

# Xref Mode support

Language	<a href="#">dumb-jump</a>	<a href="#">etags</a> (Emacs 29.3)	<a href="#">GNU Global gtags</a> with <a href="#">ggtags</a>	<a href="#">Universal Ctags</a> 6.1.0	<a href="#">rtags</a> for C/C++ only	Notes See: <a href="#">Using ctags and etags for cross referencing with TAGS</a>	Best choice
Ada		Yes		Yes			
Apex	Yes	No					
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	No					
C	Yes	Yes	Yes	Yes	Yes	<ul style="list-style-type: none"> <li>etags — declarations create tags for declarations</li> </ul>	
C++	Yes	Yes	via CTags	Yes	Yes	<ul style="list-style-type: none"> <li>etags -Q support class qualification</li> <li>etags — declarations create tags for declarations</li> </ul>	
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	No					
Eiffel		No		Yes			
Elixir	Yes	No		Yes			
Elm		No		Yes			
Emacs Lisp	Yes	Yes	via CTags	Yes		<ul style="list-style-type: none"> <li>etags supports Emacs Lisp and C, the 2 programming languages used by Emacs.</li> <li>creates a TAGS file</li> <li>etags can create a TAGS file for files in several directory trees, with files in .gz files.</li> <li>tools:               <ul style="list-style-type: none"> <li><b>etags-el</b>: create TAGS file for 1 directory</li> </ul> </li> </ul>	etags
Erlang	Yes	Yes	via CTags	Yes		<ul style="list-style-type: none"> <li>The <a href="#">ivy-erlang-complete</a> external package also provides completion and navigation for Erlang.               <ul style="list-style-type: none"> <li>See <a href="#">PEL - Erlang</a>.</li> </ul> </li> <li>etags and ctags produce the same results.               <ul style="list-style-type: none"> <li>PEL provides the <b>etags-erl</b> script that supports both but uses etags by default. It also creates tags for Erlang and C files.</li> </ul> </li> <li>the Global/gtags tool is faster than etags/ctags based searches.</li> </ul>	<ul style="list-style-type: none"> <li>etags comes with Emacs and does not require extra installation of Universal CTags. It's easy to use.</li> <li>gtags/global create larger files but provide more features if you are willing to learn global command line.</li> </ul>
F#	Yes	No					
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	No					
Haskell	Yes	No		No			
HTML		Yes					
Janet							
Java	Yes	Yes	Yes	Yes		<ul style="list-style-type: none"> <li>etags -Q support class qualification</li> <li>etags — declarations create tags for declarations</li> </ul>	
Javascript	Yes	No	via CTags	Yes			
Julia	Yes	No					
Kotlin	Yes	No					
LaTeX	Yes	Yes		No			
LFE	Yes	No		No			
Lua	Yes	Yes	via CTags	Yes			
makefile		Yes		Yes			
Matlab	Yes	No	via CTags	Yes			
Mercury		Yes					

Language	<a href="#">dumb-jump</a>	etags (Emacs 29.3)	GNU Global gtags with ggtags	<a href="#">Universal Ctags</a> 6.1.0	<a href="#">rtags</a> for C/C++ only	Notes  See: <a href="#">Using ctags and etags for cross referencing with TAGS</a>	Best choice
Nim	Yes	No		No			
Nix	Yes	No					
Objective C	Yes	Yes		Yes		<ul style="list-style-type: none"><li>etags -Q support class qualification</li><li>etags —declarations create tags for declarations</li></ul>	
OCaml	Yes	No	via CTags	Yes			
OpenSCAP	Yes	No					
Org mode	Yes	No					
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	No					
Python	Yes	Yes	via CTags	Yes			<ul style="list-style-type: none"><li>gtags/global is faster than etags for large directory tree. To use make sure to run use-gtags before you start Emacs.</li><li>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.</li><li>dumb-jump works without having to create a database or tag file.</li></ul>
R	Yes	No		Yes			
Racket	Yes						
REXX		No		Yes			
Ruby	Yes	Yes	via CTags	Yes			
Rust	Yes	Yes		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
<a href="#">rtags</a>	C, C++	Yes	Yes	Yes	Yes	No	Yes	Yes
<a href="#">hashtags</a>	Haskell							
<a href="#">fast-tags</a>	Haskell							
<a href="#">jsctags</a>	javascript							

