pyxattr Documentation

Release 0.6.1

Iustin Pop

Oct 13, 2019

Contents

This is the pyxattr module, a Python extension module which gives access to the extended attributes for filesystem objects available in some operating systems.

Downloads: go to https://pyxattr.k1024.org/downloads/. Latest version is 0.6.1. The source repository is either at http://git.k1024.org/pyxattr.git or at https://github.com/iustin/pyxattr.

CHAPTER 1

Requirements

pyxattr has been written and tested on Linux, kernel v2.4 or later, with XFS filesystems; ext2/ext3 should work also. If any other platform implements the same behavior, pyxattr could be used.

You need to have the setuptools tool installed in order to build and install the module.

CHAPTER 2

License

CHAPTER 3

Contents

3.1 Interface to extended filesystem attributes

3.1.1 Constants

XATTR_CREATE
Used as flags value, the target attribute will be created, giving an error if it already exists.

XATTR_REPLACE
Used as flags value, the target attribute will be replaced, giving an error if it doesn't exist.

NS_SECURITY
The security name space, used by kernel security modules to store (for example) capabilities information.

NS_SYSTEM
The system name space, used by the kernel to store (for example) ACLs.

NS_TRUSTED
The trusted name space, visible and accessibly only to trusted processes, used to implement mechanisms in user space.

NS_USER
The user name space; this is the name space accessible to non-privileged processes.

3.1.2 Functions

3.1.3 Deprecated functions

3.2 News

3.2.1 Version 0.6.1

released Tue, 24 Jul 2018

Minor bugfix, performance and compatibility release.

- Minor compatibility fix: on Linux, drop the use of the *attr* library, and instead switch to the glibc header *sys/xattr.h*, which is provided for a really long time (since glibc 2.3). The formerly used header *attr/xattr.h* has been removed from the *attr* library in version 2.4.48. Fix provided by Lars Wendler, many thanks!
- Release the GIL when performing I/O. Patch proposed by xwhuang, many thanks. I tested this a long while back it seemed to impact performance on local filesystems, but upon further inspection, the downsides are minor (between 0 and 5%, in many cases negligible). For remote or slow filesystems, this should allow much increased parallelism.
- Fix symlink set operation on MacOS X; bugfix provided by adamlin, much appreciated! This also uncovered testing problems related to symlinks, which are now fixed (the bug would be caught by the updated tests).

3.2.2 Version 0.6.0

released Mon, 23 Jan 2017

Bugfix and feature release (hence the version bump).

The main change is to the implementation of how attributes are listed and read. This was done due to existing race issues when attributes are modified while being read (github issue #12), but basically all various internal paths that dealt with retrieving an attribute value or listing attributes were unified in a single helper function that does handle such concurrent modifications. As a side effect, the size of the buffers used for such reads have changed, which (depending on attribute value) might change the trade-off between number of syscalls done and memory usage.

As feature release, OSX support was contributed by Adam Knight <adam@movq.us>, thanks a lot! I don't have access to OSX so the testing for it is done via Travis builds; please report any issues.

3.2.3 Version 0.5.6

released Sat, 09 Apr 2016

Small bugfix release:

- Fixes some sign-compare warnings
- Fixes potential name truncation in merge_ns()
- Fixes building on systems which don't have ENODATA

Tested with Python 2.7.11, Python 3.5.1 and PyPy 5.0.1.

3.2.4 Version 0.5.5

released Fri, 01 May 2015

Bugfix release:

- fixes some more memory leaks when handling out-of-memory in get_all() function
- improve error reporting when an attribute disappears after we asked for its length but before we managed to read it
- fix int/size_t issues found by RedHat/Fedora, https://bugzilla.redhat.com/show_bug.cgi?id=1127310; the fix is different than their fix, but it should accomplish the same thing
- convert all code to only do explicit casts after checking boundaries, making the code *-Wconversion*-clean (although that warning is not enabled by default)

3.2.5 Version 0.5.4

released Thu, 30 Apr 2015

Fix memory leaks on some of the error-handling paths of the *get()* function.

3.2.6 Version 0.5.3

released Fri, 23 May 2014

Small optimisations release:

- ari edelkind contributed a speed-up optimisation for handling of files without xattrs (which is, in general, the expected case)
- Jonas Borgström contributed a behaviour change to the handling of file names: under Python 3 and up, unicode paths are encoded/decoded using the 'surogatee' handler, instead of the 'strict' handler; while this can hide encoding errors, it mirrors what Python libraries do (e.g. see os.fsencode/fsdecode)
- Sean Patrick Santos contributed improvements to the test suite so that it can be used even on files systems which have built-in attributes (e.g. when using SELinux, or NFSv4); to enable this, define the attributes in the TEST_IGNORE_XATTRS environment variable

3.2.7 Version 0.5.2

released Thu, 03 Jan 2013

Bug-fix release. Thanks to Michał Górny, it looked like the library had problem running under pypy, but actually there was a bug in the PyArg_ParseTuple use of et# (signed vs. unsigned, and lack of compiler warnings). This was fixed, and now the test suite passed with many CPython versions and PyPy (version 1.9).

3.2.8 Version 0.5.1

released Wed, 16 May 2012

Bug-fix release. Thanks to Dave Malcolm and his cpychecker tool, a number of significant bugs (refcount leaks and potential NULL-pointer dereferences) have been fixed.

Furthermore, compatibility with Python 3 has been improved; this however required changing the meaning of the `namespace` argument to the functions: if passed, None is no longer a valid value; pass an empty string if (due to the structure of your program) you have to pass this argument but want to specify no namespace.

Also, the project home page has changed from SourceForge to GitHub, and the documentation has been converted from epydoc-based to sphinx.

3.2.9 Version 0.5

released Sun, 27 Dec 2009

Implemented support for Python 3. This required a significant change to the C module, hence the new version number.

3.2.10 Version 0.4

released Mon, 30 Jun 2008

API

The old functions ({get,set,list,remove}xattr) are deprecated and replaced with a new API that is namespace-aware and hopefully will allow other OSes (e.g. FreeBSD) to be supported more naturally.

Both the old and the new API are supported in the 0.4 versions, however users are encouraged to migrate to the new API.

New features

A new bulk get function called get_all() has been added that should be somewhat faster in case of querying files which have many attributes.

License

Since LGPLv3 is not compatible with GPLv2 (which unfortunately I didn't realize before), the license was changed to LGPLv2.1 or later.

Internals

Unittest coverage was improved.

3.2.11 Version 0.3

released Sun, 09 Mar 2008

- changed licence from GPL to LGPL (3 or later)
- changed listxattr return type from tuple to a list
- developer-related: added unittests

3.2.12 Version 0.2.2

released Sun, 01 Jul 2007

- fixed listing symlink xattrs

3.2.13 Version 0.2.1

released Sat, 11 Feb 2006

- fixed a bug when reading symlink EAs (you weren't able to do it, actually)
- fixed a possible memory leak when the actual read of the EA failed but the call to get the length of the EA didn't

Also see the search.

Index

N

X