Precize 1.9 for El Capitan, Sierra, High Sierra, Mojave & Catalina
Release Notes
Howard Oakley https://eclecticlight.co

These release notes describe the features of **Precize**, a simple GUI tool for obtaining extensive information about any file, folder or bundle, including its true size, inode data, and Bookmark information. It also lets you locate and preview any item using its Bookmark, and inspect all the information contained within any Bookmark or Finder Alias.

macOS allows files to consist of two parts: the normal **data** fork, and an unlimited number of extended attributes (**xattrs**), which contain metadata of various types. The xattrs are normally (in HFS+ and APFS file systems) stored in the file system metadata rather than with the data fork, but in some file systems such as ExFAT they are stored in hidden files.

When the Finder reports a file's size, it only gives the size of its data fork. The same is also true of the `ls` command in Terminal, unless you use its `@` option to list xattrs. In most cases, the total stored in xattrs is small, less than 1 Kbyte, but in extreme cases the total size of the xattrs can be larger than that of the data fork.

Precize addresses the problem of obtaining a full measure of any file's size, by providing the size of its data fork, the total of its xattrs, and the sum of those. At first sight, it might appear that macOS is capable of doing that itself, using a URLResourceKey claimed by Apple to include "metadata". Precize also provides the figures given in URLResourceKeys for comparison. You will quickly discover that Apple's "metadata" excludes almost all xattrs, so is of little or no value.

I initially wrote Precize to look at this problem, and the figures available from macOS. In addition to providing the only full measure of file sizes, it demonstrates that macOS doesn't appear to know the total size of any of its files. Even in Mojave running on APFS.

Precize also provides information about files and folders which is not widely available except in Terminal, in the attributes associated with the item's **inode**. These include its permissions, file type, listings of all its extended attributes, and the inode numbers for the file and the volume on which it is located.

Precize helps you use three schemes for locating files independently of where they are on any given volume: the **inode number** forming a `volfs` path, **Bookmark information**, and **File Reference URLs**.

The `volfs` path can be used as an enduring reference to the item from the command line; for example, you can enter

```
ls -la /.vol/16777224/360517297
```

and obtain a listing of the item referenced by that path, regardless of which folder you care to move that item to, on any given volume.

Bookmarks are a replacement system for inode numbers and `volfs`. Until now, they have only been accessible to compiled apps, but as support for inodes declines, they are to be preferred. Precize provides Bookmarks in Base-64 encoding, which can be used with any other tool which supports them. You can also copy a Bookmark into a Bookmark Resolver window and there use it to obtain the full path to that item, to locate it in the Finder and preview it using QuickLook, and to inspect all the information which it contains. You can now also open Finder Aliases to files and folders, and inspect all their internal data.

This works the same way as the **Open Recent** menu command does in apps, and can use the Launch Services files which make that command work as a source of Bookmarks. Precize lets you paste in Bookmarks copied from .sfl and .sfl2 files in ~/Library/Application Support/com.apple.sharedfilelist, and locate and preview the items. It is the only such utility to do so.

The third system is what Apple refers to as File Reference URLs. These use the inode number set with a different volume reference in URL format, such as
`file:///.file/id=6571367.360517297`
but cannot be used directly in Terminal, as far as I know.

What you need

- A Mac running El Capitan, Sierra, High Sierra, Mojave or Catalina. This release has been built to be fully compatible with High Sierra, including the APFS file system, with El Capitan, with Mojave, including Dark Mode and privacy protection, and Catalina.
- A copy of the latest release of Precize from https://eclecticlight.co/downloads/ (This is delivered by secure HTTPS download.)

Getting started

Precize comes compressed as a Zip file, which you should decompress, and move to your preferred folder, such as /Applications. It is not fussy where it is run from, though.

Precize not only uses my developer code signature, but is now notarized for macOS Mojave, and checks the integrity of its code against its code signature each time that it is opened. If it finds any disparity, it quits immediately. If you experience any problems running the app, and are confident that you don't have a problem with malware, try installing a fresh download. Please don't hesitate to inform me of any problems with its notarization, signature, or signature check.

