Mirror Mirror

Name:	Mirror Mirror
Author:	Mike Pasini
Version:	1.2c
Last Update:	6 April 2021
Requires:	Keyboard Maestro v9

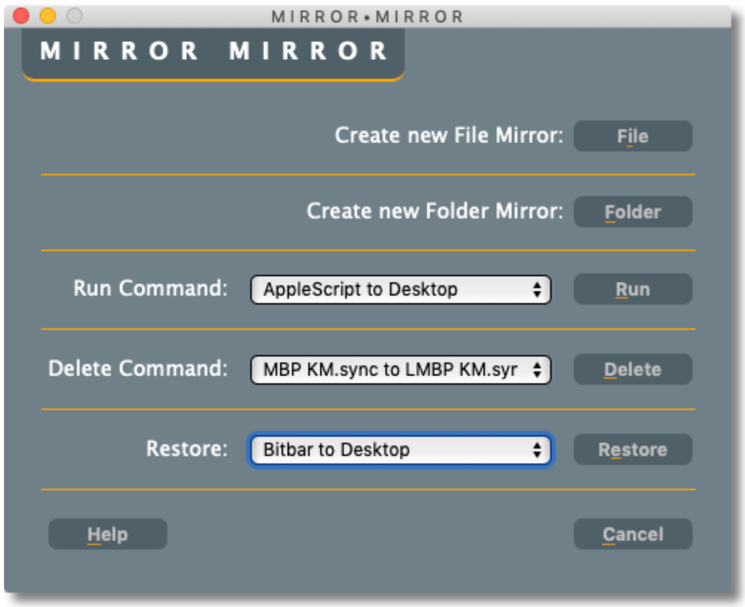
Table of Contents

INTRO

This macro uses the built-in backup utility rsync to mirror any directory or single file in a Target location. Changes made to the Source will be reflected in the Target location, effectively making an archive of the Source.

After the initial backup, which takes as long as any copy, only the changes are executed, which is much faster than a fresh copy. New files are copied, deleted files are deleted.

A log shows what was done and how long it took.



Run Command: AppleScript to Desktop

Delete Command: MBP KM.sync to LMBP KM.syr

Restore: Bitbar to Desktop

You can save and delete commands. You can also restore a Target to its Source from any saved command.

HOW IT WORKS

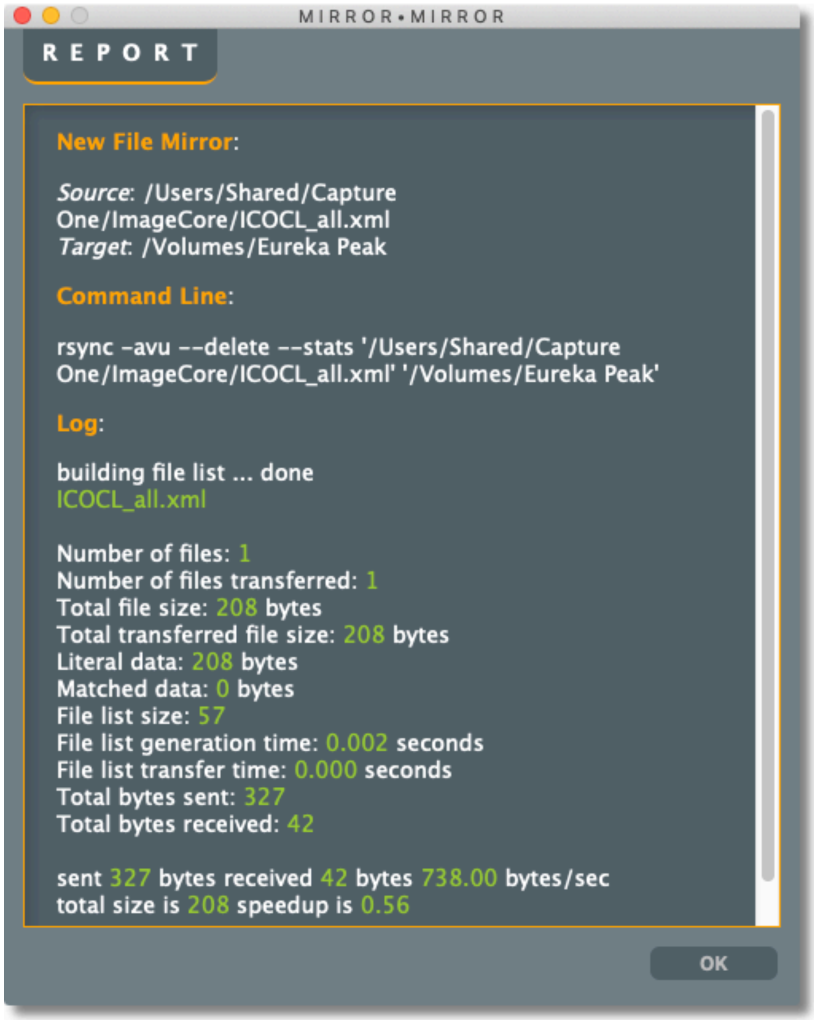
When launched, Mirror Mirror presents a Custom HTML Prompt showing all its options with the **Run** command selected so you can just press the **`Return`** key to execute that backup. The options (with keyboard shortcuts underlined) include:

