The Documentation of pylibnfs4acl

Version 0.2.0

Franz Glasner
FELDMANN media group AG

2026-06-10

Last updated on 2026-06-10 (rev 5c1ddc82db26)

Contents

CHAPTER 1

Introduction

Author
Franz Glasner

Company
FELDMANN media group AG

Version
0.2.0

Date
2026-06-10

Revision
5c1ddc82db26

pylibnfs4acl is a Python project that provides a package `nfs4acl` to manipulate NFSv4 style ACLs. In order to do this it wraps the systems's native ACL C library. For proper use some knowledge about NFSv4 ACLS is required. As such it is certainly not intended for novice users.

The library currently works on FreeBSD[1] and has been used from releases 10.3 to 14.3 regularly.

It is modelled mostly after *pylibacl* which is a package for manipulating *POSIX.1e* ACLs. You will find *pylibacl* on the Python Package Index[2] and its documentation on its homepage[3].

No external dependencies are required. Only the standard module `ctypes`[4] is needed to call into the native ACL libraries.

The package is developed and tested with Python 2.7 and Python 3.4 or later. It is expected to work with Python 3 for versions 3.2 or later. This has not been tested yet. Python 2.6 and Python 3.0 and Python 3.1 definitely do not work because they lack the `ctypes.c_ssize_t`[5] type.

It also runs on PyPy2 and PyPy3 and is tested currently on FreeBSD using the binary packages *pypy-5.10.0*, *pypy3-5.10.1*, *pypy-6.0.0_1* and *pypy3-6.0.0_1* from the *ports collection*.

[1] https://www.freebsd.org
[2] https://pypi.org/project/pylibacl/
[3] https://pylibacl.k1024.org/
[4] https://docs.python.org/3/library/ctypes.html#module-ctypes
[5] https://docs.python.org/3/library/ctypes.html#ctypes.c_ssize_t

CHAPTER 2

Overview

2.1 ACL Structure

An NFSv4 ACL has the following structure:

- An ACL (`ACL`) has zero or more entries.
- An ACL entry, in short *ACE*, (`Entry`) has these attributes:
  - The *type* of the entry (`entry_type`).

    This field determines whether the ACE *allows* or *denies* access.
  - The *tag type* of the entry (`tag_type`).

    Together with the *qualifier* field this field controls for whom this ACE describes access permissions.
  - The *principal* (aka *qualifier*) of the entry (`qualifier`).

    The ID of a user or group for whom this ACE describes access permissions. This field makes sense only for entries of *tag type* `ACL_USER` or `ACL_GROUP`.

    Typically checked against the proper effective IDs of a process.
  - The set of *permissions* (`permset`).

    This field defines what kind of access the process matching this ACE has for accessing the associated file.
  - The set of *flags* (`flagset`).

    They are used to control and/or view the inheritance of ACLs.

- A *permission set* (*Permset*) in an ACE has just methods to set and reset permission bits.
- A *flag set* (*Flagset*) in an ACE has just methods to set and reset flag bits.

2.2 Implementation Note

The implementation is fairly low-level. The provided "abstractions" are leaky. Most operations are implemented by an immediate call into the native C implementation – the given classes are just thin wrappers around the OS level handles/pointers.

2.3 Usage Examples

Get an ACL from a file with and print it:

```
import nfs4acl

with nfs4acl.ACL(file="_tmp/mnt") as acl:
    # print in the standard form
    print(acl)

    # print with numeric IDs instead
    print(acl.to_any_text(
        options=nfs4acl.const.ACL_TEXT_NUMERIC_IDS))
```

Be sure to use the `with` construct or call *close()* to cleanup any associated OS resources.

Iterate over all entries:

```
from nfs4acl import *

with ACL(file="_tmp/mnt") as acl:
    for entry in acl:
        # do something
        ...
```

Append a new entry with:

```
from nfs4acl import *

entry = Entry(acl)
```

or:

```
from nfs4acl import *

entry = acl.append()
```

Prepend a new entry at the begin:

```
from nfs4acl import *

entry = Entry(acl, pos=0)
```

Delete an existing entry from an ACL:

```
# find the entry
...

# delete it
entry.delete()
```

Warning

After adding or deleting entries all existing entries (and dependent permission and flag sets) will be invalid.

Re-using such items afterwards is outright dangerous because these items are typically implemented as descriptors (i.e. pointers) into an existing ACL structure. After manipulating the number of entries an existing object can point to another entry or to nowhere (e.g. because the ACL got reallocated completely).

While the parent ACL instance is able to account for this, the package applies proper protections against re-use of child items after such operations.

```
from nfs4acl import *

with acl:

    # get the entries as list
    entries = list(acl)

    # all entries are valid
    for e in entries:
        assert e

    # insert a new one at the beginning
    new_entry = Entry(acl, pos=0)

    #
    # ATTENTION : Re-using entries is invalid because
```

```
    #             the underlying ACL structure got
    #             rearranged.
    #             The entries therefore have been
    #             invalidated by the package.
    #
    for e in entries:
        # now all the existing old entries are invalid
        assert not e
        #
        # and some other operation like this will raise
        # a ValueError exception
        #
        e.delete()
```

Make a file executable for the owner:

```
from nfs4acl import *

with ACL(fd=myfile.fileno()) as acl:
    entries = filter(
        lambda x: x.tag_type == const.ACL_USER_OBJ)
    if entries:
        for entry in entries:
            entry.permset.add(const.ACL_EXECUTE)
    else:
        #
        # new entry for the owner:
        # add one with just the execute permission
        #
        entry = acl.append()
        entry.tag_type = const.ACL_USER_OBJ
        entry.entry_type = const.ACL_ENTRY_TYPE_ALLOW
        ps = entry.permset
        ps.add(const.ACL_EXECUTE)
        #
        # NOTE: This changed the entry inline because it
        #       is a descriptor into the entry.
        #       No need to to
        #           entry.permset = ps
        #
    # apply this to the file
    acl.applyto(myfile)
```

Copy a complete ACL entry from another ACL:

```
import nfs4acl

with nfs4acl.ACL(file="source") as source:
    with nfs4acl.ACL(file="destination") as destination:
        # find the correct entry pair
        ...

        destination_entry.copy(source_entry)

        assert destination_entry == source_entry
```

Create a new empty ACL:

```
import nfs4acl

acl = nfs4acl.ACL()

# of course...
assert acl is not None

# ... but an ACL is logically False if there are no
#     entries
assert not acl

# you have to close an empty ACL also
acl.close()
```

