 168]	`user_create`
Displaying Operator Authorizations [Page 168]	`user_getrights`
Displaying Operator Data [Page 169]	`user_get`
List of DBM Operators [Page 170]	`user_getall`
Deleting DBM Operators [Page 171]	`user_delete`

Prerequisites

You have the DBM operator authorization UserMgm [Page 18].

Changing DBM Operator Data

Use

You can modify the operator data of the DBM operator [See SAP DB Library] specified in **<userid>**.

The new values are specified with **<property>=<value>**.

In **<property>** you can specify all operator properties [Page 12] except the *Name*. The name of an operator must not be modified.

You can specify multiple operator authorizations [Page 13]. Separate these from each other with commas. Put a plus sign in front of the authorization to assign it and a minus sign to revoke it.

Prerequisites

You have the DBM operator authorization UserMgm [Page 18].

Syntax

```
user_put <userid> <property>=<value> [, <property>=<value>, ...]
```

Reply

The system outputs an OK message.

```
user_put samplename SERVERRIGHTS=+StartDB, -StopDB
OK
```

In the event of errors, see Reply Format [Page 193].

Creating DBM Operators

Use

You create a DBM operator [See SAP DB Library] having the *operator name* and *operator password* specified in **<userid>** and **<password>**.

At first, this operator has no operator authorizations [Page 13]. You must explicitly assign these to the operator (Changing the DBM Operator Data [Page 167]).

Alternatively, you can also specify a **<template_user>**, from which all operator authorizations, but not operator name and operator password are then copied.

Prerequisites

You have the DBM operator authorization UserMgm [Page 18].

Syntax

```
user_create <userid>,<password> [<template_user>]
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Displaying Operator Authorizations

Use

You display the list of operator authorizations [Page 13] for the operator specified in **<userid>**.

Prerequisites

You have the DBM operator authorization UserMgm [Page 18].

Syntax

```
user_getrights <userid>
```

Example [Page 169]

Reply

```
OK<NL>
UserMgm                 <flag>  User management
InstallMgm              <flag>  Install management
SystemCmd               <flag>  System commands
FileFull                <flag>  File full access
FileRead                <flag>  File read access
DBFileRead              <flag>  Database file read access
AccessUtility           <flag>  Utility session access
BackupRestore           <flag>  Backup restore actions
BackupSave              <flag>  Backup save actions
```

```
ParamFull               <flag>  Parameter full access
ParamCheckWrite         <flag>  Parameter check-write access
ParamRead               <flag>  Parameter read access
AccessSQL               <flag>  SQL session access
ExecLoad                <flag>  Can execute load
LoadSysTab              <flag>  Can load system tables
DBStop                  <flag>  Can stop database
DBStart                 <flag>  Can start database
DBInfoRead              <flag>  Info functions access
```

A plus symbol under `<flag>` indicates that the relevant authorization is assigned to the operator. A minus symbol under `<flag>` indicates that the relevant authorization is not assigned to the operator.

In the event of errors, see Reply Format [Page 193].

user_getrights Command: Example

Displaying the rights of the DBM operator [See SAP DB Library] with the name `DBM`, and the operator password `DBM` for using the Database Manager

```
dbmcli -d myDB -u dbm,dbm  user_getrights dbm

OK

DBInfoRead              +       Request status data
ExecLoad                +       Execute the LOAD program
SystemCmd               +       Execute operating system commands
UserMgm                 +       User management
FileFull                -       File access (read and write)
FileRead                -       File access (read only)
DBFileRead              +       Database file access (read only)
Backup                  +       Saving backups
InstallMgm              +       Installation management
LoadSysTab              +       Load the system tables
ParamCheckWrite         +       Parameter access (checked write)
ParamFull               +       Parameter access (read and write)
ParamRead               +       Parameter access (read only)
DBStart                 +       Start database instance
DBStop                  +       Stop database instance
Recovery                +       Restoring backups
AccessSQL               +       Access to SQL session
AccessUtility           +       Access to utility session
```

Result of the Request

The Database Manager operator *DBM* has all operator authorizations [Page 13] for the Database Manager, with the exception of *FileFull* and *FileRead*.

Displaying Operator Data

Use

You display the data for the DBM operator [See SAP DB Library] specified in **<userid>**.

The Database Manager displays a list of the operator authorizations [Page 13] assigned to the operator. These are separated by commas.

Prerequisites

You have the DBM operator authorization UserMgm [Page 18].

Syntax

user_get <userid>

Displaying the data for the DBM operator with the name **DBM**, password **DBM**

dbmcli -d myDB -u dbm,dbm user_get dbm

```
OK
SERVERRIGHTS=DBInfoRead,ExecLoad,SystemCmd,UserMgm,DBFileRea
d,Backup,...
SECONDPASSWORD=NO
DISABLED=NO
COMMENT=
```

Reply

