










Spell Checking

Description	Keystroke	Function	Note
<div> <div> Spell Checking in Emacs </div> <div> <ul style="list-style-type: none"> ispell man page <ul style="list-style-type: none"> ispell file format aspell man page hunspell man page enchant man page </div> </div>	Emacs support spell checking inside text files but also inside source code comments and source code docstrings! Two main modes of operation are supported and both are built-in Emacs: <ul style="list-style-type: none"> ispell: a mode where you request an explicit spell-check verification of a word or an area of the current buffer. flyspell: an active mode that runs in the background and detects spelling error on the fly, highlighting errors. <ul style="list-style-type: none"> There is also a program-mode flyspell which activates automatic spell check of source code comments and docstrings. The actual spell checking is performed by a ispell-compatible spell checking process: ispell , aspell , hunspell , or enchant programs are supported. <ul style="list-style-type: none"> ispell does not support UTF-8 encoding. It was the first spell checking program used by Emacs. Prefer aspell to ispell. aspell is preferred to ispell because it is more modern (it supports UTF-8), faster that ispell and hunspell, and quite good for suggestions. Unfortunately its maintenance slowed down since 2011. hunspell is actively maintained, used by several open source project. It support UTF-8 as well. enchant is a front-end that attempt to unify spell-checkers. It supports aspell, hunspell and several others, broadening the choice further. These programs are not bundled with Emacs; you may have to install these programs independently. 		
<div> <div> Open this PDF file. </div> <div> See also: 🔗 Help/Info </div> </div>	<f11> \$ <f1>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the 🔗 Spell Checking 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.
<div> <div> 🔗 Customize PEL spell checking control </div> </div>	<f11> \$ <f2>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL support for: spell checking. Identify which major modes will automatically activate either flyspell-mode or flyspell-prog-mode. <ul style="list-style-type: none"> If OTHER-WINDOW is non-nil (use C-u) , display in other window.
<div> <div> 🔗 Customize Emacs spell checking control </div> </div>	<f11> \$ <f3>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs spelling support. Opens the following customization groups: ispell, flyspell.
<div> <div> Using Emacs Ispell </div> </div>	Once Ispell (or Flyspell) is activated the Ispell commands are available. But you must first select the spell checking program:		
<div> <div> <div>  Select Spell checking program </div> <div> See also: 🔗 Customize </div> </div> </div>	Without PEL: <ul style="list-style-type: none"> Identify the full path of the spell checking program in the ispell-program-name user-option. When using PEL: <ul style="list-style-type: none"> Set the name only of the spell checking program accessible via PATH in the pel-spell-check-tool user-option. <ul style="list-style-type: none"> PEL sets ispell-program-name from it. Use <f11> \$ <f2> key sequence to gain access to the pel-pkg-for-spelling customization group where you can set pel-spell-check-tool. Use <f11> \$ <f3> to open the customization buffer of Ispell and Flyspell. For the changes to take effect, save the changes and execute pel-init (with M-x pel-init) or restart Emacs. 		
<div> <div> <div>  Select Spell Checking dictionaries </div> <div> <ul style="list-style-type: none"> Base dictionary Personal dictionary </div> </div> <div>  Use the same natural language in both dictionaries! </div> </div>	Emacs ispell supports two different dictionary at all times: the base dictionary and the personal dictionary: <ol style="list-style-type: none"> The base dictionary is identified by either ispell-dictionary or by ispell-local-dictionary user-options. <ul style="list-style-type: none"> The ispell-local-dictionary, if defined, takes precedence. Normally you define ispell-dictionary user-option only. You can override it with a file-local definition of ispell-local-dictionary. <ul style="list-style-type: none"> See 🔗 File/Directory Variables. Both base dictionaries are identified by their natural language code name. <ul style="list-style-type: none"> The list of available dictionaries depend on the spell checking tool you are using. Execute ispell-change-dictionary to see the list. See below. The personal dictionary , identified by the ispell-personal-dictionary user-option. <ul style="list-style-type: none"> This is identified by its filename with a complete path. The file is created when you add a word to your personal dictionary during spell checking operation. The name of the natural language is written on the first line of the dictionary file. It is best to identify the language name in the file name.  Spell checking done with aspell will fail if the natural language used by the base dictionary differs from the one inside the personal dictionary! 		
