

Extra Key Bindings

Herbert Schulz

Abstract

The `NSTextView` framework is used by many applications when text needs to be entered into a box of some kind; e.g. in many, but not all, editors, Apple's Mail and text boxes within Safari. There are many features of `NSTextView` that are available but aren't activated by default. The files supplied with this document implement some of these features as keyboard bindings that will be active anywhere that `NSTextView` is used.

Introduction

Apple supplies a fairly simple mechanism to add default key bindings to “selectors” supplied by `NSTextView`; you simply write a `DefaultKeyBinding.dict` file and place it in the `~/Library/KeyBindings/`¹ directory where `~` means your home directory. Any application that uses the `NSTextView` framework for entering text will then have these key bindings added to its repertoire.

Installing the `DefaultKeyBinding.dict` File

The `DefaultKeyBinding.dict` file can either be a XML file, similar to `plist` files used for storing preferences, or an equivalent text file that looks very much like a structure definition in C. Both versions are supplied with this document and either one can be used. The `DefaultKeyBindingText.dict` is the text version of the file while `DefaultKeyBinding.dict` is the XML version. Note: the text version must be renamed `DefaultKeyBinding.dict` before installation into `~/Library/KeyBindings/`.

What Is Added?

The following keyboard commands are added to the standard repertoire. The symbols `^`, `⌘` and `⇧` stand for the Control, Option and Shift keys respectively. The Command (`⌘`) key isn't used in any of these bindings to help avoid conflicts with keyboard shortcuts within applications.

1. Changing the “case” of a word:

| Key | Action |
|------------------------|-------------------|
| <code>^ - u</code> | Capitalize a word |
| <code>⇧ - ^ - u</code> | UPPERCASE a word |
| <code>⌘ - ^ - u</code> | lowercase a word |

¹If this directory, or others mentioned later in the document don't exist in your setup you can simply create them.

2. The command `^v`, which is already a standard binding in `NSTextView`, will scroll down one window's worth and center the insertion point within the window. The new binding `⌘v` scrolls *up* a window's worth and centers the insertion point in the window. The original binding for `⌘v` can be obtained using the two keystroke sequence `^q ⌘v`.
3. The following items use two keystrokes to execute, something unusual for Macintosh software, and are adapted from the emacs editor². The `_` stands for a Space character.

| Keys | Action |
|--------------------|---|
| <code>^x _</code> | Set Mark |
| <code>^x ^x</code> | Swap Insertion Point and Mark |
| <code>^x ^m</code> | Select between Insertion Point and Mark |

The Mark is an invisible marker set in memory, not actually saved in the file, at the position of the insertion point by the `^x _` sequence. One can then move the insertion point to another location in the file and select all of the text between the insertion point and the Mark using `^x ^m`. More useful, swap the insertion point and the Mark using `^x ^x`. Since the Mark is then where the insertion point was located you can perform the swap multiple times. Note: there is only one Mark for a given file.

Finally

I've also enclosed a `webloc` file containing a URL to get more information about building the text version of the `DefaultKeyBinding.dict` file. That location also has a link to get a list of additional functions that you might find useful.

Naturally, you can change the key bindings in the enclosed files to those you prefer. Hope you enjoy the extra functionality these files bring to your writing experience.

²As a matter of fact most of these selected keystrokes and commands were inspired by the emacs editor.