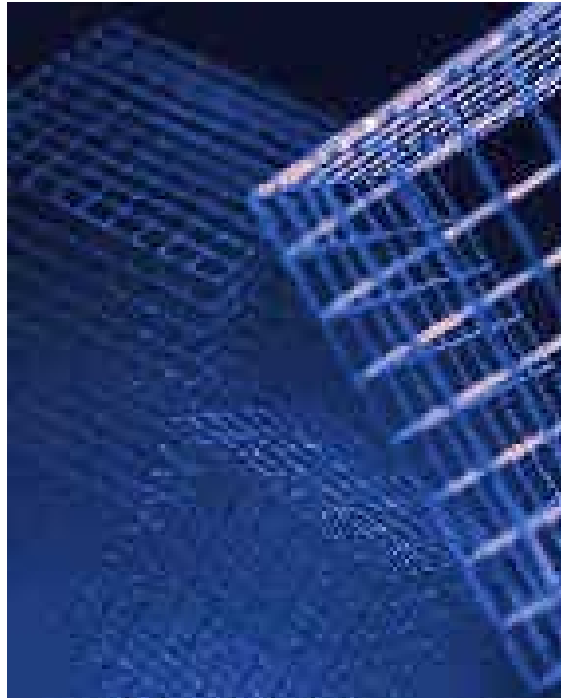


Creating the Database Manager CLI



Version 7.3









Copyright

© Copyright 2001 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
	Tip

Contents

Copyright	2
Icons	3
Content	4
Using the Database Manager CLI	5
Structure of the Database Manager CLI	6
Main Components	6
Communication Components	6
DBM Server	6
Creating the Database Manager CLI	7
Procedure	7
Link Description File dbmcli.lnk	8
Components of the Database Manager CLI	9
dbmcli.rc	9
vcn12.cpp, vcn13.cpp, vcn14.c	9
Link Description cservlib	10
Link Description splib	11
Link Description eoxlib	11
Link Descriptions sqlusr, enalib, enlib	11
Further Dependencies	11
Function Check	12
Procedure	12
Result	12

Using the Database Manager CLI

The Database Manager CLI (Database Manager Command Line Interface, DBMCLI) communicates with the DBM server and can, therefore, be used to administrate an SAP DB instance. The behavior of the Database Manager CLI depends on the call parameters and user entries.

Database Manager CLI tasks:

- The Database Manager CLI establishes the connection to the DBM server, and transfers one or more commands to it.
- The Database Manager CLI accepts and outputs the DBM server response(s).
- When the Database Manager CLI has been terminated, the connection to the DBM server is released.

Structure of the Database Manager CLI

The Database Manager CLI consists of the following components:

- Main components
- Communication components
- DBM server

Main Components

Tasks of the main components:

- Accept and evaluate the call parameters and user entries.
- Transfer the DBM server commands to the communication components.
- Receive the DBM server responses from the communication components.
- Evaluate and output the DBM server responses.

Communication Components

Tasks of the communication components:

- Communication with the local DBM server.
- Communication with a DBM server on a remote host.

The communication components are part of the SAP DB runtime environment (RTE).

DBM Server

A DBM server is integrated in the Database Manager CLI.

If the user requires, this integrated DBM server can be used. In this case, the components required to connect to a separate DBM server are not used.

Creating the Database Manager CLI

The Database Manager CLI is created in the development environment by processing a description file of the `lnk` file type, namely `dbmcli.lnk` (`$OWN/sys/desc/dbmcli.lnk`).

When a description file of the `lnk` file type is processed, an executable program is created. The link description file `dbmcli.lnk` also defines where the executable program is stored:

- Windows NT/Windows 2000: The executable program is stored in directory `$OWN\usr\pgm`.
- UNIX: The line `->$INSTROOT/bin/dbmcli` is inserted in the link description file `dbmcli.lnk`. As a result, the executable program is stored in the directory `$OWN/usr/bin`.

Procedure

Enter the following command:

```
imf dbmcli
```

This command instructs the development environment to process link description file `dbmcli.lnk`.

The link description file does not have to be specified with its type (`lnk`), since its name is unique. The development environment automatically determines the type of the link description file, and, therefore, the type of processing.

The `dbmcli` (Database Manager CLI) software component is created:

- UNIX: `$OWN/usr/bin/dbmcli`
- Windows NT / Windows 2000: `$OWN\usr\pgm\dbmcli.exe`

Link Description File `dbmcli.lnk`

The following extract from link description file `dbmcli.lnk` contains the main commands required to create the Database Manager CLI. Special commands and switches, which may be needed for the link operation on the individual platforms, have been omitted from this example.

Note

To generate the Database Manager CLI, you must always use the complete link description file `dbmcli.lnk` in the source directory.

```
&if $OS in [ WIN32, OS2 ]
    :Resource/dbmcli.rc remake
&endif
vcn12.cpp
vcn13.cpp
vcn14.c
cservlib
splib
eoxlib
&if $OS in [ WIN32, OS2 ]
    sqlusr
&else
    enalib
    enplib
&endif
```

Explanation of the Link Description File

The link description file `dbmcli.lnk` contains references to module files and link descriptions.

- References to module files cause these module files to be translated. The translation tools are assigned on the basis of the file extension of the module files.
Module files in this example:
`dbmcli.rc`, `vcn12.cpp`, `vcn13.cpp`, `vcn14.c`
- References to link descriptions cause these descriptions to be processed in accordance with their description type.
Link descriptions in this example:
`cservlib`, `splib`, `eoxlib`, `sqlusr`, `enalib`, `enplib`

You can find explanations of the module files and link descriptions in the section *Components of the Database Manager CLI*.