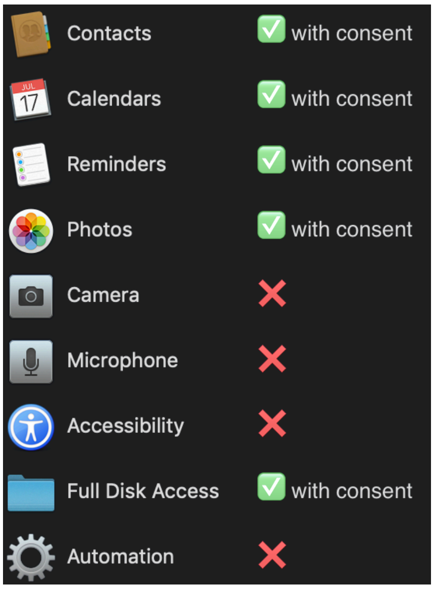
Working with Mojave's privacy protection

Precize is designed so that it will examine any item for which you have permissions. However, Mojave and Catalina add restrictions to preserve the privacy of your personal data. If you want to use Precize to examine any file or folder, including those in what Mojave/Catalina considers to be protected data, then you must add the app to the list of apps with **Full Disk Access** in the **Privacy** tab of your **Security & Privacy** pane.

If you don’t do that, but try to access protected data, then you should be prompted to give consent to Precize to do so; if you agree, it will then be added to the appropriate list of apps in the **Privacy** settings.

If Precize crashes when you try to access any protected items, this is most probably because you need to add it to the **Full Disk Access** list.

If you’re unsure what access it can have, its new **Privacy settings** command in the **Help** menu displays a window which explains succinctly:



Contacts	✓ with consent
Calendars	✓ with consent
Reminders	✓ with consent
Photos	✓ with consent
Camera	✗
Microphone	✗
Accessibility	✗
Full Disk Access	✓ with consent
Automation	✗

Use

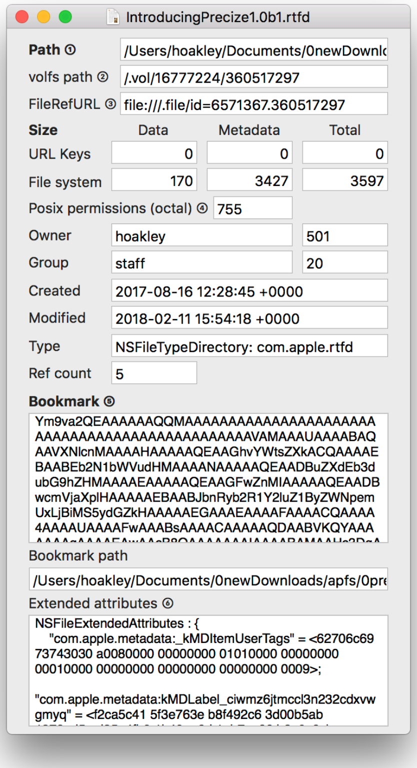
Precize currently lets you open any regular file or bundle and view extensive information about that item, including six different measures of that file’s size, each given in bytes, as well as all available inode data, `volfs` path, and Bookmark in Base-64 encoding. To open a plain folder, drag and drop that onto the app icon in the Finder or Dock; you can’t currently open a folder using the **Open…** command (this avoids ambiguity in the **Open** dialog).

To resolve a Bookmark or Alias, open a Bookmark Resolver window using the **Open Resolver** command in the Window menu, paste a Bookmark into the upper text view and click on its **Resolve** button to see the full path to that item, its `volfs` path, and a full listing of the Bookmark’s contents in the lowest scrolling text view. Click on the **Reveal in Finder** button to find a resolved Bookmark in a Finder window, or its **Preview** button to preview the document using QuickLook.

You can also open any Finder Alias, whether it points to a file or folder, by clicking on the **Open file as Base-64** button; once read in using Base-64 encoding, its contents will be shown in the lowest scrolling text view, and you can click on **Resolve** to see it resolved into paths.

Precize 1.9 for El Capitan, Sierra, High Sierra, Mojave & Catalina
Release Notes
Howard Oakley https://eclecticlight.co

Document window



Path ① /Users/hoakley/Documents/0newDownlc
volfs path ② /.vol/16777224/360517297
FileRefURL ③ file:///.file/id=6571367.360517297

Size	Data	Metadata	Total
URL Keys	0	0	0
File system	170	3427	3597

Posix permissions (octal) ④ 755
Owner hoakley 501
Group staff 20
Created 2017-08-16 12:28:45 +0000
Modified 2018-02-11 15:54:18 +0000
Type NSFileTypeDirectory: com.apple.rtfd
Ref count 5

