

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	6
5.1	Texinfo	6
6	API	7
7	Index	8

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

Write documentation for your Common Lisp project in your documentation tool of your choice (either Texinfo or Markdown at this moment). Then invoke *cl-docweaver* commands to expand Lisp definitions for either variables, functions, macros, classes, or even whole packages.

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

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

The expanded function definition 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 interpreted to extract possible references to other parts of the code; then those references are formatted as links that can be used to navigate the definitions documentations.

Finally, use See [DOCWEAVER:WEAVE-FILE function], page 7 to weave your documentation system source files.

Have a look at *cl-docweaver* documentation in `docs` directory for an example of how all this works.

4 Commands

@setup &rest options [Command]

Configures *cl-docweaver*.

OPTIONS is a *plist* with members:

- **:docsystem** The documentation system to use. Either **:texinfo** or **:markdown**. Default is **:texinfo**.
- **:parse-docstrings** A boolean that indicates if docstrings should be parsed or not. Default is T.
- **:command-prefix** The prefix character to use for commands. Default is the **#\@** character.

@clvariable variable-symbol &rest args [Command]

Expands definition for variable bound to *VARIABLE-SYMBOL*.

For example,

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

Expands to this:

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

A list of symbols is also accepted; variable definitions are expanded in sequence.

For example,

```
(@clvariable (cl:*compile-print* cl:*compile-verbose*))
```

expands to this:

```
*COMPILE-PRINT* [COMMON-LISP]
  The default for the :PRINT argument to See <undefined> [COMMON-
  LISP:COMPILE-FILE function], page <undefined>.
```

```
*COMPILE-VERBOSE* [COMMON-LISP]
  The default for the :VERBOSE argument to See <undefined> [COMMON-
  LISP:COMPILE-FILE function], page <undefined>.
```

@clfunction function-symbol &rest args [Command]

Expands definition for function bound to *FUNCTION-SYMBOL*.

For example,

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

Expands to 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.

Like with See [\[@clvariable\]](#), page [\[undefined\]](#), a list of symbols is also accepted and definitions are expanded in sequence.

@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)

*

COMPILE-PRINT	4
COMPILE-VERBOSE	4
STANDARD-OUTPUT	4

@

@clfunction	4
@clpackage	5
@clref	5
@clvariable	4
@setup	4

A

ALEXANDRIA:FLATTEN	3
ALEXANDRIA:MAP-PERMUTATIONS	4

C

COMMON-LISP:*COMPILE-PRINT*	4
COMMON-LISP:*COMPILE-VERBOSE*	4
COMMON-LISP:*STANDARD-OUTPUT*	4

D

DOCWEAVER:WEAVE-FILE	7
----------------------	---

F

FLATTEN	3
---------	---

M

MAP-PERMUTATIONS	4
------------------	---

W

WEAVE-FILE	7
------------	---