↖ Topic Index

🚦 Xref Mode support 🚧

Language	dumb-jump	etags	GNU Global gtags with ggtags	Universal Ctags	rtags for C/C++ only	Notes	Best choice
Ada		**Yes**		**Yes**			
Apex	**Yes**						
assembler		**Yes**	**Yes**	**Yes**			
autoconf		No		**Yes**			
automake		No		**Yes**			
Awk		No	**via CTags**	**Yes**			
Basic		No		**Yes**			
Batch		No	**via CTags**	**Yes**			
Bash	**Yes**						
C	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	• etags —declarations create tags for declarations	
C++	**Yes**	**Yes**	**via CTags**	**Yes**	**Yes**	• etags -Q support class qualification • etags —declarations create tags for declarations	
C#	**Yes**	**Yes**	**via CTags**	**Yes**			
Clojure	**Yes**	No		**Yes**			
CMake		No		**Yes**			
Cobol		**Yes**	**via CTags**	**Yes**			
Common Lisp	**Yes**	**Yes**	**via CTags**	**Yes**			
Coq	**Yes**						
Crystal	**Yes**						
D		No		**Yes**			
Dart	**Yes**						
Eiffel		No		**Yes**			
Elixir	**Yes**	No		**Yes**			
Elm		No		**Yes**			
Emacs Lisp	**Yes**	**Yes**	**via CTags**	**Yes**		• etags supports Emacs Lisp and C, the 2 programming languages used by Emacs. • creates a TAGS file • etags can create a TAGS file for files in several directory trees, with files in .gz files. • tools: • **etags-el**: create TAGS file for 1 directory	etags

Erlang	**Yes**	**Yes**	**via CTags**	**Yes**		• The **ivy-erlang-complete** external package also provides completion and navigation for Erlang.
						  • See **𝔓𝔈𝔏 - Erlang**.
						• etags and ctags produce the same results.
						  • PEL provides the **etags-erl** script that supports both but uses etags by default. It also creates tags for Erlang and C files.
						• the Global/gtags tool is faster than etags/ctags based searches.
							• etags comes with Emacs and does not require extra installation of Universal CTags. It's easy to use.
							• gtags/global create larger files but provide more features if you are willing to learn global command line.

F#	**Yes**						
Falcon		No		**Yes**			
Faust	**Yes**						
Fennel	**Yes**						
Flex		No		**Yes**			
Forth		**Yes**		No			
Fortran	**Yes**	**Yes**	**via CTags**	**Yes**			
Fypp		No		**Yes**			
Go	**Yes**	**Yes**		**Yes**			
Groovy	**Yes**						
Haskell	**Yes**	No		No			
Janet							
Java	**Yes**	**Yes**	**Yes**	**Yes**		• etags -Q support class qualification • etags —declarations create tags for declarations	
Javascript	**Yes**	No	**via CTags**	**Yes**			
Julia	**Yes**						
Kotlin	**Yes**						
LaTex	**Yes**	**Yes**		No			
LFE	**Yes**	No		No			
Lua	**Yes**	**Yes**	**via CTags**	**Yes**			
makefile		**Yes**		**Yes**			
Matlab	**Yes**	No	**via CTags**	**Yes**			
Nim	**Yes**	No		No			
Nix	**Yes**						