Bookmark ⑤

```
Ym9va2QEAAAAAAQQMAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAVAMAAAUAAAABAQ
AAVXNlcnMAAAAHAAAAAQEAAGhvYWtsZXkACQAAAAE
BAABEb2N1bWVudHMAAAANAAAAAQEAADBuZXdEb3d
ubG9hZHMAAAAEAAAAAQEAAGFwZnMIAAAAAQEAADB
wcmVjaXplHAAAAAEBAABJbnRyb2R1Y2luZ1ByZWNpem
UxLjBiMS5ydGZkHAAAAAEGAAAEAAAAFAAAACQAAAA
4AAAAUAAAAFwAAABsAAAACAAAAAQDAABVKQYAAA
```

Bookmark path
/Users/hoakley/Documents/0newDownloads/apfs/0pre

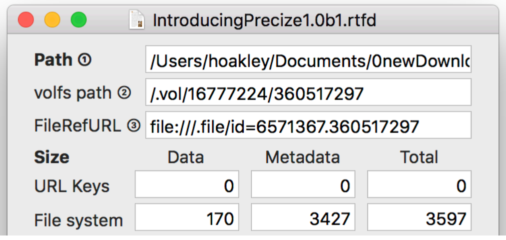
Extended attributes ⑥

```
NSFileExtendedAttributes : {
   "com.apple.metadata:_kMDItemUserTags" = <62706c69
73743030 a0080000 00000000 01010000 00000000
00010000 00000000 00000000 00000000 0009>;

"com.apple.metadata:kMDLabel_ciwmz6jtmccl3n232cdxvw
gmyq" = <f2ca5c41 5f3e763e b8f492c6 3d00b5ab
```

When you open Precize, it doesn't open a new window, but waits for you to open a document using the **Open…** menu command. Alternatively, you can drag and drop one or more documents or folders onto the app icon in a Finder window or the Dock. If you want to examine a folder, drag and drop that onto the app icon in the Finder or Dock.

When you open an item, Precize fills a deep window with information about that item. We'll step through it from the top.



Path ① /Users/hoakley/Documents/0newDownlc
volfs path ② /.vol/16777224/360517297
FileRefURL ③ file:///.file/id=6571367.360517297

Size	Data	Metadata	Total
URL Keys	0	0	0
File system	170	3427	3597

At the top is the full **path** to the item, and below that are the two other methods of giving its path, using the volfs system, and as a **FileReferenceURL**. The **`volfs` path** to the file uses its inode data to provide a reference to the file which is more enduring than its normal path and name. Simply copy that to Terminal if you want to use it in a command there. To help you copy these items quickly, each has its own menu command and keystroke shortcut, with a reminder of that shortcut next to the title of the item. So to copy the **volfs path**, simply press ⌘-2.

Precize calculates six different measures of its file size, according to two different methods.

The sizes given in the row labelled **URL Keys** are calculated using two URLResourceKeys, giving the basic size (**Data** only), and the total size (**Total**, which “includes metadata”). From those, the size of the metadata is calculated.

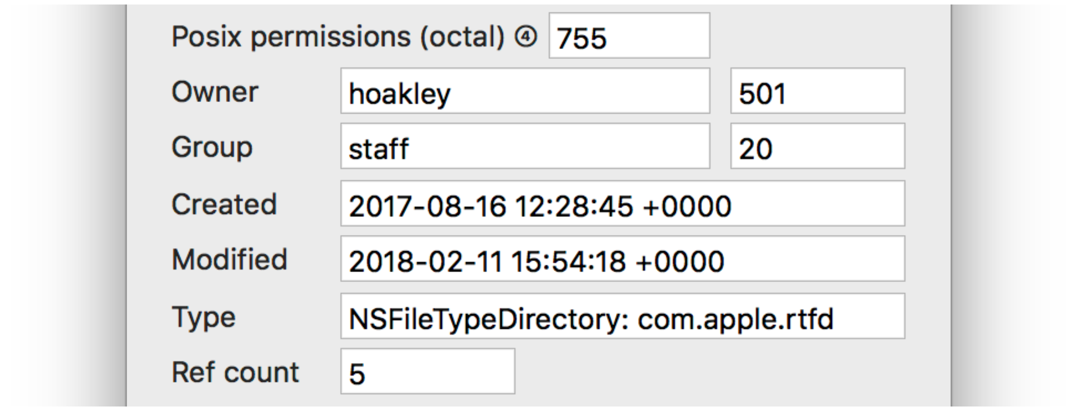
The sizes given in the row labelled **File system** are calculated using File Manager FileAttributeKeys (for **Data**), and the `getxattr` function (for **Metadata**), then they are added to return the total size of the selected file.