Copy a flag set from another ACL:

```
import nfs4acl

with nfs4acl.ACL(file="source") as source:
    with nfs4acl.ACL(file="destination") as destination:
        # find the correct entry pair
        ...

        # copy: now the flagset setter comes really handy
        destination_entry.flagset = source_entry.flagset

        assert destination_entry.flagset == source_entry.
↪flagset
```

Copy a complete ACL:

```
from nfs4acl import *
```

(continues on next page)

(continued from previous page)

```
with ACL(file="source") as given_acl:
    with ACL(acl=given_acl) as the_copy:
        # do something with the copy
        ...

        # maybe you want it to save into a different file
        the_copy.applyto(file="destination")
```

If you just want to copy an ACL from one file to another file unchanged:

```
from nfs4acl import *

with ACL(file="source") as acl:
    acl.applyto(file="destination")
```

Create an ACL from a text representation:

```
from nfs4acl import *

with ACL(file="source") as given_acl:
    with ACL(text=str(given_acl)) as the_copy:
        # it is a real copy ...
        assert not (given_acl is the_copy)

        # .. but logically identical
        assert given_acl == the_copy
```

2.4 Unit Tests

2.4.1 Preparation

The unit tests need a clean scratch area in the filesystem with support for NFSv4 ACLs. On FreeBSD the ZFS filesystem is a perfect match for this.

To setup a clean filesystem test area with NFSv4 ACL support there is a helper script `tests/zfstest.sh`. It has to be run as user *root* with the current working directory in `tests`.

The script helps to create a file-backed ZFS pool named "nfs4aclpool" and helps to mount a temporary filesystem at mountpoint `_tmp/mnt` as scratch area for the unit tests.

Usage:

1. Setup the pool:

```
cd tests
./zfstest.sh pool
```

It creates an empty imagefile in `_tmp` and uses this to create a file-backed ZFS pool named “nfs4aclpool”. A completely new image file will be created automatically. An existing file will be deleted first.

2. Setup the filesystem at the mountpoint `_tmp/mnt` and make the real user who runs the tests the owner:

```
cd tests
./zfstest.sh mount <userid>
```

3. Develop and run unit tests …
4. Unmount the working space:

```
cd tests
./zfstest.sh umount
```

5. Destroy the ZFS pool:

```
cd tests
./zfstest.sh unpool
```

Remove the image file if needed:

```
rm _tmp/zfspool.img
```

2.4.2 Running

Unit tests are executed by:

```
cd tests
python test.py
```

Unit tests use Python’s built-in `unittest`[6] package and have no other external dependencies.

[6] https://docs.python.org/3/library/unittest.html#module-unittest

CHAPTER 3

API Documentation

3.1 Package nfs4acl

A library for manipulating NFSv4 style ACLs by calling the native ACL libraries.

Author
Franz Glasner

Company
FELDMANN media group AG

Copyright

- © 2018 FELDMANN media group AG
- © 2018 Franz Glasner

License

GNU Lesser General Public License v2.1 or later (LGPLv2.1+). See *LGPL-2.1-or-later.txt* for details.

ID

@(#) $Header: nfs4acl/__init__.py 5c1ddc82db26 2026-06-10 11:04:48 +0200 fzglas.hg public $

This module contains all the public types the user will probably want to work with.

These types are implemented in platform dependent sub-packages and aliased in *nfs4acl*.

For FreeBSD[7] these are the types from *nfs4acl.freebsd.acl*:

- *nfs4acl.freebsd.acl.ACL*
- *nfs4acl.freebsd.acl.Entry*
- *nfs4acl.freebsd.acl.Permset*
- *nfs4acl.freebsd.acl.Flagset*

nfs4acl.**PLATFORM = 'freebsd'**

The name of the platform dependent module being used

3.1.1 Module nfs4acl.const

This module contains all public constants. Its content (either attributes and values) is platform dependent.

On FreeBSD[8] this is an alias for *nfs4acl.freebsd.const*.

3.1.2 Module nfs4acl.compat

A small compatibility shim for Python2 and Python3 compatibility.

nfs4acl.compat.**b**(*s*, *encoding='utf_8'*)

nfs4acl.compat.**bfs**(*s*, *encoding='utf-8'*)

nfs4acl.compat.**u**(*s*, *encoding='utf_8'*)

[7] https://www.freebsd.org
[8] https://www.freebsd.org

3.1.3 Module nfs4acl.getacl

The implementation of a *getfacl(1)*[9] alike.

Differences to *getfacl(1)*[10] are:

- Handle *only* NFSv4 ACLs – no POSIX ACLs
- Can do recursive file tree traversal with flexible symbolic link handling

See also

the *manual page*

Usage: getacl [options] [file ...]

Options:

-H, --half-logical If option *-R* is specified, follow symbolic links named on the command line only. Symbolic links encountered during traversal are not followed.

-h, --no-dereference If the command line target of the operation is a symbolic link, return the ACL from the symbolic link itself rather than following the link.

--help Print this help and exit.

-i Append the numerical ID at the end of each ACL entry containing user or group names.

-L, --logical If option *-R* is specified, all symbolic links are followed.

-n, --numeric Display user and group IDs numerically rather than converting to a user or group name.

-P, --physical If option *-R* is specified, no symbolic links are followed. This is the default.

-q, --omit-header Do not write commented information about file name and ownership.

-R, --recursive Operate recursively on files and directories.

-v Display permissions and flags in a verbose form.

--version Print the version number and exit.

-x Do not traverse file system mount points.

The following operand is available:

[9] https://www.freebsd.org/cgi/man.cgi?query=getfacl&sektion=1
[10] https://www.freebsd.org/cgi/man.cgi?query=getfacl&sektion=1

file
A pathname of a file whose ACL shall be retrieved. If file is not specified, or a file is specified as -, then getacl reads a list of pathnames, each terminated by one newline character, from the standard input.

`nfs4acl.getacl.`**`main`**`()`
The main entry point

3.1.4 Module `nfs4acl.utils`

Some utility and helper classes and functions.

`class` `nfs4acl.utils.`**`SingletonType`**
Metaclass for implementing a singleton.

Calls *__init__()* from the singleton class only once. Just overriding its *__new__* calls it always.

3.2 Package `nfs4acl.freebsd`

Sub-package that contains the implemention of NFSv4 ACLs using the FreeBSD[11] security API.

FreeBSD implements NFSv4 style ACLs on its native filesystems *ZFS* and *UFS*.

