

Cirrus 1.13 for High Sierra, Mojave, Catalina, Big Sur, Monterey, Ventura & Sonoma
Release Notes

Howard Oakley <https://eclecticlight.co>



These release notes describe the features of **Cirrus**, a utility giving you control over how items are stored in iCloud Drive, and providing tools and information to help you fix problems with iCloud.

Files stored in iCloud exist in one of three states:

- A copy of the file is in your Mac's local storage, and another is on the iCloud servers. When you look at the file in the Finder, it has no iCloud icon by it (or an icon showing just the cloud ☁), and its status is **downloaded**.
- There is no copy of the file on your Mac's local storage, but it is on the iCloud servers (and possibly stored locally on other Macs or devices). When you look at the file in the Finder, it has an iCloud icon with an arrow pointing down ☁⬇. Its status is **evicted**, as it has been evicted from local storage.
- The intermediate between those two states, in which a file is being downloaded from iCloud to provide a copy on your Mac's local storage. This is often marked by an icon showing just the cloud ☁ together with the downloading progress circle 🔄, and its status is **downloading**.

Behaviour of iCloud depends on two key settings in the iCloud pane: whether iCloud Drive is also enabled for **Desktop & Documents Folders**, and whether **Optimize Mac Storage** is enabled.

If **Optimize Mac Storage** is not enabled, then all files in iCloud Drive should either be downloaded or downloading, as without that storage optimisation, all files are stored locally. If **Optimize Mac Storage** is enabled, then files can be evicted by macOS, to save space on your local Mac storage, and that is when Cirrus is most useful, as it lets you decide which files are downloaded and which are evicted. As eviction is possible even with **Optimize Mac Storage** disabled, you can always use it to reclaim local storage.

You may also encounter problems with downloading and uploading files to iCloud. At its worst, either process can take many minutes, sometimes even several hours. Cirrus provides tools which can sometimes kickstart syncing with iCloud, and accelerate those long transfers.

Finally, Cirrus provides a lot of information about files in iCloud, and allows you to inspect log entries which can help lead you to solution of problems with iCloud and iCloud Drive. It does this by extracting those log entries most relevant to iCloud operations over a selected period of time.

What you need


- A Mac running High Sierra or later.
- An iCloud account, connected to iCloud Drive, with at least 1 MB free space on your iCloud Drive. During testing, Cirrus will create a 1 MB test file locally and copy that to your iCloud Drive. Cirrus works equally well whether you have the **Desktop & Documents Folders** setting enabled or disabled, and whether you have the **Optimize Mac Storage** option enabled or disabled.
- If you wish to use its log tools, you *must run Cirrus from an admin user account*. If you try running it from a regular user account, it will warn you when you open the app. This is a limit imposed by macOS.
- A copy of the latest release of Cirrus from <https://eclecticlight.co/downloads/> (This is delivered by secure HTTPS download.)

Getting started

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

⚠ Privacy protection in macOS Mojave and later may prevent the app from functioning properly unless you add it to the **Full Disk Access** list in the **Privacy** tab of the **Security & Privacy** pane. In Catalina and later, you may also be prompted to let Cirrus access iCloud Drive. If you are, and refuse to consent, then the app is unlikely to be able to work.

Download and Evict commands

The simplest tools in Cirrus are the **Download** and **Evict** commands in its **File** menu. When you use either of these, you will be prompted to select a file or folder in your iCloud Drive. Cirrus will then instruct macOS to download the selected item from iCloud so that you have a copy on your Mac's local storage (something you can otherwise perform by clicking on the iCloud icon  by that item), or to evict the selected item to iCloud so that it is no longer stored locally.

If an error occurs, you should be informed in an alert as to what went wrong. Note that eviction occurs very quickly, but downloading can take time, depending on the size of the item. You can select individual files or folders in both commands.

