

# The FoCus Reference Manual

---

FORMAT Customizations, version 1.0 "Kokyu Ho"

Didier Verna <[didier@didierverna.net](mailto:didier@didierverna.net)>

---

This manual was generated automatically by Declt 2.1 "Jonathan Archer" on Tue Feb 28 12:32:21 2017 GMT+1.

Copyright © 2015 Didier Verna

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>
<b>2 Systems .....</b>	<b>5</b>
2.1 net.didierverna.focus.....	5
2.2 net.didierverna.focus.flv.....	5
2.3 net.didierverna.focus.core.....	6
2.4 net.didierverna.focus.setup.....	6
<b>3 Modules .....</b>	<b>9</b>
3.1 core/src/.....	9
<b>4 Files .....</b>	<b>11</b>
4.1 Lisp .....	11
4.1.1 net.didierverna.focus.asd.....	11
4.1.2 net.didierverna.focus.flv.asd.....	11
4.1.3 net.didierverna.focus.core.asd.....	11
4.1.4 net.didierverna.focus.setup.asd.....	11
4.1.5 flv/flv.lisp.....	11
4.1.6 core/meta.lisp.....	11
4.1.7 core/src/util.lisp.....	12
4.1.8 core/src/directive.lisp.....	12
4.1.9 core/src/table.lisp.....	13
4.1.10 core/src/string.lisp.....	13
4.1.11 core/src/wrapper.lisp.....	14
4.1.12 setup/setup.lisp.....	14
<b>5 Packages .....</b>	<b>17</b>
5.1 net.didierverna.focus .....	17
5.2 net.didierverna.focus.user-functions.....	19
5.3 net.didierverna.focus.setup.....	19
<b>6 Definitions .....</b>	<b>21</b>
6.1 Exported definitions .....	21
6.1.1 Special variables .....	21
6.1.2 Macros .....	22
6.1.3 Functions .....	22
6.2 Internal definitions.....	24
6.2.1 Special variables .....	24
6.2.2 Macros .....	24
6.2.3 Functions .....	25
6.2.4 Generic functions.....	29
6.2.5 Conditions.....	31
6.2.6 Structures .....	35

<b>Appendix A</b>	<b>Indexes</b>	<b>39</b>
A.1	Concepts	39
A.2	Functions	40
A.3	Variables	42
A.4	Data types	43

## 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

**Format** is a powerful utility in the Common-Lisp standard. Format strings are written in what can be considered as a printing DSL (Domain Specific Language). However powerful that language is though, it suffers from two important limitations.

1. It is not modifiable: only a predefined set of standard directive characters can be used and it is not possible to alter their semantics.
2. It is hardly extensible. The only embryonic extension mechanism available, the `~/` directive, is extremely cumbersome to use. The called function must either reside in the `cl-user` package, or you must always use its fully qualified name in the format string, even if the corresponding code lies in the appropriate package. For instance, consider that there is a function called `my-format-function` in the package named `:my.long.package.name`. Every time you want to use this function, you need to write something like

```
(format t "~/my.long.package.name:my-format-function/" ...)
```

which essentially makes the `/` directive unusable.

**FoCus** is a library designed to circumvent those limitations. It allows you to customize the `format` DSL by adding new directive characters or modifying the standard ones. The semantics of these directive characters is specified in a so-called *format table*, a concept very close to that of *readtables*. **FoCus** ultimately translates into regular `format` calls.

This is the **FoCus** reference manual, and as such, it is not meant to be read. It may help you find sleep in case of insomnia though. See *The FoCus User Manual*, for a more human-readable guide to using **FoCus**.





## 2 Systems

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

### 2.1 net.didierverna.focus

**Long Name**

FORMAT Customizations

**Author** Didier Verna

**Contact** didier@didierverna.net

**Home Page**

<http://www.lrde.epita.fr/~didier/software/lisp/misc.php#focus>

**Source Control**

<https://github.com/didierverna/focus>

**License** BSD

**Description**

Customizable format strings and directives

**Long Description**

FoCus is a wrapper around the standard FORMAT utility. It allows you to customize the FORMAT DSL by adding new directive characters or modifying the standard ones. The semantics of these directive characters is specified in a so-called "format table", a concept very close to that of readtables. FoCus ultimately translates into regular FORMAT calls.

