NAME

Glib::ParseXSDoc – Parse POD and XSub declarations from XS files.

DESCRIPTION

This is the heart of an automatic API reference documentation system for XS-based Perl modules. FIXME more info here!!

FIXME document recognized POD directives and the output data structures

FUNCTIONS

xsdocparse (@filenames)

Parse xs files for xsub signatures and pod. Writes to standard output a data structure suitable for eval'ing in another Perl script, describing all the stuff found. The output contains three variables:

`$xspods` = ARRAYREF

array of pods found in the verbatim C portion of the XS file, listed in the order found. These are assumed to pertain to the XS/C api, not the Perl api. Any `=for apidoc` paragraphs following an `=object` paragraphs in the verbatim sections are stripped (as are the `=object` paragraphs), and will appear instead in `$data->{$package}{pods}`.

`$data` = HASHREF

big hash keyed by package name (as found in the MODULE line), containing under each key a hash with all the xsubs and pods in that package, in the order found. Packages are consolidated across multiple files.

FYI, this creates a new parser and calls `parse_file` on it for each input filename; then calls `swizzle_pods` to ensure that any `=for apidoc name` pods are matched up with their target xsubs; and finally calls Data::Dumper to write the data to stdout. So, if you want to get finer control over how the output is created, or keep all the data in-process, now you know how. :–)

METHODS

`$Glib::ParseXSDoc::verbose`

If true, this causes the parser to be verbose.

`$parser` = Glib::ParseXSDoc–>new

Create a new xsub parser.

string = `$parser`–>package

Get the current package name. Falls back to the module name. Will be undef if the parser hasn't reached the first MODULE line.

HASHREF = `$parser`–>pkgdata

The data hash corresponding to the current package, honoring the most recently encountered `=for object` directive. Ensures that it exists. Returns a reference to the member of the main data structure, so modifications are permanent and useful.

`$parser`–>parse_file (filename)

Parse one xs file. Stores all the collected data in *`$parser`*'s internal data structures.

`$parser`–>swizzle_pods

Match `=for apidoc` pods to xsubs.

`$parser`–>preprocess_pods

Honor the `__hide__` and `__function__` directives in `=for apidoc` lines.

We look for the strings anywhere, but you'll typically have it at the end of the line, e.g.:

```
  =for apidoc symname __hide__         for detached blocks
  =for apidoc __hide__                 for attached blocks

  =for apidoc symname __function__     for functions rather than methods
  =for apidoc __function__             for functions rather than methods
```

bool = `$parser`−>is_module_line ($line)

Analyze *`$line`* to see if it contains an XS MODULE directive. If so, returns true after setting the *`$parser`*'s *module*, *package*, and *prefix* accordingly.

`$pod` = `$parser`−>slurp_pod_paragraph ($firstline, `$term_regex`=/^=cut\s*/)

Slurp up POD lines from *`$filehandle`* from here to the next *`$term_regex`* or EOF. Since you probably already read a line to determine that we needed to start a pod, you can pass that first line to be included.

`$xsub` = `$parser`−>parse_xsub (\@lines)
`$xsub` = `$parser`−>parse_xsub (@lines)

Parse an xsub header, in the form of a list of lines, into a data structure describing the xsub. That includes pulling out the argument types, aliases, and code type.

Without artificial intelligence, we cannot reliably determine anything about the types or number of parameters returned from xsubs with PPCODE bodies.

OUTLIST parameters are pulled from the args list and put into an "outlist" key. IN_OUTLIST parameters are put into both.

Data type names are not mangled at all.

Note that the method can take either a list of lines or a reference to a list of lines. The flat list form is provided for compatibility; the reference form is preferred, to avoid duplicating a potentially large list of strings.

`$parser`−>clean_out_empty_pods

Looks through the data member of the parser and removes any keys (and associated values) when no pod, enums, and xsubs exist for the package.

AUTHOR

muppet <scott at asofyet dot org>

COPYRIGHT AND LICENSE