NAME

ref - Display a C function header

SYNOPSIS

ref [-t] [-c *class*]... [-f *file*]... *tag*

DESCRIPTION

ref quickly locates and displays the header of a function. To do this, *ref* looks in the "tags" file for the line that describes the function, and then scans the source file for the function. When it locates the function, it displays an introductory comment (if there is one), the function's declaration, and the declarations of all arguments.

SEARCH METHOD

ref uses a fairly sophisticated tag look-up algorithm. If you supply a filename via **-f** *file*, then elvis first scans the tags file for a static tag from that file. This search is limited to the tags file in the current directory.

If you supply a classname via **-c** *class*, then elvis searches for a tag from that class. This search is not limited to the current directory; You can supply a list of directories in the environment variable *TAGPATH*, and *ref* will search through the "tags" file in each directory until it finds a tag in the desired class.

If that fails, *ref* will then try to look up an ordinary global tag. This search checks all of the directories listed in *TAGPATH*, too.

If you've given the **-t** flag, then *ref* will simply output the tag line that it found, and then exit. Without **-t**, though, *ref* will search for the tag line. It will try to open the source file, which should be in the same directory as the tags file where the tag was discovered. If the source file doesn't exist, or is unreadable, then *ref* will try to open a file called "*refs*" in that directory. Either way, *ref* will try to locate the tag, and display whatever it finds.

INTERACTION WITH ELVIS

ref is used by *elvis*' shift-K command. If the cursor is located on a word such as "splat", in the file "foo.c", then *elvis* will invoke *ref* with the command "ref -f foo.c splat".

If *elvis* has been compiled with the -DEXTERNAL_TAGS flag, then *elvis* will use *ref* to scan the tags files. This is slower than the built-in tag searching, but it allows *elvis* to access the more sophisticated tag lookup provided by *ref*. Other than that, external tags should act exactly like internal tags.

OPTIONS

-t Output tag info, instead of the function header.

-f *file* The tag might be a static function in *file*. You can use several -f flags to have *ref* consider static tags from more than one file.

-c *class* The tag might be a member of class *class*. You can use several -c flags to have *ref* consider tags from more than one class.

FILES

tags List of function names and their locations, generated by *ctags*.

refs Function headers extracted from source files (optional).

ENVIRONMENT

TAGPATH

List of directories to be searched. The elements in the list are separated by either semicolons (for MS-DOS, Atari TOS, and AmigaDos), or by colons (every other operating system). For each operating system, *ref* has a built-in default which is probably adequate.

NOTES

You might want to generate a "tags" file the directory that contains the source code for standard C library on your system. If licensing restrictions prevent you from making the library source readable by everybody, then you can have *ctags* generate a "refs" file, and make "refs" readable by everybody.

If your system doesn't come with the library source code, then perhaps you can produce something workable from the *lint* libraries.

SEE ALSO

elvis(1), ctags(1)

AUTHOR

Steve Kirkendall
kirkenda@cs.pdx.edu