ETAGS

2021-03-30

NAME

etags, ctags - generate tag file for Emacs, vi

SYNOPSIS

etags [-aCDGIQRVh] [-i *file*] [-l *language*]
[-o *tagfile*] [-r *regexp*] [--parse-stdin=*file*]
[--append] [--no-defines] [--globals] [--no-globals] [--no-line-directive] [--include=*file*] [--ignore-indentation] [--language=*language*] [--members] [--no-members] [--output=*tagfile*] [--class-qualify] [--regex=*regexp*] [--no-regex] [--help] [--version] *file* ...

ctags [-aCdgIQRVh] [-BtTuvwx] [-l *language*]
[-o *tagfile*] [-r *regexp*] [--parse-stdin=*file*]
[--append] [--backward-search] [--cxref] [--no-defines] [--globals] [--no-globals] [--no-line-directive] [--ignore-indentation] [--language=*language*] [--members] [--no-members] [--class-qualify] [--output=*tagfile*] [--regex=*regexp*] [--update] [--help] [--version] *file* ...

DESCRIPTION

The **etags** program is used to create a tag table file, in a format understood by **emacs(1)** ; the **ctags** program is used to create a similar table in a format understood by **vi(1)** . Both forms of the program understand the syntax of C, Objective C, C++, Java, Fortran, Ada, Cobol, Erlang, Forth, Go, HTML, LaTeX, Emacs Lisp/Common Lisp, Lua, Makefile, Mercury, Pascal, Perl, Ruby, Rust, PHP, PostScript, Python, Prolog, Scheme and most assembler-like syntaxes. Both forms read the files specified on the command line, and write a tag table (defaults: **TAGS** for **etags**, **tags** for **ctags**) in the current working directory. Files specified with relative file names will be recorded in the tag table with file names relative to the directory where the tag table resides. If the tag table is in /dev or is the standard output, however, the file names are made relative to the working directory. Files specified with absolute file names will be recorded with absolute file names. Files generated from a source file--like a C file generated from a source Cweb file--will be recorded with the name of the source file. Compressed files are supported using gzip, bzip2, xz, and zstd. The programs recognize the language used in an input file based on its file name and contents. The **--language** switch can be used to force parsing of the file names following the switch according to the given language, overriding guesses based on filename extensions.

OPTIONS

Some options make sense only for the **vi** style tag files produced by ctags; **etags** does not recognize them. The programs accept unambiguous abbreviations for long option names.

-a, --append

Append to existing tag file. (For **vi**-format tag files, see also **--update**.)

-B, --backward-search

Tag files written in the format expected by **vi** contain regular expression search instructions; the **-B** option writes them using the delimiter "**?**", to search *backwards* through files. The default is to use the delimiter "**/**", to search *forwards* through files. Only **ctags** accepts this option.

--declarations

In C and derived languages, create tags for function declarations, and create tags for extern variables unless --no-globals is used. In Lisp, create tags for (defvar foo) declarations. In Mercury, declarations start a line with "**:-**" and are always tagged. In addition, this option tags predicates or functions in first rules of clauses, as in Prolog.

-D, --no-defines

Do not create tag entries for C preprocessor constant definitions and enum constants. This may make the tags file much smaller if many header files are tagged.

--globals

Create tag entries for global variables in Perl and Makefile. This is the default in C and derived languages.

--no-globals

Do not tag global variables in C and derived languages. Typically this reduces the file size by one fourth.

--no-line-directive

Ignore **#line** preprocessor directives in C and derived languages. The default is to honor those directives, and record the tags as if the file scanned was the one named in the **#line** directive. This switch is useful when the original file named by **#line** is no longer available.

-i *file*, **--include**=*file*

Include a note in the tag file indicating that, when searching for a tag, one should also consult the tags file *file* **after checking the** current file. Only **etags accepts this option.**

-I, --ignore-indentation

Don't rely on indentation as much as we normally do. Currently, this means not to assume that a closing brace in the first column is the final brace of a function or structure definition in C and C++.

-l *language*, **--language**=*language*

Parse the following files according to the given language. More than one such options may be intermixed with filenames. Use **--help** to get a list of the available languages and their default filename extensions. For example, as Mercury and Objective-C have same filename extension *.m*, **a test based on contents tries to detect** the language. If this test fails, **--language**=*mercury* **or --language**=*objc* **should be used.** The "auto" language can be used to restore automatic detection of language based on the file name. The "none" language may be used to disable language parsing altogether; only regexp matching is done in this case (see the **--regex option).**

--members

Create tag entries for variables that are members of structure-like constructs in PHP. This is the default for C and derived languages.

--no-members

Do not tag member variables.

--packages-only

Only tag packages in Ada files.