```
OK<NL>
SERVERRIGHTS=<right_list><NL>
SECONDPASSWORD=[YES|NO]<NL>
DISABLED=[YES|NO]<NL>
COMMENT=<comment><NL>
```

Values for the fields of the reply: See Operator Properties [Page 12]

In the event of errors, see Reply Format [Page 193].

List of DBM Operators

Use

You request the operator names of all DBM operators [See SAP DB Library] that are registered for the current database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization UserMgm [Page 18].

Syntax

user_getall

Display all DBM operators for the database instance:

dbmcli -d myDB -u dbm,dbm user_getall

```
OK
dbm
DBA
domain
```

Reply

```
OK<NL>
<userid><NL>
<userid><NL>
...
```

In the event of errors, see Reply Format [Page 193].

Deleting DBM Operators

Use

You delete the DBM operator [See SAP DB Library] specified in **<userid>**.

You cannot delete the DBM operator currently logged on, or the **first DBM operator** that was created when creating the database instance.

Prerequisites

You have the DBM operator authorization UserMgm [Page 18].

Syntax

```
user_delete <userid>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Accessing the Database Instance

Use

This group of DBM commands [Page 35] allows you to communicate with the database kernel.

Prerequisites

Take account of the operator authorizations [Page 13] required for the respective DBM commands.

To use these DBM commands you must generate and work in a utility session [Page 175], an SQL session [Page 181], or a service session [Page 183]. For more information see the documentation for the particular DBM commands.

Overview of Commands for Accessing the Database Instance: [Page 172]

- Access Using a Utility Session [Page 172]
- Access Using an SQL Session [Page 176]
- Access Using a Service Session [Page 183]

- Access to Database Events [Page 184]

Overview of Commands for Accessing the Database Instance

Activating a New Database Instance [Page 173]	`util_activate`
Activating a Database Event [Page 184]	`event_set`
Updating the Optimizer Statistics [Page 177]	`sql_updatestat`
Updating the Optimizer Statistics with the XPU Program [Page 59]	`exec_xpu`
Displaying the Structure of a Result Set [Page 177]	`sql_info`
Displaying a Physical Database Page [Page 173]	`util_getpage`
Displaying the Status Data for the Database Instance [Page 178]	`info`
Terminating a Database Event Session [Page 185]	`event_release`
Terminating a Service Session [Page 183]	`service_release`
Terminating an SQL Session [Page 179]	`sql_release`
Terminating a Utility Session [Page 174]	`util_release`
Scrolling in the Result Data [Page 180]	`sql_fetch`
Scrolling in the Status Information of the Database Instance [Page 180]	`info_next`
Deactivating a Database Event [Page 185]	`event_delete`
Opening a Service Session [Page 183]	`service_connect`
Opening an SQL Session [Page 181]	`sql_connect`
Opening a Utility Session [Page 175]	`util_connect`
List of Activated Database Events [Page 186]	`event_list`
Writing a Physical Database Page [Page 175]	`util_putpage`
Transferring an SQL Statement [Page 182]	`sql_execute`
Transferring a Utility or SQL Command [Page 176]	`util_execute`
Waiting for a Database Event [Page 187]	`event_wait`

Access Using a Utility Session

Use

For a utility session with the database instance [See SAP DB Library], you can use the following DBM commands [See SAP DB Library]:

Activating a New Database Instance [Page 173]	**util_activate**
Displaying a Physical Database Page [Page 173]	**util_getpage**
Terminating a Utility Session [Page 174]	**util_release**
Opening a Utility Session [Page 175]	**util_connect**
Writing a Physical Database Page [Page 175]	**util_putpage**
Transferring a Utility or SQL Command [Page 176]	**util_execute**

Prerequisites

You have the DBM operator authorization AccessUtility [Page 21].

Activating a New Database Instance

Use

You start a newly registered database instance [See SAP DB Library] for the first time and create the first database user [See SAP DB Library]. This user is the database administrator with the name *SYSDBA*.

See also: *User manual: SAP DB*

Prerequisites

You have the DBM operator authorization AccessUtility [Page 21].

Syntax

```
util_activate <sysdba>,<password>
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Displaying a Physical Database Page

Use

A physical page [See SAP DB Library] is fetched from the volume specified. Enter the number of the volume required in the parameter **<volno>,** and its type in **<volmode>**. The system displays the requested page, and before the <page_data> the length of the page <page_size> in bytes.

If you only specify one page number **<pageno>**, the corresponding logical page is output.

This command is only useful when being used by client applications.

Prerequisites

You have the DBM operator authorization AccessUtility [Page 21].

The database is in the ADMIN operational state.

You have started a utility session [Page 175].

Syntax

```
util_getpage [<volmode> <volno>] <pageno>
```

Option <volmode>

DATA	Data Volume [See SAP DB Library]
LOG	Log Volume [See SAP DB Library]
MLOG	Mirrored log volume (Log Mode [See SAP DB Library] **DUAL**)
NAME	If you use this option, do not specify a number for the following parameter **<volno>**; instead specify the name of the volume: **util_getpage [<NAME> <volname>] <pageno>**

Reply

```
OK<NL>
<page_size><NL>
<page_data>
```

Terminating a Utility Session

Use

With this command you end the active utility session. All assigned resources are released.

Prerequisites

You have the DBM operator authorization AccessUtility [Page 21] or Backup [Page 15].

You have opened a utility session [Page 175].

Syntax

```
util_release
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Opening a Utility Session

Use

You open a session with the database instance [See SAP DB Library] with which to use the utility task of the database kernel. This task is reserved solely for the administration of the database.

