

Navigation

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

12.8. dbm — Simple “database” interface¶

Platforms: Unix

Note

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

The `dbm` module provides an interface to the Unix “(n)dbm” library. Dbm objects behave like mappings (dictionaries), except that keys and values are always strings. Printing a dbm object doesn’t print the keys and values, and the `items()` and `values()` methods are not supported.

This module can be used with the “classic” `ndbm` interface, the BSD DB compatibility interface, or the GNU GDBM compatibility interface. On Unix, the **configure** script will attempt to locate the appropriate header file to simplify building this module.

The module defines the following:

exception `dbm.error`¶

Raised on dbm-specific errors, such as I/O errors. `KeyError` is raised for general mapping errors like specifying an incorrect key.

`dbm.library`¶

Name of the `ndbm` implementation library used.

`dbm.open(filename[, flag[, mode]])`¶

Open a dbm database and return a dbm object. The `filename` argument is the name of the database file (without the `.dir` or `.pag` extensions; note that the BSD DB implementation of the interface will append the extension `.db` and only create one file).

The optional `flag` argument must be one of these values:

Value	Meaning
'r'	Open existing database for reading only (default)
'w'	Open existing database for reading and writing
'c'	Open database for reading and writing, creating it if it doesn't exist
'n'	Always create a new, empty database, open for reading and writing

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`).

See also

Module [anydbm](#)

Generic interface to dbm-style databases.

Module [gdbm](#)

Similar interface to the GNU GDBM library.

Module [whichdb](#)

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

Previous topic

[12.7. whichdb — Guess which DBM module created a database](#)

Next topic

[12.9. gdbm — GNU's reinterpretation of dbm](#)

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.