mac_alias Documentation

Release 2.0.0

Alastair Houghton

Jul 27, 2017

Contents

1 What is this? 3

2 Usage 5

3 Code Documentation 7
3.1 mac_alias package 7

4 Binary Formats 11
4.1 Mac Alias Format 11
4.2 Mac Bookmark Format 12

5 Indices and tables 15

This document refers to version 2.0.0

CHAPTER 1

What is this?

The Mac OS has a special data structure it calls an Alias, which allows programs that make use of it to locate the file to which it refers more reliably than they would be able to from e.g. a filename alone.

The format of this structure is not documented, so until recently you would have used Mac OS X APIs to construct and process Alias records. Sadly, Apple has deprecated the APIs in question in favour of its new "Bookmark" functionality; this is understandable, but it makes it tricky to construct an Alias record reliably in future.

This module contains code to parse and generate Alias records from a Pythonic equivalent data structure, and does not rely on the deprecated APIs.

It also contains code to parse and generate Bookmark records, again from a Pythonic equivalent, and without relying on OS X APIs.

CHAPTER 2

Usage

To parse an Alias record given binary data:

```
from mac_alias import Alias

a = Alias.from_bytes(my_data)
```

To generate a binary Alias record:

```
a.to_bytes()
```

Finally, to build an *Alias* for a file:

```
Alias.for_file('/path/to/file.ext')
```

It's probably best to resist the temptation to mess with the Alias class too much otherwise.

Similarly, to parse a Bookmark record given binary data:

```
from mac_alias import Bookmark

b = Bookmark.from_bytes(my_data)
```

To generate a binary Bookmark record:

```
b.to_bytes()
```

And to build a *Bookmark* for a file:

```
Bookmark.for_file('/path/to/file.ext')
```

CHAPTER 3

Code Documentation

Contents:

mac_alias package

Classes

class mac_alias.**Alias**(*appinfo='x00x00x00x00'*, *version=2*, *volume=None*, *target=None*, *extra=[]*)

appinfo = None
Application specific information (four byte byte-string)

extra = None
A list of extra *(tag, value)* pairs

classmethod for_file(*path*)
Create an *Alias* that points at the specified file.

classmethod from_bytes(*bytes*)
Construct an *Alias* object given binary Alias data.

target = None
A *TargetInfo* object describing the target

to_bytes()
Returns the binary representation for this *Alias*.

version = None
Version (we support only version 2)

volume = None
A VolumeInfo object describing the target's volume

class mac_alias.**AppleShareInfo**(*zone=None*, *server=None*, *user=None*)

server = None
The AFP server

user = None
The username

zone = None
The AppleShare zone

class mac_alias.**TargetInfo**(*kind*, *filename*, *folder_cnid*, *cnid*, *creation_date*, *creator_code*, *type_code*, *levels_from=-1*, *levels_to=-1*, *folder_name=None*, *cnid_path=None*, *carbon_path=None*, *posix_path=None*, *user_home_prefix_len=None*)

carbon_path = None
The Carbon path of the target *(optional)*

cnid = None
The CNID (Catalog Node ID) of the target

cnid_path = None
The path from the volume root as a sequence of CNIDs. *(optional)*

creation_date = None
The target's *creation* date.

creator_code = None
The target's Mac creator code (a four-character binary string)

filename = None
The filename of the target

folder_cnid = None
The CNID (Catalog Node ID) of the target's containing folder; CNIDs are similar to but different than traditional UNIX inode numbers

folder_name = None
The (POSIX) name of the target's containing folder. *(optional)*

kind = None
Either ALIAS_KIND_FILE or ALIAS_KIND_FOLDER

levels_from = None
The depth of the alias? Always seems to be -1 on OS X.

levels_to = None
The depth of the target? Always seems to be -1 on OS X.

posix_path = None
The POSIX path of the target relative to the volume root. Note that this may or may not have a leading '/' character, but it is always relative to the containing volume. *(optional)*

type_code = None
The target's Mac type code (a four-character binary string)

user_home_prefix_len = None
If the path points into a user's home folder, the number of folders deep that we go before we get to that home folder. *(optional)*

class mac_alias.**Bookmark**(*tocs=None*)

classmethod for_file(*path*)
Construct a *Bookmark* for a given file.

classmethod from_bytes(*data*)
Create a *Bookmark* given byte data.

get(*key*, *default=None*)
Lookup the value for a given key, returning a default if not present.

to_bytes()
Convert this *Bookmark* to a byte representation.

tocs = None
The TOCs for this Bookmark

class mac_alias.**Data**(*bytedata=None*)

bytes = None
The bytes, stored as a byte string

class mac_alias.**URL**(*base*, *rel=None*)

absolute
Return an absolute URL.