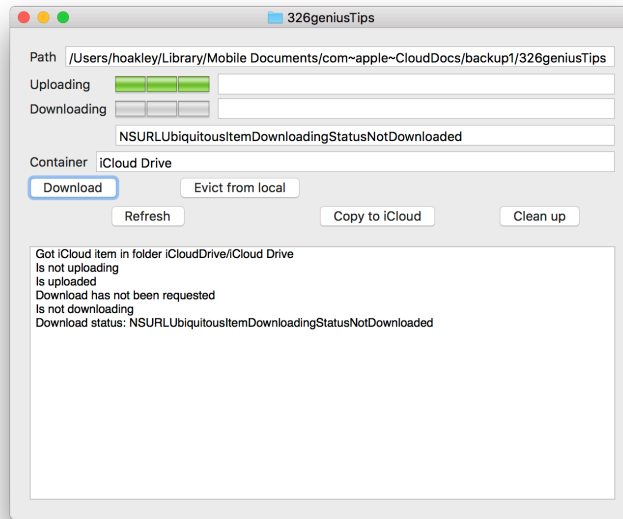
File and folder window

Cirrus also opens any file or folder stored in iCloud Drive using the **Open...** command in the File menu. If the item is not yet downloaded, the app normally calls for it, and once it has been downloaded, it is opened. However that is not always the case, and doesn't normally work when you use the **Open Recent...** command. If the item has been evicted since it was last opened, the latter is likely to result in an error reporting that the file doesn't exist. If you

Cirrus 1.13 for High Sierra, Mojave, Catalina, Big Sur, Monterey, Ventura & Sonoma
Release Notes

Howard Oakley <https://eclecticlight.co>

wait for it to be downloaded, perhaps helping that with the **Download** command, it should open normally.



Once opened, Cirrus displays information about the item and provides some tools:

- The name of the window is set to the name and type of the item opened.
- **Path** shows the full path to the item. You can copy and paste this into Terminal to access the item there, ensuring that you add any \ escape characters required for spaces.
- **Uploading** shows a level indicator of status, and any error messages resulting.
- **Downloading** shows a level indicator of status, and any error messages resulting.
- The text box below Downloading shows the official status of the item as seen by macOS
- **Container** shows the name of the iCloud container in which the item is stored.
- **Download** will request the download of the item and its entire contents.
- **Evict from local** will request macOS to evict the item. This can only work with folders, so is disabled much of the time. When not available, use the **Evict** menu command instead.
- **Refresh** fetches the above information about the item, and updates the window.
- **Copy to iCloud** runs a small test of uploading one file to your iCloud Drive.
- **Clean up** deletes any test files, named co.eclecticlight.Cirrus.data from ~/Library/Preferences and ~/Library/Mobile Documents/com~apple~CloudDocs. You should use this when you have finished performing any tests with Cirrus. If you haven't used the **Copy to iCloud** button, this is not useful. Although you can click this after each test, that shouldn't be necessary.
- The large scrolling text box at the bottom shows additional information about the item and actions.

The levels shown indicate how close that function is to completion. No bar indicates that the action has not been requested, and the full three bars indicates the function is complete, and the item uploaded or downloaded. The bar does not give any idea of the progress of an upload or download itself, though, only the stages of the process.

Change the font **size of the text** in the lower scrolling view in one point steps using the ⌘+ (Command-plus) keys to increase size up to a maximum of 24 points, or the ⌘- (Command-minus) keys to reduce size down to a minimum of 4 points. Your current font size is saved

