↖ Topic Index

Closing/Suspending and the Emacs Server

Operation	Keystroke	Function	Note
Close Emacs independent process or Emacs client Last updated on:	To close an independent Emacs process, or a client of the Emacs daemon (without terminating the daemon) you can use the following commands. 2025-05-01		
Close emacs	`C-x C-c`	(**save-buffers-kill-terminal** &optional ARG)	Offer to save each modified buffer, then kill the current connection, run all final hooks and close Emacs.
Close emacs	`⌘-q`	(**save-buffer-kill-emacs** &optional ARG)	🍎 On macOS in graphics mode only: offer to save each modified buffer, then kill the Emacs graphics application.
Suspend Emacs	• `C-x C-z` • `C-z`	(**suspend-frame**)	Suspend the current frame: • with Emacs **in terminal (tty) mode**: suspend Emacs and return to the shell. • Use the **fg** command to return to the suspended Emacs. • with Emacs in graphics mode: just hides the current frame.
Under GNU screen	• Normally `C-z` is mapped for this operation, but in cua-mode, `C-z` is mapped to (undo) and that binding is useful. The `C-x C-z` binding is also available and a good alternative to allow the use of the cua-mode. See **M CUA** • When using **GNU screen**, you may want to set the GNU screen meta key to `C-z` instead of the default `C-a`, as the PEL's **.screenrc** sets it. But then to invoke `C-z` you must type `C-z` followed by `z`. Therefore when Emacs is running in terminal mode and running inside a GNU screen session with that configuration to execute suspend-frame, type: `C-x C-z z`		

• **Emacs Server** • **Emacs Client** • **Client Options** See also: • **Emacs as Daemon @ EmacsWiki** • **Using Emacs Daemon & Client**	**Operation**	**Using PEL provided scripts : ec**	**Using Emacs explicit commands:**
	Start an Emacs daemon	• ec - -start-daemon • ec - -start-daemon —name=NAME	• emacs - -daemon • emacs - -daemon=NAME
	Start a terminal-based client and its daemon if not already running	• ec • ec - -name=NAME	• emacsclient -nw -a "" • emacsclient -nw -a "" —socket-name=NAME
	Start a GUI-based client and its daemon if not already running	• ec -g • ec -g --name=NAME	• emacsclient -c -n -a "" • emacsclient -c -n -a "" —socket-name=NAME
Stopping Emacs Daemon & clients	Inside an emacsclient instance, typing `C-x C-c` closes the client instance, but does not stop the Emacs daemon and none of its other clients. If you want to terminate the Emacs daemon and all of its currently attached Emacs client instance you can use the following commands from within one of the Emacs clients.		
Stop Emacs daemon and all clients safely	`<f11> C-x C-c`	(**pel-shutdown-server**)	Prompt to save buffers then stop/kill Emacs server. • If the current Emacs session is not using an Emacs server the command issues an error and does not stop Emacs. • If the current session is a client attached to an Emacs server it prompts further static that this will close all current clients, giving you the chance to cancel before it's too late.
Kill current Emacs daemon & all its clients ⚠️ Also kills an independent Emacs process. ⚠️ no prompt, modified buffers are not saved.	`M-x kill-emacs`	(**kill-emacs** &optional ARG RESTART)	Exit the Emacs job and kill it. • If ARG is an integer, return ARG as the exit program code. • If ARG is a string, stuff it as keyboard input. • Any other value of ARG, or ARG omitted, means return an exit code that indicates successful program termination. • If RESTART is non-nil, instead of just exiting at the end, start a new Emacs process, using the same command line arguments as the currently running Emacs process. • This function is called upon receipt of the signals SIGTERM or SIGHUP, and upon SIGINT in batch mode. • The value of 'kill-emacs-hook', if not void, is a list of functions (of no args), all of which are called before Emacs is actually killed.
Save buffers then kill current Emacs daemon & all its clients ⚠️ Also kills an independent Emacs process.	• `M-x save-buffers-kill-emacs` • `⌘-q`	(**save-buffers-kill-emacs** &optional ARG RESTART)	Offer to save each buffer, then kill this Emacs process. • With prefix ARG, silently save all file-visiting buffers without asking. • If there are active processes where 'process-query-on-exit-flag' returns non-nil and 'confirm-kill-processes' is non-nil, asks whether processes should be killed. • Runs the members of 'kill-emacs-query-functions' in turn and stops if any returns nil. If 'confirm-kill-emacs' is non-nil, calls it. • If RESTART, restart Emacs after killing the current Emacs process. 🍎 On macOS, inside a graphics mode client.

Enable/Disable Commands

Disabling/ Enabling Commands	Some Emacs commands (like `C-x n n` for **narrowing**) are disabled by default because they might be confusing for new Emacs users. Its possible to enable or disable commands using the following commands.		
Enable a command	`M-x enable-command`	(**enable-command** COMMAND)	Allow COMMAND to be executed without special confirmation from now on. COMMAND must be a symbol. • This command alters the user's .emacs file so that this will apply to future sessions. • It adds a *(put 'COMMAND 'disabled t)* inside the emacs init file.
Disable a command	`M-x disable-command`	(**disable-command** COMMAND)	Require special confirmation to execute COMMAND from now on. COMMAND must be a symbol. • This command alters your init file so that this choice applies to future sessions. • It adds a (put 'COMMAND 'disabled nil) inside the emacs init file.