

## New to using command line applications?

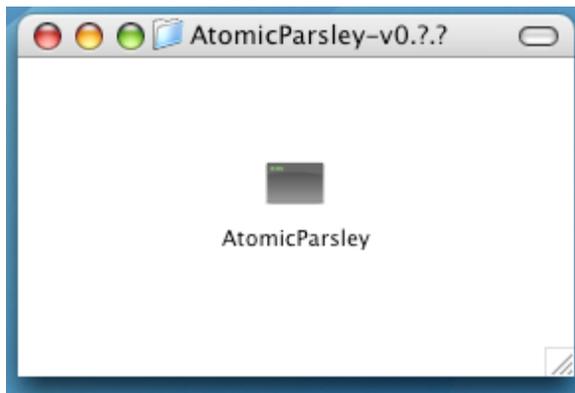
This is a quick tutorial on how to use AtomicParsley on Mac OS X. Windows behavior is a little different, but the concept is basically the same. The major differences are:

**cmd.exe is the windows equivalent of Terminal.app ("Run" in the Start menu)**

**after a drag-n-drop, a space is NOT added by cmd.exe. Switch back to cmd.exe and add a space**

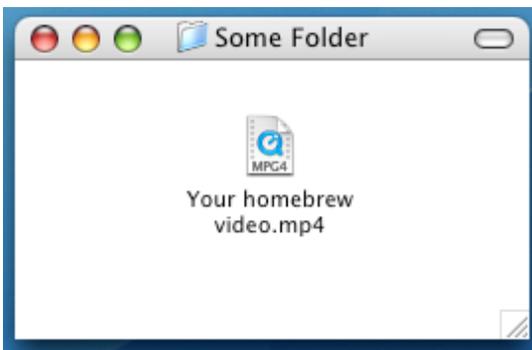
There is a part of Mac OS X that can be accessed where you don't use the mouse to get things done - you type out commands. That is the essence of a "command line interface" or CLI. AtomicParsley has a command line interface, and it has a few options, so here is how to master using the Terminal.

- **1.** In the Finder, open the folder that AtomicParsley is in. All we want is the AtomicParsley icon easily visible and accessible:

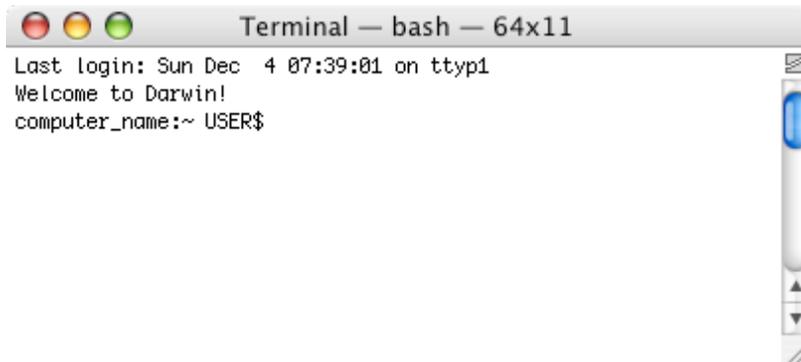


AtomicParsley is a command line program, and they essentially all look the same - just as pictured above.

- **2.** In the Finder, open the folder that contains your target file. You don't need to change the file extension to ".m4v" because this AtomicParsley sets the proper Finder Type/Creator codes, so iTunes always opens tagged files. AtomicParsley doesn't care what you name it:



- **3.** Open Terminal.app. It is in your Applications -> Utilities folder. A gawd-awful bare nekkid window will come up:

A screenshot of a Mac OS X Terminal window. The title bar reads "Terminal — bash — 64x11". The window content shows the following text: "Last login: Sun Dec 4 07:39:01 on ttty1", "Welcome to Darwin!", and "computer\_name:~ USER\$". The prompt "\$" is visible at the end of the line. The window has a scroll bar on the right side.

```
Terminal — bash — 64x11
Last login: Sun Dec 4 07:39:01 on ttty1
Welcome to Darwin!
computer_name:~ USER$
```