The main references are:

- the header in file `/usr/include/sys/acl.h`
- the manual page *acl(3)*[12]
- the manual page *acl(9)*[13]
- NFSv4 ACLs page[14] on the FreeBSD Wiki[15]

3.2.1 Module `nfs4acl.freebsd.acl`

The ACL implementation for NFSv4 ACLs on FreeBSD.

Note

The public content of this module is meant to be accessed by clients via its aliases in *nfs4acl*.

[11] https://www.freebsd.org
[12] https://www.freebsd.org/cgi/man.cgi?query=acl&sektion=3
[13] https://www.freebsd.org/cgi/man.cgi?query=acl&sektion=9
[14] https://wiki.freebsd.org/NFSv4_ACLs
[15] https://wiki.freebsd.org/

Todo

Platform-independent pickling style.

Is platform-independency really required? Use some normalized text representation for this instead (e.g. using *getfacl(1)*[16] and friends)?

class nfs4acl.freebsd.acl.**ACL**(*file=None*, *fd=None*, *follow_symlink=True*, *text=None*, *acl=None*, *_attach_ptr=None*, *_acldll=None*, *_num_entries=None*)

Represent an NFSv4 ACL.

An ACL is its own context manager because it *must* be freed with *close()* to release any OS resources associated with the ACL.

After initialization an ACL has *no* connections to filesystem objects. It exists in memory only. You have to call *applyto()* to apply the ACL to a file (either the same as used in the constructor or another one).

An ACL is its own iterator and iterates over all its entries.

An ACL in boolean context evaluates to `False` if there are no entries and to `True` otherwise. If the ACL is closed it evaluates to `False` also.

The `pickle`[17] protocol is supported. But because of the native interface: only *complete* ACLs can be pickled, not just individual entries. The *qualifier* is pickled as ID number type only. This may or may not be what you want.

Parameters

- **file** (*str*[18]) – create an ACL representing the access ACL of the specified file
- **fd** (*int*[19]) – create an ACL representing the access ACL of the given file descriptor
- **text** (*str*[20]) – create an ACL from a textual description
- **acl** (ACL) – create a copy of an existing ACL instance
- **follow_symlink** (*bool*[21]) – if `False`, act on a symlink rather than its target, if the target of the *item* is a symlink

Only one of *file*, *fd*, *text* or *acl* can be given.

If no parameters are passed, an empty ACL will be created; this makes sense only when your OS supports ACL modification, otherwise the ACL won't be useful.

INIT_COUNT = 254

The default number of entries to allocate at least when creating a new empty ACL

[16] https://www.freebsd.org/cgi/man.cgi?query=getfacl&sektion=1

__getstate__()

Return the ACL as platform-dependent-picklable state

__setstate__(*state*)

Instantiate when unpickling from pickled platform-dependent state *state*

applyto(*item*, *follow_symlink=True*)

Set an NFSv4 ACL for a file.

Parameters

- **item** – the filesystem object on which to act; it be a path string, a file-like with a `item.fileno()` or a file descriptor
- **follow_symlink** (*bool*[22]) – the method acts on a symlink rather than its target, if the target of the *item* is a symlink

__enter__()

Behave as its own context manager

__exit__(**args*, ***kwds*)

Behave as its own context manager: leave the context and call *close()* to properly free any resources

close()

Free the OS resources associated with the ACL

property is_closed

Return whether the ACL has been already closed

check_not_closed()

Check that the ACL is not closed; otherwise raise a proper `ValueError`[23] exception.

clear()

Delete all entries from the current ACL

append(*entry=None*)

Append a new Entry to the ACL and return it.

This is a convenience function to create a new *Entry* and append it to the ACL. If a parameter of type *Entry* instance is given, the entry will be a copy of that one (as if copied with *Entry.copy()*), otherwise, the new entry will be empty.

Note

The source *entry* (which effectively is a pointer) may become invalid by creating a new entry because the ACL may be relocated. So this method tries to be safe and makes a copy into a completely new and independent ACL beforehand.

delete_entry(*item*)

Delete the ACL entry in *item* from the ACL.

Parameters

item – the entry to delete to; it can be an *Entry* that belongs to *self* or an int[24] that will be interpreted as zero-based index position of the ACE to be deleted

stripped()

Return a new ACL with extended entries stripped

is_trivial()

Determine whether the ACL is trivial.

An ACL is trivial if it can be fully expressed as a file mode without losing any access rules.

For NFSv4 ACLs, an ACL is trivial if it is identical to the ACL generated by *stripped()*.

Return type

True or False

valid(*file=None*, *fd=None*, *follow_symlink=True*)

Validate an ACL.

Allow an ACL to be checked in the context of a specific file system object.

Parameters

- **file** (*str*[25]) – validate the ACL against the file specified with its name
- **fd** (*int*[26]) – validate the ACL against the file system object specified with its descriptor

Return type

True or False

Exactly one of *file* or *fd* must be given.

On FreeBSD NFSv4 ACLs can be checked only within the context of a filesystem object.

__iter__()

Iterate over all ACL entries.

An *ACL* is its own iterator and yields all entries in order.

__copy__()

Return a duplicate of the ACL.

Implementation of the copy protocol for `copy.copy()`.

__eq__(*other*)

Implement the logical == operator.

Parameters

other (*ACL* or *str*) – the thing to compare with

Converts *self* and *other* into a string with numeric ids and compares the results. If *other* is already in string form be sure to create it using numeric ids; when being generated the text representation is created with `.to_any_text(options=ACL_TEXT_NUMERIC_IDS)`.

Todo

Normalize the string representation before comparing: for each line: strip white space et al.

__ne__(*other*)

Implement the logical != operator.

Parameters

other (*ACL* or *str*) – the thing to compare with

Converts *self* and *other* into a string with numeric ids and compares the results. If *other* is already in string form be sure to create it using numeric ids; when beging generated the text representation is created with `.to_any_text(options=ACL_TEXT_NUMERIC_IDS)`.

__hash__ = None

Hashing is explicitely *not* supported because an ACL is mutable

__bool__()

Evaluate in a boolean context: an ACL in boolean context evaluates to `False` if there are no entries and to `True` otherwise.

__bytes__()

PY3 method to convert into a `bytes`[27] representation

__unicode__()

PY2 method to convert into a `unicode` representation

__str__()

Return a native string representation of the ACL using the native “acl_to_text()”

to_any_text(*prefix=''*, *separator='\n'*, *options=0*)

Return a custom text representation of the ACL.

