NAME

db_load − the DB database loader

SYNOPSIS

db_load [**-T**]
[**-f file**] [**-h home**] [**-t btree | hash | recno**] db_file

DESCRIPTION

The *db_load* utility reads from the standard input and loads it into the database *db_file*. The input to *db_load* must have been created by the *db_dump* or *db_dump185* utilities (see *db_dump*(1) for more information).

The options are as follows:

−c Specify configuration options for the DB_INFO structure provided to *db_open*(3), ignoring any value they may have based on the input. The command-line format is ‘‘name=value’’. Supported keywords are listed below.

−f Read from the specified *input* file instead of from the standard input.

−h Specify a home directory for the database.

−T The **−T** option allows non-DB applications to easily load text files into databases.

If the database to be created is of type **btree** or **hash**, the input must 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 to be created is of type **recno**, the input must be lines of text, where each line is a new data item for the database.

A simple escape mechanism, where newline and backslash (‘‘\’’) characters are special, is applied to the text input. Newline characters are interpreted as record separators. Backslash characters in the text will be interpreted in one of two ways: if the backslash character precedes another backslash character, the pair will be interpreted as a literal backslash. If the backslash character precedes any other character, the two characters following the backslash will be interpreted as hexadecimal specification of a single character, e.g., ‘‘\0a’’ is a newline character in the ASCII character set.

For this reason, any backslash or newline characters that naturally occur in the text input must be escaped to avoid misinterpretation by *db_load*.

If the **−T** option is specified, the underlying access method type must be specified using the **−t** option.

−t Specify the underlying access method. If no **−t** option is specified, the database will be loaded into a database of the same type as was dumped, e.g., a hash database will be created if a hash database was dumped.

Btree and hash databases may be converted from one to the other. Recno databases may not be converted to any other database type or from any other database type.

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

KEYWORDS

The following keywords are supported for the **−c** command-line option. See *db_open*(3) for further discussion of these keywords and what values should be specified.

The parenthetical listing specifies how the value part of the ‘‘name=value’’ pair is interpreted. Items listed as (boolean) expect value to be ‘‘1’’ (set) or ‘‘0’’ (unset). Items listed as (number) convert value to a number. Items listed as (string) use the characters of value directly.

bt_minkey (number)
The minimum number of keys per page.

db_lorder (number)
The byte order for integers in the stored database metadata.

db_pagesize (number)
: The size of pages used for nodes in the tree, in bytes.

duplicates (boolean)
: The value of the DB_DUP flag.

h_ffactor (number)
: The density within the hash table.

h_nelem (number)
: The size of the hash table.

re_len (number)
: Specify fixed-length records of the specified length.

re_pad (string)
: Specify the fixed-length record pad character.

recnum (boolean)
: The value of the DB_RECNUM flag.

renumber (boolean)
: The value of the DB_RENUMBER flag.

EXAMPLES

The *db_load* utility can be used to load text files into databases. For example, the following command loads the standard UNIX */etc/passwd* file into a database, with the login name as the key item and the entire password entry as the data item:

```
awk -F: '{print $1; print $0}' < /etc/passwd |
    sed 's/\\/\\\\/g' | db_load -T -t hash passwd.db
```

Note that backslash characters naturally occurring in the text are escaped to avoid interpretation as escape characters by *db_load*.

ENVIRONMENT VARIABLES

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

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

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)