base = None
The base URL, if any (a *URL*)

relative = None
The rest of the URL (a string)

Constants

mac_alias.**ALIAS_KIND_FILE**
mac_alias.**ALIAS_KIND_FOLDER**
Values for the kind attribute.

mac_alias.**ALIAS_HFS_VOLUME_SIGNATURE**
The volume signature for HFS+.

mac_alias.**ALIAS_FIXED_DISK**
mac_alias.**ALIAS_NETWORK_DISK**
mac_alias.**ALIAS_400KB_FLOPPY_DISK**
mac_alias.**ALIAS_800KB_FLOPPY_DISK**
mac_alias.**ALIAS_1_44MB_FLOPPY_DISK**
mac_alias.**ALIAS_EJECTABLE_DISK**
Disk type constants.

mac_alias.**ALIAS_NO_CNID**
A constant used where no CNID is present.

mac_alias.**kBookmarkPath**
mac_alias.**kBookmarkCNIDPath**
mac_alias.**kBookmarkFileProperties**
mac_alias.**kBookmarkFileName**
mac_alias.**kBookmarkFileID**
mac_alias.**kBookmarkFileCreationDate**
mac_alias.**kBookmarkTOCPath**
mac_alias.**kBookmarkVolumePath**
mac_alias.**kBookmarkVolumeURL**
mac_alias.**kBookmarkVolumeName**

mac_alias.**kBookmarkVolumeUUID**
mac_alias.**kBookmarkVolumeSize**
mac_alias.**kBookmarkVolumeCreationDate**
mac_alias.**kBookmarkVolumeProperties**
mac_alias.**kBookmarkContainingFolder**
mac_alias.**kBookmarkUserName**
mac_alias.**kBookmarkUID**
mac_alias.**kBookmarkWasFileReference**
mac_alias.**kBookmarkCreationOptions**
mac_alias.**kBookmarkURLLengths**
mac_alias.**kBookmarkSecurityExtension**

Bookmark data keys. A Bookmark holds a set of TOCs (Tables of Contents), each of which maps a set of keys to a set of values. The keys are either numeric, like the ones represented by the above constants, or strings.

Bookmarks can hold strings, byte data, numbers, dates, booleans, arrays, dicts, UUIDs, URLs and NULLs (represented by Python None). If you store data in a bookmark using the string key functionality, the documentation for CF/NSURL recommends using reverse DNS for the keys to avoid clashes.

CHAPTER 4

Binary Formats

Mac Alias Format

Everything below is big-endian.

An Alias record starts as follows:

Offset	Size	Contents
0	4	Application specific four-character code
4	2	Record size (must be >= 150 bytes)
6	2	Version (we support version 2)
8	2	Alias kind (0 = file, 1 = folder)
10	28	Volume name (Pascal-style string; first octet gives length)
38	4	Volume date (seconds since 1904-01-01 00:00:00 UTC)
42	2	Filesystem type (typically 'H+' for HFS+)
44	2	Disk type (0 = fixed, 1 = network, 2 = 400Kb, 3 = 800kb, 4 = 1.44MB, 5 = ejectable)
46	4	CNID of containing folder
50	64	Target name (Pascal-style string)
114	4	Target CNID
118	4	Target creation date (seconds since 1904-01-01 00:00:00 UTC)
122	4	Target creator code (four-character code)
126	4	Target type code (four-character code)
130	2	Number of directory levels from alias to root (or -1)
132	2	Number of directory levels from root to target (or -1)
134	4	Volume attributes
138	2	Volume filesystem ID
140	10	Reserved (set to zero)

This record is optionally followed by tag-length-value data:

Offset	Size	Contents
0	2	Tag
2	2	Length
4	Length	Value

If the length is odd, a pad byte is added at the end.

Valid tags are:

Tag	Contents
-1	Signifies the end of the alias record
0	Carbon folder name (a string)
1	CNID path (an array of CNIDs, one per directory)
2	Carbon path (a string)
3	AppleShare zone (a string)
4	AppleShare server name (a string)
5	AppleShare username (a string)
6	Driver name (a string)
9	Network mount information
10	Dial-up connection information
14	Unicode filename of target (a UTF-16 big endian string)
15	Unicode volume name (a UTF-16 big endian string)
16	High resolution volume creation date (65536ths of a second since 1904-01-01 00:00:00 UTC)
17	High resolution creation date (65536ths of a second since 1904-01-01 00:00:00 UTC)
18	POSIX path (a string)
19	POSIX path to volume mountpoint (a string)
20	Recursive alias of disk image (an alias record)
21	User home length prefix (two-byte integer, says how many directory levels to the user's home folder)

Mac Bookmark Format

Everything below is little-endian unless otherwise mentioned.

