

# The example Reference Manual

---

An example of Declt documentation system

Alexander Artemenko

This manual was generated automatically by Declt 3.0 "Montgomery Scott" on Mon Jan 25 10:54:41 2021 GMT+0.

Copyright © 2021 Alexander Artemenko

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the section entitled "Copying" is included exactly as in the original.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be translated as well.

# Table of Contents

|  |           |
|--|-----------|
| Copying .....                            | 1         |
| <b>1 Introduction .....</b>              | <b>3</b>  |
| 1.1 Pros & Cons .....                    | 3         |
| 1.1.1 Pros .....                         | 3         |
| 1.1.2 Cons .....                         | 3         |
| <b>2 Systems .....</b>                   | <b>5</b>  |
| 2.1 example .....                        | 5         |
| 2.2 example/app .....                    | 5         |
| 2.3 example/utils .....                  | 5         |
| 2.4 example/class .....                  | 5         |
| <b>3 Files .....</b>                     | <b>7</b>  |
| 3.1 Lisp .....                           | 7         |
| 3.1.1 example.asd .....                  | 7         |
| 3.1.2 example/app/file-type.lisp .....   | 7         |
| 3.1.3 example/utils/file-type.lisp ..... | 7         |
| 3.1.4 example/class/file-type.lisp ..... | 7         |
| <b>4 Packages .....</b>                  | <b>9</b>  |
| 4.1 example/app .....                    | 9         |
| 4.2 example/utils .....                  | 9         |
| 4.3 example/class .....                  | 9         |
| <b>5 Definitions .....</b>               | <b>11</b> |
| 5.1 Exported definitions .....           | 11        |
| 5.1.1 Functions .....                    | 11        |
| 5.1.2 Classes .....                      | 11        |
| 5.2 Internal definitions .....           | 13        |
| 5.2.1 Functions .....                    | 13        |
| 5.2.2 Generic functions .....            | 13        |
| 5.2.3 Classes .....                      | 14        |
| <b>6 Conclusion .....</b>                | <b>15</b> |
| <b>Appendix A Indexes .....</b>          | <b>17</b> |
| A.1 Concepts .....                       | 17        |
| A.2 Functions .....                      | 18        |
| A.3 Variables .....                      | 19        |
| A.4 Data types .....                     | 20        |



## Copying

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.



# 1 Introduction

This is the introduction to Declt (<https://www.lrde.epita.fr/~didier/software/lisp/declt/user/index.html>) example project. Declt is a documentation system used to build a Quickref (<https://quickref.common-lisp.net/>) — API reference for all libraries, included into the Quicklisp.

This project is a part of the CL Doc Systems (<https://cl-doc-systems.github.io/>) — attempt to create a comprehensive set of examples of different Common Lisp documentation systems.

## 1.1 Pros & Cons

### 1.1.1 Pros

- Generates manuals in Texinfo which can be converted to HTML, PDF, DVI and PostScript. Here is PDF version ([index.pdf](#)) of this site.
- Automatically embeds license information.
- Uses standard Texinfo format (<https://www.gnu.org/software/texinfo/manual/texinfo/texinfo.html>), which has good documentation. Theoretically, Texinfo should well suite for large documents.
- It is able to generate single or multi page HTML output. To switch the mode, change `single-page-p` variable in `docs/scripts/builder.lisp`.

### 1.1.2 Cons

- Works only under SBCL.
- Free form documentation chapters are limited by "Introduction" and "Conclusion".
- Texinfo is not so popular these days. Also, it is very verbose.
- There is no default CSS theme.
- I wasn't able to use cross referencing Texinfo tags from docstrings. Seems Texinfo markup does not work there. But it works in the introduction and conclusion. Here is example link to the `[do-the-job]`, page 11, function.





## 2 Systems

The main system appears first, followed by any subsystem dependency.

### 2.1 example

**Author** Alexander Artemenko

**License** MIT

**Description**

This description will be used only if long-description is missing

**Long Description**

Test long description. Both descriptions will be shown in the docs.

