

Customizing vim

ME 4953/5013 - Introduction to High-Performance Computing




- vim can be tailored by redefining keys or abbreviating frequently used expressions.
- Many preset customizations are available via the `:set` command.
 - Available from `ex` Mode
 - `:set all` lists all available preset settings
 - Setting can be undone by `:set no<command>`, e.g `:set nonumber`

Popular `:set` commands

Option	Significance
<code>autoindent</code>	Next line starts at previous indent level
<code>autowrite</code>	Automatically writes current file when switching files with <code>:e</code>
<code>ignorecase</code>	Ignores case when searching patterns
<code>magic</code>	Treats regular expression characters as special searching for patterns
<code>number</code>	Displays line numbers on screen
<code>showmatch</code>	Shows momentarily a match to a (or {

- The `:map` command allows you to map keyboard shortcuts.
- Example, To map the `g` key in Editor Mode to save the buffer (i.e., `:w`)
 - `:map g :w^M`
 - vim interprets the `^M` (`[Ctrl-M]`) command as `[Enter]`
- To make key sequences in Input Mode use `:map!`
- Use `:unmap` (`:unmap!`) to undo key sequence mappings

- Place all commonly used `:set` commands, keyboard mappings, and abbreviations in a file named `.vimrc` in your home directory (create if necessary)
- View or download my `.vimrc` file to use as a starting place at:
<https://github.com/johntfoster/dotvim/blob/master/vimrc>
-  **Dropbox** Users:
 - Keep a master copy of your `.vimrc` file in your Dropbox folder and symbolically link (`ln -s`) to it on all your UNIX machines so you always have the most up-to-date version.

- Great “scripts” (a.k.a. “plugins”) are available that enable special functionality when editing certain types of files (e.g., L^AT_EX, specific programming language code).
- Browse scripts at:
http://www.vim.org/scripts/script_search_results.php
- Manage scripts with **pathogen** script