The Bookmark format is a more modern alternative to the alias record. Bookmarks consist of a set of dictionaries mapping keys to values; each dictionary has its own Table of Contents (TOC) structure.

The record starts with a header:

Offset	Size	Contents
0	4	Magic number ('book')
4	4	Total size in bytes
8	4	Unknown (0x10040000) - might be a version?
12	4	Size of header (48)
16	32	Reserved

All offsets stored in the file are relative to the *end* of this header.

This is immediately followed at location 48 by a 4-byte offset to the first TOC structure. It seems odd that this is not part of the header, but for some reason best known to the engineers at Apple, it isn't.

A TOC starts with its own header:

Offset	Size	Contents
0	4	Size of TOC in bytes, minus 8
4	4	Magic number (0xfffffffe)
8	4	Identifier (just a number)
12	4	Next TOC offset (or 0 if none)
16	4	Number of entries in this TOC

This is followed by an array of TOC entries. There is code that does a binary search of the TOC structure, so they *must* be stored in *key* order. A TOC entry looks like this:

Offset	Size	Contents
0	4	Key
4	4	Offset to data record
8	4	Reserved (0)

If the key has its top bit set (0x80000000), then (key & 0x7fffffff) gives the offset of a string record.

Each data record has the following fields:

Offset	Size	Contents
0	4	Length of data (n)
4	4	Type
8	n	Data bytes

Known data types are as follows:

Code	Type	Encoding
0x0101	String	UTF-8
0x0201	Data	Raw bytes
0x0301	Number (signed 8-bit)	1-byte number
0x0302	Number (signed 16-bit)	2-byte number
0x0303	Number (signed 32-bit)	4-byte number
0x0304	Number (signed 64-bit)	8-byte number
0x0305	Number (32-bit float)	IEEE single precision
0x0306	Number (64-bit float)	IEEE double precision
0x0400	Date	*Big-endian* IEEE double precision seconds since 2001-01-01 00:00:00 UTC
0x0500	Boolean (false)	No data
0x0501	Boolean (true)	No data
0x0601	Array	Array of 4-byte offsets to data items
0x0701	Dictionary	Array of pairs of 4-byte (key, value) data item offsets
0x0801	UUID	Raw bytes
0x0901	URL	UTF-8 string
0x0902	URL (relative)	4-byte offset to base URL, 4-byte offset to UTF-8 string

The first TOC in the file generally has its identifier set to 1. As mentioned, the keys in each TOC can be strings, in which case the key field will contain the offset to the string, or they can be certain special values. Currently known values are:

Key	Meaning	Value
0x1003	Unknown	Unknown
0x1004	Target path	Array of individual path components
0x1005	Target CNID path	Array of CNIDs
0x1010	Target flags	Data - see below
0x1020	Target filename	String
0x1030	Target CNID	4-byte integer
0x1040	Target creation date	Date
0x1054	Unknown	Unknown
0x1055	Unknown	Unknown
0x1056	Unknown	Unknown
0x1101	Unknown	Unknown
0x1102	Unknown	Unknown
0x2000	TOC path	Array - see below
0x2002	Volume path	Array of individual path components
0x2005	Volume URL	URL of volume root
0x2010	Volume name	String

Continued on next page

Table 4.1 – continued from previous page

Key	Meaning	Value
0x2011	Volume UUID	String (not a UUID!)
0x2012	Volume size	8-byte integer
0x2013	Volume creation date	Date
0x2020	Volume flags	Data - see below
0x2030	Volume is root	True if the volume was the filesystem root
0x2040	Volume bookmark	TOC identifier for disk image
0x2050	Volume mount point	URL
0x2070	Unknown	Unknown
0xc001	Containing folder index	Integer index of containing folder in target path array
0xc011	Creator username	Name of user that created bookmark
0xc012	Creator UID	UID of user that created bookmark
0xd001	File reference flag	True if creating URL was a file reference URL
0xd010	Creation options	Integer containing flags passed to CFURLCreateBookmarkData
0xe003	URL length array	Array of integers - see below
0xf017	Localized name?	String?
0xf022	Unknown	Unknown
0xf080	Security extension	Unknown but looks like a hash with data and an access right
0xf081	Unknown	Unknown

The target flags (0x1010) are encoded as a Data object containing three 8-byte integers. The first contains flags describing the target; the second says which flags are valid, and the third appears to always be zero. Supported flags can be found in CFURLPriv.h, which is part of CF-Lite; for the target flags field, it's the "resource property flags" that are valid.

