

Description	Keystroke	Function	Note
Ispell - continue spell checking	C-u M-\$	(ispell-continue)	Continue a halted spelling session beginning with the current word.
Ispell Process Control			
Ispell - kill the ispell process	<f11> \$ K	(ispell-kill-ispell &optional NO-ERROR CLEAR)	Kill current Ispell process (so that you may start a fresh one). 👉 The spell check program runs as a background task connected via a pipe. It's not taking much CPU when no spelling is done, so it's normally not necessary to kill it; you can leave it running. However, it may become necessary to kill it when you want to change the dictionary or want to reduce the overhead.
Change Language Dictionary	<f11> \$ D	(ispell-change-dictionary DICT &optional ARG)	Change to dictionary DICT for ispell/aspell. • With a prefix arg, set it "globally", for all buffers. • Without a prefix arg, set it "locally", just for this buffer. 👉 Prompts supports completion: by just answering RET you can find out what the current dictionary is.
Activating Flyspell See also: 🔗 Customize	<p>Flyspell is a minor mode that performs automatic spell-checking. Flyspell can be activated without having to activate ispell, even though several of the ispell customizations also affect Flyspell. However ispell must be installed. Flyspell processes text continuously, just like a word processor, and it highlight misspelled words.</p> <p>👉 It's best to activate Flyspell mode for text buffers and Flyspell Prog mode for programming language buffers with hooks using the following code:</p> <pre>(add-hook 'text-mode-hook 'flyspell-mode) (add-hook 'prog-mode-hook 'flyspell-prog-mode)</pre> <p>👉 🖨️ PEL provides 2 user options that identify the modes where flyspell-mode and flyspell-prog-mode are automatically activated: pel-modes-activating-flyspell-mode and pel-modes-activating-flyspell-prog-mode. PEL code comes with defaults, activating flyspell and flyspell-prog modes for several major modes. Use <f11> \$ <f2> to quickly open the PEL Spelling customization group to configure these 2 user options to add or remove groups. Then 'Apply and Save' and restart Emacs for the changes to take effect.</p> <p>The spell check command bindings are only available when Flyspell (or ispell) mode is active.</p> <p>⚠️ A 3-button mouse is needed for Flyspell to access the pop-up menu of provided replacements suggestions, and the pop-ip menu does not work in terminal mode, unless code is used to fix this problem. Example of this code is shown in the FlySpell page of EmacsWiki. The PEL package incorporates that code and activates it in terminal mode using the following code:</p> <pre>(when (not (display-graphic-p)) (eval-after-load "flyspell" '(progn (fset 'flyspell-emacs-popup 'pel-spell-flyspell-emacs-popup-textual))))</pre>		
Enter/Leave Flyspell mode	<f11> \$ F	(flyspell-mode &optional ARG)	Toggles the use of Flyspell mode.
			<ul style="list-style-type: none"> Mode line shows “Fly” when Flyspell mode is active. Flyspell mode works like word processors; misspelled words are highlighted. Use Flyspell Prog mode for code; Flyspell processes all text. With a prefix argument ARG, enable Flyspell mode if ARG is positive, and disable it otherwise. Flyspell mode is a buffer-local minor mode. When enabled, it spawns a single ispell/aspell process and checks each word. The default flyspell behavior is to highlight incorrect words. <p>👉 If a hook already activate Flyspell you probably won't need this command unless you want to disable it. If you do that in a buffer for programming language you probably will want to re-activate it using the next command.</p>
Enter Flyspell Prog mode	<f11> \$ p	(flyspell-prog-mode)	Turn on Flyspell prog mode: turn on Flyspell but restricts it to comments and strings, do not spell check source code itself. Highlight misspellings only in comments or strings. 👉 If a hook activates Flyspell Prog mode, you won't need this command. ⚠️ Note that the command always enables the mode, it does not toggle it. If you want to turn spell checking off, you must use the flyspell-mode command. To re-enable Flyspell Prog mode you then use this one.
Using Flyspell	With Flyspell mode activated, the following key bindings are active and can be used to fix spelling of misspelled or incomplete words.		
Auto correct previous word See Also: 🔗 Cursor	<ul style="list-style-type: none"> C-; <f11> \ 	(flyspell-auto-correct-previous-word POSITION)	Auto correct the first misspelled word that occurs before point. ⚠️ Both iEdit and Flyspell use the C-; key as their default binding. PEL detects and reports that situation. If you see this warning modify the binding of one of the two user options.
Flyspell - complete a word	<ul style="list-style-type: none"> M-<tab> C-M-i C-. 	(flyspell-auto-correct-word)	Correct the current word in place. <ul style="list-style-type: none"> This command proposes various successive corrections for the current word. If invoked repeatedly on the same position, it cycles through the possible corrections of the current word. In most cases this is much faster than using M-\$ which always proposes choices. <p>👉 🖨️ If you want to include flyspell corrections inside the abbreviation table to automatically correct future typos you can modify the following flyspell user-options:</p> <ul style="list-style-type: none"> flyspell-abbrev-p : set it to t to automatically store flyspell corrections in local abbrev table. flyspell-use-global-abbrev-table-p : set it to t to have it store in the global abbrev table instead.
Ispell - Check a single word	M-\$	(ispell-word &optional FOLLOWING QUIETLY CONTINUE REGION)	Check spelling of word under or before the cursor. <ul style="list-style-type: none"> Opens a “Choices” buffer showing all available corrections/suggestions, similar to the way ispell does it. Several options are available at that moment: see the following “Ispell operation” lines in the above table for the single line commands that can then be used.
Flyspell - correct word See also: 🔗 Highlight	<ul style="list-style-type: none"> C-c \$ 4r 	(flyspell-correct-word-before-point &optional EVENT OPOINT)	Pop up a menu of possible corrections for misspelled word before point. 🖨️ With PEL, the 4r key-chord is also available when key-chord is available and active. See 🔗 Key-Chords . <ul style="list-style-type: none"> ⚠️ To activate this in terminal mode you must write some code. See the note in the “Activating Flyspell” row above. ⚠️ fci-mode interferes with pop-up menu displays in terminal-mode, at least with the one used by flyspell-correct-word-before-point: the menu lines become all jagged, they do not line up vertically. The problem does not affect Emacs running in graphics mode.
Using Flyspell when not activated	The following command can be used even when Flyspell mode is not activated.		
Check all text in buffer		(flyspell-buffer)	Flyspell whole buffer. <ul style="list-style-type: none"> This command is marginally useful. You can use it when Flyspell mode is not active to highlight misspelled words in the buffer. Since the other Flyspell commands bindings are not available you have to fix spelling of the words manually and re-run the command. A better way is to simply activate Flyspell and use the commands. <p>To use this command, type: M-x flyspell-buffer</p>
Querying Information	The following commands print status information inside the mini-buffer about the spell check programs being used.		
Ispell - check version	<f11> \$ v	(ispell-check-version &optional INTERACTIVEP)	Display Ispell process version as well as the version of ispell.el
Show information about used spell program	<f11> \$?	(pel-spell-show-use)	Display what spell checking program is being used, its version, the status of the spell modes and the dictionary used.

Spell Checking — References

Topic & link	Description
Make ispell automatically clear minibuffer when replacing word	
How can I change the language in Emacs when using ispell?	
Enabling spell-checking in comments	
in Emacs flyspell-mode, how to add new word to dictionary?	
Aspell Windows @ EmacsWiki	In Setup for 64-bit Windows 7
GNU Aspell (Win32 version)	