The two measures in the **Data** column should be identical, and correspond with that reported in the Finder, the `ls` command in Terminal, and everything else in MacOS. However, they ignore xattrs completely.

The two measures in the **Metadata** column should also be identical, and should match the total of individual metadata for that file. In practice, they usually don’t.

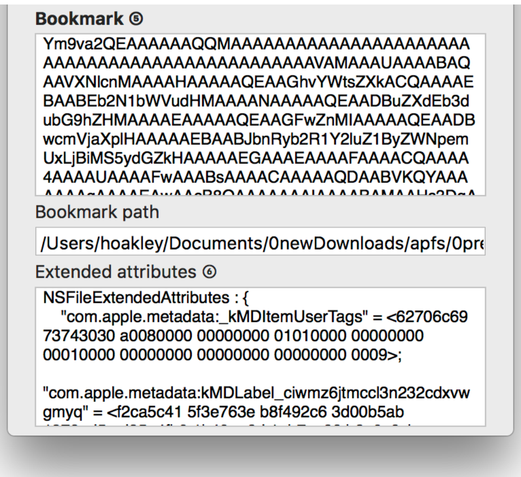
The two measures in the **Total** column should also match, but if the file has xattrs, they usually don’t, with that from **URL Keys** normally being less than that from the **File system**.

When you need to know the accurate and complete total file size, for example as will be used on a USB memory stick formatted using ExFAT, the most accurate value is that at the bottom right, for the **Total** given by the **File system**.



Posix permissions (octal) ④	755	
Owner	hoakley	501
Group	staff	20
Created	2017-08-16 12:28:45 +0000	
Modified	2018-02-11 15:54:18 +0000	
Type	NSFileTypeDirectory: com.apple.rtfd	
Ref count	5	

In the boxes below those, Precize lists each value of the inode data for that file, giving first the **Posix permissions** in octal (base 8), then the name and ID number of the **owner** and **group**. Those are followed by the date and time of **creation** and **last modification** of the file, given in UTC. The next boxes give the file **type**, using macOS terminology and as a UTI, and the **reference count**, which is the number of file system references to that item.



Bookmark ⑤

```
Ym9va2QEAAAAAAQQMAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAVAMAAAUAAAABAQ
AAVXNlcnMAAAAHAAAAAQEAAGhvYWtsZXkACQAAAAE
BAABEb2N1bWVudHMAAAANAAAAAQEAADBuZXdEb3d
ubG9hZHMAAAAEAAAAAQEAAGFwZnMIAAAAAQEAADB
wcmVjaXplHAAAAAEBAABJbnRyb2R1Y2luZ1ByZWNpem
UxLjBiMS5ydGZkHAAAAAEGAAAEAAAAFAAAACQAAAA
4AAAAUAAAAFwAAABsAAAACAAAAAQDAABVKQYAAA
```

Bookmark path

```
/Users/hoakley/Documents/0newDownloads/apfs/0pre
```

Extended attributes ⑥

```
NSFileExtendedAttributes : {
    "com.apple.metadata:_kMDItemUserTags" = <62706c69
73743030 a0080000 00000000 01010000 00000000
00010000 00000000 00000000 00000000 0009>;

"com.apple.metadata:kMDLabel_ciwmz6jtmccl3n232cdxvw
gmyq" = <f2ca5c41 5f3e763e b8f492c6 3d00b5ab
```

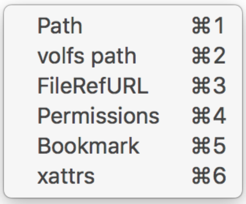
The lowest boxes provide a Base-64-encoded representation of a macOS **Bookmark** to the file, which is intended for use with the Bookmark Resolver window, and other apps which might accept such Bookmark data. Using the quick copy shortcut, press ⌘-5 and the whole of the Bookmark will be copied ready for you to paste.

That is followed by the conventional **path** contained in the Bookmark, which should be the same as the path seen in the Finder and given at the top of the window. The final box lists the **extended attributes** (xattrs) found for the file. If you want to work with those, you should use my specialist tool xattred. To change the size of the text in this last view between 4 and 24 points, use the ⌘+ keys to increase it, and the ⌘– keys to reduce it.

The **Save** command in the **File** menu lets you write all the information contained in the scrolling text box to the text file of your choice.

Quick Copy

Several of the results shown in the main document window are most useful when you can copy them, and paste them into another app such as a command in Terminal. To make this simple and efficient, Precize gives direct access to copies of six of the text boxes, through menu commands and their keystroke shortcuts. These are listed in the **Copy…** command in the **Edit** menu, and cues to their shortcuts are shown in the window.