Only one utility task exists for each database instance. Administrative operations cannot therefore be performed in parallel.

If you have not specified an operator, the DBM operator [See SAP DB Library] registered on the database server will be logged on.

Prerequisites

You have the DBM operator authorization AccessUtility [Page 21] or Backup [Page 15].

You are working in script mode [Page 195] or session mode [Page 195].

Syntax

```
util_connect
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Writing a Physical Database Page

Use

The page [See SAP DB Library] is written to the specified volume.

Specify the number of the desired volume for **<volno>** and the type of the desired volume for **<volmode>**.

If you only specify one page number **<pageno>**, this is the logical page that is written to the volume.

You can define the length of the database page **(<page_data>)** using the parameter **<page_size>**.

This command is only useful when used by client applications.

Prerequisites

You have the DBM operator authorization AccessUtility [Page 21].

The database is in the ADMIN operational state.

You have started a utility session [Page 175].

Syntax

```
util_putpage [<volmode> <volno>] <pageno> <page_size><NL><page_data>
```

Option <volmode>

DATA	Data Volume [See SAP DB Library]
LOG	Log Volume [See SAP DB Library]
MLOG	Mirrored log volume (Log Mode [See SAP DB Library] `DUAL`)
NAME	When using this option, do not enter a number for the following parameter **<volno>**; enter the name of the volume instead: **util_putpage [<NAME> <volname>] <pageno>**

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Transferring a Utility or SQL Command

Use

You transfer the specified utility command or SQL statement to the database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization AccessUtility [Page 21].

You have opened a utility session [Page 175].

Syntax

util_execute <statement>

Reply

The system outputs an OK message.

If an error occurs, the system terminates the utility session and outputs an error message number. The system sets up the session again with the current parameters.

See also: *Messages: SAP DB 7.4*

Access Using an SQL Session

Use

For an SQL session with the database instance [See SAP DB Library], you can use the following DBM commands [See SAP DB Library]:

Updating the Optimizer Statistics [Page 177]	**sql_updatestat**
Displaying the Structure of a Result Set [Page 177]	**sql_info**

Displaying the Status Data for the Database Instance [Page 178]	`info`
Terminating an SQL Session [Page 179]	`sql_release`
Scrolling in the Result Data [Page 180]	`sql_fetch`
Scrolling in the Status Information of the Database Instance [Page 180]	`info_next`
Opening an SQL Session [Page 181]	`sql_connect`
Transferring an SQL Statement [Page 182]	`sql_execute`

Prerequisites

Note the operator authorizations required for the particular DBM commands [Page 35] (Operator Authorizations [Page 13]).

Updating the Optimizer Statistics

Use

You update the statistics for the Optimizer.

These statistics enable the Optimizer to choose the best strategy for retrieving data. Obsolete statistics impair the performance of the database instance.

You can update all the statistics or restrict the update function to a set of statistics using a table specification **<tablespec>**.

See also: *Optimizer: SAP DB 7.4*

Prerequisites

You have the DBM operator authorization AccessSQL [Page 20].

You have started an SQL session [Page 181].

Syntax

```
sql_updatestat [<tablespec>]
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Displaying the Structure of a Result Set

Use

You display the structure of the result set for an SQL statement.

Prerequisites

You have the DBM operator authorization AccessSQL [Page 20].

You have started an SQL session [Page 181].

Syntax

sql_info <statement>

sql_info select username,connectmode,templimit, user_id from users

```
OK
USERNAME;CONNECTMODE;TEMPLIMIT;USER_ID
```

Reply

The names of all columns are output separated by semicolons.

```
OK<NL>
<col_1>;<col_2>;...;<col_n><NL>q
```

Displaying Status Information for the Database Instance

Use

You display the information given under **<info_id>**.

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

You have started an SQL session [Page 181].