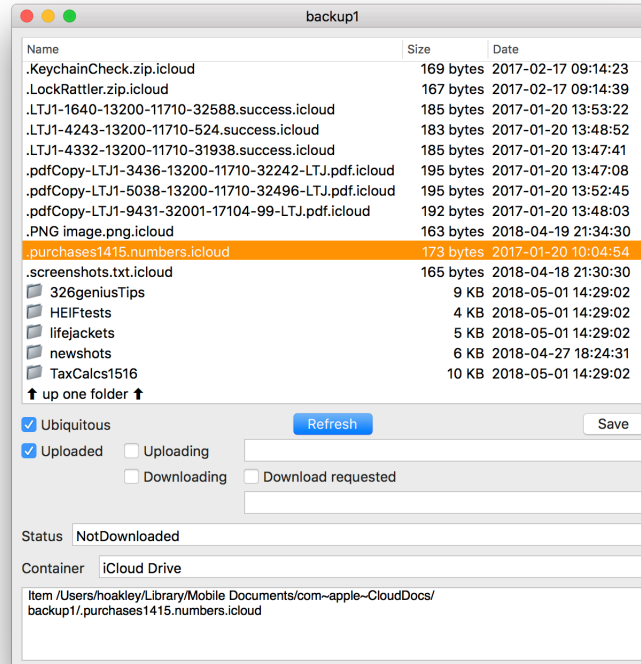
Cirrus 1.13 for High Sierra, Mojave, Catalina, Big Sur, Monterey, Ventura & Sonoma
Release Notes

Howard Oakley <https://eclecticlight.co>

together with window size and position in Cirrus's preferences to use as the default when you next open the app.

The **Save as...** menu command saves the contents of the scrolling text box to an RTF file.

iCloud Browser window



The problem with the File and Folder window is that it is subject to the laws of quantum physics: when you observe a system, you or rather macOS changes it. Merely browsing files in the **Open File** dialog is usually sufficient to download them from iCloud, which makes it impossible to check their status when they have been evicted. This is where the **iCloud Browser** window comes into its own.

To open an iCloud Browser, use the **Open Browser** command in the **Window** menu. Initially, this opens at the level of your iCloud Drive folder, but in Mojave and earlier can also show and access all the files and folders in your Mobile Documents folder. This doesn't include iCloud's shared databases, such as your address book or calendar, or special shared content such as photos (from the Photos app), music, or purchased content from the iTunes Store.

Rows in the browser represent one of four types:

- **files**, whose name often begins with a period/stop and ends with .icloud,
- **folders**, which are distinguished with the 📁 emoji symbol,
- **symbolic links** (symlinks), distinguished with the 🔗 emoji symbol,
- the final **navigation** row, containing the text '⬆ up one folder ⬆'.

Click on any file or folder row or use the **↑↓** cursor keys to see details of that item, or **double-click** on a folder row to open that folder. To move back up the folder hierarchy, double-click on the navigation row at the end of the scrolling list of items. When you double-click on a symbolic link, the browser will follow that link; however, moving up the folder hierarchy from that link's destination will not return you to the folder in which the link is, but will take you up to the parent of that folder.

Selecting a file or folder with a single click displays information about its iCloud status in the checkboxes and text boxes below. These are:

- **window title** – the current file or folder name
- **Ubiquitous** – the item is in iCloud,
- **Uploaded** – the item has been uploaded from your Mac to iCloud,
- **Uploading** – the item is currently being uploaded to iCloud; the box after that contains any error messages relating to uploading,
- **Downloading** – the item is currently being downloaded from iCloud; the box below and to the right contains any error messages relating to downloading,
- **Download requested** – the item has been requested for download, but that has not yet started,
- **Status** – the iCloud downloading status of the item,
- **Container** – the name of the iCloud container in which the item is stored,
- **lowest scrolling textbox** – the normal and full path to the item.

You will notice that iCloud files which have been evicted from local storage have names which start with a period/stop, and end with the extension .icloud. When those are downloaded to a local copy, macOS removes the initial period/stop, making the item visible, and strips the extension. On the other hand, folders do not change their names, and are effectively unchanged when 'downloaded'.