**Version** 1.0

**Dependencies**

- [net.didierverna.focus.setup], page 6, (system)
- [net.didierverna.focus.core], page 6, (system)
- [net.didierverna.focus.flv], page 5, (system) (for feature net.didierverna.focus.flv)

**Source** [net.didierverna.focus.asd], page 11, (Lisp file)

### 2.2 net.didierverna.focus.flv

**Long Name**

FORMAT Customizations, FLV library

**Author** Didier Verna

**Contact** didier@didierverna.net

**Home Page**

<http://www.lrde.epita.fr/~didier/software/lisp/misc.php#focus>

**Source Control**

<https://github.com/didierverna/focus>

**License** BSD

**Description**

Focus's support for file-local variables

**Long Description**

FoCus's FLV library provides one additional macro: IN-FORMAT-TABLE. This macro is only available when \*FORMAT-TABLE\* can be made file-local.

**Version** 1.0

**Dependencies**

- [net.didierverna.focus.core], page 6, (system)
- net.didierverna.asdf-flv

**Source** [net.didierverna.focus.flv.asd], page 11, (Lisp file)

**Component**

[flv], page 11, (Lisp file)

**2.3 net.didierverna.focus.core****Long Name**

FORMAT Customizations, core library

**Author** Didier Verna

**Contact** didier@didierverna.net

**Home Page**

<http://www.lrde.epita.fr/~didier/software/lisp/misc.php#focus>

**Source Control**

<https://github.com/didierverna/focus>

**License** BSD

**Description**

Customizable format strings and directives

**Long Description**

FoCus's core functionality. For a more complete description of FoCus, see the net.didierverna.focus system.

**Version** 1.0

**Dependency**

[net.didierverna.focus.setup], page 6, (system)

**Source** [net.didierverna.focus.core.asd], page 11, (Lisp file)

**Components**

- [meta], page 11, (Lisp file)
- [src], page 9, (module)

**2.4 net.didierverna.focus.setup****Long Name**

FORMAT Customizations, setup library

**Author** Didier Verna

**Contact** didier@didierverna.net

**Home Page**

<http://www.lrde.epita.fr/~didier/software/lisp/misc.php#focus>

**Source Control**

<https://github.com/didierverna/focus>

**License** BSD

**Description**

FoCus's preload setup library

**Long Description**

FoCus's setup library provides support for various preload configuration parameters. For a more complete description of FoCus, see the `net.didierverna.focus` system.

**Source** [`net.didierverna.focus.setup.asd`], page 11, (Lisp file)

**Component**

[`setup`], page 14, (Lisp file)



## 3 Modules

Modules are listed depth-first from the system components tree.

### 3.1 core/src/

**Dependency**

[meta], page 11, (Lisp file)

**Parent** [net.didierverna.focus.core], page 6, (system)

**Location** core/src/

**Components**

- [util], page 12, (Lisp file)
- [directive], page 12, (Lisp file)
- [table], page 13, (Lisp file)
- [string], page 13, (Lisp file)
- [wrapper], page 14, (Lisp file)



## 4 Files

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

### 4.1 Lisp

#### 4.1.1 net.didierverna.focus.asd

**Location** net.didierverna.focus.asd

**Systems** [net.didierverna.focus], page 5, (system)

#### 4.1.2 net.didierverna.focus.flv.asd

**Location** flv/net.didierverna.focus.flv.asd

**Systems** [net.didierverna.focus.flv], page 5, (system)

#### 4.1.3 net.didierverna.focus.core.asd

**Location** core/net.didierverna.focus.core.asd

**Systems** [net.didierverna.focus.core], page 6, (system)

#### 4.1.4 net.didierverna.focus.setup.asd

**Location** setup/net.didierverna.focus.setup.asd

**Systems** [net.didierverna.focus.setup], page 6, (system)

#### 4.1.5 flv/flv.lisp

**Parent** [net.didierverna.focus.flv], page 5, (system)

**Location** flv/flv.lisp

##### Exported Definitions

[in-format-table], page 22, (macro)