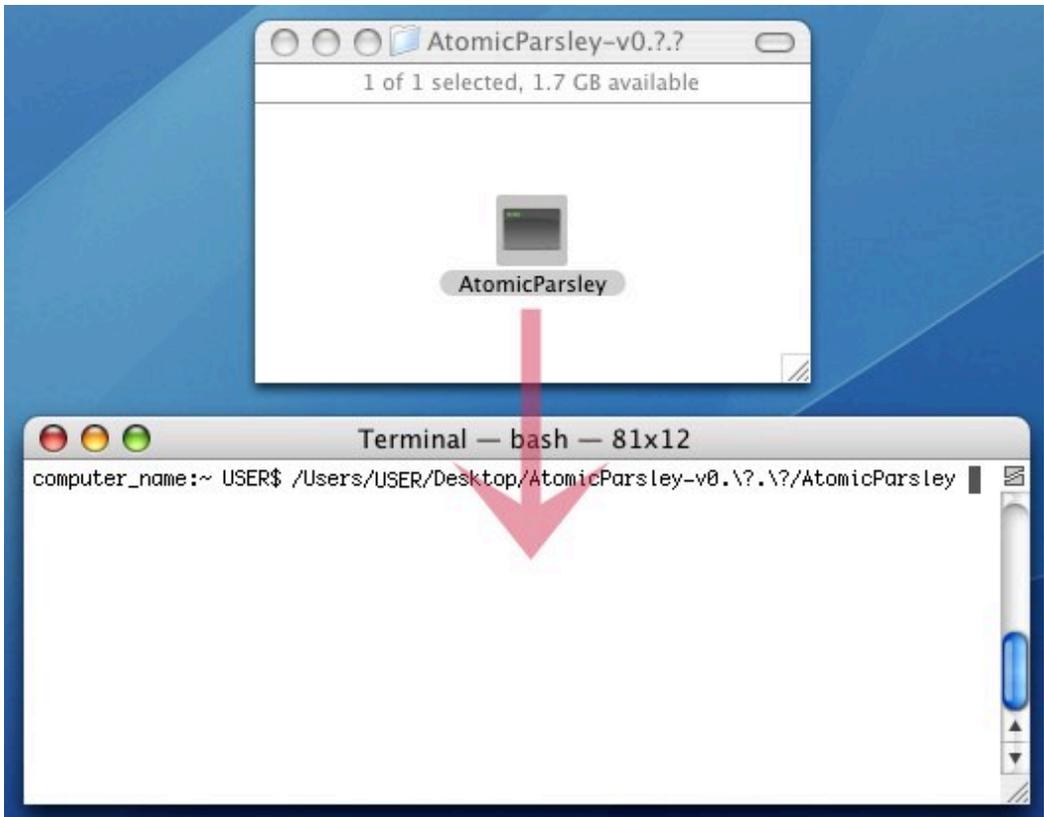
Here we see whatever your `computer_name` is set in SystemPreferences->Sharing, and whatever your short username is, and the "shell prompt". Here the prompt is "\$" because our shell is "bash" - seen in the window title.

- **4.** AtomicParsley has a particular "syntax" of understanding commands. For example, in English you say "You, fetch that newspaper". In this particular English syntax, the pronoun comes first, then the verb, and then.... Well enough of high-school english. AtomicParsley works similarly (though other programs differ widely):

```
AtomicParsley [file] [options]
```

We see here that its `program_name` first, then the file to be modified, and any & all options last. And precisely which options are there? Well...

- **5.** To get Mac OS X to use AtomicParsley, it has to know where it is. There are some special folders (hidden from everyday eyes) where if a program is in, it is automatically available to the entire OS (you can see exactly which folders yourself by this command "echo \$PATH" and hitting return - without the quotes). AtomicParsley isn't in one of those special folders, so you need to tell Mac OS X its exact location. This is easily accomplished by dragging & dropping the AtomicParsley program icon (seen in step 1) onto the Terminal.app window (seen in step 3):



The path to AtomicParsley will automatically be entered into the Terminal.app window - and without having to type a single letter. Then, in Terminal.app, hit the "return" or "enter" key. You should see.... a variety of options. In this partial picture:

