

Emacs Tab Bar

Operation	Keystroke	Function	Note
Emacs Tab Bar Emacs >= 27.1	Emacs 27.1 introduced a tab bar that is quite useful to maintain an Emacs window “context”. PEL key bindings for the management of tab bar are placed within the pel:window prefix, and some are inside the PEL Window Hydra.		
Open this PDF file. See also: 🔗 Help/Info	<code><M-f11> M-= <f1></code>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the 🔗 Tab Bar 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.
🔗 Customize Emacs Tab Bar control	<code><M-f11> M-= <f3></code>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs tab bar management support.
Toggle tab-bar-mode	<code><M-f11> M-= M-=</code>	(tab-bar-mode &optional ARG)	Toggle the tab bar in all frames.
Create/delete tabs			
Create a new tab	<ul style="list-style-type: none"> C-x t 2 <code><M-f11> M-= M-\</code> 	(tab-new &optional ARG FROM-NUMBER)	Create a new tab ARG positions to the right. <ul style="list-style-type: none"> If a negative ARG, create a new tab ARG positions to the left. If ARG is zero, create a new tab in place of the current tab. If no ARG is specified, then add a new tab at the position specified by ‘tab-bar-new-tab-to’. Argument addressing is relative in contrast to ‘tab-bar-new-tab-to’, where argument addressing is absolute. If FROM-NUMBER is a tab number, a new tab is created from that tab.
Tear window in tab bar See: 🔗 Windows	C-x w ^ t <ul style="list-style-type: none"> <code><f11> w i t</code> * <f7> T 	(tab-window-detach)	Move the selected window to a new tab . <ul style="list-style-type: none"> This command removes the selected window from the configuration stored on the current tab, and makes a new tab with that window in its configuration.
Create a tab for specified buffer	<ul style="list-style-type: none"> C-x t b <code><M-f11> M-= M-b</code> 	(switch-to-buffer-other-tab BUFFER-OR-NAME)	Prompt for a buffer and switch to buffer BUFFER-OR-NAME in another tab. <ul style="list-style-type: none"> Like C-x 5 b (which see), but creates a new tab.
Visit a file in a new tab	<ul style="list-style-type: none"> C-x t f C-x t C-f <code><M-f11> M-= M-f</code> 	(find-file-other-tab FILENAME &optional WILDCARDS)	Prompt for FILENAME. , edit file FILENAME, in another tab. <ul style="list-style-type: none"> Like C-x 5 C-f (which see), but creates a new tab. If WILDCARDS is non-nil, FILENAME can include widcards, and all matching files will be visited.
Open a dired buffer for specified directory	<ul style="list-style-type: none"> C-x t f C-x t C-f <code><M-f11> M-= M-d</code> 	(dired-other-tab DIRNAME &optional SWITCHES)	"Edit" directory DIRNAME. Like ‘dired’ but make a new tab.
Close tab	<ul style="list-style-type: none"> C-x t 0 <code><M-f11> M-= M-c</code> 	(tab-close &optional TAB-NUMBER TO-NUMBER)	Close the tab specified by its absolute position TAB-NUMBER. <ul style="list-style-type: none"> If no TAB-NUMBER is specified, then close the current tab and switch to the tab specified by ‘tab-bar-close-tab-select’. Interactively, TAB-NUMBER is the prefix numeric argument, and defaults to 1. TAB-NUMBER counts from 1. Optional TO-NUMBER could be specified to override the value of ‘tab-bar-close-tab-select’ programmatically with a position of an existing tab to select after closing the current tab. TO-NUMBER counts from 1.
Close tab by name	<code><M-f11> M-= M-k</code>	(tab-bar-close-tab-by-name NAME)	Prompt for the tab name and close it.
Navigate across tabs	Move across (select) the tabs in the current Emacs frame with the following commands. <ul style="list-style-type: none"> PEL supports shortcut keys for Emacs running in graphics mode on macOS using the Fn key which it configures as the Hyper modifier. This is not available when emacs is running in terminal mode. 		
Select tab by number	<ul style="list-style-type: none"> Fn 1 Fn 9 M-1 <M-f11> M-t M-9 <M-f11> M-t 	(tab-select)	<ul style="list-style-type: none"> On macOS with Emacs running in graphical mode, PEL dynamically sets the tab-bar-select-tab-modifiers user-option variable to ‘(hyper) , which effectively activates these quick Fn key bindings. On other systems, and for Emacs running in terminal mode, use <u>numeric prefix argument</u> key(s) followed by <code><M-f11> M-t</code>. This provides the extra flexibility of being able to select a larger range of tabs; you can compose an integer using <u>numeric prefix argument</u> keys.
Select next tab	<code><M-f11> M-= M-]</code>	(tab-next &optional ARG)	Switch to next tab. <ul style="list-style-type: none"> With n numeric prefix argument select next ARG tab
Select previous tab	<ul style="list-style-type: none"> C-x t 0 <code><M-f11> M-= M-[</code> 	(tab-previous &optional ARG)	
Select last tab	<code><M-f11> M-= M-l</code>	(tab-last &optional ARG)	Switch to the last tab or ARGth tab from the end of the tab bar. <ul style="list-style-type: none"> Interactively, ARG is the prefix numeric argument; it defaults to 1, which means the last tab on the tab bar. For example, C-u 2 or M-2 prefix selects the tab before the last tab.
Select recent tab	<code><M-f11> M-= M-r</code>	(tab-recent &optional ARG)	Switch to ARGth most recently visited tab. <ul style="list-style-type: none"> Interactively, ARG is the prefix numeric argument and defaults to 1.
Select tab by name • Possibly create it	<ul style="list-style-type: none"> C-x t RET <code><M-f11> M-= M-s</code> 	(tab-switch NAME)	Switch to the tab by NAME. <ul style="list-style-type: none"> Default values are tab names sorted by recency, so you can use M-n to get the name of the most recently visited tab, the second most recent, and so on. When the tab with that NAME doesn’t exist, create a new tab and rename it to NAME.
Manage tabs			
Rename a tab	<code><M-f11> M-= M-a</code>	(tab-rename NAME &optional TAB-NUMBER)	Give the tab specified by its absolute position TAB-NUMBER a new NAME. <ul style="list-style-type: none"> If no TAB-NUMBER is specified, then rename the current tab. Interactively, TAB-NUMBER is the prefix numeric argument, and defaults to the current tab. TAB-NUMBER counts from 1. Interactively, prompt for the new NAME. If NAME is the empty string, then use the automatic name function ‘tab-bar-tab-name-function’.
Move tab to the right/left	<ul style="list-style-type: none"> C-x t m <code><M-f11> M-= M-m</code> 	(tab-move &optional ARG)	Move the current tab ARG positions to the right. <ul style="list-style-type: none"> Interactively, ARG is the prefix numeric argument and defaults to 1. If ARG is negative, move the current tab ARG positions to the left. Argument addressing is relative in contrast to ‘tab-bar-move-tab-to’, where argument addressing is absolute.
Restore tab previous window configuration	<code><M-f11> M-= M-p</code>	(tab-bar-history-back)	Restore a previous window configuration used in the current tab. <ul style="list-style-type: none"> This navigates back in the history of window configurations.
Cancel restoration of tab window layout	<code><M-f11> M-= M-n</code>	(tab-bar-history-forward)	Cancel restoration of the previous window configuration. <ul style="list-style-type: none"> This navigates forward in the history of window configurations.