#### 4.1.6 core/meta.lisp

**Parent** [net.didierverna.focus.core], page 6, (system)

**Location** core/meta.lisp

**Packages** [net.didierverna.focus], page 17,

##### Exported Definitions

[nickname-package], page 23, (function)

##### Internal Definitions

- [\*readtable\*], page 24, (special variable)
- [clindent], page 25, (function)
- [defindent], page 24, (macro)
- [i-reader], page 27, (function)
- [in-readtable], page 24, (macro)
- [tilde-reader], page 29, (function)

### 4.1.7 core/src/util.lisp

**Parent** [src], page 9, (module)

**Location** core/src/util.lisp

**Internal Definitions**

- [focus-error], page 31, (condition)
- [other-case], page 27, (function)

### 4.1.8 core/src/directive.lisp

**Dependency**

[util], page 12, (Lisp file)

**Parent** [src], page 9, (module)

**Location** core/src/directive.lisp

**Internal Definitions**

- [\*standard-directives\*], page 24, (special variable)
- [copy-format-directive], page 25, (function)
- [copy-function-directive], page 25, (function)
- [copy-standard-closing-directive], page 25, (function)
- [copy-standard-delimiting-directive], page 25, (function)
- [copy-standard-directive], page 25, (function)
- [copy-standard-opening-directive], page 25, (function)
- [directive-character], page 26, (function)
- [directive-closing-character], page 26, (function)
- [directive-function-name], page 26, (function)
- [(setf directive-function-name)], page 26, (function)
- [directive-opening-character], page 26, (function)
- [find-standard-directive], page 26, (function)
- [format-directive], page 35, (structure)
- [format-directive-p], page 26, (function)
- [function-directive], page 36, (structure)
- [function-directive-p], page 26, (function)
- [make-format-directive], page 27, (function)
- [make-function-directive], page 27, (function)
- [make-standard-closing-directive], page 27, (function)
- [make-standard-delimiting-directive], page 27, (function)
- [make-standard-directive], page 27, (function)
- [make-standard-opening-directive], page 27, (function)
- [nonstandard-directive-character], page 30, (method)
- [nonstandard-directive-character], page 33, (condition)
- [standard-closing-directive], page 36, (structure)
- [standard-closing-directive-p], page 28, (function)
- [standard-delimiting-directive], page 36, (structure)
- [standard-delimiting-directive-p], page 28, (function)



- [standard-directive], page 36, (structure)
- [standard-directive-lookup-error], page 34, (condition)
- [standard-directive-p], page 28, (function)
- [standard-opening-directive], page 37, (structure)
- [standard-opening-directive-p], page 28, (function)

#### 4.1.9 core/src/table.lisp

##### Dependency

[directive], page 12, (Lisp file)

**Parent** [src], page 9, (module)

**Location** core/src/table.lisp

**Packages** [net.didierverna.focus.user-functions], page 19,

##### Exported Definitions

- [\*format-table\*], page 21, (special variable)
- [make-format-table], page 22, (function)
- [register-format-table], page 23, (function)
- [set-format-directive], page 23, (function)
- [unregister-format-table], page 23, (function)
- [with-format-table], page 22, (macro)

##### Internal Definitions

- [%make-format-table], page 25, (function)
- [copy-format-table], page 25, (function)
- [find-table], page 26, (function)
- [format-table], page 35, (structure)
- [format-table-error], page 31, (condition)
- [format-table-p], page 26, (function)
- [lookup-table], page 27, (function)
- [missing-table], page 32, (condition)
- [name], page 29, (method)
- [table], page 30, (method)
- [table-character], page 30, (method)
- [table-collision], page 34, (condition)
- [table-directive-collision], page 34, (condition)
- [table-directive-error], page 34, (condition)
- [table-lookup-error], page 35, (condition)
- [table-mappings], page 29, (function)
- [(setf table-mappings)], page 29, (function)
- [table-name], page 29, (function)

#### 4.1.10 core/src/string.lisp

##### Dependency

[table], page 13, (Lisp file)