	<ul style="list-style-type: none"> PEL activates flyspell-mode and fix issues in terminal mode. When running in terminal mode, the function modifies ‘flyspell-emacs-popup’ with ‘pel-spell-flyspell-emacs-popup-textual’ to allow the flyspell pop-up menu to work in terminal mode. 		
<div> <div> Querying Information </div> </div>	The following commands print status information inside the mini-buffer about the spell check programs being used.		
<div> <div> Ispell - check version </div> </div>	<f11> \$ v	(ispell-check-version &optional INTERACTIVEP)	Display Ispell process version as well as the version of ispell.el
<div> <div> Show spell checking programs and status </div> </div>	<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. It displays something like what is shown below:
	<pre> -UU-:----F1 App.hs Top (1,0) Git-master (Haskell WK Anzu Fly ElDoc) 3:09pm 1.59 ----- ispell: off, flyspell: on, flyspell-prog: on. Spell check program used : aspell (using: @(#) International Ispell Version 3.1.20 (but really Aspell 0.60.8)) Spell main dictionary : aspell default dictionary. (using LANG: en_CA.UTF-8) Spell personal dictionary: ~/.emacs.d/.ispell Flyspell prevention lock : off </pre>		
<div> <div> <div> Process Control </div> <div> <div> Change Language Dictionary </div> <div>  Use the same natural language in both dictionaries! </div> </div> </div> </div>	<f11> \$ D	(ispell-change-dictionary DICT &optional ARG)	Change to dictionary DICT for ispell/aspell. <ul style="list-style-type: none"> By default set the dictionary for the current buffer (locally). With a prefix arg (C-u or M-- or something else), set it "globally", for all buffers.
	 Prompts supports completion: hit TAB to see the complete list of available dictionaries. <ul style="list-style-type: none"> To see the name of the current dictionary type RET when prompted.  If you use a personal dictionary, it must use the same natural language as the base dictionary you select with this command, otherwise the spell checking operation may fail . <ul style="list-style-type: none"> Aspell will fail, issuing a message like this: Error: Expected language "fr" but got "en". <ul style="list-style-type: none"> The above error occurs when the base dictionary language code is "fr" but the personal dictionary has word in the language code "en". 		
<div> <div> Ispell - kill the ispell process </div> </div>	<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.		
<div> <div> <div> Manual spell check </div> <div> Ispell - complete a word </div> </div> </div>	<ul style="list-style-type: none"> M-<tab> C-M-i C-. 	(ispell-complete-word &optional INTERIOR-FRAG)	Try to complete the word before or at point. <ul style="list-style-type: none"> If optional INTERIOR-FRAG is non-nil, then the word may be a character sequence inside of a word. Standard ispell choices are then available. Notes: <ol style="list-style-type: none"> this also works in Org-Mode, even though the binding is not available. If flyspell is activated, the keys are bound to flyspell-auto-correct-word (see below).
<div> <div> Ispell - Check a single word </div> </div>	M-\$	(ispell-word &optional FOLLOWING QUIETLY CONTINUE REGION)	Check spelling of word under or before the cursor. Several options are available at that moment: see the following “ Ispell operation ” lines below for the single line command that can then be used. A list of replacement is shown in a buffer. Use the letter i to include the word into the dictionary.