**Dependency**

[example/app], page 5, (system)

**Source** [example.asd], page 7, (file)

**Directory** /home/runner/work/declt/declt/

### 2.2 example/app

**Author** Alexander Artemenko

**License** MIT

**Dependencies**

- [example/utils], page 5, (system)
- [example/class], page 5, (system)

**Source** [example.asd], page 7, (file)

**Directory** /home/runner/work/declt/declt/

**Component**

[file-type.lisp], page 7, (file)

### 2.3 example/utils

**Author** Alexander Artemenko

**License** MIT

**Source** [example.asd], page 7, (file)

**Directory** /home/runner/work/declt/declt/

**Component**

[file-type.lisp], page 7, (file)

### 2.4 example/class

**Author** Alexander Artemenko

**License** MIT

**Source** [example.asd], page 7, (file)

**Directory** /home/runner/work/declt/declt/

**Component**

[file-type.lisp], page 7, (file)



## 3 Files

Files are sorted by type and then listed depth-first from the systems components trees.

### 3.1 Lisp

#### 3.1.1 example.asd

**Location** /home/runner/work/declt/declt/example.asd

**Systems**

- [example], page 5, (system)
- [example/app], page 5, (system)
- [example/utils], page 5, (system)
- [example/class], page 5, (system)

#### 3.1.2 example/app/file-type.lisp

**Parent** [example/app], page 5, (system)

**Location** app.lisp

**Packages** [example/app], page 9,

**Exported Definitions**

[foo], page 11, (function)

#### 3.1.3 example/utils/file-type.lisp

**Parent** [example/utils], page 5, (system)

**Location** utils.lisp

**Packages** [example/utils], page 9,

**Exported Definitions**

[do-the-job], page 11, (function)

**Internal Definitions**

[concat], page 13, (function)

#### 3.1.4 example/class/file-type.lisp

**Parent** [example/class], page 5, (system)

**Location** class.lisp

**Packages** [example/class], page 9,

**Exported Definitions**

- [admin], page 11, (class)
- [non-documented-user], page 12, (class)
- [user], page 12, (class)

**Internal Definitions**

- [email], page 13, (method)
- [inner-documented-user], page 14, (class)
- [is-admin], page 13, (generic function)
- [is-admin], page 13, (method)

- `[is-admin]`, page 13, (method)
- `[last-login-at]`, page 13, (method)
- `[(setf last-login-at)]`, page 13, (method)
- `[name]`, page 13, (method)

## 4 Packages

Packages are listed by definition order.

### 4.1 example/app

This is docstring for the EXAMPLE/APP package.

The package contains a FOO function which does it's job by applying transformation to the first and second arguments.

**Source** [file-type.lisp], page 7, (file)

**Use List** common-lisp

#### Exported Definitions

[foo], page 11, (function)

### 4.2 example/utils

The utils package.

The only exported function DO-THE-JOB will be show in a separate section "Exported Definitions".

**Source** [file-type.lisp], page 7, (file)

**Use List** common-lisp

#### Exported Definitions

[do-the-job], page 11, (function)

#### Internal Definitions

[concat], page 13, (function)

### 4.3 example/class

This package demonstrates how Declt displays classes and generic functions.

The key concept is USER class

It is possible to check if user has admin privileges, using this IS-ADMIN function.

Right now, IS-ADMIN returns T only for objects of class ADMIN.

**Source** [file-type.lisp], page 7, (file)

**Use List** common-lisp

#### Exported Definitions

- [admin], page 11, (class)
- [non-documented-user], page 12, (class)
- [user], page 12, (class)

#### Internal Definitions

- [email], page 13, (generic function)
- [email], page 13, (method)
- [inner-documented-user], page 14, (class)

- [is-admin], page 13, (generic function)
- [is-admin], page 13, (method)
- [is-admin], page 13, (method)
- [last-login-at], page 13, (generic function)
- [last-login-at], page 13, (method)
- [(setf last-login-at)], page 13, (method)
- [(setf last-login-at)], page 13, (generic function)
- [name], page 13, (generic function)
- [name], page 13, (method)

## 5 Definitions

Definitions are sorted by export status, category, package, and then by lexicographic order.

### 5.1 Exported definitions

#### 5.1.1 Functions

`do-the-job` *FIRST SECOND* [Function]  
 The function does the job.

It CONCATENATES first and second arguments calling internal function `concat`.

On this multiline we'll check how does documentation system processes docstrings.