**Parent** [src], page 9, (module)

**Location** core/src/string.lisp

**Internal Definitions**

- [delimiter], page 29, (method)
- [directive-body-position], page 26, (function)
- [format-string], page 29, (method)
- [format-string-error], page 31, (condition)
- [missing-delimiter], page 32, (condition)
- [missing-string-directive], page 32, (condition)
- [missing-table-directive], page 33, (condition)
- [modifier], page 29, (method)
- [next-directive-position], page 27, (function)
- [spurious-modifier], page 33, (condition)
- [spurious-parameter], page 34, (condition)
- [standard-directive], page 28, (function)
- [standard-directive-body], page 30, (generic function)
- [standard-directive-body], page 30, (method)
- [standard-directive-body], page 30, (method)
- [standard-directive-body], page 30, (method)
- [standard-format-string], page 28, (function)
- [string-position], page 30, (method)
- [table], page 30, (method)

#### 4.1.11 core/src/wrapper.lisp

**Dependency**

[string], page 13, (Lisp file)

**Parent** [src], page 9, (module)

**Location** core/src/wrapper.lisp

**Exported Definitions**

- [\*compile\*], page 21, (special variable)
- [format], page 22, (macro)
- [formatter], page 22, (macro)

#### 4.1.12 setup/setup.lisp

**Parent** [net.didierverna.focus.setup], page 6, (system)

**Location** setup/setup.lisp

**Packages** [net.didierverna.focus.setup], page 19,

**Exported Definitions**

- [\*release-major-level\*], page 21, (special variable)
- [\*release-minor-level\*], page 21, (special variable)
- [\*release-name\*], page 21, (special variable)
- [\*release-status\*], page 21, (special variable)
- [\*release-status-level\*], page 21, (special variable)
- [configuration], page 22, (function)

- [configure], page 22, (function)
- [setup-flv], page 23, (function)
- [version], page 23, (function)

**Internal Definitions**

- [%version], page 25, (function)
- [\*configuration\*], page 24, (special variable)
- [release-status-number], page 28, (function)
- [restrict-because], page 28, (function)



## 5 Packages

Packages are listed by definition order.

### 5.1 net.didierverna.focus

The FoCus package.

**Source**     [meta], page 11, (Lisp file)

#### Use List

- [net.didierverna.focus.setup], page 19,
- common-lisp

#### Exported Definitions

- [\*compile\*], page 21, (special variable)
- [\*format-table\*], page 21, (special variable)
- [format], page 22, (macro)
- [formatter], page 22, (macro)
- [in-format-table], page 22, (macro)
- [make-format-table], page 22, (function)
- [nickname-package], page 23, (function)
- [register-format-table], page 23, (function)
- [set-format-directive], page 23, (function)
- [unregister-format-table], page 23, (function)
- [with-format-table], page 22, (macro)

#### Internal Definitions

- [%make-format-table], page 25, (function)
- [\*readtable\*], page 24, (special variable)
- [\*standard-directives\*], page 24, (special variable)
- [clindent], page 25, (function)
- [copy-format-directive], page 25, (function)
- [copy-format-table], page 25, (function)
- [copy-function-directive], page 25, (function)
- [copy-standard-closing-directive], page 25, (function)
- [copy-standard-delimiting-directive], page 25, (function)
- [copy-standard-directive], page 25, (function)
- [copy-standard-opening-directive], page 25, (function)
- [defindent], page 24, (macro)
- [delimiter], page 29, (generic function)
- [delimiter], page 29, (method)
- [directive-body-position], page 26, (function)
- [directive-character], page 26, (function)
- [directive-closing-character], page 26, (function)
- [directive-function-name], page 26, (function)
- [(setf directive-function-name)], page 26, (function)