- **Create new File Mirror** will prompt for a Source file and Target Location, copy the Source to the Target and as you if you want to save the command to run again later.
- **Create new Folder Mirror** will prompt for a Source folder and Target Location, copy the Source to the Target and as you if you want to save the command to run again later.
- **Run Command** lists all saved commands, allowing you to select one to run with the last run command selected.
- **Delete Command** lists all saved commands, allowing you to delete one.
- **Restore** reverses the copy of the saved command.
- **Help** provides a brief overview.
- **Cancel** quits the macro.

The first two, used to create new rsync commands, prompt for a Source and then for a Target location. **Run**, **Delete** and **Restore** all act on the selected option in the pulldown list.

All of those options except **Delete** produce a report listing the Source and Target, the complete rsync command and the verbose output from the rsync run.

The two options that create new rsync commands will then give you the opportunity to save the command. This makes



New File Mirror:

Source: /Users/Shared/Capture One/ImageCore/ICOCL_all.xml
Target: /Volumes/Eureka Peak

Command Line:

```
rsync -avu --delete --stats '/Users/Shared/Capture One/ImageCore/ICOCL_all.xml' '/Volumes/Eureka Peak'
```

Log:

```
building file list ... done
ICOCL_all.xml

Number of files: 1
Number of files transferred: 1
Total file size: 208 bytes
Total transferred file size: 208 bytes
Literal data: 208 bytes
Matched data: 0 bytes
File list size: 57
File list generation time: 0.002 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 327
Total bytes received: 42

sent 327 bytes received 42 bytes 738.00 bytes/sec
total size is 208 speedup is 0.56
```

it very easy to update an ongoing project's Target location.

Mirror Mirror uses local variables whenever it can but it wants to remember the last used Source and Target as a convenient starting place for new commands. It also needs to remember the saved commands. So it uses these global variables:

- **MM Source** is the last used Source file or directory
- **MM Target** is the last used Target location
- **MM List** is the list of saved Sources and Targets separated by a tab
- **MM ChoiceRun** is the last run command chosen and the default for the next run

Mirror Mirror does not store the rsync command in **MM List** because it never changes.

The built-in keyboard shortcuts make it possible to use only the keyboard to run Mirror Mirror. They do not use the **`accesskey`** tag so you don’t need to press any modifiers. Note that on the Main Window, the **`Return`** key is assigned to the **Run** option.

USAGE NOTES

Mirror Mirror performs a one-way sync from the Source to the Target. Any changes made to the Source will be reflected in the Target. So if you delete a file in the Source, it will be deleted in the Target the next time you run the backup command.

The Command Line section of the log shows you the actual command used. If you want to create a separate macro for that, you can paste the command line into an Execute Shell Script action.

All rsync operations use the "**`-avu --delete --stats`**" options. That enables archive mode, full reporting, skips existing files (especially if they are newer) and cleanup.

Mirror Mirror performs a one-way sync but a two-way sync only requires a second run using the **Restore** option. You would do this if you have modified files in the Target location that you want to update the Source with. Using the usual **Run** command won't update the Source with the new Target file but it won't revert it either. But using **Restore** will copy the newer Target file to the Source.