For example, if you want to copy the volfs path to use in Terminal's command line, you can use the **Edit** > **Copy…** > **volfs path** command, or even better just press ⌘-2. You don't need to select any text, as Precize will copy the text content of the **volfs path** text box automatically, ready for you to paste.

To help you remember which text boxes have quick copy support, and their keystroke shortcuts, they are marked in the window titles using the shortcut number in a circle, e.g. ② for the **volfs path**.

How the sizes are obtained

The values listed in **URL Keys** are obtained using the following Swift 4.0 code:

```
let theRes = try url.resourceValues(forKeys: [.fileSizeKey, .totalFileSizeKey])
if let theAll = theRes.totalFileSize {
   uTotalSize = theAll
}
if let theData = theRes.fileSize {
   uDataSize = theData
}
```

The value listed in **File system** for **Data** is obtained using the following Swift 4.0 code:

```
let fAttributes = try FileManager.default.attributesOfItem(atPath: url.path)
fmDataSize = fAttributes[FileAttributeKey.size] as! Int
```

The value listed in **File system** for **Metadata** is obtained using the following Swift 4.0 code:

```
let length = getxattr(fileSystemPath, item, nil, 0, 0, 0)
```

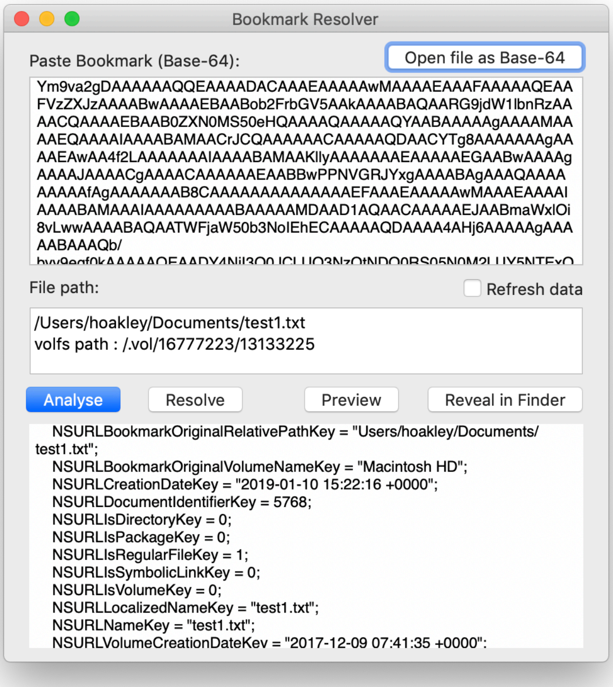
for each of the xattrs of that file, totalling those values.

The value listed in **URL Keys** for **Metadata** is obtained by subtracting the **Data** value from that of the **Total**. The value listed in **File system** for **Total** is obtained by adding the **Data** and **Metadata** values.

Because the URL and FileManager calls for the parent folder of a bundle only return values for that enclosing folder, and not its contents, those are the values displayed by Precize. Thus, it does not show the total size of an RTFD bundle, only that of the bundle folder itself.

Data shown in the other boxes are obtained by listing the whole dictionary returned in the `fAttributes` value above.