Parameters

- **prefix** (*str*[28]) – optional string that will be pre-pended to all lines.

- **separator** (*str*[29]) – a string that will be used to separate the entries in the ACL.
- **options** (*int*[30]) – a combination (with or'ing) of the following values:
  - ***ACL_TEXT_VERBOSE***
    Format ACL using verbose form
  - ***ACL_TEXT_NUMERIC_IDS***
    Do not resolve IDs into user or group names
  - ***ACL_TEXT_APPEND_ID***
    In addition to user and group names, append numeric IDs

For NFSv4 ACLs the format of the text string shall be the compact form, unless the *ACL_TEXT_VERBOSE* flag is given.

__repr__()
Return repr(self).

class nfs4acl.freebsd.acl.**Entry**(*acl*, *pos=-1*, *_attach_ptr=None*, *_attach_generation=None*)

Represents an NFSv4 entry in an ACL (aka ACE).

A newly created ACE is initialized in the following ways: the ACE *tag_type* component contains *ACL_UNDEFINED_TAG*, the *qualifier* component contains *ACL_UNDEFINED_ID*, and the set of permissions has no permissions enabled. Any existing ACL entry descriptors that refer to entries in the ACL continue to refer to those entries.

In boolean context an entry evaluates to `True` iff its parent ACL is not closed.

Create a new ACL entry (ACE) within ACL *acl* at position *pos*.

Parameters

- **acl** (ACL) – the ACL this ACE belongs to
- **pos** (*int*[31]) – optional position where to create the new ACL; if the value is -1 then the new entry will be appended at the end

[17] https://docs.python.org/3/library/pickle.html#module-pickle
[18] https://docs.python.org/3/library/stdtypes.html#str
[19] https://docs.python.org/3/library/functions.html#int
[20] https://docs.python.org/3/library/stdtypes.html#str
[21] https://docs.python.org/3/library/functions.html#bool
[22] https://docs.python.org/3/library/functions.html#bool
[23] https://docs.python.org/3/library/exceptions.html#ValueError
[24] https://docs.python.org/3/library/functions.html#int
[25] https://docs.python.org/3/library/stdtypes.html#str
[26] https://docs.python.org/3/library/functions.html#int
[27] https://docs.python.org/3/library/stdtypes.html#bytes
[28] https://docs.python.org/3/library/stdtypes.html#str
[29] https://docs.python.org/3/library/stdtypes.html#str
[30] https://docs.python.org/3/library/functions.html#int

copy(*src*)

Copy the content of the ACL entry in *src* into *self*.

The source entry can be one of the same ACL or belong to another ACL. Both entries remain independent afterwards.

delete()

Delete this ACL entry from the ACL

property parent

The parent *ACL* of this entry

property tag_type

The tag type of the current entry as int.

Together with the *qualifier* field this field controls for whom this entry describes access permissions.

Valid values are:

- ***ACL_USER_OBJ***
  Permissions apply to file owner (aka *user@*)
- ***ACL_USER***
  Permissions apply to additional user specified by qualifier
- ***ACL_GROUP_OBJ***
  Permissions apply to file group (aka *group@*)
- ***ACL_GROUP***
  Permissions apply to additional group specified by qualifier
- ***ACL_MASK***
  Permissions specify mask (**not** to be used on NFSv4 ACLs)
- ***ACL_OTHER***
  Permissions apply to other (**not** to be used on NFSv4 ACLs)
- ***ACL_OTHER_OBJ***
  Same as *ACL_OTHER*
- ***ACL_EVERYONE***
  Permissions apply to virtually everyone (aka *everyone@*)

property qualifier

The qualifier (aka principal) of the entry as int[32].

This is the id of a user or group for whom this entry describes access permissions.

If the *tag_type* is *ACL_USER*, this should be a user id. If the tag type if *ACL_GROUP*, this should be a group id.

On a newly created entry this is *ACL_UNDEFINED_ID*.

property entry_type

Get the NFSv4 entry type of the entry as `int`[33].

Valid values are:

- ***ACL_ENTRY_TYPE_ALLOW***
  allow type entry
- ***ACL_ENTRY_TYPE_DENY***
  deny type entry

Not yet supported are:

- *ACL_ENTRY_TYPE_AUDIT*
- *ACL_ENTRY_TYPE_ALARM*

property permset

Return a *Permset* instance that operates on the current entry.

This field defines what kind of access the process matching this entry has for accessing the associated file.

> **Note**
>
> When operating only on the permset of the ACE the setter is not needed because the returned *Permset* operates directly on the entry you got the permset from.
>
> The setter can be used to copy a permset as a whole from another one (of the same or another ACL).

property flagset

Return a *Flagset* instance that operates on the current entry.

Flage are used to control and/or view the inheritance of ACLs.

> **Note**
>
> When operating only on the flagset of the ACE the setter is not needed because the returned *Flagset* operates directly on the entry you got the flagset from.
>
> The setter can be used to copy a flagset as a whole from another one (of the same or another ACL).

__bool__()

An Entry evaluates to `True` iff its parent ACL is not yet closed and the number of entries has not changed since the creation of *self*.

check_valid()

Raise a `ValueError`[34] exception iff the entry evals to `False`.

So – this method passes if the parent ACL is not closed and the current entry's generation number matches the generation number of the parent ACL.

__eq__(*other*)

Logical equality check for an ACE: all of its fields must be equal.

__ne__(*other*)

Logical inequality check for an ACE: if any of its fields differs then the complete entry differs

__hash__ = None

Hashing is explicitely *not* supported: the type is mutable

__str__()

Return a native string representation of the entry.

Simulate "acl_to_text()" for an entry only.

__repr__()

Return repr(self).

class nfs4acl.freebsd.acl.**Permset**(*entry*)

Type which represents the permission set in an ACL entry.

An instance of this class is just a descriptor into the permset of the entry. Thus all operations operate *directly* on the entry.

In boolean context a Permset evaluates to `True` iff any of its permission bits is set.

Because an instance of this class is *mutable* hashing is explicitely *not supported*.

Parameters

entry (`Entry`) – the ACE this permission set belongs to.

property parent

The parent ACL entry

add(*perm*)

Add a permission to the permission set.

This function adds the permission contained in the argument *perm* to the permission set. An attempt to add a permission that is already contained in the permission set is not considered an error.

Only a single permission can be added per call. FreeBSD checks this.

