

**NAME**

*dbm* – general *dbm*(3) database management tool

**SYNOPSIS**

**dbm** [ **-AILNRSU** ] [ **-d** *dbm\_file* ] [ **-m** *mode* ] [ **-o** *output\_file* ] *command* [ *args* ... ]

**DESCRIPTION**

*Dbm* is used to manage *dbm*(3) type databases. Its function is controlled by the *dbm command* given on the command line, possibly with additional arguments. Its typical usage is to load a *dbm* database from a input file or to dump it to a readable format. It may also be used to probe for selected keys or add specific key/value pairs. In addition, *dbm* provides parsing routines for an extended textual format, suitable for building tables or various kinds. The format is further described in the section about the **parse** command below.

**OPTIONS**

- A** Append mode, don't automatically clear the database on the **load** and **make** operations.
- I** Insert mode; will supply the DBM\_INSERT flag to all *dbm\_store* operations. This means that if two entries with equal keys are given, only the first will actually be entered to the database (default) and no warning will be given.
- L** Lowercase mode. Change all keys to lowercase before reading from or writing to the database.
- R** Replace mode; will supply the DBM\_REPLACE flag to all *dbm\_store* operations. This means that if two entries with equal keys are given, only the last will actually be entered to the database and no warning messages are given.
- S** Add a @@@ senteniel after the last entry has been written to the database.
- U** Uppercase mode. Turn all keys to uppercase before reading from or writing to the database.
- d** *dbm\_file*  
Perform all operations on the named database file. If no **-d** option is given, the last argument after the *command* will be used as the *dbm\_file*.
- m** *mode*  
Use the given *mode* when creating new databases.
- o** *output\_file*  
Send all output from the **dump** and **parse** operations to the named output file instead of stdout.

**COMMANDS**

- clear** Creates an empty *dbm* database, either by clearing an old one or by creating a new.
- delete** *key* [ ... ]  
Removes entries with the specified keys from the database.
- dump** Dumps the *dbm* database to stdout (or to *output\_file*, if the **-o** option is used). The output will consist of one entry per line with a tab between each key and value.
- fetch** *key* [ ... ]  
**Fetch** will search for the *key* in the database and print both key and value in **dump** format if found on the standard output. Non-existing *keys* will be signalled by a [NOT\_FOUND] message.
- load** [*file* ... ]  
Load the database with entries from the specified *files*. If no *files* are given or if a file is specified as '-', the database will be loaded from standard input. Each line of the file should have a key and value separated by a tab. (Incidentally, this is the same format as **dump** and *pathalias*(1) will produce.) The database is first cleared unless the append (**-A**) switch has been given.
- make** [*file* ... ]  
**Make** combines the operations of **parse** and **load** (q.v.), by storing each record after it has been parsed.

**parse** [*file* ...]

This command will parse the contents of the specified *files* (or stdin if no *files* are given or when a file is '-'), according to the following syntax:

value key key ...

Whitespace delimit tokens and sharp signs (#) anywhere on a line begins comments unless any of them are quoted by a backslash (\) or put inside double quotes (" ... ") or angle brackets (<...>).

Lines beginning with whitespace are considered to be continuations of the previous line.

**store** *key value* [ *key value* ...]

Store one or more key/value pairs explicitly mentioned on the command line.

**EXAMPLES**

dbm -d foo clear	<i>create the database foo</i>
cat infile   dbm load foo	<i>load it from the infile</i>
dbm parse xfile   dbm -AI load foo	<i>add keys from the xfile...</i>
<i>(or, shorter)</i>	<i>...not already present...</i>
dbm -I make xfile foo	<i>...in the database</i>
dbm fetch keya keyb foo	<i>fetch values for the keys</i>
dbm -R store keyc valuec foo	<i>overwrite previous value for keyc</i>

**AUTHOR**

Lennart Lovstrand <l@ida.liu.se>  
CIS Dept, Univ of Linköping, Sweden

**SEE ALSO**

*pathalias(1), dbm(3), ndbm(3)*

**BUGS**

Should probably remove the sentinel when opening the database for write access (provided that the **-S** flag has been given).

Requires *ndbm(3)* or *mdbm(3)* support for no real reason.