

Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [12. Data Persistence](#) »

12.12. dumbdbm — Portable DBM implementation¶

Note

The `dumbdbm` module has been renamed to `dbm.dumb` in Python 3.0. The [2to3](#) tool will automatically adapt imports when converting your sources to 3.0.

Note

The `dumbdbm` module is intended as a last resort fallback for the [anydbm](#) module when no more robust module is available. The `dumbdbm` module is not written for speed and is not nearly as heavily used as the other database modules.

The `dumbdbm` module provides a persistent dictionary-like interface which is written entirely in Python. Unlike other modules such as [gdbm](#) and [bsddb](#), no external library is required. As with other persistent mappings, the keys and values must always be strings.

The module defines the following:

exception `dumbdbm.error`¶

Raised on dumbdbm-specific errors, such as I/O errors. [KeyError](#) is raised for general mapping errors like specifying an incorrect key.

`dumbdbm.open(filename[, flag[, mode]])`¶

Open a dumbdbm database and return a dumbdbm object. The *filename* argument is the basename of the database file (without any specific extensions). When a dumbdbm database is created, files with `.dat` and `.dir` extensions are created.

The optional *flag* argument is currently ignored; the database is always opened for update, and will be created if it does not exist.

The optional *mode* argument is the Unix mode of the file, used only when the database has to be created. It defaults to octal `0666` (and will be modified by the prevailing `umask`).

Changed in version 2.2: The *mode* argument was ignored in earlier versions.

See also

Module [anydbm](#)

Generic interface to `dbm`-style databases.

Module [dbm](#)

Similar interface to the DBM/NDBM library.

Module [gdbm](#)

Similar interface to the GNU GDBM library.

Module [shelve](#)

Persistence module which stores non-string data.

Module [whichdb](#)

Utility module used to determine the type of an existing database.

12.12.1. Dumbdbm Objects¶

In addition to the methods provided by the [UserDict.DictMixin](#) class, `dumbdbm` objects provide the following methods.

`dumbdbm.sync()`¶

Synchronize the on-disk directory and data files. This method is called by the [sync\(\)](#) method of `Shelve` objects.

[Table Of Contents](#)

[12.12. dumbdbm — Portable DBM implementation](#)

- [12.12.1. Dumbdbm Objects](#)

Previous topic

Next topic

[12.13. sqlite3 — DB-API 2.0 interface for SQLite databases](#)

This Page

- [Show Source](#)

Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [12. Data Persistence](#) »

© [Copyright](#) 1990-2010, Python Software Foundation.

The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Feb 26, 2010. Created using [Sphinx](#) 0.6.3.