Similarly the volume flags (0x2020) are encoded in the same manner, but this time it's the "volume property flags" that are interesting.

The TOC path (0x2000) is only used if there are multiple volumes between the target and the filesystem root. In that case, it contains an array, with every other item holding a TOC ID for a dictionary describing a volume; the values between TOC IDs appear to be zero. The array starts from the filesystem root.

The URL length array (0xe003) is used to indicate how the path components were originally broken up; if the URL encoded by the bookmark has a base URL, each entry in the length array gives the number of path elements that come from that base URL.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Index

A

absolute (mac_alias.URL attribute), 9
Alias (class in mac_alias), 7
ALIAS_1_44MB_FLOPPY_DISK (in module mac_alias), 9
ALIAS_400KB_FLOPPY_DISK (in module mac_alias), 9
ALIAS_800KB_FLOPPY_DISK (in module mac_alias), 9
ALIAS_EJECTABLE_DISK (in module mac_alias), 9
ALIAS_FIXED_DISK (in module mac_alias), 9
ALIAS_HFS_VOLUME_SIGNATURE (in module mac_alias), 9
ALIAS_KIND_FILE (in module mac_alias), 9
ALIAS_KIND_FOLDER (in module mac_alias), 9
ALIAS_NETWORK_DISK (in module mac_alias), 9
ALIAS_NO_CNID (in module mac_alias), 9
appinfo (mac_alias.Alias attribute), 7
AppleShareInfo (class in mac_alias), 7

B

base (mac_alias.URL attribute), 9
Bookmark (class in mac_alias), 8
bytes (mac_alias.Data attribute), 9

C

carbon_path (mac_alias.TargetInfo attribute), 8
cnid (mac_alias.TargetInfo attribute), 8
cnid_path (mac_alias.TargetInfo attribute), 8
creation_date (mac_alias.TargetInfo attribute), 8
creator_code (mac_alias.TargetInfo attribute), 8

D

Data (class in mac_alias), 9

E

extra (mac_alias.Alias attribute), 7

F

filename (mac_alias.TargetInfo attribute), 8
folder_cnid (mac_alias.TargetInfo attribute), 8
folder_name (mac_alias.TargetInfo attribute), 8
for_file() (mac_alias.Alias class method), 7
for_file() (mac_alias.Bookmark class method), 8
from_bytes() (mac_alias.Alias class method), 7
from_bytes() (mac_alias.Bookmark class method), 8

G

get() (mac_alias.Bookmark method), 9

K

kBookmarkCNIDPath (in module mac_alias), 9
kBookmarkContainingFolder (in module mac_alias), 9
kBookmarkCreationOptions (in module mac_alias), 9
kBookmarkFileCreationDate (in module mac_alias), 9
kBookmarkFileID (in module mac_alias), 9
kBookmarkFileName (in module mac_alias), 9
kBookmarkFileProperties (in module mac_alias), 9
kBookmarkPath (in module mac_alias), 9
kBookmarkSecurityExtension (in module mac_alias), 9
kBookmarkTOCPath (in module mac_alias), 9
kBookmarkUID (in module mac_alias), 9
kBookmarkURLLengths (in module mac_alias), 9
kBookmarkUserName (in module mac_alias), 9
kBookmarkVolumeCreationDate (in module mac_alias), 9
kBookmarkVolumeName (in module mac_alias), 9
kBookmarkVolumePath (in module mac_alias), 9
kBookmarkVolumeProperties (in module mac_alias), 9
kBookmarkVolumeSize (in module mac_alias), 9
kBookmarkVolumeURL (in module mac_alias), 9
kBookmarkVolumeUUID (in module mac_alias), 9
kBookmarkWasFileReference (in module mac_alias), 9
kind (mac_alias.TargetInfo attribute), 8

L

levels_from (mac_alias.TargetInfo attribute), 8
levels_to (mac_alias.TargetInfo attribute), 8

P

posix_path (mac_alias.TargetInfo attribute), 8

R

relative (mac_alias.URL attribute), 9

S

server (mac_alias.AppleShareInfo attribute), 7

T

target (mac_alias.Alias attribute), 7
TargetInfo (class in mac_alias), 8
to_bytes() (mac_alias.Alias method), 7
to_bytes() (mac_alias.Bookmark method), 9
tocs (mac_alias.Bookmark attribute), 9
type_code (mac_alias.TargetInfo attribute), 8

U

URL (class in mac_alias), 9
user (mac_alias.AppleShareInfo attribute), 8
user_home_prefix_len (mac_alias.TargetInfo attribute), 8

V

version (mac_alias.Alias attribute), 7
volume (mac_alias.Alias attribute), 7

Z

zone (mac_alias.AppleShareInfo attribute), 8