Components of the Database Manager CLI

The Database Manager CLI consists of the following components:

- Module file `dbmcli.rc`.
- Module file `vcn12.cpp`.
- Module file `vcn13.cpp`.
- Module file `vcn14.c`.
- Link description `cservlib`
- Link description `splib`
- Link description `eaxlib`
- Link description `sqlusr`
- Link description `enalib`
- Link description `enblib`

`dbmcli.rc`

The module file `dbmcli.rc` is only relevant for Windows NT / Windows 2000. For this reason, the module is encapsulated in a condition in the link description file `dbmcli.lnk`.

The module file `dbmcli.rc` is a resource file and is stored in the subdirectory `$OWN\sys\src\Resource`. If the module file `dbmcli.rc` is specified in the link description file `dbmcli.lnk`, the `Resource` directory must also be specified explicitly:

```
:Resource/dbmcli.rc
```

The module file `dbmcli.rc` contains source code that is specific to Windows NT/Windows 2000. When this file is specified in the link description file, the relevant resources are integrated in the Database Manager CLI.

`vcn12.cpp, vcn13.cpp, vcn14.c`

The module files `vcn12.cpp`, `vcn13.cpp`, and `vcn14.c` are stored in the directory `$OWN/sys/src/cn`. This directory name is determined implicitly from the module file name. These module files are the main components of the Database Manager CLI, and contain C or C++ source codes.

- The `main` function is stored in module file `vcn12.cpp`.
- The module file `vcn13.cpp` contains certain functions required for file access and character set conversion.
- The module file `vcn14.c` is used to connect main components with communication components.

Compilation Description

In the module files `vcn12.cpp`, `vcn13.cpp` and `vcn14.c`, `#include` instructions are used to refer to include files.

The dependency of a module file on an include file is described in the compilation description. The compilation description appropriate to the module file is determined and evaluated implicitly using the directory in which the module file to be translated is stored.

In addition to the dependency of a module file on an include file, the compilation description specifies further properties of the module files that control how they are translated by the development environment.

Include Files

The dependencies of a module file on include files is explained in the following example:

Example

The dependencies for the module file `vcn12.cpp` in the directory `$OWN/sys/src/cn` are contained in compilation description `$OWN/sys/desc/cn.com`. These are described by the following lines in the compilation description:

```
vcn12.cpp inc=gsp09.h,heo02.h,hcn13.h,hcn14.h,hcn90.h,...
```

This example only contains the keyword `inc`. This keyword refers to the required include files.

The include files listed after the keyword `inc` are created by the development environment in include directory `$OWN/wrk/incl` before the module file is translated. In doing so, the development environment processes the include files according to their file type.

Only type `h` include files are shown in the example above. Type `h` include files are copied from the relevant subdirectory under `$OWN/sys/src` to a subdirectory of the include directory. The name of this subdirectory is determined implicitly by the development environment from the name of the include file.

Link Description `cservlib`

The link description `cservlib` (`$OWN/sys/desc/cservlib.lib`) is of the type `lib`. The link description type `lib` generates a static library.

A `lib` type link description consists of several module files that are grouped after translation to form the appropriate library. The dependencies and properties of the individual module files are stored in the directory-specific compilation descriptions.

If `cservlib` is entered in link description file `dbmcli.lnk`, the `cservlib` library is integrated in the Database Manager CLI.

The `cservlib` library contains the functionality of the DBM server integrated in the Database Manager CLI.

Link Description splib

The link description **splib** (`$OWN/sys/desc/splib.lib`) is of the type **lib**.
The **splib** library contains modules with functions that can be used generally.

Link Description eoxlib

The link description **eoxlib** (`$OWN/sys/desc/eoxlib.lib`) is of the type **lib**.
The **eoxlib** library contains communication components of the SAP DB runtime environment.

Link Descriptions sqlusr, enalib, enlib

The link descriptions **sqlusr**, **enalib**, **enlib** (`$OWN/sys/desc/sqlusr.lib`,
`$OWN/sys/desc/enalib.lib`, `$OWN/sys/desc/enlib.lib`) are of the type **lib**.
The **sqlusr**, **enalib**, **enlib** libraries contain communication components of the SAP DB runtime environment. Since the SAP DB runtime environment is platform specific, different libraries are used on Windows NT/Windows 2000 and UNIX. For this reason, the link description file is encapsulated in a condition in the link description file `dbmcli.lnk`.

Further Dependencies

If you are using Windows NT or Windows 2000, note that the Database Manager CLI requires a dynamic library. This library must be created explicitly.

To do so, enter the following command:

```
imf sqltcp
```

Function Check

You can check the function of the Database Manager CLI by executing a DBM server command.

Procedure

1. Check whether you can actually access the software component you have just created, the Database Manager CLI (`$OWN/usr/bin/dbmcli`, Windows NT/Windows 2000: `$OWN\usr\pgm\dbmcli.exe`).
2. Enter the DBM server command `dbm_version`:

```
dbmcli [-n <server_node>] -d <database_name> -u  
<dbm_userid>,<dbm_password> dbm_version
```

Example

```
dbmcli -d TST -u dbm,dbm dbm_version
```

Result

The output should roughly be as follows:

```
OK  
VERSION      = 7.2.4  
BUILD        = DBMServer 7.2.4      Build 009-000-220-192  
OS           = UNIX  
INSTROOT     = /usr/sapdb-srv  
LOGON        = False  
CODE         = ASCII  
SWAP         = full
```