- [directive-opening-character], page 26, (function)
- [find-standard-directive], page 26, (function)
- [find-table], page 26, (function)
- [focus-error], page 31, (condition)
- [format-directive], page 35, (structure)
- [format-directive-p], page 26, (function)
- [format-string], page 29, (generic function)
- [format-string], page 29, (method)
- [format-string-error], page 31, (condition)
- [format-table], page 35, (structure)
- [format-table-error], page 31, (condition)
- [format-table-p], page 26, (function)
- [function-directive], page 36, (structure)
- [function-directive-p], page 26, (function)
- [i-reader], page 27, (function)
- [in-readtable], page 24, (macro)
- [lookup-table], page 27, (function)
- [make-format-directive], page 27, (function)
- [make-function-directive], page 27, (function)
- [make-standard-closing-directive], page 27, (function)
- [make-standard-delimiting-directive], page 27, (function)
- [make-standard-directive], page 27, (function)
- [make-standard-opening-directive], page 27, (function)
- [missing-delimiter], page 32, (condition)
- [missing-string-directive], page 32, (condition)
- [missing-table], page 32, (condition)
- [missing-table-directive], page 33, (condition)
- [modifier], page 29, (generic function)
- [modifier], page 29, (method)
- [name], page 29, (generic function)
- [name], page 29, (method)
- [next-directive-position], page 27, (function)
- [nonstandard-directive-character], page 29, (generic function)
- [nonstandard-directive-character], page 30, (method)
- [nonstandard-directive-character], page 33, (condition)
- [other-case], page 27, (function)
- [spurious-modifier], page 33, (condition)
- [spurious-parameter], page 34, (condition)
- [standard-closing-directive], page 36, (structure)
- [standard-closing-directive-p], page 28, (function)
- [standard-delimiting-directive], page 36, (structure)
- [standard-delimiting-directive-p], page 28, (function)
- [standard-directive], page 28, (function)

- [standard-directive], page 36, (structure)
- [standard-directive-body], page 30, (generic function)
- [standard-directive-body], page 30, (method)
- [standard-directive-body], page 30, (method)
- [standard-directive-body], page 30, (method)
- [standard-directive-lookup-error], page 34, (condition)
- [standard-directive-p], page 28, (function)
- [standard-format-string], page 28, (function)
- [standard-opening-directive], page 37, (structure)
- [standard-opening-directive-p], page 28, (function)
- [string-position], page 30, (generic function)
- [string-position], page 30, (method)
- [table], page 30, (generic function)
- [table], page 30, (method)
- [table], page 30, (method)
- [table-character], page 30, (generic function)
- [table-character], page 30, (method)
- [table-collision], page 34, (condition)
- [table-directive-collision], page 34, (condition)
- [table-directive-error], page 34, (condition)
- [table-lookup-error], page 35, (condition)
- [table-mappings], page 29, (function)
- [(setf table-mappings)], page 29, (function)
- [table-name], page 29, (function)
- [tilde-reader], page 29, (function)

## 5.2 net.didierverna.focus.user-functions

The FoCUS user functions package.

This package is used to store anonymous user functions (as provided to SET-FORMAT-DIRECTIVE) under gentemp'ed names.

**Source** [table], page 13, (Lisp file)

## 5.3 net.didierverna.focus.setup

Format Customizations, setup.

**Source** [setup], page 14, (Lisp file)

**Use List** common-lisp

**Used By List**

[net.didierverna.focus], page 17,

**Exported Definitions**

- [\*release-major-level\*], page 21, (special variable)
- [\*release-minor-level\*], page 21, (special variable)
- [\*release-name\*], page 21, (special variable)

- [`*release-status*`], page 21, (special variable)
- [`*release-status-level*`], page 21, (special variable)
- [`configuration`], page 22, (function)
- [`configure`], page 22, (function)
- [`setup-flv`], page 23, (function)
- [`version`], page 23, (function)

**Internal Definitions**

- [`%version`], page 25, (function)
- [`*configuration*`], page 24, (special variable)
- [`release-status-number`], page 28, (function)
- [`restrict-because`], page 28, (function)