[31] https://docs.python.org/3/library/functions.html#int
[32] https://docs.python.org/3/library/functions.html#int
[33] https://docs.python.org/3/library/functions.html#int
[34] https://docs.python.org/3/library/exceptions.html#ValueError

For NFSv4 ACLs, valid values are:

- ***ACL_READ_DATA***
  Read permission
- ***ACL_LIST_DIRECTORY***
  Same as *ACL_READ_DATA*
- ***ACL_WRITE_DATA***
  Write permission, or permission to create files
- ***ACL_ADD_FILE***
  Same as *ACL_WRITE_DATA*
- ***ACL_APPEND_DATA***
  Permission to create directories. Ignored for files
- ***ACL_ADD_SUBDIRECTORY***
  Same as *ACL_APPEND_DATA*
- ***ACL_READ_NAMED_ATTRS***
  Ignored
- ***ACL_WRITE_NAMED_ATTRS***
  Ignored
- ***ACL_EXECUTE***
  Execute permission
- ***ACL_DELETE_CHILD***
  Permission to delete files and subdirectories
- ***ACL_READ_ATTRIBUTES***
  Permission to read basic attributes
- ***ACL_WRITE_ATTRIBUTES***
  Permission to change basic attributes
- ***ACL_DELETE***
  Permission to delete the object this ACL is placed on
- ***ACL_READ_ACL***
  Permission to read ACL
- ***ACL_WRITE_ACL***
  Permission to change the ACL and file mode
- ***ACL_SYNCHRONIZE***
  Ignored

clear()

Clear all permissions from the permission set

delete(*perm*)

Delete a permission from the permission set.

This function deletes the permission contained in the argument perm from the permission set. An attempt to delete a permission that is not contained in the permission set is not considered an error.

For the valid permissions see *add()*.

test(*perm*)

Test if a permission exists in the permission set.

For the permissions see *add()*.

__int__()

Convert the permission set losslessly to an `int`[35]

__index__()

Convert losslessly to an `int`[36] and support `bin()`[37], `hex()`[38] or `oct()`[39]

__eq__(*other*)

Implement self != other by value

__ne__(*other*)

Implement self != other by value

__hash__ = None

Hashing is explcitely *not* supported: the type is not immutable

__bool__()

A Permset evaluates to `True` iff any of its permission bits is set.

__str__()

Return str(self).

__repr__()

Return repr(self).

class nfs4acl.freebsd.acl.**Flagset**(*entry*)

Type which represents the flagset in an NSFv4 ACL entry.

An instance of this class is just a descriptor into the flagset of the entry. Thus all operations operate *directly* on the entry.

In boolean context a Permset evaluates to `True` iff any of its flag bits is set.

This is exactly the behaviour of *Permset*.

[35] https://docs.python.org/3/library/functions.html#int
[36] https://docs.python.org/3/library/functions.html#int
[37] https://docs.python.org/3/library/functions.html#bin
[38] https://docs.python.org/3/library/functions.html#hex
[39] https://docs.python.org/3/library/functions.html#oct

Because an instance of this class is *mutable* hashing is explicitely *not supported*.

Parameters
entry (Entry) – the ACE this flagset belongs to.

property parent
The parent ACL entry

add(*flag*)
Add a flag to the flagset.

This function adds the Flag contained in the argument *flag* to the flag set.

It is not considered an error to attempt to add flags that already exist in the flagset.

Valid values are:

- ***ACL_ENTRY_FILE_INHERIT***
  Will be inherited by files
- ***ACL_ENTRY_DIRECTORY_INHERIT***
  Will be inherited by directories
- ***ACL_ENTRY_NO_PROPAGATE_INHERIT***
  Will not propagate
- ***ACL_ENTRY_INHERIT_ONLY***
  Inherit-only
- ***ACL_ENTRY_INHERITED***
  Inherited from parent

clear()
Clear all NFSv4 ACL flags from the current flagset

delete(*flag*)
Delete a NFSv4 flag from the flagset.

An attempt to delete a permission that is not contained in the permission set is not considered an error.

For the valid flags see *add()*

test(*flag*)
Test if a flag exists in the flagset.

For the valid flags see *add()*.

__int__()
Get the flagset as native int

__index__()
Convert loslessly to an int[40] and support bin()[41], hex()[42] or oct()[43]

__eq__(*other*)
Implement self == other by value

__ne__(*other*)
Implement self != other by value

__hash__ = None
Hashing is explcitely *not* supported: the type is not immutable

__bool__()
A Flagset evaluates to `True` iff any of its flag bits is set.

__str__()
Return str(self).

__repr__()
Return repr(self).

3.2.2 Module `nfs4acl.freebsd.const`

FreeBSD specific public constants from the `/usr/include/sys/acl.h` header.

> **Note**
> This module is meant to be accessed by clients via its alias `nfs4acl.const`.

`nfs4acl.freebsd.const.`**ACL_UNDEFINED_TAG = 0**
A newly created entry has this tag

`nfs4acl.freebsd.const.`**ACL_USER_OBJ = 1**
Permissions apply to file owner (aka *owner@*)

`nfs4acl.freebsd.const.`**ACL_USER = 2**
Permissions apply to additional user specified by qualifier

`nfs4acl.freebsd.const.`**ACL_GROUP_OBJ = 4**
Permissions apply to file group (aka *group@*)

`nfs4acl.freebsd.const.`**ACL_GROUP = 8**
Permissions apply to additional group specified by qualifier

`nfs4acl.freebsd.const.`**ACL_MASK = 16**
Permissions specify mask.

Invalid for NFSv4 ACLs. Do *not* use *ACL_MASK*.

[40] https://docs.python.org/3/library/functions.html#int
[41] https://docs.python.org/3/library/functions.html#bin
[42] https://docs.python.org/3/library/functions.html#hex
[43] https://docs.python.org/3/library/functions.html#oct

nfs4acl.freebsd.const.**ACL_OTHER = 32**

Permissions apply to other

nfs4acl.freebsd.const.**ACL_OTHER_OBJ = 32**

Same as *ACL_OTHER*

nfs4acl.freebsd.const.**ACL_EVERYONE = 64**

Permissions apply to everyone (aka *everyone@*)

nfs4acl.freebsd.const.**ACL_UNDEFINED_ID = -1**

A newly created ACL has this principal (qualifier) which is **not** a real id on the system.

For entries other than *ACL_USER* and *ACL_GROUP*, this field should be set to *ACL_UNDEFINED_ID*.

nfs4acl.freebsd.const.**ACL_ENTRY_TYPE_ALLOW = 256**

allow type entry.