Syntax

info <info_id>

Option <info_id>

BADVOLUMES	Damaged volumes [See SAP DB Library]
BADIDX	Damaged indexes
CACHES	Accesses to the caches [See SAP DB Library]
CLSCON	liveCache [See SAP DB Library] class container (only for the liveCache instance type [See SAP DB Library])
COLUMNS	Columns of database instance [See SAP DB Library] tables
DATA	Data volumes [See SAP DB Library]
VOLUMES	Volumes
DISABLEDIDX	Deactivated indexes
EVENTS	Active events

INDEXES	Database instance indexes
INFOS	Available information
IO	Read and write operations
LOCKS	Locks [See SAP DB Library]
LOG	Log status (Log Concept [See SAP DB Library])
OMSCACHES	liveCache caches (only for liveCache instance type)
OMSMON	liveCache monitors (only for liveCache instance type)
PARAMS	Database Parameters [See SAP DB Library]
STATE	Short description of the status of the database instance
SYSDDCONFIG	Internal configuration
SYSDDMONITOR	Internal monitors
SYSDDSRVDB	Internal database status
TABLES	Database instance tables
UNUSEDIDX	Unused indexes
UPDSTAT	Update of the statistics for the Optimizer is possible (**See also:** *Optimizer: SAP DB 7.4*)
UPDSTATW	Statistics must be updated for the Optimizer
USERS	Logged on database users [See SAP DB Library]
VERSIONS	Version of the database software

Reply

```
OK<NL>
(END|CONTINUE)<NL>
<description_record><NL>
<info_record><NL>
<info_record><NL>
```

`END`	Result set output complete
`Continue`	More result data ready

The description line `<description_record>` contains the names of the fields of the following information lines `<info_record>`.

Terminating an SQL Session

Use

You exit the active SQL session. All assigned resources are released.

Prerequisites

You have the DBM operator authorization AccessSQL [Page 20] or DBInfoRead [Page 14].

You have opened an SQL session [Page 181].

Syntax

```
sql_release
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Scrolling in the Result Data

Use

You request the output of additional result data.

Prerequisites

You have the DBM operator authorization AccessSQL [Page 20].

You have opened an SQL session [Page 181] and transferred an SQL statement [Page 182]. In the response, `CONTINUE` indicated that additional result data was available.

Syntax

```
sql_fetch
```

Reply

The fields in a record are separated by semicolons.

Character strings are displayed in single quotation marks.

```
OK<NL>
(END|CONTINUE)<NL>
<record><NL>
<record><NL>
```

Values for the individual fields of the reply

`END`	The contents of the file have been transferred in full. The file is closed automatically.
`CONTINUE`	The file contains further entries that were not transferred due to the limited storage available for replies. You can display these by entering the above command again.

Scrolling in the Status Information of the Database

Instance

Use

You can display additional information on the status of the database instance [See SAP DB Library].

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

When displaying status information for the database instance [Page 178] or scrolling in the status information for the database instance, the system informs you that you can request more information using the keyword `CONTINUE`.

Syntax

```
info_next
```

Reply

```
OK<NL>
(END|CONTINUE)<NL>
<description_record><NL>
<info_record><NL>
<info_record><NL>
```

Values for the individual fields of the reply

`END`	Result set output complete
`Continue`	More result data ready
`<description_record>`	Name of the fields of the subsequent information lines
`<info_record>`	Information

Opening an SQL Session

Use

You create an SQL session for the specified DBM operator [See SAP DB Library] with the database kernel. The operator of an SQL session shares the performance of the database instance [See SAP DB Library] with up to `n` additional operators.

If you have not specified an operator, the DBM operator registered on the server is logged on.

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14] or AccessSQL [Page 20].

You are working in script mode [Page 195] or session mode [Page 195].

Syntax

```
sql_connect [<userid>,<password>]
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Transferring an SQL Statement

Use

When you enter a SELECT statement, the system displays the results. The system outputs an OK message for other statements.

Prerequisites

You have the DBM operator authorization AccessSQL [Page 20].

You have started an SQL session [Page 181].

Syntax

sql_execute <statement>

```
sql_execute SELECT username,connectmode,templimit,user_id
from users

OK
END
'PUBLIC';'SINGLE';(null);3
'DBA';'SINGLE';(null);10
'DOMAIN';'SINGLE';(null);11
'SYS';'SINGLE';(null);12
'CONTROL';'MULTIPLE';(null);0
```

Reply

```
OK<NL>
(END|CONTINUE)<NL>
<record><NL>
<record><NL>
```

The fields in an answer record are separated by semicolons.

Character strings are displayed in single quotation marks.

END	Result set output complete
Continue	More result data ready

In the event of errors, see Reply Format [Page 193].

Access Using a Service Session

Use

To use the DBM Server commands for checking backups [Page 149] and restoring parameter files from the data backup [Page 161], you must have a service session running.

For a service session with the database instance [See SAP DB Library], you can use the following DBM commands [See SAP DB Library]:

Terminating a Service Session [Page 183]	`service_release`
Opening a Service Session [Page 183]	`service_connect`

Prerequisites

You have the DBM operator authorization Backup [Page 15].

Terminating a Service Session

Use

This command ends the active service session. All assigned resources are released. When Ending the Database Manager [Page 58], this command is implicitly executed.

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You have started a service session [Page 183].

