

Start

Alifix is a simple utility to detect and help fix broken Finder Aliases.

Since Apple overhauled them in macOS 10.12, Finder Aliases have coped well with what we do to them – moving the Alias around, changing the name and path of the item to which it points. But they do get silently broken, unable to find that item, a problem which comes to a head when you copy large folders to another volume, or clone a whole drive.

Alifix traverses your folders and files deeply to look for Aliases. If the item to which an Alias points can be located by macOS, Alifix will optionally rewrite that Alias to ensure that it is up to date. For those Aliases which Alifix can't get to point to the original item, thus being broken, Alifix identifies them and optionally writes an adjacent file revealing their internal data in plain text. If they're important, that should help you fix them.

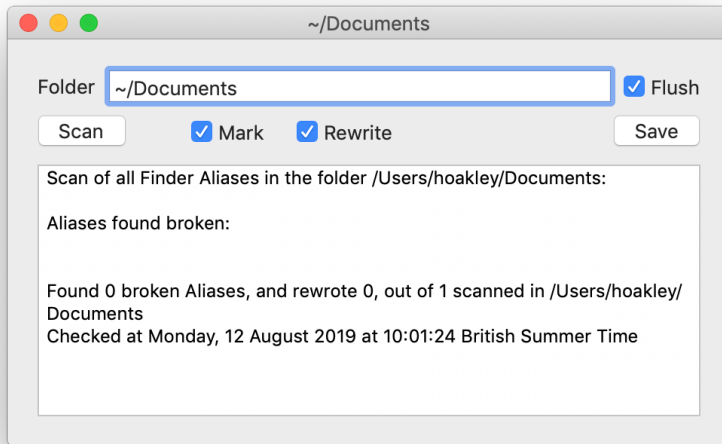
Alifix is valuable for anyone preparing a volume to be copied or cloned, and for cleaning up the copy afterwards. Fixing your Finder Aliases is also an excellent part of general disk housekeeping worth performing every few months.

→ [Details](#)

→ [Technical Information](#)

→ [Start](#)

Details



When you open Alifix, a new document window is opened; to open one yourself, use the **New** command in the **File** menu.

Select a folder to scan either by typing the path in the text box marked **Folder**, or by leaving that blank and clicking on **Scan**. Paths typed in can use the shorthand *tilde* symbol ~ meaning the Home folder, so ~/Documents refers to the Documents folder in your Home folder. If the **Flush** box is ticked, the contents of the **Folder** box will be emptied when the scan is undertaken; if you want to leave a typed-in path there, ensure the **Flush** box is empty (not ticked) before clicking on **Scan**.

To run a scan on the folder entered, click on the **Scan** button. To select a folder to scan using a standard **Open File** dialog, leave the Folder box empty and just click on **Scan**.

→ [Details continued](#)

Details continued

Two options control scan behaviour:

- If the **Mark** box is ticked, then alongside each broken Alias file, Alifix writes a new text file containing the contents of the Alias, including the path to the item to which the Alias pointed.
- If the **Rewrite** box is ticked, then each unbroken Alias is rewritten.
- If both **Mark** and **Rewrite** boxes are ticked, alongside each rewritten Alias, Alifix writes a new text file containing the contents of the Alias, including the path to the item to which the Alias pointed.

Scanning goes deep: although Alifix doesn't follow Aliases when checking files, it tries to examine every file in the folder you have selected to scan.

 ***You shouldn't normally try scanning the top-level or root folder of your startup volume, as that will take a long time***, and will consume a lot of memory. The most important folder to scan is normally your Home folder, or just the Documents folder within it.

Once scanning is complete, the lower scrolling text box displays details of all the broken Alias files found in the folder which it has scanned. Click on the **Save** button to save this as a text file if you wish.

When scans are undertaken with the **Mark** box ticked, Alifix performs the same scan and detection. However, when a broken alias is discovered, a new file is created in the same folder as that alias, with the same name with -alifix-broken.text appended to its name. That file contains the full details of the broken Alias in text form, which includes the original path to the folder or file to which the Alias pointed. This helps you create a replacement Alias if you wish. If you have the **Rewrite** box also ticked, Alifix also writes new files alongside those Aliases which are rewritten, with -alifix-rewrite.text appended to their names.

→ [Details concluded](#)

→ [Start](#)

Details concluded

⚠ If you wish Alifix to have access to files in folders whose privacy is protected, add the app to the **Full Disk Access** list in **Privacy & Security** settings, or macOS will block it from doing so.

Alifix *doesn't* deal with symbolic links (symlinks), which are quite different from Finder Aliases. It does, though, include Bookmark files which are created by my free command tool `alisma`.

You can resize text in the scrolling text box between 4 and 24 points using ⌘+ to enlarge and ⌘– to reduce the size.

Further information about and support for Alifix is available from its product page, easily accessed through the **Alifix Support** command in the **Help** menu, which opens that page in your default browser.

→ [Technical information](#)

Technical Information

Scans consist of a deep traversal of the file system using `FileManager.default.enumerator(atPath:)`. Aliases are identified using the item UTI, obtained using `NSURL.getResourceValue(&theUTI, forKey: URLResourceKey.typeIdentifierKey)`. This apparently has a long-standing bug in which some files lack a UTI and return NULL, but that appears to be handled gracefully. The UTI for Finder Aliases and alisma Bookmark files is `com.apple.alias-file`.

The Alias is read in as a Bookmark using `NSURL.bookmarkData(withContentsOf:)`, then resolved using `URL.init(resolvingBookmarkData: options: relativeTo: bookmarkDataIsStale:)`. The options for that are set as `URL.BookmarkResolutionOptions.init(arrayLiteral: NSURL.BookmarkResolutionOptions.withoutUI, NSURL.BookmarkResolutionOptions.withoutMounting)` to ensure the user isn't invited to resolve them, and missing volumes aren't sought over the network. If that returns an error, the Alias is considered to be broken, and internal Bookmark data is obtained using `NSURL.resourceValues(forKeys: [URLResourceKey(rawValue: "NSURLBookmarkAllPropertiesKey")], fromBookmarkData:)`.

When the Rewrite box is ticked, Alifix writes out the refreshed Alias using

```
let theNewBMData = try URL.bookmarkData(options:
[URL.BookmarkCreationOptions.suitableForBookmarkFile], includingResourceValuesForKeys:
nil, relativeTo: nil)
try URL.writeBookmarkData(theNewBMData, to:)
```

→ [Change list](#)

Change list

1.4:

- Window controls adjusted, new app icon and tweaks for Tahoe.

1.3:

- Rebuilt for 10.13 and later, with interface tweaks.

1.2:

- Universal App.

1.1:

- added bookmark resolution options to tackle user interaction and long timeouts.

1.0:

- added code integrity check when opening
- added Help book
- added support link
- added saving of prefs
- added control of text size
- tidied menus
- supported saving window sizes and positions
- ported to Swift 5 and Xcode 10.3.

1.0b2:

- made scrolling text view editable, to allow font and size changing
- added Rewrite option to force rewrite of unbroken Aliases
- tidied handling of broken Aliases.

1.0b1:

- initial release, without feature to delete broken Aliases.

1 July 2025.