The sizes given for folders are not the total folder size, but the total size of the items immediately in the folder, as obtained from a shallow traversal of the folder contents. This is because the file system doesn't store folder sizes, so they have to be obtained by inspecting every folder and file contained within a folder. To do this for large folders, such as ~/Documents, can take a long time, sometimes several minutes. Restricting this to a shallow traversal of the contents makes this much quicker, although less informative.

In addition to the navigation controls, there are two buttons, one to **Refresh** the content of the browser window. This is the default button for that window, and is also conveniently activated by pressing the **Return** key, if you prefer. As is usual with such views, you can adjust the width of the columns using the header rows. You cannot currently change the sort order for items, though, or re-order the columns/rows.

The other button, **Save**, generates and saves a text summary of the files and folders currently shown in the browser. The default file name contains the name of the folder being browsed and the date and time at which the information was obtained from the items listed. Saves always start at the top level of your Home folder, as the most recently accessed folder

(normally the default when saving) is likely to be one of the iCloud folders which you have been examining.



Change the font **size of the text** in the lower scrolling view in one point steps using the ⌘+ (Command-plus) keys to increase size up to a maximum of 24 points, or the ⌘- (Command-minus) keys to reduce size down to a minimum of 4 points. Your current font size is saved together with window size and position in Cirrus's preferences to use as the default when you next open the app.

You can open as many iCloud Browser windows as you wish. You can close them using their normal controls, such as the red Close button.

Viewing files and folders in the iCloud Browser doesn't alter any of them in the slightest: Cirrus can only read their data, and cannot write to them in any way.

iCloud Browser Reports

Text reports saved using the iCloud Browser's Save button use a notational system not unlike that of the Terminal command `ls` to give information about each file and folder. Extracts from an example look like this:

```
Listing of folder /Users/hoakley/Library/Mobile Documents/com~apple~CloudDocs on Thursday, 3
May 2018 at 17:16:03 British Summer Time
----- .DS_Store
iU--- .201603-297geniusTips.txt.icloud      iCloud Drive   NotDownloaded
iU--- .201604-298geniusTips.txt.icloud      iCloud Drive   NotDownloaded
iU--d  .Trash      iCloud Drive   Current
iU---  backup1      iCloud Drive   NotDownloaded
```

The opening line gives the full path of the folder being listed, followed by the timestamp of when this information was obtained from the file system (not when the file was generated or saved). Then follows a list of all the files and folders found there.

Each line starts with five characters:

- **i** at the start indicates the item is 'ubiquitous', i.e. held in iCloud
- **U** in the second position indicates that the item is uploaded to iCloud
- **u** in the third position indicates that the item is uploading to iCloud
- **D** in the fourth position indicates that the item is downloading from iCloud
- **d** in the fifth position indicates the the item has been requested to be downloaded from iCloud.

The next item in each line is the current name of the item. If it is a folder, that is prefaced by the folder emoji. Any uploading or downloading errors are then reported, and the final item reports the iCloud downloading status.

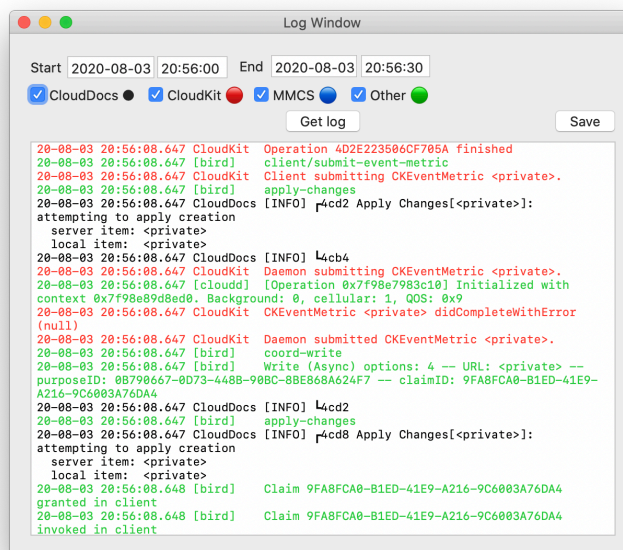
Items which are not in iCloud at all, such as the hidden `.DS_Store` file listed above, have five hyphens as their status, making them easy to distinguish.

