↖ Topic Index

🚦 Cross-Reference Front-End Capabilities 🚧

Back-end ➠ / Feature ⬇ (recommendations)	Default xref selector	ivy-xref interface	helm-xref interface
Built-in	**Yes , built-in.**	**No**. Requires **ivy-xref** and **ivy** . PEL activates with: **pel-use-ivy-xref**	**No**. Requires: • **helm-xref** . PEL activates with: **pel-use-helm-xref** : this automatically activates **pel-use-helm** 1. **Helm mode**. PEL activates with : **pel-use-helm**
Case sensitivity in tag search?	**Possible:** controlled by **tags-case-fold-search** user-option. Part of etags. • With PEL toggle it with `<f11> X X`	**Possible:** controlled by **tags-case-fold-search** user-option. Part of etags. • With PEL toggle it with `<f11> X X`	**Possible:** controlled by **tags-case-fold-search** user-option. Part of etags. • With PEL toggle it with `<f11> X X`
Directly jump to unique target?	**No**, always show a list in the xref buffer. You have to select it to jump to target.	**Yes. M-.** jumps right to the unique target. It only prompts on multiple targets.	**Yes. M-.** jumps right to the unique target. It only prompts on multiple targets.
Uses external program?	**No**. Internal to Emacs. Uses elisp-mode.el	Yes: etags or ctags (Universal-CTags) • To create the TAGS file	Yes. • To create cscope.files and cscope.out files, the list of files to index and the index database.
Support moving point to definition	**Yes** • `M-.`	**Yes** • `M-.`	**Yes** • with **xcscope**: `C-c s d` • with **helm-cscope**: `M-.`
Can perform extra filtering on multiple targets	No, but use navigation and search command in the buffer.	**Yes**	**Yes**
Case Sensitivity			
Emacs package support it			• **xcscope**, **helm-cscope**
Support finding all callers of a function	**Yes** • `M-?`		Yes
Emacs command to run the external command?			Yes:
Tags-based?	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
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) .
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. • 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)
Automatically activates cross-reference mode when opening a file via a cross-reference		**No** PEL will have to add a mechanism to do that	
Support finding all function called by a function			Yes

Back-end ➡ / Feature⬇ (recommendations)	Default xref selector	ivy-xref interface	helm-xref interface
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
Can fin use of identifiers?	Yes, with M-? But it uses find and grep: it is slow.		Yes
Show results in helm buffer?			Yes, only some commands with helm-cscope
Show results in ivy list?			No

Cross Reference Tools by major-modes

Recommendations	Default xref selector	ivy-xref interface	helm-xref interface	Others	Description and Recommendation
	The following provides recommendations for setting effective cross-reference mechanism for various major-modes. This is a work in progress 🚧 and it will evolve over time, based on my experience and feedback I might get.				
Ada 🚧future					
𝔓𝔈 - Arc ⓕⓜ					
𝔓𝔈 - C			• C (CC-mode)		
𝔓𝔈 - C++			• C++ partly (old - 2012)		
𝔓𝔈 - Chez ⓕⓜ					
𝔓𝔈 - Chibi ⓕⓜ					
𝔓𝔈 - Chicken ⓕⓜ					
𝔓𝔈 - Clojure ⓕⓜ					
Common Lisp ⓕⓜ					
Crystal 🚧future					
𝔓𝔈 - D ⓘⓕⒶ					
Dart 🚧future					
Eiffel 🚧future					
𝔓𝔈 - Elm 🚧future Ⓕ					
𝔓𝔈 - Elixir ⓒⓜⓕⒶ					

Recommendations	Default xref selector	ivy-xref interface	helm-xref interface	Others	Description and Recommendation

Emacs Lisp				With: **Lispy**:
				• `M-.` To jump to symbol definition.
				• `M-,` To jump back. (**pop-tag-mark**)
					• Lispy binds `M-.` to **lispy-goto-symbol**.
					  • It can find the location of **any symbol** that is **already bound**:
					  • any function, variable, customized user-option, even C function if you have the C source installed. It also opens .el.gz compressed source files.
					  • Does **not** find the location of functions that are part of files that are not yet loaded.

Erlang ⓒⓕⒶ					
Factor Ⓚⓕ∞ⓜ					
Forth Ⓚ					
Fortran future					
Gambit ⓕⓜ					
Gerbil ⓕⓜⒶ					
GNU Guile ⓕⓜ					
Gleam					
Go					
Groovy future					
Haskell Ⓕ					
Haxe future					
Hy *(python)* ⓜ					
Janet ⓘⓕⓜ					
Java future			• Java partly (old - 2012)		
Javascript					
Julia ⓜ					
Kotlin future					
LFE ⓒⓜⓕⒶ					
Lua future					
Modula future					
NetRexx					
Nim ⓜ					
Objective-C future					
OCaml ⓘⓕ					
Pascal future					
Perl					
Python					
Purescript Ⓕ					
Racket ⓕⓜ					
ReasonML					
REXX					

Recommendations	Default xref selector	ivy-xref interface	helm-xref interface	Others	Description and Recommendation
𝔓𝔩 - Ruby					
𝔓𝔩 - Rust					
Scala 🚧future					
𝔓𝔩 - Scheme ⓕⓜ					
Seed7 🚧future					
Swift 🚧future					
𝔓𝔩 - Tcl 🚧future ⓕⓘ					
𝔓𝔩 - Typescript 🚧					
𝔓𝔩 - UNIX Shell					
𝔓𝔩 - V					
Zig 🚧future					