# Vintage Mode

Contents

### Overview

Vintage is a vi mode editing package for Sublime Text. It allows you to combine vi's command mode with Sublime Text's features, including multiple selections.

Vintage mode is developed in the open, and patches are more than welcome. If you'd like to contribute, details are on the GitHub page.

## Enabling Vintage

Vintage is disabled by default, via the `ignored_packages` setting. If you remove "Vintage" from the list of ignored packages, you'll be able to edit with vi keys:

1. Select the Preferences/Settings - Default menu item
2. Edit the ignored_packages setting, changing it from:

   ```
           "ignored_packages": ["Vintage"]
   ```

   to:

   ```
           "ignored_packages": []
   ```

   now save the file.
3. Vintage mode is now enabled - you'll see "INSERT MODE" listed in the status bar

Vintage starts in insert mode by default. This can be changed by adding:

```
        "vintage_start_in_command_mode": true
```

to your User Settings.

## What's Included

Vintage includes most basic actions: d (delete), y (copy), c (change), gu (lower case), gU (upper case), g~ (swap case), g? (rot13), < (unindent), and > (indent).

It also includes many motions, including l, h, j, k, W, w, e, E, b, B, alt+w (move by sub-words), alt+W (move backwards by sub-words), $, ^, %, 0, G, gg, f, F, t, T, ^f, ^b, H, M, and L.

Text objects are supported, including words, quotes, brackets and tags.

Repeat ('.') is in there, as is specifying counts for commands and motions. Registers are supported, as are macros and bookmarks. Many other miscellaneous commands are supported too, such and *, /, n, N, s, S and more.

## What's Not

Insert mode is regular Sublime Text editing, with the usual Sublime Text key bindings: vi insert mode key bindings are not emulated.

Ex commands are not implemented, apart from :w and :e, which work via the command palette.

## Under the Hood

Vintage mode is implemented entirely via key bindings and the plugin API – feel free to browse through the Vintage package and see how it's put together. As an example, if you'd like to bind "jj" to exit insert mode, you can add this key binding:

```
{ "keys": ["j", "j"], "command": "exit_insert_mode",
        "context":
        [
                { "key": "setting.command_mode", "operand": false },
                { "key": "setting.is_widget", "operand": false }
        ]
}
```

## OS X Lion

In Lion, holding down a key won't repeat it, but will instead show a popup menu to select between character variations. This doesn't work well with command mode, so you may want to disable it. This can be done via entering this at the terminal:

```
defaults write com.sublimetext.2 ApplePressAndHoldEnabled -bool false
```

## Ctrl Keys

Vintage supports these ctrl key bindings:

- *Ctrl+[*: Escape
- *Ctrl+R*: Redo
- *Ctrl+Y*: Scroll down one line
- *Ctrl+E*: Scroll up one line
- *Ctrl+F*: Page Down
- *Ctrl+B*: Page Up

However, because they conflict with other Sublime Text key bindings, these are disabled by default on Windows and Linux. They can be enabled with the vintage_ctrl_keys setting:

```
        "vintage_ctrl_keys": true
```

## Ex Mode

Please take a look at VintageEx for an Ex mode for Vintage