↖ Topic Index

Xref Mode support 🚧

Language	etags	GNU Global gtags with ggtags	Universal Ctags	rtags for C/C++ only	Notes	Best choice
Ada	**Yes**		**Yes**			
assembler	**Yes**	**Yes**	**Yes**			
autoconf	No		**Yes**			
automake	No		**Yes**			
Awk	No	**via CTags**	**Yes**			
Basic	No		**Yes**			
Batch	No	**via CTags**	**Yes**			
C	**Yes**	**Yes**	**Yes**	**Yes**	• etags —declarations create tags for declarations	
C++	**Yes**	**via CTags**	**Yes**	**Yes**	• etags -Q support class qualification • etags —declarations create tags for declarations	
C#	**Yes**	**via CTags**	**Yes**			
Clojure	No		**Yes**			
CMake	No		**Yes**			
Cobol	**Yes**	**via CTags**	**Yes**			
Common Lisp	**Yes**	**via CTags**	**Yes**			
D	No		**Yes**			
Eiffel	No		**Yes**			
Elixir	No		**Yes**			
Elm	No		**Yes**			

Emacs Lisp	**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
					  • **tags-for-pel** : create 1 TAGS file for all PEL and Emacs elisp and C code.
						etags

Erlang	**Yes**	**via CTags**	**Yes**			
Falcon	No		**Yes**			
Flex	No		**Yes**			
Forth	**Yes**		No			
Fortran	**Yes**	**via CTags**	**Yes**			
Fypp	No		**Yes**			
go	**Yes**		**Yes**			
Haskell	No		No			
Java	**Yes**	**Yes**	**Yes**		• etags -Q support class qualification • etags —declarations create tags for declarations	
Javascript	No	**via CTags**	**Yes**			
LaTex	**Yes**		No			
LFE	No		No			
lua	**Yes**	**via CTags**	**Yes**			
makefile	**Yes**		**Yes**			
Matlab	No	**via CTags**	**Yes**			
Nim	No		No			
Objective C	**Yes**		**Yes**		• etags -Q support class qualification • etags —declarations create tags for declarations	
OCaml	No	**via CTags**	**Yes**			
Pascal	**Yes**	**via CTags**	**Yes**			
Perl	**Yes**		**Yes**			
php	**Yes**		**Yes**			
Postscript	**Yes**		No			
proc	**Yes**		No			
Prolog	**Yes**		No			
Python	**Yes**	**via CTags**	**Yes**			
R	No		**Yes**			
REXX	No		**Yes**			
Ruby	**Yes**	**via CTags**	**Yes**			
Rust	No		**Yes**			
Scheme	**Yes**	**via CTags**	**Yes**			
SQL	No		**Yes**			
TCL	No	**via CTags**	**Yes**			
Tex	**Yes**		**Yes**			
Texinfo	**Yes**		?			

Language	etags	GNU Global gtags with ggtags	Universal Ctags	rtags for C/C++ only	Notes	Best choice
V	No		No			
Verilog	No	**via CTags**	**Yes**			
VHDL	No	**via CTags**	**Yes**			
yacc	**Yes**	**Yes**	**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.
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

	C			
Modules	• Header files:			
functions	Yes			
structures				
Macros	• pre-processor symbols: • pre-processor macros: yes			