## 6 Definitions

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

### 6.1 Exported definitions

#### 6.1.1 Special variables

- \*compile\*** [Special Variable]  
 Whether to compile to standard format calls.  
 If non-nil, the translation to standard format strings occurs at compile-time (hence, the current format table must also be known at compile-time). Otherwise (the default), the translation occurs at run-time.  
**Package** [net.didierverna.focus], page 17,  
**Source** [wrapper], page 14, (Lisp file)
- \*format-table\*** [Special Variable]  
 The current format table.  
**Package** [net.didierverna.focus], page 17,  
**Source** [table], page 13, (Lisp file)
- \*release-major-level\*** [Special Variable]  
 The major level of this release.  
**Package** [net.didierverna.focus.setup], page 19,  
**Source** [setup], page 14, (Lisp file)
- \*release-minor-level\*** [Special Variable]  
 The minor level of this release.  
**Package** [net.didierverna.focus.setup], page 19,  
**Source** [setup], page 14, (Lisp file)
- \*release-name\*** [Special Variable]  
 The name of this release.  
 The general naming theme for FoCus is "Akido movements".  
**Package** [net.didierverna.focus.setup], page 19,  
**Source** [setup], page 14, (Lisp file)
- \*release-status\*** [Special Variable]  
 The status of this release.  
**Package** [net.didierverna.focus.setup], page 19,  
**Source** [setup], page 14, (Lisp file)
- \*release-status-level\*** [Special Variable]  
 The status level of this release.  
**Package** [net.didierverna.focus.setup], page 19,  
**Source** [setup], page 14, (Lisp file)

### 6.1.2 Macros

`format` *DESTINATION FORMAT-CONTROL &rest ARGS* [Macro]

Wrapper around the standard `FORMAT` function.

When `FORMAT-CONTROL` is a string, it is interpreted according to the current format table. Such translation may occur at compile or run time, according to the value of `*COMPILE*`.

**Package** [net.didierverna.focus], page 17,

**Source** [wrapper], page 14, (Lisp file)

`formatter` *FORMAT-STRING* [Macro]

Wrapper around the standard `FORMATTER` macro.

`FORMAT-STRING` is interpreted according to the current (compile-time) format table.

**Package** [net.didierverna.focus], page 17,

**Source** [wrapper], page 14, (Lisp file)

`in-format-table` *TABLE-OR-NAME* [Macro]

Set the current format table to `TABLE-OR-NAME` in the current file.

This macro behaves like `IN-PACKAGE` or `IN-READTABLE` and also sets `*COMPILE*` to `t`, so as to enable compile-time behavior in FoCus.

**Package** [net.didierverna.focus], page 17,

**Source** [flv], page 11, (Lisp file)

`with-format-table` *TABLE-OR-NAME &body BODY* [Macro]

Execute `BODY` with the current format table bound to `TABLE-OR-NAME`.

**Package** [net.didierverna.focus], page 17,

**Source** [table], page 13, (Lisp file)

### 6.1.3 Functions

`configuration` *KEY* [Function]

Return `KEY`'s value in the current FoCus configuration.

**Package** [net.didierverna.focus.setup], page 19,

**Source** [setup], page 14, (Lisp file)

`configure` *KEY VALUE* [Function]

Set `KEY` to `VALUE` in the current FoCus configuration.

**Package** [net.didierverna.focus.setup], page 19,

**Source** [setup], page 14, (Lisp file)

`make-format-table` *&optional INITIALLY &aux TABLE MAPPINGS* [Function]

Create and return a new format table.

The table may be `INITIALLY` `:standard` (the default), `:standard-upcase`, `:standard-downcase` or `:blank`.

**Package** [net.didierverna.focus], page 17,

**Source** [table], page 13, (Lisp file)

**nickname-package** *&optional NICKNAME* [Function]  
 Add NICKNAME (:FOCUS by default) to the :NET.DIDIERVERNA.FOCUS package.

**Package** [net.didierverna.focus], page 17,

**Source** [meta], page 11, (Lisp file)

**register-format-table** *TABLE NAME &optional FORCE* [Function]

Register TABLE under NAME (a symbol) and return it.

FORCE means overwrite an already existing registration under that name. Otherwise (the default), throw a TABLE-COLLISION error.

**Package** [net.didierverna.focus], page 17,

**Source** [table], page 13, (Lisp file)

**set-format-directive** *CHAR &key STANDARD (FUNCTION FUNCTION-OR-NAME) BOTH-CASE FORCE (TABLE TABLE-OR-NAME) &aux TABLE* [Function]

Set a ~CHAR directive in TABLE.

- TABLE (the current format table by default) may be a table or a table name. - When BOTH-CASE (the default), operate on both case versions of CHAR.