PEL script Tools to help with tag creation programs

Tool	PEL Tool	Description
<a href="#">GNU Global gtags</a> with <a href="#">ggtags</a>	<a href="#">envfor-gtags</a>	<p>A shell script that must be sourced: add an alias to turn this into a shell command. I call the alias: <i><b>use-gtags</b></i></p> <ul style="list-style-type: none"><li>It sets up environment variables required by gtags.<ul style="list-style-type: none"><li>You will have to update it to conform to your environment.</li></ul></li><li>Run it before issuing the gtags command to create the Global tag files.</li></ul> <p>👉</p> <p>Unlike etags/ctags, gtags creates 3 files that can then be used on the command line by the <a href="#">global command line tool</a> 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.</p> <ul style="list-style-type: none"><li>From within Emacs access global <a href="#">man</a> page with <code>&lt;f11&gt; ? m global RET</code></li></ul>

Tool	PEL Tool	Description
<b>etags</b>	<b><u>etags-c</u></b>	Build a etags TAGS file for C source code files. <ul style="list-style-type: none"> <li>Type it without arguments to get help.</li> </ul>
	<b><u>etags-cpp</u></b>	Build a etags TAGS file for C++ source code files. <ul style="list-style-type: none"> <li>Type it without arguments to get help.</li> </ul>
	<b><u>etags-el</u></b>	Build a etags TAGS file for Emacs Lisp source code files. <ul style="list-style-type: none"> <li>Type it without arguments to get help.</li> </ul>
	<b><u>etags-erl</u></b>	Build a etags TAGS file for Erlang source code files. <ul style="list-style-type: none"> <li>Type it without arguments to get help.</li> </ul>
	<b><u>etags-lisp</u></b>	Build a etags TAGS file for Common Lisp source code files. <ul style="list-style-type: none"> <li>Type it without arguments to get help.</li> </ul>
	<b><u>etags-py</u></b>	Build a etags TAGS file for Python source code files. <ul style="list-style-type: none"> <li>Type it without arguments to get help.</li> </ul>
<b>Universal Ctags</b>	The above etags-XX commands also support the Universal CTags. <ul style="list-style-type: none"> <li>To use them with Universal CTags set the <b>ETAGS_USE_UCTAGS</b> environment variable in the shell before running the etags-XX command.</li> <li>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: <div> <b>ETAGS_USE_UCTAGS=1 etags-py .</b> </div> </li> </ul>	

xref-backend-functions

		local	Global	
<b>C</b>	c-mode		<ul style="list-style-type: none"> <li>etags--xref-backend</li> </ul>	With nothing activated
			<ul style="list-style-type: none"> <li>dumb-jump-xref-activate</li> <li>etags--xref-backend</li> </ul>	With pel-use-dumb-jump = t
		<ul style="list-style-type: none"> <li>ggtags--xref-backend</li> <li>t</li> </ul>	<ul style="list-style-type: none"> <li>etags--xref-backend</li> </ul>	with ggtags-mode active in buffer
<b>Python</b>			<ul style="list-style-type: none"> <li>etags--xref-backend</li> </ul>	With nothing activated
		<ul style="list-style-type: none"> <li>ggtags--xref-backend</li> <li>t</li> </ul>	<ul style="list-style-type: none"> <li>etags--xref-backend</li> </ul>	with ggtags-mode active in buffer
<b>Erlang</b>		<ul style="list-style-type: none"> <li>erlang-etags--xref-backend</li> <li>t</li> </ul>	etags--xref-backend	With nothing activated
		<ul style="list-style-type: none"> <li>ggtags--xref-backend</li> <li>erlang-etags--xref-backend</li> <li>t</li> </ul>		with ggtags-mode active in buffer. Tested a little on jabbed source, but could not get many hits.
<b>Rust</b>	rust-mode		etags--xref-backend	With nothing activated
		<ul style="list-style-type: none"> <li>ggtags--xref-backend</li> <li>t</li> </ul>	etags--xref-backend	with ggtags-mode active in buffer
<b>Javascript</b>	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		
<b>C</b>	<ul style="list-style-type: none"> <li>Header files:</li> </ul>			<ul style="list-style-type: none"> <li>functions: yes</li> <li>pre-processor macros: yes</li> </ul>	<ul style="list-style-type: none"> <li>pre-processor symbols:</li> <li></li> </ul>		
<b>C++</b>							
<b>Erlang</b>							