NOTE: pay attention the Texinfo markup in the second paragraph does not supported in docstrings and I was not able to make CONCATENATES word bold :(

**Package** `[example/utils]`, page 9,

**Source** `[file-type.lisp]`, page 7, (file)

`foo` *FIRST &key OTHER* [Function]  
 This is example function.

\* FIRST - Just a first argument.

\* OTHER - Optional keyword argument. Default is 100500.

Returns: A string with first and other concatenated.

Internally it calls `EXAMPLE/UTILS:DO-THE-JOB` to do the real job.

Note, that the link above is broken, but Coo does not warn us when building the docs. Sphinx issues a warning inn such case.

**Package** `[example/app]`, page 9,

**Source** `[file-type.lisp]`, page 7, (file)

#### 5.1.2 Classes

`admin` () [Class]  
 Admins should have additional priveleges.

**Package** `[example/class]`, page 9,

**Source** `[file-type.lisp]`, page 7, (file)

**Direct superclasses**

`[user]`, page 12, (class)

**Direct methods**

`[is-admin]`, page 13, (method)

|  |         |
|--|---------|
| <code>non-documented-user ()</code>                                | [Class] |
| <b>Package</b> [example/class], page 9,                            |         |
| <b>Source</b> [file-type.lisp], page 7, (file)                     |         |
| <b>Direct superclasses</b>   |         |
| [user], page 12, (class)   |         |
| <code>user ()</code>   | [Class] |
| All users in the system have this class.                           |         |
| Last login slot is updated automatically.                          |         |
| <b>Package</b> [example/class], page 9,                            |         |
| <b>Source</b> [file-type.lisp], page 7, (file)                     |         |
| <b>Direct superclasses</b>   |         |
| standard-object (class)  |         |
| <b>Direct subclasses</b>   |         |
| • [admin], page 11, (class)  |         |
| • [non-documented-user], page 12, (class)                          |         |
| • [inner-documented-user], page 14, (class)                        |         |
| <b>Direct methods</b>  |         |
| • [is-admin], page 13, (method)                                    |         |
| • last-login-at (method)   |         |
| • [last-login-at], page 13, (method)                               |         |
| • [email], page 13, (method)                                       |         |
| • [name], page 13, (method)  |         |
| <b>Direct slots</b>  |         |
| name   | [Slot]  |
| A full username.   |         |
| <b>Type</b> string   |         |
| <b>Initargs</b> :name  |         |
| <b>Readers</b> [name], page 13, (generic function)                 |         |
| email  | [Slot]  |
| Correct email address.   |         |
| <b>Type</b> string   |         |
| <b>Initargs</b> :email   |         |
| <b>Readers</b> [email], page 13, (generic function)                |         |
| last-login-at  | [Slot]  |
| <b>Readers</b> [last-login-at], page 13, (generic function)        |         |
| <b>Writers</b> [(setf last-login-at)], page 13, (generic function) |         |



## 5.2 Internal definitions

### 5.2.1 Functions

`concat` *FIRST SECOND* [Function]  
 This function is not exported and should not be showed in the API reference.  
**Package** [example/utils], page 9,  
**Source** [file-type.lisp], page 7, (file)

### 5.2.2 Generic functions

`email` *OBJECT* [Generic Function]  
**Package** [example/class], page 9,  
**Methods**  
     `email` (*USER* user) [Method]  
     Correct email address.  
     **Source** [file-type.lisp], page 7, (file)

`is-admin` *USER* [Generic Function]  
 Returns t if user can modify the system.  
**Package** [example/class], page 9,  
**Source** [file-type.lisp], page 7, (file)  
**Methods**  
     `is-admin` (*USER* admin) [Method]  
     `is-admin` (*USER* user) [Method]

`last-login-at` *OBJECT* [Generic Function]  
`(setf last-login-at)` *NEW-VALUE OBJECT* [Generic Function]  
**Package** [example/class], page 9,  
**Methods**  
     `last-login-at` (*USER* user) [Method]  
     automatically generated reader method  
     **Source** [file-type.lisp], page 7, (file)  
     `(setf last-login-at)` *NEW-VALUE* (*USER* user) [Method]  
     automatically generated writer method  
     **Source** [file-type.lisp], page 7, (file)

`name` *OBJECT* [Generic Function]  
**Package** [example/class], page 9,  
**Methods**  
     `name` (*USER* user) [Method]  
     A full username.  
     **Source** [file-type.lisp], page 7, (file)

### 5.2.3 Classes

`inner-documented-user ()` [Class]

This class only to demonstrate how Declt's separates exported symbols from internal.

