newLISPdoc - the newLISP documentation program

last edited 2013-11-07

Comments in newLISP source files can be converted to HTML documentation using only a few tags in comments. The newLISPdoc system is designed to use a minimum of tags and leave the tagged comments still readable.

newLISPdoc also generates an index page for all newLISP source files generated.

See here for the source of newLISPdoc . The program and this documentation are also part of the source distribution of newLISP since version 9.0. Since newLISP version 9.1 syntax highlighting is built into `newlispdoc` which is installed in the same directory as the main newLISP executable program. The script `syntax.cgi` is still available for web site installations, but is not required anymore for `newlispdoc`.

Usage

From within the directory where the modules are, execute with all module filenames to process on the commandline. For example to process the files `mysql.lsp`, `odbc.lsp` and `sqlite3.lsp` do:

```
newlispdoc mysql.lsp odbc.lsp sqlite.lsp
```

Or on Win32:

```
newlisp newlispdoc mysql.lsp odbc.lsp sqlite.lsp
```

This will generate `index.html`, `mysql.lsp.html`, `odbc.lsp.html` and `sqlite.lsp.html` all in the same directory, where the command was executed. The page `index.html` contains links to all other pages.

If the current directory contains the file `newlispdoc.css`, HTML output will be formatted accordingly. For a sample file see `util/newlispdoc.css` in the source distribution.

The command line flag `-s` can be supplied to additionally generate syntax highlighted HTML versions of the source files and put a link to the highlighted version of the source file on the documentation page:

```
newlispdoc -s mysql.lsp odbc.lsp sqlite.lsp
newlispdoc -s *.lsp
```

The `-d` flag supplies a download link to the raw source:

```
newlispdoc -d *.lsp
newlispdoc -s -d *.lsp
```

One or both options can be supplied.

Or on Win32:

```
newlisp newlispdoc -s mysql.lsp odbc.lsp sqlite.lsp
```

Since version 1.3 of newLISPdoc a file containing URLs of source file locations can be specified. This allows indexing and documenting of newLISP source code distributed on different sites:

```
newlispdoc -url file-with-urls.txt
newlispdoc -s -url file-with-urls.txt
```

Or on Win32:

```
newlisp newlispdoc -url file-with-urls.txt
newlisp newlispdoc -s -url file-with-urls.txt
```

`http://` and `file://` URLs can be used. Like with individual files, the `-s` switch can be specified to generate also syntax highlighted source files. A URL file contains one URL per line. No other information is allowed in the file. The following is a sample URL file:

```
http://asite.com/code/afile.lsp
http://othersite.org/somefile.lsp
file:///usr/home/joe/program.lsp
```

The last line shows a file URL from the local filesystem.

All generated files will be written to the current directory.

Tags

The following tags start at the beginning of a line with 2 semicolons and one space before the tag:

```
;; @module one word for module name
;; @index Title and URL for index page
;; @description one line description of the module
;; @location the original URL location of the source file
;; @version one line version info
;; @author one line for author info
```

```
;; @syntax one line syntax pattern
;; @param name description on one line
;; @return description on one line
;; @example multiline code example starting on next line
```

The only required tag is either the `@module` tag or alternatively the `@index` tag. If neither one tag is present in the file, it will not be processed. All other tags are optional. Only lines starting with ;; (2 semicolons) are processed. Program comment text which should not appear in the documentation should start with only one semicolon.

The one line description of the `@description` tag will be put under the module name on the index and module doc page. This and the `@location` where added in June 2007, and are not part of the newlispdoc program in the newLISP v. 9.1 release.

A function may have multiple `@syntax` tags each on consecutive lines.

The following is the only tag, which can be embedded anywhere in the text. Between the tag link specification and description is exactly one space:

```
@link link description
```

Custom tags can be made up by just prepending the custom name with a `@`. The text after the custom tag will be translated as usual, e.g. it may contain a `@link` tag. Like in most other tags, text is limited to the same line.

All words between `<...>` angle braces are displayed in italic. Internally newLISPdoc uses `<em>,</em>` tags for formatting. They should be used for parameter specs after the @param tag and in text referring to those parameters..

All words between single quotes `' ... '` are printed in monospace. Internally newLISPdoc uses `<tt>,</tt>` tags for formatting.

All other lines starting with 2 semicolons contain descriptive text. An empty line with only 2 semicolons at the beginning is a break between paragraphs of text.

Lines not starting with 2 semicolons are ignored by newLISPdoc. This allows doing code comments with just one semicolon.

If more formatting is required than what is offered by newLISPdoc, the following simple HTML tags and their closing forms may also be used: `<h1>,<h2>,<h3>,<h4>, <i>,<em>,<b>,<tt>,<p>, <br>,<pre>,<center>,<blockquote>` and `<hr>`.

Linking to other module collections

newLISPdoc generates and index page for all modules documented. A special tag `@index` can be used to show a link on the index page to an index of other module collections. This way multilevel indices of modules can be created. To display a link to another module collection on the index page, create a file containing a `@index` and a `@description` link:

```
; - other-collection.txt -
;; @index OtherCollection http://example.com/modules
;; @description Modules from OtherCollection
```

Use one or more of these files on the newLISPdoc command line as any other source file:

```
newlispdoc -s other-collection.txt *.lsp
```

This will show the index entry for OtherCollection on the module index before listing all modules in `*.lsp`.

Examples

The following is the commented source of and example program followed by the pages generated in HTML:

```
;; @module example.lsp
;; @author John Doe, johndoe@example.com
;; @version 1.0
;;
;; This module is an example module for the newlispdoc
;; program, which generates automatic newLISP module
;; documentation.


;; @syntax (example:foo <num-repeat> <str-message>)
;; @param <num-repeat> The number of times to repeat.
;; @param <str-message> The message string to be printed.
;; @return Returns the message in <str-message>
;;
;; The function 'foo' repeatedly prints a string to
;; standard out terminated by a line feed.
;;
;; @example
;; (example:foo 5 "hello world")
;; =>
;; "hello world"
;; "hello world"
;; "hello world"
;; "hello world"
;; "hello world"

(context 'example)
```

```
(define (foo n msg)
        (dotimes (i n)
                (println msg))
)

;; See the @link http://example.com/example.lsp source .
```

Below the `example.lsp.html` and `index.html` page generated:

Module index

Module: example.lsp

Author: John Doe, johndoe@example.com
Version: 1.0

This module is an example module for the newlispdoc program, which generates automatic newLISP module documentation.

- § -

Syntax: (example:foo *num-repeat str-message*)

parameter: *num-repeat* - The number of times to repeat.
parameter: *str-message* - The message string to be printed.

return: Returns the message in *str-message*

The function `foo` repeatedly prints a string to standard out terminated by a line feed.

example:

```
(example:foo 5 "hello world")
=>
"hello world"
```

```
"hello world"
"hello world"
"hello world"
"hello world"
```

See the source.

#nbsp;

- ∂ -

generated with newLISP and newlispdoc

Below the index page `index.html` which was generated. When more than one module is specified on the command line, the index page will show one link line for each module.

Index

Module: example.lsp

foo

- ∂ -

generated with newLISP and newlispdoc