Allow principal (qualifier) to perform actions requiring permissions.

nfs4acl.freebsd.const.**ACL_ENTRY_TYPE_DENY = 512**

deny type entry.

Deny principal (qualifier) from performing actions requiring permissions.

nfs4acl.freebsd.const.**ACL_ENTRY_TYPE_AUDIT = 1024**

Audit.

Log any attemted access by the principal (qualifier) which requires permissions. Requires one or both of the successful-access and failed-access flags.

Not yet supported on FreeBSD.

nfs4acl.freebsd.const.**ACL_ENTRY_TYPE_ALARM = 2048**

Alarm.

Generate a system alarm at any attempted access by the principal (qualifier) which requires permissions. Requires one or both of the successful-access and failed-access flags.

Not yet supported on FreeBSD

nfs4acl.freebsd.const.**ACL_EXECUTE = 1**

Execute (files) / change-directory (directories)

nfs4acl.freebsd.const.**ACL_READ_DATA = 8**

Read-data (files) / list-directory (directories)

nfs4acl.freebsd.const.**ACL_LIST_DIRECTORY = 8**

The same as *ACL_READ_DATA*

nfs4acl.freebsd.const.**ACL_WRITE_DATA = 16**
: Write-data (files) / create-file (directories)

nfs4acl.freebsd.const.**ACL_ADD_FILE = 16**
: The same as *ACL_WRITE_DATA*

nfs4acl.freebsd.const.**ACL_APPEND_DATA = 32**
: Append-data (files) / create-subdirectory (directories)

nfs4acl.freebsd.const.**ACL_ADD_SUBDIRECTORY = 32**
: The same as *ACL_APPEND_DATA*

nfs4acl.freebsd.const.**ACL_READ_NAMED_ATTRS = 64**
: Read the named attributes of the file/directory.

  Ignored on FreeBSD.

nfs4acl.freebsd.const.**ACL_WRITE_NAMED_ATTRS = 128**
: Write the named attributes of the file/directory.

  Ignored on FreeBSD.

nfs4acl.freebsd.const.**ACL_DELETE_CHILD = 256**
: Remove a file or subdirectory from within the given directory (directories only)

nfs4acl.freebsd.const.**ACL_READ_ATTRIBUTES = 512**
: Read the attributes of the file/directory

nfs4acl.freebsd.const.**ACL_WRITE_ATTRIBUTES = 1024**
: Write the attributes of the file/directory

nfs4acl.freebsd.const.**ACL_DELETE = 2048**
: Delete the file/directory this ACL is placed on.

  Some servers will allow a delete to occur if either this permission is set in the file/directory or if the delete-child permission is set in its parent direcory.

nfs4acl.freebsd.const.**ACL_READ_ACL = 4096**
: Allow to read the file/directory NFSv4 ACL

nfs4acl.freebsd.const.**ACL_WRITE_ACL = 8192**
: Allow to write the file/directory NFSv4 ACL and the file mode

nfs4acl.freebsd.const.**ACL_WRITE_OWNER = 16384**
: Allow change of ownership of the file or directory

nfs4acl.freebsd.const.**ACL_SYNCHRONIZE = 32768**
: Allow clients to use synchronous I/O with the server.

  Ignored on FreeBSD.

nfs4acl.freebsd.const.**NFS4_STR_PERM_NAMES = [(8, 'read_data', 'r'), (8, 'list_directory', None), (16, 'write_data', 'w'), (16, 'add_file', None), (1, 'execute', 'x'), (32, 'append_data', 'p'), (32, 'add_subdirectory', None), (256, 'delete_child', 'D'), (2048, 'delete', 'd'), (512, 'read_attributes', 'a'), (1024, 'write_attributes', 'A'), (64, 'read_xattr', 'R'), (128, 'write_xattr', 'W'), (4096, 'read_acl', 'c'), (8192, 'write_acl', 'C'), (16384, 'write_owner', 'o'), (32768, 'synchronize', 's'), (65529, 'full_set', ''), (40953, 'modify_set', ''), (4680, 'read_set', ''), (1200, 'write_set', '')]**

The long and short descriptions for the valid permissions in a *Permset* in the canonical order of their output in the text representation

nfs4acl.freebsd.const.**ACL_ENTRY_FILE_INHERIT = 1**

Newly-created files will inherit the ACE, minus its inheritance flags. Newly-created subdirectories will inherit the ACE; if directory-inherit is not also specified in the parent ACE, inherit-only will be added to the inherited ACE.

nfs4acl.freebsd.const.**ACL_ENTRY_DIRECTORY_INHERIT = 2**

Newly created subdirectories will inherit the ACE

nfs4acl.freebsd.const.**ACL_ENTRY_NO_PROPAGATE_INHERIT = 4**

Newly-created subdirectories will inherit the ACE, minus its inheritance flags.

nfs4acl.freebsd.const.**ACL_ENTRY_INHERIT_ONLY = 8**

The ACE is not considered in permissions checks, but it is heritable; however, the inherit-only flag is stripped from inherited ACEs.

nfs4acl.freebsd.const.**ACL_ENTRY_SUCCESSFUL_ACCESS = 16**

Trigger an alarm/audit when the principal (qualifier) is allowed to perform an action covered by permissions (only valid for AUDIT and ALARM type ACEs)

nfs4acl.freebsd.const.**ACL_ENTRY_FAILED_ACCESS = 32**

Trigger an alarm/audit when the principal (qualifier) is prevented from performing an action covered by permissions (only valid for AUDIT and ALARM type ACEs)

nfs4acl.freebsd.const.**ACL_ENTRY_INHERITED = 128**

Inherited from parent.

This flag is set on an ACE that has been inherited from its parent.

