


Cross Reference Back-End Capabilities

Back-end ➡ / Feature ¶	elisp	etags (pure etags or Universal -CTags with etags format)	cscope	GNU global gtags with support from Universal-Ctags and Pygments	gxref	xref with Universal-CTags	RTags	dumb-jump
Uses external program?	No. <ul style="list-style-type: none"> Internal to Emacs. Uses elisp-mode.el 	Yes: etags or ctags (Universal-CTags) <ul style="list-style-type: none"> To create the TAGS file 	Yes. <ul style="list-style-type: none"> To create cscope.files and cscope.out files, the list of files to index and the index database. 					
Emacs command to run the external command?			Yes:					
Tags-based?	No		No	No				
Can use Tags?			No					
Requires interpretation/load of examined source code?	Yes. Only able to detect identifiers that have already been defined from .el files that have been loaded.		No					
(can) use external database file(s)?	No		Yes - requires it	Yes				
Support multiple definitions in code	Yes, Honours xref front-end selection.		Yes					
Support list the use of identifier			Yes					
Support multiple directory trees of source code?	Yes: the etags command must be given files from several directory trees with their full pathnames to get these paths in the TAGS file.		No? (I have not find how,  need to investigate the idea of symlinks and file list) .	<ul style="list-style-type: none"> Yes but they have to all be under the same root! One way around this is to use the <code>—file</code> option and symlinks: <ol style="list-style-type: none"> create symlinks to the external into the directory where the GTAGS files will be created. Use find or fd to create a listfile of all files you need to parse. Inside that generated listfile replace the path of files in external directories with the path using the symlinks. Execute gtags <code>—file listfile</code> 				
Support compressed archives?	Yes	Yes, etags process .gz files and list the file name without the .gz extension. This way, generated TAGS can work even if a file was compressed or de-compressed after the creation of the TAGS file, as long as the emacs code that handles the TAGS file is able to detect the .gz file even if the reference is the name of the uncompressed file. <ul style="list-style-type: none"> Emacs 25, 26 and 27.1 xref-etags-backend fails (see GNU bug report #44494). PEL has a work-around for this bug. 	No? (I have not found how)	No , gtags has no option to handle compressed files like .gz files. <p>Work-around: decompress all files before running gtags. For example, using fd, rags and gunzip, keeping the .gz files:</p> <pre>fd -e .gz --print0 xargs -0 gunzip -k</pre>				
Automatically activates mode when opening a file via an cross-reference		No PEL will have to add a mechanism to do that		Yes, seems to work.				
Supported Programming Languages	Emacs Lisp		<ul style="list-style-type: none"> C (CC-mode) C++ partly (old - 2012) Java partly (old - 2012) 	A very large set, including all languages PEL supports. Global has direct support for some but it also uses Universal CTags and Pygments to help it include support for more.				

Back-end ➡ / Feature ⚡	elisp	etags (pure etags or Universal -CTags with etags format)	cscope	GNU global gtags with support from Universal-Ctags and Pygments	gxref	xref with Universal-CTags	RTags	dumb-jump
Support moving point to definition	Yes • M- .		Yes • with xcscope: C-c s d • with helm-cscope: M- .	Yes • M- .				
Support finding all callers of a function	Yes • M- ?		Yes	Yes • M- ?				
Support finding all function called by a function			Yes					
Support finding all assignment to a symbol			Yes					
Support finding #include files (C, C+ +)			Yes					
Has command to refresh tags/ database file(s)			Yes					
Support automatic refresh of tags/ database file(s)			No					
Loads tags/ database file inside Emacs to use			No					
PEL shell file to create the TAGS/ database file(s)	• ~/bin/tags-for- pel • ~/bin/etags- el		None for the moment					
Support finding use of identifiers?	Yes, with M-? But it uses find and grep: it is slow.		Yes					
Emacs package support it			• xcscope • helm-cscope					
Support showing results in helm buffer?			Yes, only some commands with helm-cscope					
Support showing results in ivy list			No					