<div> <div> <div> Fix spelling mistake before point </div> <div> <ul style="list-style-type: none"> Add old->new in the abbreviation table </div> </div> <div> See also: 🔗 Abbreviations </div> </div>	<ul style="list-style-type: none"> <f11> a \$ <f11> M-\$ 	(pel-ispell-word-then-abbrev &optional LOCALLY)	Fix spelling mistake in text before point. <ul style="list-style-type: none"> Create an ‘abbrev’ abbreviation for it. Store the abbreviation globally unless the LOCALLY argument is non-nil, in which case store it in the local abbreviation list. If there's nothing wrong with the word at point, keep looking for a typo until the beginning of buffer. You can skip typos you don't want to fix with ‘ SPC ’, and you can abort completely with ‘ C-g ’.  A similar operation is possible with flyspell. See flyspell-auto-correct-word .

Description	Keystroke	Function	Note
Flyspell - complete a word	<ul style="list-style-type: none"> M-<code><tab></code> 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. 🙄🗿 If you want to include flyspell corrections inside the abbreviation table to automatically correct future typos you can modify the following flyspell user-options: <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 \$ <u>4r</u> 	(flyspell-correct-word-before-point &optional EVENT OPOINT)	Pop up a menu of possible corrections for misspelled word before point. <div>🗿🔗 With PEL, the <u>4r</u> key-chord is also available when key-chord is available and active. See 🗿Key-Chords.</div> <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	M-x flyspell-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.
Selecting a dictionary for the file using file local variables	To select a local dictionary, use the following: Local IspellDict: To select a local personal dictionary, use the following: Local IspellPersDict:		
Example	<pre>;; Local Variables: ;; mode: emacs-lisp ;; comment-column: 40 ;; ispell-check-comments: exclusive ;; ispell-local-dictionary: "american" ;; End:</pre>		

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?	
GNU Aspell - latest version: 0.60.8	Aspell is a very good spell checking program and library. Unfortunately maintenance has severely slowed down. See: <ul style="list-style-type: none"> Aspell and Hunspell: The Tale of Two Spell Checkers, by Sumit Khanna, Sep 27, 2016. <ul style="list-style-type: none"> GNU Aspell @ GNU GNU Aspell @ Github GNU Aspell @ Wikipedia
Aspell 0.61 Manual	The latest version of the Aspell manual as of Nov 2021. Formatting is not as nice as the manual for version 0.60.9
Gnu Apell 0.60.9 Manual	The manual of the version currently available under Homebrew (as of Nov. 2021).
GNU Aspell - Mailing Lists	The place to get support. The following lists are available: <ul style="list-style-type: none"> aspell-announce - archives aspell-devel - archives aspell-user - archives
Aspell Dictionaries	<ul style="list-style-type: none"> Aspell dictionary files @ GNU, list organized by the ISO 639-1 2-letter language codes. Files are .tar.bz2 Note however that aspell dictionary files are environment dependent compiled files. Read the section on aspell dictionary files listed below first. On macOS use homebrew to install aspell. it also installs the dictionary files.
Aspell directory files See: <ul style="list-style-type: none"> Aspell Manual - Working With Dictionaries 	To list aspell configuration, use the following command: aspell config <ul style="list-style-type: none"> This lists all aspell configuration information, including the data-dir that identifies the location of the aspell dictionary files. <ul style="list-style-type: none"> On a macOS system with aspell installed with homebrew, the dictionary files are stored inside the following directory: <code>/usr/local/Cellar/aspell/0.60.8/lib/aspell-0.60</code> File types: <ul style="list-style-type: none"> .alias : aspell dictionary alias name, a list of aspell commands identifying another .rws or .multi file. .amf : aspell mode filter control file .cmap : aspell character map file .cset : aspell character set data file .dat : language data file, uses the same format as aspell configuration file. <ul style="list-style-type: none"> The *-phonet.dat files are the <i>soundlike</i> files used for phonetic comparisons. The *-affix.dat files are affix compression files. .info : aspell filter option files .kbd : keyboard layout files (identifies side-by-side keys that may cause mis-typing). .multi: multi-dictionary compound instructions which refer to multiple .rws files. .rws : compiled dictionary platform dependent file. Created by the aspell create master command.