Syntax

```
service_release
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Opening a Service Session

Use

You start the appropriate service kernel for the version of the database software and open a session with the utility task.

The service kernel is a specially configured database instance [See SAP DB Library] without a dataset of its own. You can use it to carry out tasks for other database instances, such as Checking a Backup [Page 149], or Restoring the Parameter File from a Data Backup [Page 161].

Prerequisites

You have the DBM operator authorization Backup [Page 15].

You are working in script mode [Page 195] or session mode [Page 195].

Syntax

```
service_connect
```

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Access to Database Events

Use

You can use the following commands to access events of the database kernel:

Activating a Database Event [Page 184]	`event_set`
Terminating a Database Event Session [Page 185]	`event_release`
Deactivating a Database Event [Page 185]	`event_delete`
List of Activated Database Events [Page 186]	`event_list`
Waiting for a Database Event [Page 187]	`event_wait`

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

Activating a Database Event

Use

You activate a database event. In accordance with the set parameters, you can display messages about the occurrence of the activated events in the Database Manager CLI using the commands for waiting for a database event [Page 187].

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

Syntax

```
event_set <name> [<prio>] [[<value>] <value>]
```

You activate the event `BACKUP_PAGES` with the value `1000`.

```
dbmcli -d myDB -u dbm,dbm -uUTL -c event_set BACKUP_PAGES
1000
```

```
OK
```

During the running backup process, the database instance will now generate a message after every thousand backed up pages [See SAP DB Library]. You can display the generated messages using the commands for waiting for a database event [Page 187].

Options

<name>	Name of the event
<prio>	Priority of the event NIL: No priority defined MEDIUM: Medium priority HIGH: High priority
<value>	Additional value whose meaning depends on the event

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Terminating a Database Event Session

Use

When waiting for a database event [Page 187], a database instance [See SAP DB Library] session is implicitly created. You end that session with this command.

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

Syntax

event_release

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

Deactivating a Database Event

Use

You deactivate a database event. The occurrence of the event will now no longer be logged by the Database Manager [Page 1].

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

Syntax

event_delete <name> [<prio>] [[<value>] <value>]

You deactivate the event BACKUP_PAGES, for which the value 1000 was defined.

dbmcli -d myDB -u dbm,dbm -uUTL -c event_delete BACKUP_PAGES 1000

```
OK
```

Options

<name>	Name of the event
<prio>	Priority of the event **NIL**: No priority defined **MEDIUM**: Medium priority **HIGH**: High priority
<value>	Additional value whose meaning depends on the event

Reply

The system outputs an OK message.

In the event of errors, see Reply Format [Page 193].

List of the Activated Database Events

Use

You request a list of activated database events.

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

Syntax

event_list

dbmcli -d myDB -u dbm,dbm event_list

```
OK
Name             State    Prio     1st  2nd   Description
BACKUP_PAGES     OFF      NIL      0          Next <n> pages
moved
DB_ABOVE_LIMIT   ON       MEDIUM   90         Database above
limit
DB_ABOVE_LIMIT   ON       HIGH     95         Database above
limit
```

```
DB_ABOVE_LIMIT  ON      HIGH    96          Database above
limit
DB_ABOVE_LIMIT  ON      HIGH    97          Database above
limit
DB_ABOVE_LIMIT  ON      HIGH    98          Database above
limit
DB_ABOVE_LIMIT  ON      HIGH    99          Database above
limit
DB_BELOW_LIMIT  OFF     NIL     0           Database below
limit
DW_COMPRESS     OFF     NIL                 Compress device
starts
DW_DISPLACE     OFF     NIL                 Displace device
starts
LOG_ABOVE_LIMIT OFF     NIL     0           Log above limit
```

Reply

```
OK<NL>
Name        State   Prio    1st     2nd     Description<NL>
<name>      <state> <prio>  <value> <value> <description><NL>
<name>      <state> <prio>  <value> <value> <description><NL>
...
```

Values in the individual fields of the reply

<name>	Name of the event
<state>	Status of the event ON: Event is activated OFF: Event is not activated The database kernel reports the occurrence of an activated event while Waiting for a Database Event [Page 187].
<prio>	Priority of the event NIL: No priority defined MEDIUM: Medium priority HIGH: High priority
<value>	Additional value whose meaning depends on the event
<description>	Description of the event

In the event of errors, see Reply Format [Page 193].

Waiting for a Database Event

Use

You request notification from the Database Manager about the occurrence of an activated event and the data associated with it.

Once you have received the notification, you evaluate it, and wait for the next notification from the Database Manager of the occurrence of the activated event by entering the command again.

You only receive a reply to this command when the activated event occurs.

Use this command through a Database Manager programming interface.

If you want to use the command with the Database Manager CLI, work in session mode [Page 195].

Prerequisites

You have the DBM operator authorization DBInfoRead [Page 14].

You have activated a database event (Activating a Database Event [Page 184]).

Syntax

event_wait