- Attempting to override an existing directive throws a TABLE-DIRECTIVE-COLLISION error, unless FORCE is non-nil.

The operation to perform is as follows:

- If FUNCTION is provided (either a function or a function name), associate CHAR with it.

- If STANDARD is provided, associate CHAR with the standard directive denoted by STANDARD character (case does not matter).

- Otherwise, remove the ~CHAR directive from TABLE.

**Package** [net.didierverna.focus], page 17,

**Source** [table], page 13, (Lisp file)

**setup-flv** [Function]

Autodetect support for file-local variables.

Update FoCus configuration and \*FEATURES\* accordingly.

**Package** [net.didierverna.focus.setup], page 19,

**Source** [setup], page 14, (Lisp file)

**unregister-format-table** *NAME* [Function]

Unregister NAMED table.

**Package** [net.didierverna.focus], page 17,

**Source** [table], page 13, (Lisp file)

**version** *&optional TYPE* [Function]

Return the current version of FoCus.

TYPE can be one of :number, :short or :long.

A version number is computed as major\*10000 + minor\*100 + patchlevel, leaving two digits for each level. Alpha, beta and rc status are ignored in version numbers.

A short version is something like 1.3{a,b,rc}4, or 1.3.4 for patchlevel. Alpha, beta or rc levels start at 1. Patchlevels start at 0 but are ignored

in the output, so that 1.3.0 appears as just 1.3.

A long version is something like  
 1.3 {alpha,beta,release candidate,patchlevel} 4 "Kokyu Ho". As for  
 the short version, a patchlevel of 0 is ignored in the output.

**Package** [net.didierverna.focus.setup], page 19,

**Source** [setup], page 14, (Lisp file)

## 6.2 Internal definitions

### 6.2.1 Special variables

**\*configuration\*** [Special Variable]

The FoCus configuration settings.  
 This variable contains a property list of configuration options.  
 Current options are:  
 - :swank-eval-in-emacs (Boolean)

See section A.1 of the user manual for more information.

**Package** [net.didierverna.focus.setup], page 19,

**Source** [setup], page 14, (Lisp file)

**\*readtable\*** [Special Variable]

The FoCus readtable.

**Package** [net.didierverna.focus], page 17,

**Source** [meta], page 11, (Lisp file)

**\*standard-directives\*** [Special Variable]

The list of standard format directives.

**Package** [net.didierverna.focus], page 17,

**Source** [directive], page 12, (Lisp file)

### 6.2.2 Macros

**defindent** *SYMBOL INDENT* [Macro]

Set *SYMBOL*'s indentation to *INDENT* in (X)Emacs.  
*SYMBOL* and *INDENT* need not be quoted.  
 See *CLINDENT* for more information.

**Package** [net.didierverna.focus], page 17,

**Source** [meta], page 11, (Lisp file)

**in-readtable** *NAME* [Macro]

Set the current readtable to the value of *NAME*::\*READTABLE\*.