Your Source and Target can be anything, including other volumes on your network including iCloud Drive, eternal drives, USB cards, etc.

This macro started as a simple way to copy a file to an external drive. It isn't really intended as an alternative to Time Machine or utilities like FreeFileSync, which offer many more options, including two-way syncs. But it makes backups pretty painless and very fast, whether they're temporary copies, working projects constantly being updated or archives.

Audio signals are used to reinforce different processes. Tink indicates completion of a task while Blow indicates Mirror Mirror has quit. Mirror Mirror will stay active until you click the Cancel button.

Rsync is a powerful utility whose documentation explains a number of arguments and combinations of arguments whose consequences can be difficult to grasp. Mirror Mirror hides all of that from you with a safe one-way command. I've used this command for files and folders for years to make backups to external drives.

A TEST DRIVE

I recommend experimenting with Mirror Mirror on dummy files before using it on live ones so you understand what it does. Here's a simple setup inspired by Manny Fernandez at InfoSec Monkey (https://www.infosecmonkey.com/2020/01/19/2-way-sync-with-rsync/) that you can manipulate to test Mirror Mirror.

- Launch Terminal or your favorite equivalent (like iTerm)

- Create some some folders on the Desktop:

```
cd ~/Desktop
mkdir mysource mytarget
cd mysource
touch file1 file2 file3
cd ..
```

That creates three empty files in mysource.

To make a second folder mirror mysource, run Mirror Mirror with the **Create New Folder** Mirror command. Select mysource in the Desktop folder as the source. Select the mytarget folder as the Target location.

Note that it copies three files in a new mysource folder in the mytarget folder, which are listed by name at the top of the log as the file list.

Peek at mytarget from the Finder to see for yourself. They're there:

```
ls mytarget/mysource
```

Now let's add some text to file2 in mysource (and display it):

```
echo "hello" > mysource/file2
cat mysource/file2
```

Run Mirror Mirror again but this time use the **Run** command with the name you gave the initial backup showing in the popup list. Note in the report that Mirror Mirror has copied only one file this time and it's file2.

Peek at file2 in mytarget. Is your text there? Yep:

```
cat mytarget/mysource/file2
```

Now make a change to file1 in mytarget:

```
echo "goodbye" > mytarget/mysource/file1
```

You wouldn't normally edit a file in an archive or backup directory, of course. But you might edit a file on another machine that you want to keep in sync. For this exercise, think of mytarget as your portable machine on which you've done some work at the library and now want to update your home machine (mysource).

If you run Mirror Mirror using the **Run** command with the only option you've created, nothing is copied ("Number of files transferred: 0"). The empty file1 in mysource does not overwrite the changed file2 in mytarget because mytarget's version of the file is newer.

Instead, to copy the newer file from mytarget, use the **`Restore`** command with that same saved command. Mirror Mirror reverses the operation, swapping the directory name from the Source to the Target and making the Target the Source. This effectively gives you a two-way sync.

Where you might get in trouble is editing the same file in two locations without backing up between edits. In that case, you would have unique edits in each file. Go ahead. Try it. You'll see Mirror Mirror won't overwrite the conflicted file. Either way you try to copy it. So no damage is done but you might be confused about which file contains what until you reconcile the two versions.

RELEASE NOTES

Mirror Mirror

6 April 2021

- Implement key shortcuts for every command in main window
- Add visual key (orange underline) to button labels for the key shortcuts
- Revised JavaScript to recognize Return for OK/Save buttons in any window
- Removed accesskey option for ^ ⌥ **h** and ^ ⌥ **c** in main window to avoid conflicts with other possible macros

2 April 2021

- Recognize Return for OK buttons in any window
- Recognize ^ ⌥ **h** for Help in main window
- Recognize ^ ⌥ **c** for Cancel in main window

29 March 2021

- Initial release.

CONTACT

I'm happy to promptly address any concerns you may have. I can’t do anything about the modules themselves but I can address interface issues.

You can reach me at http://mikepasini.com or on the Keyboard Maestro forum as mrpasini.