Where to find iCloud shared files and folders

If you are using iCloud Drive without putting **Desktop & Documents Folders** into iCloud, your own iCloud Drive folders will be found in ~/Library/Mobile Documents/com~apple~CloudDocs, which is the set starting location for the iCloud Browser. This roughly corresponds to the top level of iCloud Drive as seen in the Finder and the Open... command in Cirrus. However, macOS adds other folders which are actually located in ~/Library/Mobile Documents/, which you may also need to browse.

If you have enabled **Desktop & Documents Folders** as well as using iCloud Drive, those folders don't include your Documents and Desktop folders. There are several ways to locate those, including following the symlink placed in ~/Library/Mobile Documents/com~apple~CloudDocs (useful for Documents only, as no symlink is created for Desktop), or by opening those folders from the top level of your Home folder. This can become extremely confusing, particularly because of discrepancies between what is shown in the Finder and **Open...**, and what is shown in the iCloud Browser. This is all part of the new illusion created by macOS with iCloud.

Log window



Open a Log window at any time using the **Open Log Window** command in the **Window** menu. Cirrus provides three rows of controls at the top of this window:

- **Start** provides the start date (as YYYY-MM-DD) and local time (as HH:MM:SS) for the start of the log extract created when you click on the **Get log** button.
- **End** provides the end date and local time (same formats as Start) for the end of the log extract created when you click on the **Get log** button.

- The **CloudDocs**, **CloudKit**, **MMCS**, and **Other** checkboxes set whether log entries from those subsystems are shown in the text output area. These do not determine whether those log entries are collected, only whether they are displayed and saved to file.
- **Get log** obtains an extract from the log for the set period, and displays its messages in the lower text output view. Getting the log extract can take a significant period of time. During this wait, Cirrus keeps that activity in the background and displays the ‘busy spinner’ next to this button.
- **Save** opens a file save dialog in which you should select a file into which to save the current contents of the text output in the large scrolling text area. These are saved in RTF format to preserve their colour coding.

Below the buttons is the scrolling text output area. You can copy from that, or save using the **Save** button.

Currently, the default dates and times are set to capture logs for a period of one minute prior to opening the log window. In a future release, they will be linked instead to your last iCloud action in Cirrus.

Four types of log message are displayed:

- from the `com.apple.clouddocs` subsystem, displayed in black/white, controlled by the **CloudDocs** checkbox,
- from the `com.apple.cloudkit` subsystem, displayed in red, controlled by the **CloudKit** checkbox,
- from the `MMCS` subsystem, displayed in green, controlled by the **MMCS** checkbox,
- those from `cloudd` and `bird` processes and the kernel, displayed in blue, and controlled by the **Other** checkbox.

Those checkboxes do *not* affect the log entries collected by or stored in Cirrus, only the display of those types of log message. They also determine what is saved when you save the output view to an RTF file: types which are not displayed are not saved in the file.

Each log entry is displayed in the following fixed format:

YY-MM-DD HH:MM:SS.sss [subsystem] eventMessage

where [subsystem] is the name of the subsystem, e.g. `CloudDocs`, the process (either `cloudd` or `bird`), or `[other]`.

Change the font **size of the text** in the lower scrolling view in one point steps using the ⌘+ (Command-plus) keys to increase size up to a maximum of 24 points, or the ⌘– (Command-minus) keys to reduce size down to a minimum of 4 points. Your current font size is saved together with window size and position in Cirrus’s preferences to use as the default when you next open the app. Cirrus uses the system monospace font for display of log entries.

You can now open as many Log Windows as you wish. Close them using the normal red Close button on that window.

