

December 05, 2014

Making the clipboard work between iTerm2, tmux, vim and OS X.

Getting copying and pasting to behave sanely when working with the terminal has been a constant struggle, probably ever since I started working with PuTTY, slackware and what must have been Windows 98.

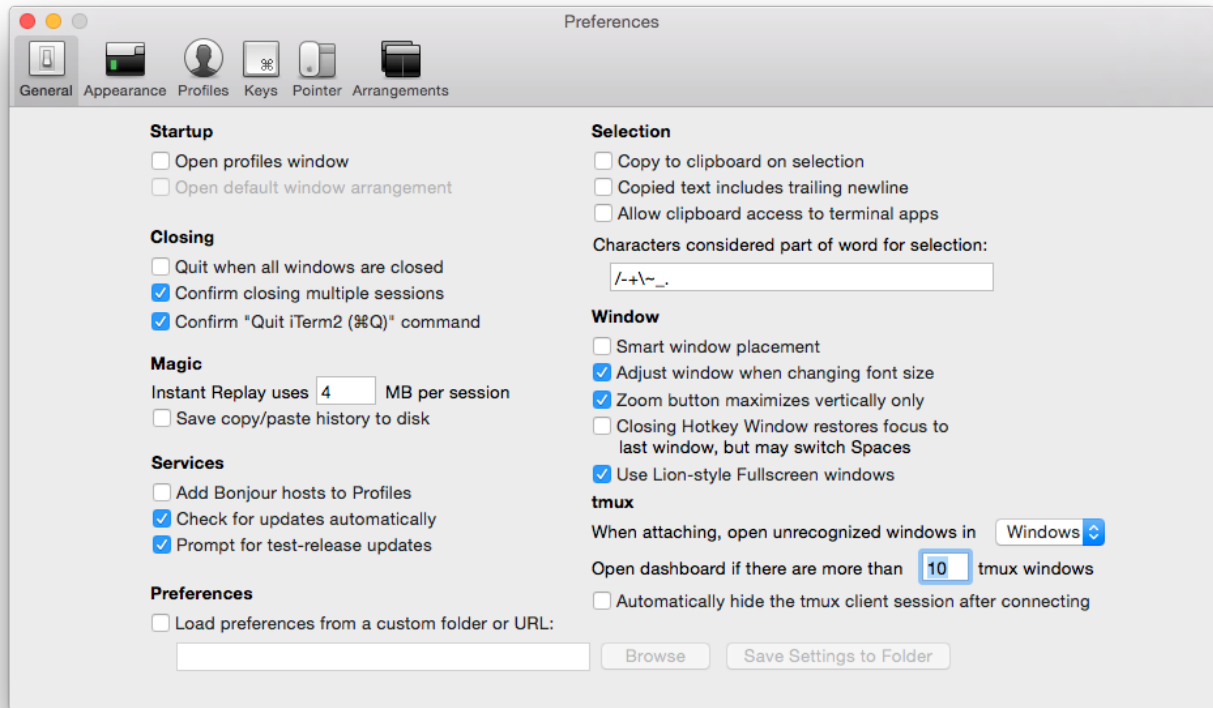
These days my environment consists of OS X 10.10, [tmux](#), [iTerm2](#) and well, [Vim](#) has never gone away. Neither have my issues with copy-pasting though. It seems that every time I have a solution, a few months later a cog in the machine changes and breaks the whole set-up again, which then takes me months to fix due to my lazyness.

I finally took the time again to look into this, and figured I should share. So this is the December 2014 solution! Keep in mind that in January 2015 this may no longer work though...

iTerm2

First: iTerm2. iTerm2 has a setting that allows terminal applications to access the clipboard.

This works through ANSI escape sequences, and you actually want this **OFF**. Before today I had this setting on, which kinda worked but it always truncated the clipboard, which made it completely frustrating for copy-pasting larger things such as logfiles.



So to repeat this, the "Allow clipboard access to terminal apps" must be **OFF**

tmux

OS X has two command-line utilities to interact with the clipboard, `pbcopy` and `pbpaste`. These utilities break when running inside of tmux.

To fix this, we need a small utility that can be installed using [brew](#):

```
brew install reattach-to-user-namespace
```

If you would like to know exactly how this works and why it's needed, or if you want to install this without using brew, the [github project page](#) has all the answers.

