SB-TEXINFO

Documentation String to Texinfo Converter

Table of Contents

Introduction

SB-TEXINFO converts Common Lisp documentation strings for inclusion in a Texinfo manual—which can then be converted into eg. HTML and PDF.

While current implementation is SBCL-only, there is no fundamental reason why support for other Common Lisps could not be added.

SB-TEXINFO was originally written for processing the SBCL docstrings by Rudi Sclatte in 2004, and has been maintained as part of SBCL since then. This version was split from the SBCL sources in 2011 in order to generalize it for documenting other software. Like SBCL, this manual is in the Public Domain.

SB-TEXINFO is maintained in Git:

```
git clone git://github.com/nikodemus/sb-texinfo.git
```

will get you a local copy.

http://github.com/nikodemus/sb-texinfo/

is the GitHub project page, where the issue tracker is located.

1 Overview

There are two main ways of using SB-TEXINFO

- **The Right Way:** The recommended way is to write a manual in Texinfo, and use the `@`include directive to include processed docstrings generated using [Function generate-includes], page 4.
- **Quick and Dirty:** Call [Function document-package], page 4 with the name of the package or packages to document. This will produce a Texinfo file containing extracted documentation for the project.

  This can be a convenient way to produce a template for further development using the first method, and perhaps more importantly quickly allows you to generate reference documentation for packages with docstrings but no manual.

2 Examples

2.1 Using generate-includes

This manual is produced using the [Function generate-includes], page 4 method. ‘sb-texinfo.texinfo’ is the main Texinfo source file, where docstrings are included.

https://raw.github.com/nikodemus/sb-texinfo/master/doc/sb-texinfo.texinfo

A ‘Makefile’ is responsible for running SBCL and the Texinfo toolchain.

https://raw.github.com/nikodemus/sb-texinfo/master/doc/Makefile

Finally, ‘style.css’ is used to prettify the HTML version.

https://raw.github.com/nikodemus/sb-texinfo/master/doc/style.css

This produces the following HTML and PDF files:

http://nikodemus.github.com/sb-texinfo/sb-texinfo.html
http://nikodemus.github.com/sb-texinfo/sb-texinfo.pdf

and a GNU info file as well.

2.2 Using document-package

An alternate version of this manual, produced using

```
(sb-texinfo:document-package
 :sb-texinfo
 :output-file "document-package-sample.texinfo")
```

and further processed using the ‘Makefile’ linked above:

http://nikodemus.github.com/sb-texinfo/document-package-sample.html
http://nikodemus.github.com/sb-texinfo/document-package-sample.pdf

3 Dictionary

`document-package` *package &key output-file standalone title* [Function]

Creates Texinfo documentation for `package` in `output-file`, which defaults to the <shortest-package-name>.texinfo. Returns the pathname of the created file as the primary value. An "include/" directory is created in the same location as the Texinfo file, which contains parts included in the Texinfo file.

If `standalone` is true (the default), a standalone Texinfo file is created. Otherwise a file suitable for including in other Texinfo files is created.

The `title` is used for the documentation, defaulting to capitalized shortest package name.

The generated Texinfo uses include files generated by `generate-includes`, making this function a convenient way to generate an initial template for a manually maintained Texinfo manual.

`document-package-pathname` *package &key output-file &allow-other-keys* [Function]

Returns the pathname used for `output-file` by `document-package` when called with the same arguments.

`generate-includes` *directory packages &key base-package* [Function]

Create files in `directory` containing Texinfo markup of all docstrings of each exported symbol in `packages`. The `directory` is created if it does not exist yet. Trailing slash is required.

The names of the generated files are of the form

```
<doc-type>-<packagename>-<symbol-name>.texinfo
```

and can be included in Texinfo source via `@`include statements. Texinfo syntax-significant characters are escaped in symbol names, but if a docstring contains invalid Texinfo markup, you lose.

Appendix A Function Index