Language	dumb-jump	etags	GNU Global gtags with ggtags	Universal Ctags	rtags for C/C++ only	Notes	Best choice
Objective C	**Yes**	**Yes**		**Yes**		• etags -Q support class qualification • etags —declarations create tags for declarations	
OCaml	**Yes**	No	**via CTags**	**Yes**			
OpenSCAP	**Yes**						
Org mode	**Yes**						
Pascal	**Yes**	**Yes**	**via CTags**	**Yes**			
Perl	**Yes**	**Yes**		**Yes**			
PHP	**Yes**	**Yes**		**Yes**			
Postscript		**Yes**		No			
proc		**Yes**		No			
Prolog		**Yes**		No			
Protocol Buffers	**Yes**						
Python	**Yes**	**Yes**	**via CTags**	**Yes**			• gtags/global is faster than etags for large directory tree. To use make sure to run use-gtags before you start Emacs. • etags comes with Emacs and does not require extra installation of Universal CTags. It's easy to use but like gtags/global requires the creation of a file: the TAGS file. • dumb-jump works without having to create a database or tag file.
R	**Yes**	No		**Yes**			
Racket	**Yes**						
REXX		No		**Yes**			
Ruby	**Yes**	**Yes**	**via CTags**	**Yes**			
Rust	**Yes**	No		**Yes**			
Sass	**Yes**						
Scala	**Yes**						
Scheme	**Yes**	**Yes**	**via CTags**	**Yes**			
SML	**Yes**						
Solidity	**Yes**						
Swift	**Yes**						
SQL	**Yes**	No		**Yes**			
TCL	**Yes**	No	**via CTags**	**Yes**			
Terraform / HCL	**Yes**						
Tex		**Yes**		**Yes**			
Texinfo		**Yes**		?			
V		No		No			
Vala	**Yes**						
Verilog	**Yes**	No	**via CTags**	**Yes**			
VHDL	**Yes**	No	**via CTags**	**Yes**			
yacc		**Yes**	**Yes**	**Yes**			
Zig	**Yes**						

Integration with Language Specific Tools

Tool	Programming Language	xref	auto-complete	company	flycheck	flymake	ivy	helm
rtags	**C, C++**	Yes	Yes	Yes	Yes	No	Yes	Yes
hashtags	**Haskell**							
fast-tags	**Haskell**							

PEL script Tools to help with tag creation programs

Tool	PEL Tool	Description

GNU Global gtags with ggtags	**envfor-gtags**	A shell script that must be sourced: add an alias to turn this into a shell command. I call the alias: ***use-gtags***
		• It sets up environment variables required by gtags.
		  • You will have to update it to conform to your environment.
		• Run it before issuing the gtags command to create the Global tag files.
		☝
		Unlike etagcs/ctags, gtags creates 3 files that can then be used on the command line by the **global command line tool** which provides a large set of searching capabilities. Inside Emacs the Global database is accessed via the ggtags Emacs external package. But you can also learn global command line and use it from the command line to perform more complex operations.
		  • From within Emacs access global man page with **<f11> ? m global RET**

Tool	PEL Tool	Description
etags	**etags-c**	Build a etags TAGS file for C source code files. • Type it without arguments to get help.
	etags-cpp	Build a etags TAGS file for C++ source code files. • Type it without arguments to get help.
	etags-el	Build a etags TAGS file for Emacs Lisp source code files. • Type it without arguments to get help.
	etags-erl	Build a etags TAGS file for Erlang source code files. • Type it without arguments to get help.
	etags-lisp	Build a etags TAGS file for Common Lisp source code files. • Type it without arguments to get help.
	etags-py	Build a etags TAGS file for Python source code files. • Type it without arguments to get help.
Universal Ctags	The above etags-XX commands also support the Universal CTags. • To use them with Universal CTags set the **ETAGS_USE_UCTAGS** environment variable in the shell before running the etags-XX command. • For shells that support ability to set the environment variable for the duration of the next command you can issue the command like this at the root of a Python source code directory tree: `ETAGS_USE_UCTAGS=1 etags-py .`	

xref-backend-functions

		local	Global	
C	c-mode		• etags--xref-backend	With nothing activated
			• dumb-jump-xref-activate • etags--xref-backend	With pel-use-dumb-jump = t
		• ggtags--xref-backend • t	• etags--xref-backend	with ggtags-mode active in buffer
Python			• etags--xref-backend	With nothing activated
		• ggtags--xref-backend • t	• etags--xref-backend	with ggtags-mode active in buffer
Erlang		• erlang-etags--xref-backend • t	etags--xref-backend	With nothing activated
		• ggtags--xref-backend • erlang-etags--xref-backend • t		with ggtags-mode active in buffer. Tested a little on jabbed source, but could not get many hits.
Rust	rust-mode		etags--xref-backend	With nothing activated
		• ggtags--xref-backend • t	etags--xref-backend	with ggtags-mode active in buffer
Javascript	js-mode		etags--xref-backend	With nothing activated
				with ggtags-mode active in buffer

GGTags Programming Language Support

Language	Modules	Types	Classes	Functions/Macros	Symbols		
C	• Header files:			• functions: yes • pre-processor macros: yes	• pre-processor symbols: •		
C++							
Erlang							