NAME

db_dump − the DB database dump utility

SYNOPSIS

db_dump [**-d**] [**-f output**] db_file
db_dump [**-p**] [**-f output**] [**-h home**] db_file
db_dump185 [**-p**] [**-f output**] db_file

DESCRIPTION

The *db_dump* utility reads the database file *db_file* and writes it to the standard output using a portable flat-text format understood by the *db_load*(1) utility. The argument *db_file* must be a file produced using the *DB* library functions.

The *db_dump185* utility is similar to the *db_dump* utility except that it reads databases in the format used by DB version 1.85 and earlier.

The options are as follows:

−d Dump the specified database in a format helpful for debugging the DB library routines. **The output format of the −d option is not standard and may change, without notice, between releases of the DB library.**

−f Write to the specified *file* instead of to the standard output.

−h Specify a home directory for the database. As DB versions before 2.0 did not support the concept of a database ''home'', *db_dump185* does not support this option.

−p If characters in either the key or data items are printing characters (as defined by *isprint*(3)), use printing characters in *file* to represent them. This option permits users to use standard text editors to modify the contents of databases.

Note, different systems may have different notions as to what characters are ''printing'', and databases dumped in this manner may be less portable to external systems.

Dumping and reloading hash databases that use user-defined hash functions will result in new databases that use the default hash function. While using the default hash function may not be optimal for the new database, it will continue to work correctly.

Dumping and reloading btree databases that use user-defined prefix or comparison functions will result in new databases that use the default prefix and comparison functions. **In this case, it is quite likely that the database will be damaged beyond repair permitting neither record storage or retrieval.**

The only available workaround for either case is to modify the sources for the *db_load*(1) utility to load the database using the correct hash, prefix and comparison functions.

The *db_dump185* utility may not be available on your system as it is not always installed when the DB libraries and utilities are installed. If you are unable to find it, see your system administrator for further information.

The *db_dump* utility exits 0 on success, and >0 if an error occurs. The *db_dump185* utility exits 0 on success, and >0 if an error occurs.

OUTPUT FORMATS

There are two output formats used by *db_dump* and *db_dump185*.

In both output formats, the first few lines of the output contain header information describing the underlying access method, filesystem page size and other bookkeeping information. This information is output in ''name=value'' pairs, where ''name'' may be any of the keywords listed in the *db_load*(1) manual page, and ''value'' will be its value. While this header information can be edited before the database is reloaded, there is rarely any reason to do so, as all of this information can be overridden by command-line arguments to *db_load*.

Following the header information are the key/data pairs from the database. If the database being dumped is of type **btree** or **hash**, the output will be paired lines of text, where the first line of the pair is the key item, and the second line of the pair is its corresponding data item. If the database being dumped is of type **recno**, the output will be lines of text, where each line is a new data item for the database.

If the **−p** option was specified, each output line will consist of single characters representing any characters from the database that were ''printing'', and backslash (''\'') escaped characters for any that were not. Backslash characters appearing in the output mean one of two things: if the backslash character precedes another backslash character, it means that a literal backslash character occurred in the key or data item. If the backslash character precedes any other character, the next two characters should be interpreted as hexadecimal specification of a single character, e.g., “\0a” is a newline character in the ASCII character set.

If the **−p** option was not specified, each output line will consist of paired hexadecimal values, e.g., the line “726f6f74” is the string “root” in the ASCII character set.

In both output formats, a single newline character ends both the key and data items.

ENVIRONMENT VARIABLES

The following environment variables affect the execution of *db_dump*:

DB_HOME

If the **−h** option is not specified and the environment variable *DB_HOME* is set, it is used as the path of the database home, as described in *db_appinit*(3).

SEE ALSO

isprint(3)

The DB library is a family of groups of functions that provides a modular programming interface to transactions and record-oriented file access. The library includes support for transactions, locking, logging and file page caching, as well as various indexed access methods. Many of the functional groups (e.g., the file page caching functions) are useful independent of the other DB functions, although some functional groups are explicitly based on other functional groups (e.g., transactions and logging). For a general description of the DB package, see *db_intro*(3).

db_archive(1), *db_checkpoint*(1), *db_deadlock*(1), *db_dump*(1), *db_load*(1), *db_recover*(1), *db_stat*(1), *db_intro*(3), *db_appinit*(3), *db_cursor*(3), *db_dbm*(3), *db_internal*(3), *db_lock*(3), *db_log*(3), *db_mpool*(3), *db_open*(3), *db_thread*(3), *db_txn*(3)