Topic & link	Description
<p>Testing aspell on the command line with the available dictionaries:</p> <p>Aspell produces better results than hunspell: Note that the aspell results for the French language is superior to what hunspell is able to detect (see the results for the same test run with hunspell below).</p>	<p>Testing in English:</p> <pre>> echo htink aspell -a --sug-mode=ultra --lang=en_US @(#) International Ispell Version 3.1.20 (but really Aspell 0.60.8) & htink 4 0: think, stink, ht ink, ht-ink</pre> <p>Test en français:</p> <pre>> echo francais aspell -a --sug-mode=ultra --lang=fr_CA @(#) International Ispell Version 3.1.20 (but really Aspell 0.60.8) & francais 7 0: français, française, fiançais, François, fronçais, franc ais, franc-ais</pre> <pre>> echo francias aspell -a --sug-mode=ultra --lang=fr @(#) International Ispell Version 3.1.20 (but really Aspell 0.60.8) & francias 5 0: francisa, francisas, français, franciens, francien</pre>
Aspell Windows @ EmacsWiki	In Setup for 64-bit Windows 7
GNU Aspell (Win32 version)	
Hunspell	<p>Hunspell is more popular than aspell because it is currently (in 2021) actively maintained and used in several Open Source programs such as LibreOffice, Firefox, Chrome, and several others. Unfortunately it is not as good as aspell in some respect. The two sets of tests in French here show one situation where aspell is better.</p> <ul style="list-style-type: none">Hunspell Home PageHunspell @ GithubHunspell @ Wikipedia
Hunspell-compatible dictionary files	<ul style="list-style-type: none">libreoffice/dictionaries - libre-office dictionary wiki - git repository<ul style="list-style-type: none">French: Grammalecte-dic(fr)<ul style="list-style-type: none">Dictionnaires Hunspell 7.0 , Lexique 7.0, Thésaurus et Césures (téléchargement)
Hunspell files: dictionary and affix files.	<p>The document titled “Editing the spell checking dictionaries” from the Chromium Project, describes the format and purpose of the files used by hunspell:</p> <ul style="list-style-type: none">the .dic files: dictionary files: the list of words.the .aff files: the affix rules files: a list of rules and other options.
Location of Hunspell directories	<p>The hunspell -D command lists the hunspell directories it is able to find and lists the searched directories.</p> <ul style="list-style-type: none">On my macOS system the directories listed include the following:<ul style="list-style-type: none">/usr/share/hunspell/usr/share/myspell/usr/share/myspell/dicts/Library/Spelling~/Library/Spelling... and several directories for OpenOffice, even though I have LibreOffice and several files are stored inside the ~/Library/Application Support/LibreOffice/... directory tree. <p>I installed several dictionaries using LibreOffice and they are not listed by hunspell -D.</p> <ul style="list-style-type: none">So I searched for them using the fd -g *.aff and the fd -g *.dic commands.Then I copied the files into my ~/Library/Spelling directory. <p>Now the hunspell -D command lists the directories available.</p>
Testing hunspell com the command line wit available dictionaries:	<p>Testing in English:</p> <pre>> echo htink hunspell -a -d en_US @(#) International Ispell Version 3.2.06 (but really Hunspell 1.7.0) & htink 4 0: think, stink, ht ink, ht-ink</pre> <p>Test en français:</p> <pre>> echo francais hunspell -a -d fr-classique @(#) International Ispell Version 3.2.06 (but really Hunspell 1.7.0) & francais 5 0: français, francisa, franchis, franc ais, franc-ais</pre> <pre>> > echo francias hunspell -a -d fr-classique @(#) International Ispell Version 3.2.06 (but really Hunspell 1.7.0) & francias 5 0: francisa, francisas, franciens, franchisas, francs</pre>
Language Codes	
ISO 639 Language Codes	<ul style="list-style-type: none">ISO 639-1 @ Wikipedia. ISO 639-1 : the 2-letter language codesISO 639.2 Language Code List