
















Emacs support for the Tcl Programming Language



Description	Keystroke	Function	Note
Tcl Editing	<p>Emacs built-in support for Tcl is provided by the tcl.el file which provides the tcl-mode to edit Tcl source code and the inferior-tcl-mode to run a Tcl REPL in a buffer.</p> <p>PEL provides extra support, described in this table, when  the pel-use-tcl user-option is set to t.</p>		
<div>Last updated on:</div> 2025-03-19			
Open this PDF file. See also: 🔗 Help/Info	<div><f11> SPC t <f1></div> <div><f12> <f1></div>	<p>(pel-help-pdf &optional OPEN-WEB-PAGE)</p>	<p>Open the pel - Tcl local PDF. If the prefix argument (like C-u or M--) is used, then it opens the remote GitHub hosted raw PDF instead. If the pel-flip-help-pdf-arg user-option is set it's the other way around.</p>
🔗 Customize PEL Tcl support	<div><f11> SPC t <f2></div> <div><f12> <f2></div>	<p>(pel-customize-pel &optional OTHER-WINDOW)</p>	<p>Customize PEL Tcl support.</p> <ul style="list-style-type: none"> If OTHER-WINDOW is non-nil (use C-u), display in another window.
🔗 Customize Emacs Tcl support	<div><f11> SPC t <f3></div> <div><f12> <f3></div>	<p>(pel-customize-library &optional OTHER-WINDOW)</p>	<p>Customize Emacs Tcl support (which is currently placed in C group): C</p> <ul style="list-style-type: none"> If OTHER-WINDOW is non-nil (use C-u), display in another window.
Select tcl-mode for extension-less file  The <f12> key is available only until a PEL controlled major mode is activated. Then it becomes a buffer prefix key.	<div><f12></div>	<p>(pel-as &optional FORCE)</p>	<p>Inside a fundamental-mode buffer, interactively select major mode for the buffer. Re-do it with arg.</p> <p>This command is mostly used to set the major mode of a buffer in fundamental-mode', when the <f12> key binding is available for it. After being used once in a buffer the major mode is selected and the PEL key binding will not be available when PEL supports the major mode.</p> <ul style="list-style-type: none"> For Tcl file, select tcl. It will insert a shebang line specified by  pel-tcl-shebang-line user option. For Tcl expect script file, select expect. It will insert a shebang line specified by  pel-tcl-expect-shebang-line user option. <p>PEL defines the (as &optional FORCE) alias unless  pel-has-alias-as user-option is set to nil. You can use M-x as to invoke it.</p>
Comments			
Toggle display of comments in buffer or active region See also: 🔗 Comments	<div><f11> ; ;</div>	<p>(hide/show-comments-toggle &optional START END)</p>	<p>Toggle hiding/showing of comments in the active region or whole buffer.</p> <ul style="list-style-type: none"> If the region is active then toggle in the region. Otherwise, in the whole buffer. <p> This requires the hide-comnt.el package (see 🔗 Comments).  PEL activates it when the pel-use-hide-comnt user option is t.</p>
Generic code skeletons • tempo skeletons See also: • 🔗 Inserting Text • T Templates	<p>Several mechanisms have been developed to allow easy insertion of predefined text in Emacs.  PEL does not yet define skeletons for Tcl. You can use the generic one.</p> <ul style="list-style-type: none"> Emacs provides the built-in skeleton mechanism and the tempo skeletons. <ul style="list-style-type: none"> PEL supports both. They are used a little bit differently. PEL provides generic tempo skeletons you can use for Tcl until PEL adds Tcl-specific skeletons. <ul style="list-style-type: none"> PEL provides key bindings to the tempo skeletons: the generic code templates, accessible via the <f6> prefix key, and the language-specific code templates, accessible via the <f12> key prefix. 		
🔗 Customize PEL Text Insertions control for Tcl code skeletons.	<div><f6> <f2></div> <div><f12> <f12> <f2></div>	<p>(pel-customize-pel &optional OTHER-WINDOW)</p> <p>(pel-customize-generic-skels &optional OTHER-WINDOW)</p>	<p>Open the customization groups that control the format of the various skeletons including the generic skeleton used by the <f6> h key and the <f12><f12> h key (see below).</p> <ul style="list-style-type: none"> If OTHER-WINDOW is non-nil (use C-u), display in other window.
Insert generic file module header block — Language agnostic After inserting the template, navigate though areas that must be filled with: <ul style="list-style-type: none"> forward: C-c . backward: C-c , 	<div><f6> h</div> <div><f12> <f12> h</div>	<p>(pel-generic-file-header)</p>	<p>Insert a file header block at the top of the file. Works only for buffer visiting a file.</p> <p> The command key binding <f6> h is available only 1 second after Emacs has started.</p> <p> As mentioned above PEL does not yet define Tcl-specific skeletons, this uses the generic one.</p> <p> Specify the format of the header via the user-options in the pel-pkg-generic-code-style customization group accessible via <f6> <f2></p> <ul style="list-style-type: none"> Inside a Tcl buffer, <f12> <f2> provides access to the following customization groups: <p> After inserting a template, use tempo-forward-mark and tempo-backward-mark to move to the beginning of each section that must be filled.</p>
Toggle pel-tempo-mode	<div><f6> SPC</div> <div><f12> <f12> SPC</div>	<p>(pel-tempo-mode &optional ARG)</p>	<p>Toggle PEL tempo mode on/off.</p> <p>PEL tempo mode activates C-c . and C-c ,, as well as to C-c C-. and C-c C-, key bindings to navigate across tempo mark hot-spots. When pel-tempo-mode is active the pel-tempo-mode lighter (⚡) is shown on the status bar. The second set of keys are only available in graphics mode.</p> <p> The pel-generic-file-header command inserts the text using a tempo skeleton: the PEL tempo mode is automatically activated by typing <f6> h.</p>
Expand any tag in template Note: PEL default skeleton does not use tags.	<div><f6> <f12></div> <div><f12> <f12> <f12></div>	<p>(tempo-complete-tag &optional SILENT)</p>	<p>Look for a tag and expand it. All the tags in the tag lists in 'tempo-local-tags' (this includes 'tempo-tags') are searched for a match for the text before the point. The way the string to match for is determined can be altered with the variable 'tempo-match-finder'. If 'tempo-match-finder' returns nil, then the results are the same as no match at all.</p> <ul style="list-style-type: none"> If a single match is found, the corresponding template is expanded in place of the matching string. If a partial completion or no match at all is found, and SILENT is non-nil, the function will give a signal. If a partial completion is found and 'tempo-show-completion-buffer' is non-nil, a buffer containing possible completions is displayed.

Emacs & Tcl — References

Document	Notes
The Tcl Programming Language	<ul style="list-style-type: none"> Tcl @ Wikipedia Tcl home