```
dbmcli -d myDB -u dbm,dbm
dbmcli on myDB>event_wait
OK
NAME=BACKUP_PAGES
PRIORITY=LOW
DATE=20010712
TIME=00151725
VALUE1=99
COUNT=188
DESCRIPTION=Next <n> pages moved
```

You receive notification about the occurrence of the event `BACKUP_PAGES`.

Overview of All DBM Commands

archive_stage	Creating a Single Archive of the Version Files [Page 124]
archive_stage_repeat	Creating Multiple Archives of the Version Files [Page 126]
autolog_cancel	Canceling the Automatic Log Backup [Page 121]
autolog_off	Deactivating the Automatic Log Backup [Page 128]
autolog_on	Activating the Automatic Log Backup [Page 129]
autolog_show	Displaying the Automatic Log Backup Function [Page 121]
backup_cancel	Terminating an Interrupted Backup [Page 128]
backup_ext_ids_forget	Releasing the Memory Occupied by the External Backup IDs [Page 143]
backup_ext_ids_get	Fetching External Backup IDs [Page 143]

backup_ext_ids_list	Displaying External Backup IDs [Page 141]
backup_ext_ids_listnext	Scrolling in the External Backup IDs [Page 142]
backup_history_close	Closing the Backup History [Page 140]
backup_history_date	Change Date of the Backup History [Page 136]
backup_history_list	Displaying the Backup History [Page 137]
backup_history_listnext	Scrolling in the Backup History [Page 139]
backup_history_open	Fetching the Backup History [Page 140]
backup_ignore	Continuing the Backup Without Last Known Full Backup Medium [Page 129]
backup_replace	Backing Up to a Succeeding Medium [Page 130]
backup_start	Backing Up the Database Instance [Page 132]
backup_state	Displaying the Current Backup Status [Page 122]
bye	Ending the Database Manager [Page 58]
db_addvolume	Adding a Volume [Page 104]
db_admin	Transferring to the ADMIN Operational State [Page 70]
db_clear	Deleting Runtime Information After a Database Error [Page 66]
db_cons	Using the Database Console [Page 62]
db_create	Create a Database Instance [Page 78]
db_drop	Deleting the Current Database Instance [Page 76]
db_enum	List of All Registered Database Instances [Page 74]
db_offline	Transferring to the OFFLINE Operational State [Page 71]
db_online	Transferring to the ONLINE Operational State [Page 71]
db_reg	Registering a Variant of the Current Database Instance [Page 82]
db_reg -R	Changing the Software Version of the Current Database Instance [Page 73]
db_restart	Restarting the Database Instance [Page 66]
db_restartinfo	Displaying Current Information About the Database Instance [Page 153]
db_speed	Displaying the Database Kernel Variant [Page 60]
db_start	Starting the Database Instance [Page 67]
db_state	Displaying the Operational State of the Database Instance [Page 60]
db_stop	Stopping the Database Instance [Page 69]
db_unreg	Deleting the Registration of a Variant of the Current Database Instance [Page 77]
dban_start	Calling the Database Analyzer [Page 42]
dban_state	Determining the Status of the Database Analyzer [Page 46]

dban_stop	Ending the Database Analyzer [Page 44]
dbm_configget	Displaying a Configuration Parameter of the Database Manager [Page 84]
dbm_configset	Changing a Configuration Parameter of the Database Manager [Page 84]
dbm_getpath	Displaying the Version-Independent Directories [Page 73]
dbm_setpath	Defining the Version-Independent Directories [Page 74]
dbm_version	Displaying the Version Information of the Database Manager [Page 56]
diag_histlist	Displaying the Diagnosis History [Page 48]
diag_pack	Packing Database Files [Page 53]
event_delete	Deactivating a Database Event [Page 185]
event_list	List of Activated Database Events [Page 186]
event_release	Terminating a Database Event Session [Page 185]
event_set	Activating a Database Event [Page 184]
event_wait	Waiting for a Database Event [Page 187]
exec_lcinit	Executing the liveCache Initialization Script [Page 57]
exec_load	Starting the PythonLOAD Program [Page 68]
exec_xload	Starting the LOAD Program [Page 68]
exec_xpu	Updating the Optimizer Statistics with the XPU Program [Page 59]
exit	Ending the Database Manager [Page 58]
file_backup	Backing Up a Database File [Page 54]
file_getfirst	Opening a Database File [Page 52]
file_getlist	List of the Database Files [Page 51]
file_getnext	Scrolling Through a Database File [Page 50]
file_operation	Editing Log Files of the Database Instance [Page 49]
file_restore	Restoring a Database File [Page 55]
help	List of DBM Commands [Page 112]
info	Displaying Status Information for the Database Instance [Page 178]
info_next	Scrolling in the Status Information of the Database Instance [Page 180]
inst_enum	List of All Registered Versions of the Database Software [Page 75]
inst_reg	Registering a Version of the Database Software [Page 83]
inst_unreg	Deleting the Registration of a Version of the Database Software [Page 78]
load_r3tab	Loading SAP-Specific Tables [Page 65]
load_systab	Loading the System Tables [Page 64]

medium_date	Change Date of the Media Definition File [Page 114]
medium_delete	Deleting a Backup Medium [Page 119]
medium_get	Displaying the Media Data [Page 117]
medium_getall	List of All Defined Backup Media [Page 118]