It will not be shown in the separate section "Internal Definitions".

**Package** [example/class], page 9,

**Source** [file-type.lisp], page 7, (file)

**Direct superclasses**

[user], page 12, (class)

## 6 Conclusion

`Declt` can be used when you need to generate API reference for third-party libraries as `Quickref` (<https://quickref.common-lisp.net/>) does for all Quicklisp libraries.

Ability to generate docs in different formats also might be interesting.

But the lack of markup support for docstrings and cross-reference helpers along with limited ability to create free form documentation chapters makes `Declt` useless for documenting 40Ants projects (<https://40ants.com>).



## Appendix A Indexes

### A.1 Concepts

#### E

|                                    |   |
|------------------------------------|---|
| example.asd .....                  | 7 |
| example/app/file-type.lisp .....   | 7 |
| example/class/file-type.lisp ..... | 7 |
| example/utils/file-type.lisp ..... | 7 |

#### F

|  |   |
|--|---|
| File, Lisp, example.asd .....                  | 7 |
| File, Lisp, example/app/file-type.lisp .....   | 7 |
| File, Lisp, example/class/file-type.lisp ..... | 7 |
| File, Lisp, example/utils/file-type.lisp ..... | 7 |

#### L

|   |   |
|---|---|
| Lisp File, example.asd .....                  | 7 |
| Lisp File, example/app/file-type.lisp .....   | 7 |
| Lisp File, example/class/file-type.lisp ..... | 7 |
| Lisp File, example/utils/file-type.lisp ..... | 7 |

## A.2 Functions

|  |    |  |
|--|----|--|
| <b>(</b>                                     |    |  |
| (setf last-login-at) .....                   | 13 |  |
| <b>C</b>                                     |    |  |
| concat .....                                 | 13 |  |
| <b>D</b>                                     |    |  |
| do-the-job .....                             | 11 |  |
| <b>E</b>                                     |    |  |
| email .....                                  | 13 |  |
| <b>F</b>                                     |    |  |
| foo .....                                    | 11 |  |
| Function, concat .....                       | 13 |  |
| Function, do-the-job .....                   | 11 |  |
| Function, foo .....                          | 11 |  |
| <b>G</b>                                     |    |  |
| Generic Function, (setf last-login-at) ..... | 13 |  |
| Generic Function, email .....                | 13 |  |
| Generic Function, is-admin .....             | 13 |  |
| Generic Function, last-login-at .....        | 13 |  |
| Generic Function, name .....                 | 13 |  |
| <b>I</b>                                     |    |  |
| is-admin .....                               | 13 |  |
| <b>L</b>                                     |    |  |
| last-login-at .....                          | 13 |  |
| <b>M</b>                                     |    |  |
| Method, (setf last-login-at) .....           | 13 |  |
| Method, email .....                          | 13 |  |
| Method, is-admin .....                       | 13 |  |
| Method, last-login-at .....                  | 13 |  |
| Method, name .....                           | 13 |  |
| <b>N</b>                                     |    |  |
| name .....                                   | 13 |  |

### A.3 Variables

#### E

email ..... 12

#### L

last-login-at ..... 12

#### N

name ..... 12

#### S

Slot, email ..... 12

Slot, last-login-at ..... 12

Slot, name ..... 12

## A.4 Data types

### A

`admin` ..... 11

### C

Class, `admin` ..... 11  
 Class, `inner-documented-user` ..... 14  
 Class, `non-documented-user` ..... 12  
 Class, `user` ..... 12

### E

`example` ..... 5  
`example/app` ..... 5, 9  
`example/class` ..... 5, 9  
`example/utils` ..... 5, 9

### I

`inner-documented-user` ..... 14

### N

`non-documented-user` ..... 12

### P

Package, `example/app` ..... 9  
 Package, `example/class` ..... 9  
 Package, `example/utils` ..... 9

### S

System, `example` ..... 5  
 System, `example/app` ..... 5  
 System, `example/class` ..... 5  
 System, `example/utils` ..... 5

### U

`user` ..... 12