Bookmark and Alias Resolver window

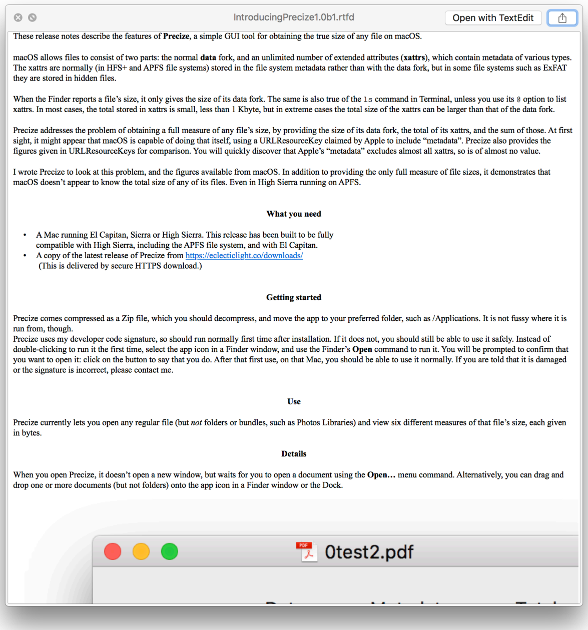


Ym9va2gDAAAAAAQQEAAAADACAAAEAAAAAwMAAAAEAAAFAAAAAQEAA
FVzZXJzAAAABwAAAAEBAABob2FrbGV5AAkAAAABAQAARG9jdW1lbnRzAA
AACQAAAAEBAAB0ZXN0MS50eHQAAAAQAAAAAQYAABAAAAAgAAAAMAA
AAEQAAAAIAAAABAMAACrJCQAAAAAACAAAAAQDAACYTg8AAAAAAAgAA
AAEAwAA4f2LAAAAAAAIAAAABAMAAKllyAAAAAAAEAAAAAEGAABwAAAAg
AAAAJAAAACgAAAACAAAAAAEAABBwPPNVGRJYxgAAAABAgAAAQAAAA
AAAAAfAgAAAAAAAB8CAAAAAAAAAAAAAAAEFAAAEAAAAAwMAAAEAAAAI
AAAABAMAAAIAAAAAAAAABAAAAAMDAAD1AQAACAAAAAEJAABmaWxlOi
8vLwwAAAABAQAATWFjaW50b3NoIEhECAAAAAQDAAAA4AHj6AAAAAgAAA
AABAAAQb/

File path:
/Users/hoakley/Documents/test1.txt
volfs path : /.vol/16777223/13133225

```
    NSURLBookmarkOriginalRelativePathKey = "Users/hoakley/Documents/
test1.txt";
    NSURLBookmarkOriginalVolumeNameKey = "Macintosh HD";
    NSURLCreationDateKey = "2019-01-10 15:22:16 +0000";
    NSURLDocumentIdentifierKey = 5768;
    NSURLIsDirectoryKey = 0;
    NSURLIsPackageKey = 0;
    NSURLIsRegularFileKey = 1;
    NSURLIsSymbolicLinkKey = 0;
    NSURLIsVolumeKey = 0;
    NSURLLocalizedNameKey = "test1.txt";
    NSURLNameKey = "test1.txt";
    NSURLVolumeCreationDateKey = "2017-12-09 07:41:35 +0000";
```

A Bookmark is of no use if you can't use it to locate the file/folder which it references. Open a Bookmark Resolver window, paste any Base-64-encoded Bookmark into the upper text view, then click on the **Resolve** button. If Precize can resolve the Bookmark data, the lower text box will then display its regular and `volfs` paths, which you can copy to Terminal or elsewhere. To locate that item, click on the **Reveal in Finder** button after resolving the Bookmark, and to preview the document using QuickLook, click on the **Preview** button.



These release notes describe the features of **Precize**, a simple GUI tool for obtaining the true size of any file on macOS.

macOS allows files to consist of two parts: the normal **data** fork, and an unlimited number of extended attributes (**xattrs**), which contain metadata of various types. The xattrs are normally (in HFS+ and APFS file systems) stored in the file system metadata rather than with the data fork, but in some file systems such as ExFAT they are stored in hidden files.

When the Finder reports a file's size, it only gives the size of its data fork. The same is also true of the `ls` command in Terminal, unless you use its `@` option to list xattrs. In most cases, the total stored in xattrs is small, less than 1 Kbyte, but in extreme cases the total size of the xattrs can be larger than that of the data fork.

Precize addresses the problem of obtaining a full measure of any file's size, by providing the size of its data fork, the total of its xattrs, and the sum of those. At first sight, it might appear that macOS is capable of doing that itself, using a URLResourceKey claimed by Apple to include "metadata". Precize also provides the figures given in URLResourceKeys for comparison. You will quickly discover that Apple's "metadata" excludes almost all xattrs, so is of almost no value.

I wrote Precize to look at this problem, and the figures available from macOS. In addition to providing the only full measure of file sizes, it demonstrates that macOS doesn't appear to know the total size of any of its files. Even in High Sierra running on APFS.

What you need

- A Mac running El Capitan, Sierra or High Sierra. This release has been built to be fully compatible with High Sierra, including the APFS file system, and with El Capitan.
- A copy of the latest release of Precize from https://eclecticlight.co/downloads/
  (This is delivered by secure HTTPS download.)

Getting started

Precize comes compressed as a Zip file, which you should decompress, and move the app to your preferred folder, such as /Applications. It is not fussy where it is run from, though.
Precize uses my developer code signature, so should run normally first time after installation. If it does not, you should still be able to use it safely. Instead of double-clicking to run it the first time, select the app icon in a Finder window, and use the Finder's **Open** command to run it. You will be prompted to confirm that you want to open it: click on the button to say that you do. After that first use, on that Mac, you should be able to use it normally. If you are told that it is damaged or the signature is incorrect, please contact me.