Testing

Cirrus provides a small, standardised test involving the copying of a 1 MB file from local storage (in `~/Library/Preferences/co.eclecticlight.Cirrus.data`) to your iCloud Drive. This is ideal for observing the transactions which take place in the iCloud subsystems and services, and should help diagnose problems.

To run the test, ensure that you can see a Finder window in which the iCloud progress icon will be visible. Then press **⌘1** to invoke the **Test Upload** command in the Window menu, or click on the **Copy to iCloud** button in any document window, and watch that progress icon.

Cirrus then creates a 1 MB test file consisting of the byte `0x88` repeated throughout and saves it to `~/Library/Preferences/co.eclecticlight.Cirrus.data`. The app next checks to see if there is already a test file in your iCloud Drive, at `~/Library/Mobile Documents/com~apple~CloudDocs/co.eclecticlight.Cirrus.data`. If there is, that is deleted.

Cirrus then tries to copy the test file from `~/Library/Preferences/co.eclecticlight.Cirrus.data` to `~/Library/Mobile Documents/com~apple~CloudDocs/co.eclecticlight.Cirrus.data`, which should then appear at the top level of your iCloud Drive folder.

Once the test has completed, open a new Log window. This will have the 60 seconds prior to opening set in its **Start** and **End** boxes ready for you to use.

Observe when you click to start the test, and when iCloud's progress icon completes and disappears, marking the end of the copying process, and compare those times against the times saved to the **Start** and **End** boxes. In the great majority of cases, the actual completion of copying occurs well before the time set in the **End** boxes, and you won't need to adjust that. If copying were to take longer, adjust the **End** time to a few seconds after its completion. If you make the interval between **Start** and **End** too short, less than about 20 seconds, some log entries may be omitted.

Once the copying is complete, you can press **⌘2** to invoke the **Clean Up Test** command, or click on the **Clean up** button, which removes the two copies of the test file.

Once the test copy is finished and you are happy with the **Start** and **End** times, and ensuring the the real clock time is now later than the **End** time, click on the **Get log** button to view the log during the copying process.

You can also perform a manual test, in which you drag-copy a file to your iCloud Drive, and set the **Start** and **End** time according to your Mac's clock. You can use a similar technique to assess log entries when saving a document from an app to iCloud, or for any other function that you wish, although you will have to enter **Start** and **End** times manually for those.

Cirrus 1.13 for High Sierra, Mojave, Catalina, Big Sur, Monterey, Ventura & Sonoma
Release Notes

Howard Oakley <https://eclecticlight.co>

How Cirrus Works

Cirrus obtains its log extracts using `log show`, in a command of the form

```
log show --predicate 'subsystem == "com.apple.clouddocs" OR  
subsystem == "com.apple.cloudkit" OR subsystem == "com.apple.mmcs"  
OR processImagePath CONTAINS[c] "cloudd" OR processImagePath  
CONTAINS[c] "bird" OR processID = 0' --style json --info  
for the time period specified by the Start and End.
```

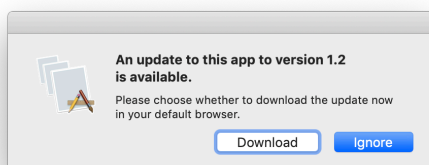
This returns the log entries in JSON format, which are then parsed entry by entry to create a styled attributed string containing all the log entries in sequence.

Checking for updates

Whenever you open Cirrus, 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 started up, 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, Cirrus then checks that its location is on my WordPress blog, and posts a dialog which invites you to download the update.



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 there. 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, Cirrus 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 Cirrus'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, Cirrus will reset it automatically to that default.

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

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

which works properly through the preferences server `cfprefsd`.

Oddities

iCloud Drive works quite differently from other storage systems. Although Cirrus shouldn't crash as a result of its quirks, it is easy to become confused by iCloud's behaviour. Normally, when working with files and folders, if you can see them in a dialog, they exist and can be opened and used. This is not the case with iCloud Drive.