medium_label	Displaying the Backup Information in the Operational State ONLINE or ADMIN [Page 146]
medium_labeloffline	Displaying the Backup Information in the Operational State OFFLINE [Page 144]
medium_migrate	Copying an Existing Media Definition [Page 120]
medium_put	Defining or Changing a Backup Medium [Page 115]
param_abortsession	Terminating a Parameter Session [Page 88]
param_addvolume	Changing Volume Parameters [Page 89]
param_checkall	Checking All Database Parameters [Page 111]
param_commitsession	Confirming Changes to the Database Parameter File [Page 100]
param_copy	Copying a Database Parameter File [Page 105]
param_delvolume	Deleting Volume Parameters [Page 108]
param_directdel	Deleting a Database Parameter [Page 107]
param_directget	Displaying a Value from the Database Parameter File [Page 99]
param_directgetall	Displaying All Parameters of the Current Database Parameter File [Page 92]
param_directput	Directly Changing a Value in the Database Parameter File [Page 102]
param_extget	Displaying Individual Data for a Database Parameter [Page 99]
param_extgetall	Displaying the Data for All Database Parameters [Page 93]
param_getdefault	Displaying the System Default Value [Page 98]
param_getexplain	Displaying the Stored Explanatory Text [Page 97]
param_getfull	Displaying All Data for a Database Parameter [Page 90]
param_gethelp	Displaying the Stored Help Text [Page 98]
param_gethistory	Opening the Database Parameter History [Page 109]
param_gethistorynext	Scrolling in the Database Parameter History [Page 101]
param_getproperties	Displaying All Properties of a Database Parameter [Page 92]
param_gettype	Displaying the Data Type of a Database Parameter [Page 97]
param_getvalue	Displaying the Current Value of a Database Parameter [Page 96]
param_getvolsall	Displaying the Data for All Volume Parameters [Page 94]

param_getvolume	Displaying the Data for Individual Volume Parameters [Page 95]
param_init	Initializing the Database Parameters for a New Database Instance [Page 104]
param_put	Changing the Value of a Database Parameter [Page 88]
param_putconfirm	Correcting Database Parameters [Page 106]
param_restore	Resetting the Parameter File to a Previous Version [Page 112]
param_rmfile	Deleting the Database Parameter File [Page 107]
param_startsession	Opening a Parameter Session [Page 103]
param_versions	List of Available Parameter Files [Page 106]
quit	Ending the Database Manager [Page 58]
recover_cancel	Terminating an Interrupted Restore or Backup Check [Page 156]
recover_check	Checking a Backup [Page 149]
recover_config	Restoring the Parameter File from a Data Backup [Page 161]
recover_ignore	Continuing a Parallel Restore or Backup Check [Page 156]
recover_replace	Restoring or Checking a Backup with Succeeding Medium [Page 164]
recover_start	Restoring the Database Instance [Page 158]
recover_state	Displaying the Current Recovery Status [Page 154]
recover_state_check	Displaying the Current Status of a Backup Check [Page 148]
release	Ending the Database Manager [Page 58]
service_connect	Opening a Service Session [Page 183]
service_release	Terminating a Service Session [Page 183]
show	Displaying Information About the Database Instance [Page 61]
show_list	List of the Information About the Database Instance [Page 65]
show_next	Scrolling in the Information About the Database Instance [Page 63]
sql_connect	Opening an SQL Session [Page 181]
sql_execute	Transferring an SQL Statement [Page 182]
sql_fetch	Scrolling in the Result Data [Page 180]
sql_info	Displaying the Structure of a Result Set [Page 177]
sql_recreateindex	Recreating a Damaged Index [Page 163]
sql_release	Terminating an SQL Session [Page 179]
sql_updatestat	Updating the Optimizer Statistics [Page 177]
trace_off	Deactivating the Database Trace [Page 39]

trace_prot	Creating the Text Version of the Database Trace [Page 41]
trace_protopt	Displaying Options for the Text Version of the Database Trace [Page 37]
trace_show	Displaying the Logged Activities of the Database Trace [Page 38]
user_create	Creating DBM Operators [Page 168]
user_delete	Deleting DBM Operators [Page 171]
user_get	Displaying Operator Data [Page 169]
user_getall	List of DBM Operators [Page 170]
user_getrights	Displaying Operator Authorizations [Page 168]
user_put	Changing DBM Operator Data [Page 167]
util_activate	Activating a New Database Instance [Page 173]
util_connect	Opening a Utility Session [Page 175]
util_execute	Transferring a Utility or SQL Command [Page 176]
util_getpage	Displaying a Physical Database Page [Page 173]
util_putpage	Writing a Physical Database Page [Page 175]
util_release	Terminating a Utility Session [Page 174]

Reply Format

Normally communication is in ASCII format. Only in special cases, such as binary data access, do the request and reply also contain binary data.

This ensures that the replies on the client side can be easily read. It also largely avoids dependence on the hardware architecture.

Successful Reply

If the request is successful, an OK message is given in the following format:

