

Common Lisp Documentation Weaver

Mariano Montone (marianomontone@gmail.com)

Copyright © 2021 Mariano Montone

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Table of Contents

1	Introduction	1
2	Installation	2
3	Usage	3
4	Commands	4
5	Documentation systems	5
5.1	Texinfo	5
6	API	6
7	Index	7

1 Introduction

CL-DOCWEAVER is a document weaver for Common Lisp.

Documentation for a Lisp project is written with the user's tool of choice (like Texinfo, Markdown, etc). Then, Common Lisp definitions are expanded into the documentation source using DocWeaver commands.

DOCWEAVER commands give the user control on how definitions are to be expanded, either via command options or by choosing a different set of commands.

CL-DOCWEAVER is easy to extend to support different documentation tools.

Texinfo and Markdown are the ones with best support at this moment.

2 Installation

3 Usage

First choose your documentation system (either Texinfo or Markdown at this moment).

Write your documentation as you normally would for the chosen documentation system; then, whenever you need to reference to something in your Common Lisp system, use DocWeaver commands to expand Lisp definitions.

Commands have the following syntax: (`@command-name &rest args`).

For example, use (`@clfunction alexandria:flatten`) to expand the definition of `ALEXANDRIA:FLATTEN` function.

That looks like this:

```
FLATTEN (tree) [ALEXANDRIA]
  Traverses the tree in order, collecting non-null leaves into a list.
```

Note that commands usually receive options in order to be able to control different aspects of the expanded definition.

By default, docstrings are parsed to extract possible references to other parts of the code; then those references are formatted as links that can be used to navigate the source code.

Finally, use See [DOCWEAVER:WEAVE-FILE function], page 6 to weave your documentation system source files.

4 Commands

@clvariable (*function-symbol* &rest *args*) [Command]

Expands definition for variable bound to FUNCTION-SYMBOL.

For example,

```
(@clvariable cl:*standard-output*)
```

Looks like this:

```
*STANDARD-OUTPUT* [COMMON-LISP]
  default output stream
```

@clfunction (*function-symbol* &rest *args*) [Command]

Expands definition for function bound to FUNCTION-SYMBOL.

For example,

```
(@clfunction alexandria:map-permutations)
```

Looks like this:

```
MAP-PERMUTATIONS (function sequence &key (start 0) end [ALEXANDRIA]
  length (copy t))
  Calls function with each permutation of LENGTH constructable
  from the subsequence of SEQUENCE delimited by START and END. START
  defaults to 0, END to length of the sequence, and LENGTH to the
  length of the delimited subsequence.
```

@clpackage (*package-name* &key (*include-external-definitions t*) [Command]
include-internal-definitions (categorized t))

Expands definition for Common Lisp package named *PACKAGE-NAME*.

If *INCLUDE-EXTERNAL-DEFINITIONS* is T, then all package external definitions are expanded.

If *INCLUDE-INTERNAL-DEFINITIONS* is T, then all package internal definitions are expanded.

If *CATEGORIZED* is T, then package definitions appear categorized in sections (variables, functions, etc).

Example:

```
(@clpackage :alexandria)
```

@clref (*symbol type*) [Command]

Creates a reference to *SYMBOL*. *TYPE* should be one of **variable**, **function**, **class**, etc.

For example, to reference ALEXANDRIA:FLATTEN function, do this:

```
(@clref alexandria:flatten, function)
```

And this is the resulting link: See [ALEXANDRIA:FLATTEN function], page 3

5 Documentation systems

5.1 Texinfo

The Texinfo output needs to include `common-lisp.texi` file, that is shipped with CL-DOCWEAVER.

The `common-lisp.texi` file contains a set of Texinfo macros that are used by CL-DOCWEAVER for expanding Common Lisp definitions.

You can have a look at CL-DOCWEAVER own documentation in `docs/cl-docweaver.texi` for an example for how this should be used.

Also you may want to invoke `makeinfo` and `texi2any` Texinfo commands with `--no-validate` option, as some of the generated references in docstrings may not appear in your final document, and without that option you would get an error.

See `docs/Makefile` in CL-DOCWEAVER source for an example of how Texinfo tools should be used.

6 API

DOCWEAVER

[PACKAGE]

External definitions

Functions

WEAVE-FILE (*file output-file &rest options &key docsystem* [DOCWEAVER]
modules command-prefix (parse-docstrings t))

Weaves documentation source in *FILE* and writes the result to *OUTPUT-FILE*.

Arguments:

- DOCSYSTEM: specify the documentation tool that is being used (*:texinfo*, *:markdown*, etc.).
- MODULES: is the list of *modules* (or ASDF system names) that need to be loaded to be able to read definition descriptions.
- COMMAND-PREFIX: is the character to use as prefix for commands. The character ‘at’ is the default.
- PARSE-DOCSTRINGS: if *T*, then docstings are parsed and highlighted and references to code from it created.

7 Index

(Index is nonexistent)

*

STANDARD-OUTPUT 4

@

@clfunction 4

@clpackage 4

@clref 4

@clvariable 4

A

ALEXANDRIA:FLATTEN 3

ALEXANDRIA:MAP-PERMUTATIONS 4

D

DOCWEAVER:WEAVE-FILE 6

C

COMMON-LISP:*STANDARD-OUTPUT* 4

F

FLATTEN 3

M

MAP-PERMUTATIONS 4

W

WEAVE-FILE 6