Customizing Emacs with PEL

Operation	Keystroke	Function	Notes

PEL: Control Emacs via Easy Customization	PEL is designed to help you get going quickly with Emacs. Instead of having to write Emacs Lisp code, you use Emacs easy-to-use customization system.
	• PEL provides a growing set of customization groups and user option variables that control several aspects of Emacs:
	  • The "**pel-use-**" activation user options identify what built-in or external Emacs Lisp package to use. PEL has logic to autoload the packages only when you need them. This way your Emacs will start quickly even if you have identified a large number of packages.
	  • Once a package or feature is activated with the "pel-use-" user option, the other options control different behaviour of the activated package.
	
	☝ Once you have modified the configuration, execute `M-x pel-init`. PEL will activate the new configuration.		

Emacs Easy Customization	With the following command you can customize anything. with the first command you open the customization buffer and then you can search or browse what you want to customize. The second command allow you to open the buffer at a specific customization group and the third one at a specific user option. These commands prompt for the information you are looking for. You can always use completion by typing `<tab>` at any point to get a list of available groups or variables.		
Customize Emacs	`<M-f11> E`	**(customize)**	Select a customization buffer which you can use to set user options. • User options are structured into "groups". • Initially the top-level group 'Emacs' and its immediate subgroups are shown; the contents of those subgroups are initially hidden.
Customize a specific group	`<M-f11> G`	**(customize-group** &optional GROUP OTHER-WINDOW)	Customize GROUP, which must be a customization group. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Customize a user option	`<M-f11> O`	**(customize-option** SYMBOL)	Customize SYMBOL, which must be a user option.
Customize PEL	The following commands opens the Emacs customization group related to a PEL topic. None of these commands prompt; they open the customization buffer at the requested group.		
All PEL	`<M-f11> !`	**(pel-cfg** &optional OTHER-WINDOW)	Customize PEL support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Speedbar Management	`<M-f11> S`	**(pel-cfg-pkg-speedbar** &optional OTHER-WINDOW)	Customize PEL Speedbar support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
User Identification	`<M-f11> i`	**(pel-cfg-identification** &optional OTHER-WINDOW)	Customize PEL Identification support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Grep	`<M-f11> g`	**(pel-cfg-pkg-grep** &optional OTHER-WINDOW)	Customize PEL Grep support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Session management	`<M-f11> s`	**(pel-cfg-pkg-session** &optional OTHER-WINDOW)	Customize PEL Session support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Window management	`<M-f11> w`	**(pel-cfg-pkg-window** &optional OTHER-WINDOW)	Customize PEL Window support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Configure PEL Programming Language support	The following commands opens the Emacs configuration group to configure PEL support for the specified programming language. You should be able to control most of the important features of the programming languages through these customizations including the activation of important packages as well as aspects of programming language styles like indentation style and width.		
AppleScript & Text-to-Speech (audio narration)	`<M-f11> SPC a`	**(pel-cfg-pkg-applescript** &optional OTHER-WINDOW)	Customize PEL Applescript support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
C	`<M-f11> SPC c`	**(pel-cfg-pkg-c** &optional OTHER-WINDOW)	Customize PEL C support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
C++	`<M-f11> SPC C`	**(pel-cfg-pkg-c++** &optional OTHER-WINDOW)	Customize PEL C++ support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
D	`<M-f11> SPC D`	**(pel-cfg-pkg-d** &optional OTHER-WINDOW	Customize PEL D support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Emacs Lisp	`<M-f11> SPC l`	**(pel-cfg-pkg-elisp** &optional OTHER-WINDOW)	Customize PEL Elisp support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Elixir	`<M-f11> SPC x`	**(pel-cfg-pkg-elixir** &optional OTHER-WINDOW)	Customize PEL Elixir support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Erlang	`<M-f11> SPC e`	**(pel-cfg-pkg-erlang** &optional OTHER-WINDOW)	Customize PEL Erlang support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Julia	`<M-f11> SPC j`	**(pel-cfg-pkg-julia** &optional OTHER-WINDOW)	Customize PEL Julia support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Common Lisp	`<M-f11> SPC L`	**(pel-cfg-pkg-lisp** &optional OTHER-WINDOW)	Customize PEL Lisp support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Python	`<M-f11> SPC p`	**(pel-cfg-pkg-python** &optional OTHER-WINDOW	Customize PEL Python support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Configure PEL Markup support	The following commands opens the Emacs customization group related to configure PEL support for the specific markup language.		
Graphviz DOT	`<M-f11> SPC g`	**(pel-cfg-pkg-graphviz-dot** &optional OTHER-WINDOW)	Customize PEL Graphviz-Dot support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
reStructuredText	`<M-f11> SPC r`	**(pel-cfg-pkg-reST** &optional OTHER-WINDOW)	Customize PEL Rest support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.