For example, if you try to open an item which has been evicted in a normal Files and Folders window, one of two things will happen: you may simply see an error alert reporting that an item with a slightly different name doesn't exist so cannot be opened, or they may be a long pause, during which macOS downloads the item automatically, then suddenly displays the item as if nothing had happened.

If you open a downloaded item, and leave its document window open, you cannot evict it, as macOS recognises that the local copy of that file is open, so it cannot be removed from local storage.

If you open a downloaded folder in your iCloud Drive, then close that folder and evict it, you will be unable to open it again using the **Open...** menu command until it has been downloaded again, but you should still be able to open it from the **Open Recent...** command. It will then show that it is not downloaded, and the **Downloading** bar will be empty. You can check these out in better detail in an iCloud Browser window.

To observe how macOS works with stub and real files, open an iCloud Browser window and navigate to a folder in your iCloud Drive. Try downloading or evicting some files there, then click on the **Refresh** button in the browser window to see how they have changed there.

Change list

1.13:

- addressed deprecations
- added support for Ventura and Sonoma
- removed startup integrity check
- removed log privacy controls
- updated Help book.

1.12:

- changed log checks during startup to detect errors in time formatting
- corrected all time formatting to allow for 12-hour clocks
- improved speed and reduced memory requirement for logs.

1.11:

- added log checks during startup to address macOS bugs.

1.10:

- changed text colours in the log view, and added colour dots to the dialog
- correct use of system version numbers for compatibility
- corrected app ID in Help book
- Universal App which runs native on both Intel and Apple Silicon Macs, with 11.0 support.

1.9:

- changed the font for log displays to system monospace
- added check for admin user status to offer log display
- hid privacy setting control in Catalina
- added kernel entries to the log extract obtained
- ported to Xcode 11.3.1, 10.15 SDK and Swift 5.1.

1.8:

- reverted all paths as Catalina now uses the same as before. Thanks, Apple.

1.7:

- put browser starting paths into preferences so they can be changed when needed.

1.6:

- fixed bug which stopped upload test from occurring
- added menu commands for upload test and cleanup
- added Catalina paths, making Browser functional in beta 5
- constrained upper Browser paths to prevent possible crash
- tidied windows and menus
- updated Help book for Catalina and previous changes.

1.5:

- added changing of text size in all text views, and saving as defaults
- tweaked window size and position saving and defaults
- added support for automatic checking and downloading of updates.

1.4:

- ported to Swift 5 and Xcode 10.2.1
- tidied text views and improved their controls
- added code signature check on opening app
- added Help menu item for Cirrus support.

1.3:

- added support for cursor key navigation in iCloud Browser.

1.2:

- ported to Swift 4.2.1 and Xcode 10.1
- notarized.

1.1:

- added Browse updates menu command
- added full support for Dark Mode
- built using Xcode 10β.

1.0:

- moved getting log extract into background, with busy spinner
- added Tooltips
- added Help book.

1.0b5:

- added following symlinks to iCloud Browser
- changed folder size calculation to shallow rather than deep traversal
- released iCloud Browser to cover entire Home folder
- added Save to iCloud Browser to generate text report
- added datestamping to iCloud Browser's data
- changed file sizes in iCloud Browser to report more flexibly
- changed expression of iCloud Download Status in iCloud Browser
- anchored the Refresh button properly so that it doesn't float into the view above.

1.0b4:

- added iCloud Browser window and features
- fixed bug in incomplete updating of attributes in document refresh
- made Browser and Log windows unlimited, removing Close menu commands.

1.0b3:

- major rewrite
- moved log-related actions and info to new log window
- rebuilt to be compatible with El Capitan
- disabled opening of log window when running in El Capitan
- added new code to open files and folders as main document type
- added Download and Evict menu commands.

1.0b2:

- fixed a bug in parsing and classifying Other log entries.

1.0b1:

- initial release.

15 July 2023.