It may also be set programmatically, and is valid on both files and directories.

nfs4acl.freebsd.const.**NFS4_STR_FLAG_NAMES = [(1, 'file_inherit', 'f'), (2, 'dir_inherit', 'd'), (8, 'inherit_only', 'i'), (4, 'no_propagate', 'n'), (16, 'successfull_access', 'S'), (32, 'failed_access', 'F'), (128, 'inherited', 'I')]**

The long and short descriptions for the valid flags in a *Flagset* in the canonical order of their output in the text representation

nfs4acl.freebsd.const.**ACL_TEXT_VERBOSE = 1**

Format ACL using verbose form

nfs4acl.freebsd.const.**ACL_TEXT_NUMERIC_IDS = 2**

Do not resolve IDs into user or group names

nfs4acl.freebsd.const.**ACL_TEXT_APPEND_ID = 4**

In addition to user and group names, append numeric IDs

3.2.3 Module nfs4acl.freebsd.types

Definitions of low-level types to access the native ACL library.

nfs4acl.freebsd.types.**uid_t**

alias of c_uint[44]

nfs4acl.freebsd.types.**gid_t**

alias of c_uint[45]

nfs4acl.freebsd.types.**acl_tag_t**

alias of c_uint[46]

nfs4acl.freebsd.types.**acl_perm_t**

alias of c_uint[47]

nfs4acl.freebsd.types.**acl_entry_type_t**

alias of c_ushort[48]

nfs4acl.freebsd.types.**acl_flag_t**

alias of c_ushort[49]

nfs4acl.freebsd.types.**acl_type_t**

alias of c_int[50]

nfs4acl.freebsd.types.**acl_permset_t**

alias of LP_c_int

nfs4acl.freebsd.types.**acl_flagset_t**

alias of LP_c_ushort

nfs4acl.freebsd.types.**acl_entry_t**

alias of c_void_p[51]

[44] https://docs.python.org/3/library/ctypes.html#ctypes.c_uint
[45] https://docs.python.org/3/library/ctypes.html#ctypes.c_uint
[46] https://docs.python.org/3/library/ctypes.html#ctypes.c_uint
[47] https://docs.python.org/3/library/ctypes.html#ctypes.c_uint
[48] https://docs.python.org/3/library/ctypes.html#ctypes.c_ushort
[49] https://docs.python.org/3/library/ctypes.html#ctypes.c_ushort
[50] https://docs.python.org/3/library/ctypes.html#ctypes.c_int
[51] https://docs.python.org/3/library/ctypes.html#ctypes.c_void_p

nfs4acl.freebsd.types.**acl_t**

alias of c_void_p[52]

3.2.4 Module nfs4acl.freebsd.ffi

The low-level call interface to the native FreeBSD ACL library in its *libc*.

All the heavy interaction with the ctypes[53] package is done here.

[52] https://docs.python.org/3/library/ctypes.html#ctypes.c_void_p
[53] https://docs.python.org/3/library/ctypes.html#module-ctypes

CHAPTER 4

Utilities

These are the applications provided as part of pylibnfs4acl.

4.1 getacl

Get NFSv4 ACL information

4.1.1 Synopsis

getacl [*options*] [*file* ...]

4.1.2 Description

The **getacl** utility writes discretionary access control information associated with the specified file(s) to standard output. If an ACL is not in effect for a *file* then the standard discretionary access permissions are interpreted as an ACL containing only the required (i.e. trivial) ACL entries.

4.1.3 Options

-H, --half-logical

If *--recursive* is specified, follow symbolic links named on the command line only. Symbolic links encountered during traversal are not followed.

-h, --no-dereference

If the command line target of the operation is a symbolic link, return the ACL from the symbolic link itself rather than following the link.

Only effective if *--recursive* is not specified.

--help

Print this help message to stdout and exit.

-i

Append the numerical ID at the end of each ACL entry containing user or group names.

-L, --logical

If *--recursive* is specified, all symbolic links are followed.

-n, --numeric

Display user and group IDs numerically rather than converting to a user or group name.

-P, --physical

If *--recursive* is specified, no symbolic links are followed. This is the default.

-q, --omit-header

Do not write commented information about file name and ownership. This is useful when dealing with filenames with unprintable characters.

-R, --recursive

Operate recursively on files and directories.

-v

Display permissions and flags in a verbose form.

--version

Print the version of **getacl** to stdout and exit.

-x

Do not traverse file system mount points. Only effective if *--recursive* is given also.

The following operand is available:

file

A pathname of a file whose ACL shall be retrieved. If file is not specified, or a file is specified as -, then **getacl** reads a list of pathnames, each terminated by one newline character, from the standard input.

4.1.4 See Also

getfacl(1)[54], *setfacl(1)*[55], *acl(3)*[56], *acl(9)*[57], *symlink(7)*[58]

[54] https://www.freebsd.org/cgi/man.cgi?query=getfacl&sektion=1
[55] https://www.freebsd.org/cgi/man.cgi?query=setfacl&sektion=1
[56] https://www.freebsd.org/cgi/man.cgi?query=acl&sektion=3
[57] https://www.freebsd.org/cgi/man.cgi?query=acl&sektion=9
[58] https://www.freebsd.org/cgi/man.cgi?query=symlink&sektion=7

4.1.5 Standards

The **getacl** utility is intended to be compliant with the general symbolic link handling for tools as described in *symlink(7)*[59].

[59] https://www.freebsd.org/cgi/man.cgi?query=symlink&sektion=7

CHAPTER 5

Changes

All major changes over the versions are listed here. For breaking changes have a look at *Breaking Changes*, they are listed there in detail.

5.1 Changelog

5.1.1 Pre-1.0 Series

0.2.0 (2026-06-10)

- **[feature]**
  Implemented a **`getacl`** utility to get NFSv4 ACL information. This is a somewhat extended version of *`getfacl(1)`*[60] with regard to recursive file tree traversal and symbolic link handling. But it does *not* handle POSIX ACLs.
- **[feature]**
  Implemented *`nfs4acl.freebsd.acl.ACL.__getstate__()`* and *`nfs4acl.freebsd.acl.ACL.__setstate__()`* for proper `pickle`[61] support. This is obviously platform-dependent!
- **[feature]**
  Implemented proper checks to protect against dependent item misuse:
  - Check for operations on closed ACLs.
  - Check whether an ACL could have been structurally rearranged by the low-level functions; if so invalidate previous entries and their associated permission and

[60] https://www.freebsd.org/cgi/man.cgi?query=getfacl&sektion=1
[61] https://docs.python.org/3/library/pickle.html#module-pickle

flag sets.

The checks raise a `ValueError`[62] in a similar way to Python operations on closed files.

- **[doc]**
  Documentation is fairly completed.
- **[doc]**
  Updated to more modern documentation standards and tool versions; documentation is now in `docs/`.
- **[misc]**
  Update license information to modern REUSE standards and use SPDX identifiers.

0.1.0 (2018-07-27)

- **[feature]**
  First fairly complete and really usable release.

5.2 Breaking Changes

None yet.

[62] https://docs.python.org/3/library/exceptions.html#ValueError

CHAPTER 6

Authors

The package *nfs4acl* is written and maintained by Franz Glasner.

