

**NAME**

**dbm\_clearerr**, **dbm\_close**, **dbm\_delete**, **dbm\_dirfno**, **dbm\_error**, **dbm\_fetch**, **dbm\_firstkey**,  
**dbm\_nextkey**, **dbm\_open**, **dbm\_store** - database access functions

**SYNOPSIS**

```
#include <fcntl.h>
```

```
#include <ndbm.h>
```

*DBM \**

```
dbm_open(const char *base, int flags, mode_t mode);
```

*void*

```
dbm_close(DBM *db);
```

*int*

```
dbm_store(DBM *db, datum key, datum data, int flags);
```

*datum*

```
dbm_fetch(DBM *db, datum key);
```

*int*

```
dbm_delete(DBM *db, datum key);
```

*datum*

```
dbm_firstkey(DBM *db);
```

*datum*

```
dbm_nextkey(DBM *db);
```

*int*

```
dbm_error(DBM *db);
```

*int*

```
dbm_clearerr(DBM *db);
```

*int*

```
dbm_dirfno(DBM *db);
```

**DESCRIPTION**

Database access functions. These functions are implemented using `dbopen(3)` with a `hash(3)` database.

*datum* is declared in `<ndbm.h>`:

```
typedef struct {
    void *dptr;
    int dsize;
} datum;
```

The **dbm\_open**(*base, flags, mode*) function opens or creates a database. The *base* argument is the basename of the file containing the database; the actual database has a *.db* suffix. I.e., if *base* is `"/home/me/mystuff"` then the actual database is in the file `/home/me/mystuff.db`. The *flags* and *mode* arguments are passed to `open(2)`. `(O_RDWR | O_CREAT)` is a typical value for *flags*; `0660` is a typical value for *mode*. `O_WRONLY` is not allowed in *flags*. The pointer returned by **dbm\_open**() identifies the database and is the *db* argument to the other functions. The **dbm\_open**() function returns `NULL` and sets *errno* if there were any errors.

The **dbm\_close**(*db*) function closes the database.

The **dbm\_store**(*db, key, data, flags*) function inserts or replaces an entry in the database. The *flags* argument is either `DBM_INSERT` or `DBM_REPLACE`. If *flags* is `DBM_INSERT` and the database already contains an entry for *key*, that entry is not replaced. Otherwise the entry is replaced or inserted. The **dbm\_store**() function normally returns zero but returns 1 if the entry could not be inserted (because *flags* is `DBM_INSERT`, and an entry with *key* already exists) or returns -1 and sets *errno* if there were any errors.

The **dbm\_fetch**(*db, key*) function returns `NULL` or the *data* corresponding to *key*.

The **dbm\_delete**(*db, key*) function deletes the entry for *key*. The **dbm\_delete**() function normally returns zero or returns -1 and sets *errno* if there were any errors.

The **dbm\_firstkey**(*db*) function returns the first key in the database. The **dbm\_nextkey**(*db*) function returns subsequent keys. The **dbm\_firstkey**() function must be called before **dbm\_nextkey**(). The order in which keys are returned is unspecified and may appear random. The **dbm\_nextkey**() function returns `NULL` after all keys have been returned.

The **dbm\_error**(*db*) function returns the *errno* value of the most recent error. The **dbm\_clearerr**(*db*) function resets this value to 0 and returns 0.

The **dbm\_dirfno**(*db*) function returns the file descriptor to the database.

## SEE ALSO

open(2), dbopen(3), hash(3)

## STANDARDS

These functions (except **dbm\_dirfno()**) are included in the Version 2 of the Single UNIX Specification ("SUSv2").

## HISTORY

The functions **dbm\_init()**, **fetch()**, **store()**, **delete()**, **firstkey()**, and **nextkey()** first appeared in Version 7 AT&T UNIX.