--parse-stdin=*file*

May be used (only once) in place of a file name on the command line. **etags will read from standard input and mark the produced tags** as belonging to the file **FILE.**

-Q, --class-qualify

Qualify tag names with their class name in C++, ObjC, Java, and Perl. This produces tag names of the form *class*::*member* for C++ and Perl, *class***(***category***) for Objective C, and** *class.member* **for Java.** For Objective C, this also produces class methods qualified with their arguments, as in *foo*:*bar*:*baz*:*more*.

-o *tagfile*, **--output**=*tagfile*

Explicit name of file for tag table; for **etags only, a file name** of - means standard output; overrides default **TAGS or tags.** (But ignored with **-v or -x.)**

-r *regexp*, **--regex**=*regexp*

Make tags based on regexp matching for the files following this option, in addition to the tags made with the standard parsing based on language. May be freely intermixed with filenames and the **-R** option. The regexps are cumulative, i.e., each such option will add to the previous ones. The regexps are of one of the forms:
[{*language***}]**/*tagregexp*/**[***nameregexp*/**]***modifiers*
@*regexfile*

where *tagregexp* **is used to match the tag. It should not match** useless characters. If the match is such that more characters than needed are unavoidably matched by *tagregexp*, **it may be useful to** add a *nameregexp*, **to narrow down the tag scope. ctags** ignores regexps without a *nameregexp*. **The syntax of regexps is** the same as in emacs. The following character escape sequences are supported: \a, \b, \d, \e, \f, \n, \r, \t, \v, which respectively stand for the ASCII characters BEL, BS, DEL, ESC, FF, NL, CR, TAB, VT.
The *modifiers* **are a sequence of 0 or more characters among** *i*, **which means to ignore case when matching;** *m*, **which means** that the *tagregexp* **will be matched against the**

whole file contents at once, rather than line by line, and the matching sequence can match multiple lines; and *s*, **which implies** *m* **and means that the** dot character in *tagregexp* **matches the newline char as well.**
The separator, which is / **in the examples, can be any character** different from space, tab, braces and **@. If the separator** character is needed inside the regular expression, it must be quoted by preceding it with \\.
The optional **{***language***} prefix means that the tag** should be created only for files of language *language***, and ignored** otherwise. This is particularly useful when storing many predefined regexps in a file.
In its second form, *regexfile* **is the name of a file that contains** a number of arguments to the *--regex=* **option,** one per line. Lines beginning with a space or tab are assumed to be comments, and ignored.

Here are some examples. All the regexps are quoted to protect them from shell interpretation.

Tag the DEFVAR macros in the emacs source files:
--regex='/[\t]*DEFVAR_[A-Z_ \t(]+"\([^"]+\)"/'

Tag VHDL files (this example is a single long line, broken here for formatting reasons):
--language=none --regex='/[\t]*\(ARCHITECTURE\|\| CONFIGURATION\) +[^]* +OF/' --regex='/[\t]*\ \(ATTRIBUTE\|ENTITY\|FUNCTION\|PACKAGE\(BODY\)?\ \|PROCEDURE\|PROCESS\|TYPE\)[\t]+\([^ \t(]+\)/\3/'

Tag TCL files (this last example shows the usage of a *tagregexp***):**
--lang=none --regex='/proc[\t]+\([^ \t]+\)/\1/'

A regexp can be preceded by {*lang*}**, thus restricting it to match** lines of files of the specified language. Use **etags --help to obtain** a list of the recognized languages. This feature is particularly useful inside **regex files. A regex file contains one regex per line. Empty lines,** and those lines beginning with space or tab are ignored. Lines beginning with @ are references to regex files whose name follows the @ sign. Other lines are considered regular expressions like those following **--regex.**
For example, the command
*etags --regex=@regex.file *.c*
reads the regexes contained in the file regex.file.

-R, --no-regex

Don't do any more regexp matching on the following files. May be freely intermixed with filenames and the **--regex option.**

-u, --update

Update tag entries for *files* **specified on command line, leaving** tag entries for other files in place. Currently, this is implemented by deleting the existing entries for the given files and then rewriting the new entries at the end of the tags file. It is often faster to simply rebuild the entire tag file than to use this. Only **ctags accepts this option.**

-v, --vgrind

Instead of generating a tag file, write index (in **vgrind format)** to standard output. Only **ctags accepts this option.**

-x, --cxref

Instead of generating a tag file, write a cross reference (in **cxref format) to standard output. Only ctags accepts this option.**

-h, -H, --help

Print usage information. Followed by one or more --language=LANG prints detailed information about how tags are created for LANG.

-V, --version

Print the current version of the program (same as the version of the emacs **etags is shipped with).**

SEE ALSO

" **emacs " entry in info;** *GNU Emacs Manual*, **Richard** Stallman.
cxref(**1**), **emacs**(**1**), **vgrind**(**1**), **vi**(**1**).

COPYING