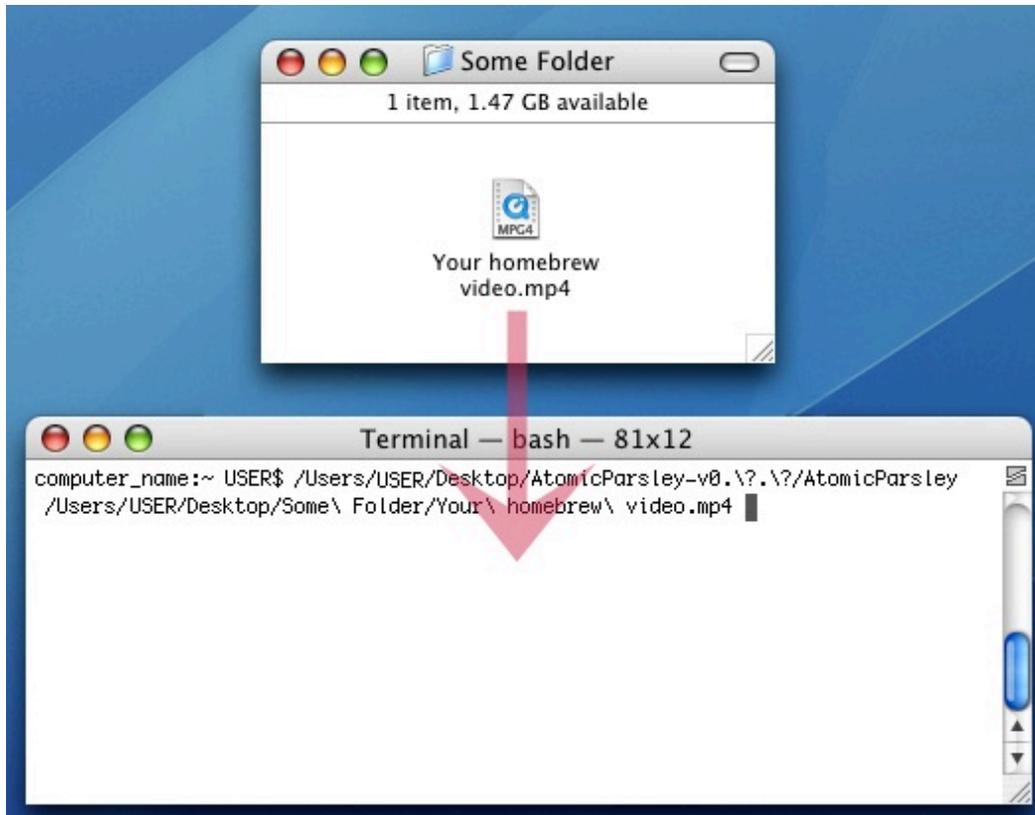
```

--grouping      , -G (str)  Set the grouping tag: "moov.udta.meta.ilst.grp.data"
--artwork       , -A (/path) Set (multiple) artwork (jpeg or png) tag: "covr.data"
--bpm           , -B (num)  Set the tempo/bpm tag: "moov.udta.meta.ilst.tempo.data"
--compilation   , -C (bool) Sets the "cpil" atom (true or false to delete the atom)
--advisory      , -y (1of3) Sets the iTunes lyrics advisory ('remove', 'clean', 'explicit')

```

We see here that the "--grouping" command line option takes a string ("str") as its argument. "--artwork" takes a path (the /slash/delimited/folder/structure/to/AtomicParsley seen above is a /path). "--bpm" takes a number as its argument (as in "--bpm 120"). Compilation takes a bool - which is short for a boolean value of true or false. And "--advisory" (as in "--advisory 'clean'") takes 1 of 3 specific arguments

- 6. To put it all together, we drag the AtomicParsley program onto the Terminal.app window each and every time we want to use the program. It is followed by the path to the file we want to tag:



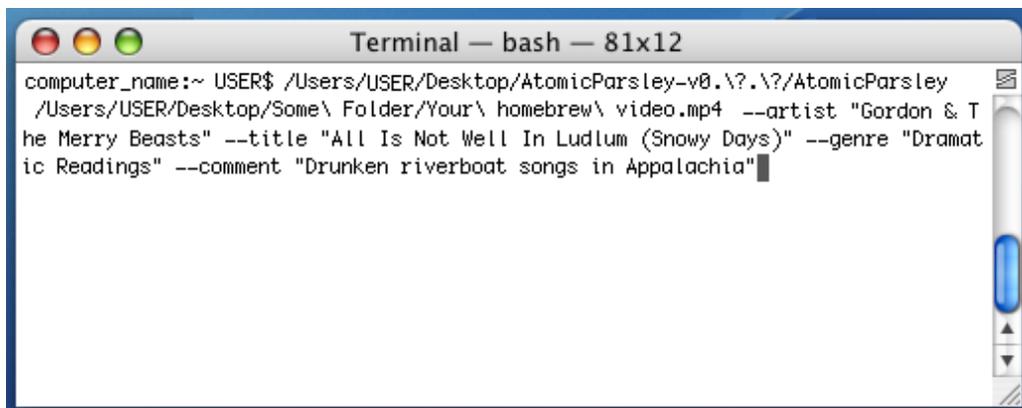
You will see the path to your video file will be automatically placed in that window (with a trailing space, which is what we want also - arguments are separated by spaces).

- **7.** We add our command line flags, whatever & however many of them we want. For example, to set the artist we want to append :

```
--artist "Gordon & The Merry Beasts"
```

to do more than 1 tag, we just add a space and use another option:

```
--artist "Gordon & The Merry Beasts" --title "All Is Not Well In Ludlum (Snowy Days)" --genre "Dramatic Readings" --comment "Drunken riverboat songs in Appalachia"
```



The text will wrap around if it is too long, but don't worry - it's supposed to do that. When we have all the tags listed as we want, hit return or enter. Your original file will be unaltered, and a new one will be written to next beside it.

When AtomicParsley has finished, you will get another shell prompt exactly as in Step 3 - that is how you tell when it is finished. Rename as necessary. Open the new file in iTunes - the old file you should save as a backup.

- **8.** Please remember this is beta software.