Search projects

Help Sponsor Log in Register

python-magic 0.4.18

✓ Latest version

`pip install python-magic`

Released: May 5, 2020

File type identification using libmagic

Navigation

Project description

Release history

Download files

Project links

Homepage

Statistics

GitHub statistics:

Stars: 1,769

Project description

python-magic

pypi package 0.4.18 build passing

python-magic is a Python interface to the libmagic file type identification library. libmagic identifies file types by checking their headers according to a predefined list of file types. This functionality is exposed to the command line by the Unix command `file`.

Usage

```
>>> import magic
>>> magic.from_file("testdata/test.pdf")
'PDF document, version 1.2'
# recommend using at least the first 2048 bytes, as less
>>> magic.from_buffer(open("testdata/test.pdf").read(204
'PDF document, version 1.2'
```

Forks: 209

Open issues/PRs: 28

View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery

Meta

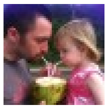
License: MIT License (MIT)

Author: Adam Hupp

mime, magic, file

Requires: Python >=2.7, !=3.0.*, !=3.1.*, !=3.2.*, !=3.3.*, !=3.4.*

Maintainers



ahupp

Classifiers

Intended Audience

- Developers

License

- OSI Approved :: MIT License

Programming Language

- Python
- Python :: 2
- Python :: 2.7

```
>>> magic.from_file("testdata/test.pdf", mime=True)
'application/pdf'
```

There is also a `Magic` class that provides more direct control, including overriding the magic database file and turning on character encoding detection. This is not recommended for general use. In particular, it's not safe for sharing across multiple threads and will fail throw if this is attempted.

```
>>> f = magic.Magic(uncompress=True)
>>> f.from_file('testdata/test.gz')
'ASCII text (gzip compressed data, was "test", last modi
21:32:52 2008, from Unix)'
```

You can also combine the flag options:

```
>>> f = magic.Magic(mime=True, uncompress=True)
>>> f.from_file('testdata/test.gz')
'text/plain'
```

Installation

The current stable version of python-magic is available on PyPI and can be installed by running `pip install python-magic`.

Other sources:

- PyPI: http://pypi.python.org/pypi/python-magic/
- GitHub: https://github.com/ahupp/python-magic

This module is a simple wrapper around the libmagic C library, and that must be installed as well:

Debian/Ubuntu

$ sudo apt-get install libmagic1

Windows

- Python :: 3
- Python :: 3.5
- Python :: 3.6
- Python :: 3.7
- Python :: 3.8
- Python :: Implementation :: CPython

You'll need DLLs for libmagic. @julian-r has uploaded a version of this project that includes binaries to PyPI: https://pypi.python.org/pypi/python-magic-bin/0.4.14

Other sources of the libraries in the past have been File for Windows . You will need to copy the file `magic` out of `[binary-zip]\share\misc`, and pass its location to `Magic(magic_file=...)`.

If you are using a 64-bit build of python, you'll need 64-bit libmagic binaries which can be found here: https://github.com/pidydx/libmagicwin64. Newer version can be found here: https://github.com/nscaife/file-windows.

OSX

- When using Homebrew: `brew install libmagic`
- When using macports: `port install file`

Troubleshooting

- 'MagicException: could not find any magic files!': some installations of libmagic do not correctly point to their magic database file. Try specifying the path to the file explicitly in the constructor: `magic.Magic(magic_file="path_to_magic_file")`.
- 'WindowsError: [Error 193] %1 is not a valid Win32 application': Attempting to run the 32-bit libmagic DLL in a 64-bit build of python will fail with this error. Here are 64-bit builds of libmagic for windows: https://github.com/pidydx/libmagicwin64
- 'WindowsError: exception: access violation writing 0x00000000 ' This may indicate you are mixing Windows Python and Cygwin Python. Make sure your libmagic and python builds are consistent.

Bug Reports

python-magic is a thin layer over the libmagic C library. Historically, most bugs that have been reported against python-magic are actually bugs in libmagic; libmagic bugs can be reported on their tracker here: https://bugs.astron.com/my_view_page.php. If you're not sure where the bug lies feel free to file an issue on GitHub and I can triage it.

Running the tests

To run the tests across 3 recent Ubuntu LTS releases (depends on Docker):

```
$ ./test_docker.sh
```

To run tests locally across all available python versions:

```
$ ./test/run.py
```

To run against a specific python version:

```
$ LC_ALL=en_US.UTF-8 python3 test/test.py
```

Versioning

Minor version bumps should be backwards compatible. Major bumps are not.

Name Conflict

There are, sadly, two libraries which use the module name `magic`. Both have been around for quite a while. If you are using this module and get an error using a method like `open`, your code is expecting the other one. Hopefully one day these will be reconciled.

Author

Written by Adam Hupp in 2001 for a project that never got off the ground. It originally used SWIG for the C library bindings, but switched to ctypes once that was part of the python standard library.

You can contact me via my website or GitHub.

Contributors

Thanks to these folks on github who submitted features and bug fixes.

- Amit Sethi
- bigben87
- fallgesetz
- FlaPer87
- Hugo van Kemenade
- lukenowak
- NicolasDelaby
- sacha@ssl.co.uk
- SimpleSeb
- tehmaze

License

python-magic is distributed under the MIT license. See the included LICENSE file for details.

I am providing code in the repository to you under an open source license. Because this is my personal repository, the license you receive to my code is from me and not my employer (Facebook).

Help

Installing packages

Uploading packages

User guide

FAQs

About PyPI

PyPI on Twitter

Infrastructure dashboard

Package index name retention

Our sponsors

Contributing to PyPI

Bugs and feedback

Contribute on GitHub

Using PyPI

Code of conduct

Report security issue

Translate PyPI

Development credits

Privacy policy

Terms of use

Status: All Systems Operational

Developed and maintained by the Python community, for the Python community.
Donate today!

Site map

Switch to desktop version

English español français 日本語 português (Brasil) українська Ελληνικά Deutsch 中文 (简体) русский
עברית

Pingdom
Monitoring

Google
Object Storage and
Download Analytics

Sentry
Error logging

AWS
Cloud computing

DataDog
Monitoring

Fastly
CDN

DigiCert
EV certificate

StatusPage
Status page