

NAME

AnyDBM_File - provide framework for multiple DBMs

NDBM_File, DB_File, GDBM_File, SDBM_File, ODBM_File - various DBM implementations

SYNOPSIS

```
use AnyDBM_File;
```

DESCRIPTION

This module is a "pure virtual base class"--it has nothing of its own. It's just there to inherit from one of the various DBM packages. It prefers ndbm for compatibility reasons with Perl 4, then Berkeley DB (See *DB_File*), GDBM, SDBM (which is always there--it comes with Perl), and finally ODBM. This way old programs that used to use NDBM via dbmopen() can still do so, but new ones can reorder @ISA:

```
BEGIN { @AnyDBM_File::ISA = qw(DB_File GDBM_File NDBM_File) }
use AnyDBM_File;
```

Having multiple DBM implementations makes it trivial to copy database formats:

```
use POSIX; use NDBM_File; use DB_File;
tie %newhash, 'DB_File', $new_filename, O_CREAT|O_RDWR;
tie %oldhash, 'NDBM_File', $old_filename, 1, 0;
%newhash = %oldhash;
```

DBM Comparisons

Here's a partial table of features the different packages offer:

	odbm	ndbm	sdbm	gdbm	bsd-db
Linkage comes w/ perl	yes	yes	yes	yes	yes
Src comes w/ perl	no	no	yes	no	no
Comes w/ many unix os	yes	yes[0]	no	no	no
Builds ok on !unix	?	?	yes	yes	?
Code Size	?	?	small	big	big
Database Size	?	?	small	big?	ok[1]
Speed	?	?	slow	ok	fast
FTPable	no	no	yes	yes	yes
Easy to build	N/A	N/A	yes	yes	ok[2]
Size limits	1k	4k	1k[3]	none	none
Byte-order independent	no	no	no	no	yes
Licensing restrictions	?	?	no	yes	no

[0]

on mixed universe machines, may be in the bsd compat library, which is often shunned.

[1]

Can be trimmed if you compile for one access method.

[2]

See *DB_File*. Requires symbolic links.

[3]

By default, but can be redefined.

SEE ALSO

`dbm(3)`, `ndbm(3)`, `DB_File(3)`, *perldbfilter*