Use

Precize currently lets you open any regular file (but *not* folders or bundles, such as Photos Libraries) and view six different measures of that file's size, each given in bytes.

Details

When you open Precize, it doesn't open a new window, but waits for you to open a document using the **Open…** menu command. Alternatively, you can drag and drop one or more documents (but not folders) onto the app icon in a Finder window or the Dock.

Bookmarks which are saved in .sfl and .sfl2 files in ~/Library/Application Support/com.apple.sharedfilelist pad the Base-64 data with superfluous spaces, newlines, and tabs. Precize strips those characters from the text pasted in before trying to use it as a Bookmark. It replaces the text that you pasted into the box with the cleaned-up text, so that you can save that as a more concise Bookmark if you wish.

The **Reveal in Finder** button runs an AppleScript to do its work. If you have disabled AppleScript for any reason, that is not likely to work. The **Preview** button has one unexpected side-effect: because of a bug (I think in QuickLook support libraries), it also selects all the text in the lower text window. This is a workaround which I had to introduce, or QuickLook proved unable to find the document.

To inspect a Finder Alias (or a Bookmark file created by `alisma`) to a file or folder, click on the **Open file as Base-64** button. This drops down an Open File sheet. Select the Alias there and click on the **Open** button. The Alias will be read in, its contents converted into Bookmark format and displayed in the upper text view encoded in Base-64. They will automatically be analysed to show the internal information in the lowest text view. If you want to see the Alias formally resolved, click on the **Resolve** button.

Resolving Aliases can update their internal information, for example changing paths if the item to which the Alias points has been moved and can still be located. By default, the **Refresh data** checkbox is not ticked, and resolving an Alias will *not* then update its internal data. If you want that data to be updated, tick that box and the next time that you click on the Resolve button, that action will result in the data being updated in the Alias itself.

When you click on the **Analyse** or **Resolve** button, the lowest (scrolling) text view displays all the information available in that Bookmark. Even if a Bookmark can't be resolved, for example if its path and other details no longer exist, this information will still be shown and provides valuable historical information about the original file or folder. Note that the internal information is extracted before resolution is performed. As resolution may cause the internal data in Finder Aliases such as paths to be updated (if they have changed), what you see in the lowest text view reflects the state of the Alias at the time that it was first opened, not that after the resolution performed by Precize.

Two separate listings of information are provided. The first is in a structured format, and the second shows how that data is stored within the Bookmark. You can copy and paste this into other apps as you wish. The first section of the listing includes the results of the `NSURLBookmarkAllPropertiesKey`, and the second those from `NSURLBookmarkDetailedDescription`.

The Refresh data checkbox sets the bookmarkDataIsStale parameter passed when resolving the Bookmark using `URL.init(resolvingBookmarkData:bookmarkDataIsStale:)`.

Change the text size in this view between 4 and 24 points using ⌘+ to enlarge it, and ⌘– to reduce it.

How to inspect an Alias without updating its contents

A surprising number of actions, including obtaining a QuickLook thumbnail, can result in resolution of an Alias and update the information contained within it. There are circumstances in which you won't want that to happen, so that the information is left as it was originally. Here is a suggested sequence of actions which should achieve that.

1. Open the Resolver window.
2. Click on the **Open file as Base-64** button.
3. Switch the **Open File** dialog to display items in **List** format rather than in **Columns**. This works around QuickLook's thumbnail display.
4. Select the Alias which you want to examine.
5. Click on the **Open** button.

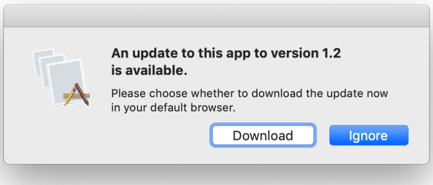
The information displayed in the lowest scrolling text view the reflects that at that moment. If you then tick the **Refresh data** box and click on **Resolve**, then **Analyse**, you may see the paths and other information updated.

Checking for updates

Whenever you open Precize, it may check to see if an update to the app is available. This doesn't use the popular Sparkle mechanism for updating in place, but works as detailed here.