Bonus fact: this also fixes `launchctl` in tmux.

After this, we need to make a few modifications to your `~/.tmux.conf`. Add the following lines:

```
# Copy-paste integration
set-option -g default-command "reattach-to-user-namespace -l bash"

# Use vim keybindings in copy mode
setw -g mode-keys vi
```

```
# Setup 'v' to begin selection as in Vim
bind-key -t vi-copy v begin-selection
bind-key -t vi-copy y copy-pipe "reattach-to-user-namespace pbcopy"

# Update default binding of `Enter` to also use copy-pipe
unbind -t vi-copy Enter
bind-key -t vi-copy Enter copy-pipe "reattach-to-user-namespace pbcopy"

# Bind ']' to use pbpaste
bind ] run "reattach-to-user-namespace pbpaste | tmux load-buffer - && tmux past"
```

The previous lines sets up tmux correctly, and binds various copy and paste keys to use pbpaste and pbcopy.

The one thing I have not figured out here, is that I haven't been able to get tmux mouse support with this. So if you want to copy and paste to the system clipboard, you need to do this with the keyboard short-cuts.

Know how to extend this feature to get mouse support? Do let me know!

Don't forget to restart tmux or run:

```
tmux source-file ~/.tmux.conf
```

To start using this new configuration.

Vim

Lastly, vim! It would be awesome if we can just 'yank' and paste using `y` and `p` from Vim as well.

This is rather easy. Open up your `.vimrc`, or `.vim/vimrc` file and simply add this line:

```
set clipboard=unnamed
```

However, this did not work with the Vim version that OS X shipped with (which is 7.3). To upgrade to 7.4, just use Homebrew again:

```
brew install vim
```

Sources

A big thank you to the sources. I got most of my information from the following links:

- <http://brentvatne.ca/tmux-copy-paste/>
- <http://apple.stackexchange.com/questions/41412/using-tmux-and-pbpaste-pbcopy-and-launchctl>
- <http://robots.thoughtbot.com/tmux-copy-paste-on-os-x-a-better-future>
- <https://github.com/ChrisJohnsen/tmux-MacOSX-pasteboard>
- <http://robots.thoughtbot.com/how-to-copy-and-paste-with-tmux-on-mac-os-x>

Bonus tmux feature

Add the following lines in `~/.tmux.conf` to automatically open new windows and tabs in the directory from where you opened them:

```
# New window with default path set to last path  
bind '"' split-window -c "#{pane_current_path}"  
bind % split-window -h -c "#{pane_current_path}"  
bind c new-window -c "#{pane_current_path}"
```

Respond

7 Comments

Evert's blog

 Ernestas Poškus ▾ Recommend 1 Share

Sort by Best ▾



Join the discussion...

Manoj Mahalingam · 5 months ago

You can select with mouse, and before letting go, press y to copy.

1 ^ | ▾ · Reply · Share ›

Evert Mod → **Manoj Mahalingam** · 4 months ago

I noticed this a little later as well, and this is exactly how I work around that issue. Thanks!

^ | ▾ · Reply · Share ›

Tanner Jones · 2 months ago

Just an fyi, after 'brew install vim' I lost what I feel like is the correct behavior for the backspace key on Mac OSX. After some searching, it seems that the default behavior for 'backspace' is to only work on characters typed during the current insertion mode. I don't like this... The fix is to put the following in ~/.vimrc
set backspace=indent,eol,start

<http://stackoverflow.com/quest...>

^ | ▾ · Reply · Share ›

**Rickard von Essen** · 4 months ago

Awesome! Nice bonus.

^ | ▾ · Reply · Share ›

**Mingliang Liu** · 5 months ago

Very nice post!

^ | ▾ · Reply · Share ›

Brian Dant · 5 months ago

Nice work! Thank you.

^ | ▾ · Reply · Share ›

**Marcin Kasinski** · 6 months ago

Awesome post! Thank you very much for sharing. That is exactly what I was looking for!

^ | ▾ · Reply · Share ›

ALSO ON EVERT'S BLOG

WHAT'S THIS?

Written by *Evert Pot*. Find me on *Twitter* or *Github*.