```
OK[,<description>]<NL>
[<answer>]
```

The character string `OK` is the first token of the reply. There is an optional explanatory text on the same line. The meta character `<NL>` stands for a line feed.

The reply data is provided after the line feed. The structure of the reply data depends on the concrete DBM command [Page 35].

Error Message

```
ERR[,<description>]<NL>
[<errcode>,<errdescription><NL>
[<subcode>,<subdescription><NL>]
[<extended_description><NL>]]
```

An error always starts with the character string `ERR`. The same line may also contain a description.

The number of the error (<errcode>) and a relevant text (<errdescription>) can be included on the next line
(**See also:** *Messages: SAP DB 7.4*).

If the system has the error number and relevant text of a subcomponent (for example, runtime environment or operating system), these are transferred as <subcode> and <subdescription>.

Possible additional information follows in the <extended_description>. Special cases in which the <extended description> has a structured format are described with the relevant DBM commands.

Internal Error of the Database Manager:

```
ERR
-24979,ERR_XPNOTFOUND_CN00 : parameter not found
```

Error in the Runtime Environment:

```
ERR
-24994,ERR_RTE_CN00 : runtime environment error
3,cannot access PARAM file [32]
```

Error when analyzing the description file (see also: Configuring Database Instances [Page 85]):

```
ERR
-24978,ERR_XPSYNTAX_CN00 : xparam syntax error
16,XPERR_IVSEQ_CN21 : error in IF-ELSIF-ELSE sequence
D:\d628\usr\env\cserv.pcf(60) :     ELSE
```

Operating Modes

The Database Manager CLI [Page 1] program provides three operating modes:

- Command Mode [Page 194]
- Script mode [Page 195]
- Session Mode [Page 195]

All modes are implicitly activated by the way the Database Manager CLI is called.

Command Mode

The Database Manager CLI [Page 1] works in command mode if a DBM command [Page 35] is specified when the program is called.

The command is executed. The session and the Database Manager CLI program then end.

Script Mode

Use

Use this command to activate the script mode.

Specify the name of the input script [Page 33] **<file_name>** that contains the DBM commands [Page 35] that are to be processed by the Database Manager.

Use the following input options in the script:

Local System Call: ! [Page 24]
Comment: # [Page 24]
Linking Lines: / [Page 24]
Linking Lines: < [Page 24]

Syntax

```
dbmcli -i <file_name>
```

Session Mode

The Database Manager CLI works in the session mode if no DBM command [Page 35] and no input script are specified.

Within a session of the Database Manager CLI you can execute several commands.

Use the following input methods for this:

Local System Call: ! [Page 24]
Comment: # [Page 24]
Linking Lines: / [Page 24]
Linking Lines: < [Page 24]

To end the session mode, enter **release**.

Database Manager Files

For each database instance, the Database Manager creates a range of files that are stored in different locations in the directory structure, depending on the file type.

See also:

Directory Structure of the Database System for SAP Systems

Directory Structure of the Database System for Open Source Systems

There are two types of Database Manager file:

Log Files of the Database Manager [Page 196]

Configuration Files of the Database Manager [Page 196]

Database Manager Log Files

Specify the file ID and/or the file class when packing the diagnosis package `diagpkg.tgz` (Packing Database Files [Page 53]).

General Log Files (File Class `protocol`)

File Content	File Name	File ID
DBM Server log	`dbm.prt`	`DBMPRT`
Log of the utility command	`dbm.utl`	`UTLPRT`
Log of the system table load	`dbm.ins`	`INSTPRT`
Log of database administration activities	`dbahist.prt`	`DBAHIST`
Log of the system table load	`dbm.ins`	`INSTPRT`

Log Files for Backups and Recoveries (File Class `backup`)

File Content	File Name	File ID
History of backups and recoveries	`dbm.knl`	`BACKHIST`
History of the media used for backups	`dbm.mdf`	`BACKMDF`
History of backups with external backup tools	`dbm.ebf`	`BACKEBF`
Log of last backup with an external backup tool	`dbm.ebp`	`BACKEBP`

For information on the location of files in the database system, see also:

Directory Structure of the Database System for SAP Systems [See SAP DB Library] or

Directory Structure of the Database System for Open Source Systems [See SAP DB Library]

Database Manager Configuration Files

The configuration files of the Database Manager are located in the directory `<independent_data_path>/wrk/<database_name>`.

Configuration files have the file class `config`.

File Content	File Name	File ID
Configuration file of the Database Manager Created by the system for each database instance	`dbm.cfg`	-

Database Manager backup media	dbm.mmm	DBMMDF

Displaying a Configuration Parameter of the Database Manager [Page 84]
Changing a Configuration Parameter of the Database Manager [Page 84]