CHAPTER 7

License

The license is the GNU Lesser General Public License v2.1 or later (SPDX identifier: `LGPL-2.1-or-later`).

```
                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it.  You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
```

(continues on next page)

(continued from previous page)

```
strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use,
not price.  Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights.  These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you.  You must make sure that they, too, receive or can get the source
code.  If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it.  And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that
there is no warranty for the free library.  Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

  Finally, software patents pose a constant threat to the existence of
any free program.  We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder.  Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

  Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License.  This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License.  We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.
```

(continues on next page)

(continued from previous page)

```
  When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library.  The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom.  The Lesser General
Public License permits more lax criteria for linking other code with
the library.

  We call this license the ”Lesser” General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License.  It also provides other free software developers Less
of an advantage over competing non-free programs.  These disadvantages
are the reason we use the ordinary General Public License for many
libraries.  However, the Lesser license provides advantages in certain
special circumstances.

  For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it becomes
a de-facto standard.  To achieve this, non-free programs must be
allowed to use the library.  A more frequent case is that a free
library does the same job as widely used non-free libraries.  In this
case, there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

  In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software.  For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

  Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is
linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

  The precise terms and conditions for copying, distribution and
modification follow.  Pay close attention to the difference between a
”work based on the library” and a ”work that uses the library”.  The
former contains code derived from the library, whereas the latter must
be combined with the library in order to run.

                  GNU LESSER GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License Agreement applies to any software library or other
```

(continues on next page)

(continued from previous page)

```
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called ”this License”).
Each licensee is addressed as ”you”.

  A ”library” means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

  The ”Library”, below, refers to any such software library or work
which has been distributed under these terms.  A ”work based on the
Library” means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language.  (Hereinafter, translation is
included without limitation in the term ”modification”.)

  ”Source code” for a work means the preferred form of the work for
making modifications to it.  For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

  Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope.  The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
writing it).  Whether that is true depends on what the Library does
and what the program that uses the Library does.

  1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact
all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

  You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.

  2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:
```

(continues on next page)

(continued from previous page)

```
    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices
    stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no
    charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a
    table of data to be supplied by an application program that uses
    the facility, other than as an argument passed when the facility
    is invoked, then you must make a good faith effort to ensure that,
    in the event an application does not supply such function or
    table, the facility still operates, and performs whatever part of
    its purpose remains meaningful.

    (For example, a function in a library to compute square roots has
    a purpose that is entirely well-defined independent of the
    application.  Therefore, Subsection 2d requires that any
    application-supplied function or table used by this function must
    be optional: if the application does not supply it, the square
    root function must still compute square roots.)

These requirements apply to the modified work as a whole.  If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works.  But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library.  To do
```

(continues on next page)

(continued from previous page)

```
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License.  (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.)  Do not make any other change in
these notices.

  Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library".  Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library".  The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library.  The
threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data
```

(continues on next page)

(continued from previous page)

```
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work.  (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

  Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

  6. As an exception to the Sections above, you may also combine or
link a ”work that uses the Library” with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

  You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License.  You must supply a copy of this License.  If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License.  Also, you must do one
of these things:

    a) Accompany the work with the complete corresponding
    machine-readable source code for the Library including whatever
    changes were used in the work (which must be distributed under
    Sections 1 and 2 above); and, if the work is an executable linked
    with the Library, with the complete machine-readable ”work that
    uses the Library”, as object code and/or source code, so that the
    user can modify the Library and then relink to produce a modified
    executable containing the modified Library.  (It is understood
    that the user who changes the contents of definitions files in the
    Library will not necessarily be able to recompile the application
    to use the modified definitions.)

    b) Use a suitable shared library mechanism for linking with the
    Library.  A suitable mechanism is one that (1) uses at run time a
    copy of the library already present on the user's computer system,
    rather than copying library functions into the executable, and (2)
    will operate properly with a modified version of the library, if
    the user installs one, as long as the modified version is
    interface-compatible with the version that the work was made with.

    c) Accompany the work with a written offer, valid for at
```

(continues on next page)

(continued from previous page)

```
    least three years, to give the same user the materials
    specified in Subsection 6a, above, for a charge no more
    than the cost of performing this distribution.

    d) If distribution of the work is made by offering access to copy
    from a designated place, offer equivalent access to copy the above
    specified materials from the same place.

    e) Verify that the user has already received a copy of these
    materials or that you have already sent this user a copy.

  For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it.  However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

  It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system.  Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

  7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

    a) Accompany the combined library with a copy of the same work
    based on the Library, uncombined with any other library
    facilities.  This must be distributed under the terms of the
    Sections above.

    b) Give prominent notice with the combined library of the fact
    that part of it is a work based on the Library, and explaining
    where to find the accompanying uncombined form of the same work.

  8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License.  Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License.  However, parties who have received copies,
```

(continues on next page)

(continued from previous page)

```
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

  9. You are not required to accept this License, since you have not
signed it.  However, nothing else grants you permission to modify or
distribute the Library or its derivative works.  These actions are
prohibited by law if you do not accept this License.  Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

  10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions.  You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

  11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all.  For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices.  Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.
```

(continues on next page)

(continued from previous page)

```
This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded.  In such case, this License incorporates the limitation as if
written in the body of this License.

  13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number.  If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation.  If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

  14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission.  For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this.  Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

                            NO WARRANTY

  15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU.  SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
```

(continues on next page)

(continued from previous page)

```
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

                     END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms
of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It
is safest to attach them to the start of each source file to most
effectively convey the exclusion of warranty; and each file should
have at least the "copyright" line and a pointer to where the full
notice is found.

<one line to give the library's name and an idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify
it under the terms of the GNU Lesser General Public License as
published by the Free Software Foundation; either version 2.1 of the
License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License
along with this library; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or
your school, if any, to sign a "copyright disclaimer" for the library,
```

(continues on next page)

```
if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library
`Frob' (a library for tweaking knobs) written by James Random Hacker.
```

```
<signature of Ty Coon >, 1 April 1990
Ty Coon, President of Vice
That's all there is to it!
```

CHAPTER 8

TODO List

Todo

Platform-independent pickling style.

Is platform-independency really required? Use some normalized text representation for this instead (e.g. using *getfacl(1)*[1] and friends)?

original entry

Todo

Normalize the string representation before comparing: for each line: strip white space et al.

original entry

[1] https://www.freebsd.org/cgi/man.cgi?query=getfacl&sektion=1

Python Module Index

n

Index

Symbols

A

B

C

D

E

F

G

I

M

N

P

Q

S

T

U

V