

Reading Writing ".DS_Store" Files

Version 9.0.0.10

December 15, 2025

A ".DS_Store" file is a metadata file on Mac OS X that holds information about folder and icons as viewed and manipulated in Finder. One common reason to manipulate ".DS_Store" files is to create a nice-looking disk image for a Mac OS X installer.

".DS_Store" reading and writing is based on a reverse-engineered description of the file format [DS_Store].

1 ".DS_Store" Files and Entries

```
(require ds-store)      package: ds-store-lib

| (read-ds-store path [#:verbose verbose?]) → (listof ds?)
|   path : path-string?
|   verbose? : any/c = #f
```

Reads the ".DS_Store" file at *path* returning a list of store items.

```
| (write-ds-store path dses) → void?
|   path : path-string?
|   dses : (listof ds?)
```

Writes *dses* to the ".DS_Store" file at *path*, replacing the file's current content.

```
| (struct ds (path id type data)
|   #:transparent)
|   path : (or/c path-element? 'same)
|   id : symbol?
|   type : (or/c 'long 'shor 'bool 'type 'ustr 'blob)
|   data : (or/c exact-integer? boolean? symbol? string?
|           bytes? iloc? fwind?)
```

Represents a entry in a ".DS_Store" file. A ".DS_Store" file typically has multiple entries for a single file or directory in the same directory as the ".DS_Store".

The *path* should be '*same*' only for a volume root directory; information about a directory is otherwise recorded in its parent directory's ".DS_Store" file.

The *id* symbols should each have four ASCII characters. See the ".DS_Store" format description [DS_Store] for more information *id* and *type* values.

The *data* field *long* should be an exact integer for '*long*' and '*shor*' types, a boolean for the '*bool*' type, a 4-character ASCII symbol for the '*type*' type, a string for the '*ustr*' type, and either a byte string, *iloc*, or *fwind* for the '*blob*' type.

```
| (struct iloc (x y)
|   #:transparent)
|   x : exact-integer?
|   y : exact-integer?
```

Represents an icon location for an '*Iloc*' entry.

```
| (struct fwind (t l b r mode sideview?))
```

```
#:transparent)
t : exact-integer?
l : exact-integer?
b : exact-integer?
r : exact-integer?
mode : symbol?
sideview? : any/c
```

Represent a window location for a `'fwi0` entry. The `mode` field should have four ASCII characters, and recognized `modes` include `'icnv`, `'clmv`, and `'Nlsv`.

2 Finder Aliases

A 'pict' entry in a ".DS_Store" file references a file through a Finder alias. See also [ds-store/cross-alias](#).

```
(require ds-store/alias)      package: ds-store-lib
(path->alias-bytes path [#:wrt wrt-dir]) → (or/c bytes? #f)
  path : path-string?
  wrt-dir : (or/c #f path-string?) = #f
```

Constructs a byte string to represent a Finder alias but using the "CoreFoundation" library on Mac OS.

See also [path->synthesized-alias-bytes](#).

3 Cross-Built Finder Aliases

```
(require ds-store/cross-alias)      package: ds-store-lib
```

Added in version 1.1 of package `ds-store-lib`.

```
(path->synthesized-alias-bytes
  #:volume-name volume-name
  #:file-name file-name
  #:file-inode file-inode
  #:parent-name parent-name
  #:parent-inode parent-inode
  #:file-absolute-name file-absolute-name
  #:file-absolute-path-within-volume file-absolute-path-within-volume
  #:volume-maybe-absolute-path volume-maybe-absolute-path)
→ bytes?
volume-name : string?
file-name : string?
file-inode : exact-integer?
parent-name : string?
parent-inode : exact-integer?
file-absolute-name : string?
file-absolute-path-within-volume : string?
volume-maybe-absolute-path : string?
```

Like `path->alias-bytes`, but creates alias bytes without using Mac OS libraries, which requires specifying details of the filesystem for the alias:

- `volume-name`: The name of the volume.
- `file-name`: The name of a file referenced by the alias, not including its path.
- `file-inode`: The inode the referenced file (in the same sense as the `'inode` result of `file-or-directory-stat`).
- `parent-name`: The name of the directory containing the referenced file, not including the directory's path. If the referenced file is in the volume's root directory, `parent-name` will be `volume-name`.
- `parent-inode`: The inode of the file's enclosing directory (in the same sense as the `'inode` result of `file-or-directory-stat`).
- `file-absolute-name`: The full path to the referenced file, but using Mac OS Classic path syntax, so path elements are separated by `:`s. This path starts with `volume-name` and ends with `file-name`.

- *file-absolute-path-within-volume*: The full path to the referenced file using Unix path conventions. If the referenced file is in the volume's root directory, this path is *file-name* prefixed with `/`.
- *volume-maybe-absolute-path*: A prediction of how the volume will be mounted, normally *volume-name* prefixed with `/Volumes/`.

Alias synthesis is based on a reverse-engineered description of the alias format [Alias].

Bibliography

[DS_Store] Wim Lewis and Mark Mentovai, “DS_Store Format.” <http://search.cpan.org/~wiml/Mac-Finder-DSStore/DSStoreFormat.pod>

[Alias] Wikipedia, “Alias (Mac OS).” [https://en.wikipedia.org/wiki/Alias_\(Mac_OS\)](https://en.wikipedia.org/wiki/Alias_(Mac_OS))