Once the app has successfully completed its integrity check, it looks at whether update checking has been turned off in its preferences file. If that has, it abandons any attempt to check for updates. If checking is allowed, it then checks when it last checked for updates. If that was more than 12 hours ago, it continues to perform the check. It then connects to my GitHub server, from where it downloads a list of current versions of my apps. It doesn't upload any data to the GitHub server at all, and no statistics beyond GitHub normal connection figures are collected either: no personal identifiers are recorded.

If there is an update available, Precize then checks that its location is on my WordPress blog, and posts a dialog which invites you to download the update.



An update to this app to version 1.2 is available.

Please choose whether to download the update now in your default browser.

If you click on the **Download** button, it then points your default browser at that update, which should trigger the update to be downloaded to your normal downloads folder. The update is received as a regular Zip archive, and is exactly the same as you would download from the Downloads page here. It also carries a quarantine flag, so that when you unZip it and install the app inside, it undergoes normal first run ‘Gatekeeper’ security checks. If you click on the **Ignore** button, Precize won’t remind you about it again for another 12 hours.

An additional item at the end of the Help menu explains the update status. If no update check is performed, or the check fails, the last item reads **Update not checked**. If the check is performed and update information is obtained, even when no update is available or you decline to download it, that menu item reads **Checked for update** and is ticked (but still disabled).

You can customise this behaviour by changing Precize’s preferences. The keys to use are:

- `noUpdateCheck`, a Boolean. When set to `true`, this disables all update checking. Default is `false`.
- `updateCheckInt`, a real number (Double). When set to a value greater than 1.0, the minimum time interval between checks, in seconds. Default is 43200, which is 12 hours. If you set it to any value less than 1, Precize will reset it automatically to that default.

To change either of these, use a Terminal command of the form

```
defaults write co.eclecticlight.Precize updateCheckInt '10'
```

which works properly through the preferences server `cfprefsd`..

Limitation

Precize is also intended to be offered as a Finder Service, enabling you to select file(s) in the Finder and use its contextual menu to open them in Precize. Although it currently offers itself as a Service, the current version doesn’t yet open any files when used as a Service. If you find this annoying, you can disable Precize from Services in the normal way, through the Keyboard pane.

Acknowledgements

Code to generate the `FileReferenceURL` is adapted from an NSURL extension for Swift 3 written by Frédéric Blondiau. I am very grateful to Michael Lynn for providing the information which enables full data from Bookmarks to be shown when they are resolved.

Change list

Precize 1.9:
- added auto-update feature
- added changing text size in two text views
- improved memory of window sizes and positions.

Precize 1.8:
- added code integrity check when opening the app
- added product support page link to Help menu
- improved functionality of text views.

Precize 1.7:
- added option to set whether to refresh data when resolving Aliases
- split analyse and resolve into separate actions in Resolver
- added analyse to occur automatically whenever opening an Alias
- improved information format in lowest text view of Resolver window
- added and updated Tooltips for Resolver window
- updated Help book.

Precize 1.6:
- added feature to open, analyse and resolve Finder Aliases.

Precize 1.5:
- added full Bookmark info to Resolver
- Built in Xcode 10.1.

Precize 1.4:
- notarized for Mojave, to give you assurance that it is safe to run
- ported to Swift 4.2, built in Xcode 10ß6
- added privacy usage strings to Info.plist
- added Privacy settings window to Help.

Precize 1.3:
- added UTI to Type textbox, and expanded the box.

Precize 1.2:
- added Browse updates command to Help menu
- built using Xcode 10ß2
- fixed potential data access conflict in xattrs
- fixed Dark Mode.

Precize 1.1:
- added Copy… menu and submenu items
- added copying of six items from document window
- adjusted labels to indicate copied items
- fixed Save command so that it is enabled and works
- updated Help book.

Precize 1.0:
- adjusted document window size and resizing
- added Tooltips
- added Help book.

Precize 1.0b6:

- rolled Preciziun into Precize as Bookmark Resolver
- fixed crashing bug in Bookmark Resolver
- redesigned interface into separate items in window
- rewrote save function completely.

Precize 1.0b5:

- added FileReferenceURL output.

Preciziun 1.0b2:

- added Quick Look preview feature.

Preciziun 1.0b1:

- initial release.

Precize 1.0b4:

- converted permissions value to octal
- replaced hex and text of Bookmark with Base-64 version, for use with Preciziun.

Precize 1.0b3:

- added volfs path and Bookmark data
- added Save command for text from scrolling view.

Precize 1.0b2:

- added inode data
- added document type to support bundles.

Precize 1.0b1:

- initial release, without function as a Service.

22 August 2019.