**Package** [net.didierverna.focus], page 17,

**Source** [meta], page 11, (Lisp file)

### 6.2.3 Functions

<code>%make-format-table</code> <b>&amp;key</b> ( <i>MAPPINGS MAPPINGS</i> )	[Function]
<b>Package</b> [net.didierverna.focus], page 17,	
<b>Source</b> [table], page 13, (Lisp file)	
<code>%version</code> <i>TYPE MAJOR MINOR STATUS LEVEL NAME</i>	[Function]
<b>Package</b> [net.didierverna.focus.setup], page 19,	
<b>Source</b> [setup], page 14, (Lisp file)	
<code>clindent</code> <i>SYMBOL INDENT</i>	[Function]
Set <i>SYMBOL</i> 's indentation to <i>INDENT</i> in (X)Emacs. This function sets <i>SYMBOL</i> 's common-lisp-indent-function property. If <i>INDENT</i> is a symbol, use its indentation definition. Otherwise, <i>INDENT</i> is considered as an indentation definition.	
<b>Package</b> [net.didierverna.focus], page 17,	
<b>Source</b> [meta], page 11, (Lisp file)	
<code>copy-format-directive</code> <i>INSTANCE</i>	[Function]
<b>Package</b> [net.didierverna.focus], page 17,	
<b>Source</b> [directive], page 12, (Lisp file)	
<code>copy-format-table</code> <i>INSTANCE</i>	[Function]
<b>Package</b> [net.didierverna.focus], page 17,	
<b>Source</b> [table], page 13, (Lisp file)	
<code>copy-function-directive</code> <i>INSTANCE</i>	[Function]
<b>Package</b> [net.didierverna.focus], page 17,	
<b>Source</b> [directive], page 12, (Lisp file)	
<code>copy-standard-closing-directive</code> <i>INSTANCE</i>	[Function]
<b>Package</b> [net.didierverna.focus], page 17,	
<b>Source</b> [directive], page 12, (Lisp file)	
<code>copy-standard-delimiting-directive</code> <i>INSTANCE</i>	[Function]
<b>Package</b> [net.didierverna.focus], page 17,	
<b>Source</b> [directive], page 12, (Lisp file)	
<code>copy-standard-directive</code> <i>INSTANCE</i>	[Function]
<b>Package</b> [net.didierverna.focus], page 17,	
<b>Source</b> [directive], page 12, (Lisp file)	
<code>copy-standard-opening-directive</code> <i>INSTANCE</i>	[Function]
<b>Package</b> [net.didierverna.focus], page 17,	
<b>Source</b> [directive], page 12, (Lisp file)	

- directive-body-position** *STRING START* [Function]  
 Return a *STRING*'s directive body position from *START*.  
*START* is the position of the tilde character, so this function essentially parses the directive arguments in order to skip them.  
**Package** [net.didierverna.focus], page 17,  
**Source** [string], page 13, (Lisp file)
- directive-character** *INSTANCE* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- directive-closing-character** *INSTANCE* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- directive-function-name** *INSTANCE* [Function]  
 (setf directive-function-name) *VALUE INSTANCE* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- directive-opening-character** *INSTANCE* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- find-standard-directive** *CHARACTER &aux CHARACTER* [Function]  
*DIRECTIVE*  
 Return the standard directive corresponding to *CHARACTER*.  
 Throw a *NONSTANDARD-DIRECTIVE-CHARACTER* error otherwise.  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- find-table** *TABLE-OR-NAME* [Function]  
 Return its table argument directly, or look it up by name.  
**Package** [net.didierverna.focus], page 17,  
**Source** [table], page 13, (Lisp file)
- format-directive-p** *OBJECT* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- format-table-p** *OBJECT* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [table], page 13, (Lisp file)
- function-directive-p** *OBJECT* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)

- i-reader** *STREAM SUBCHAR ARG* [Function]  
 Read an argument list for the `DEFINDENT` macro.
- Package** [net.didierverna.focus], page 17,  
**Source** [meta], page 11, (Lisp file)
- lookup-table** *NAME &optional ERRORP &aux TABLE* [Function]  
 Look for a format table registered under *NAME*.  
*ERRORP* (the default) means to throw a `TABLE-NOT-REGISTERED` error if no such table is found. Otherwise, just return `nil`.
- Package** [net.didierverna.focus], page 17,  
**Source** [table], page 13, (Lisp file)
- make-format-directive** *&key* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- make-function-directive** *&key (FUNCTION-NAME FUNCTION-NAME)* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- make-standard-closing-directive** *&key (CHARACTER CHARACTER) (OPENING-CHARACTER OPENING-CHARACTER)* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- make-standard-delimiting-directive** *&key (CHARACTER CHARACTER)* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- make-standard-directive** *&key (CHARACTER CHARACTER)* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- make-standard-opening-directive** *&key (CHARACTER CHARACTER) (CLOSING-CHARACTER CLOSING-CHARACTER)* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- next-directive-position** *STRING START* [Function]  
 Return the next directive position in *STRING* from *START*, or `nil`.
- Package** [net.didierverna.focus], page 17,  
**Source** [string], page 13, (Lisp file)
- other-case** *CHAR* [Function]  
 Return the other case version of *CHAR* if it exists.
- Package** [net.didierverna.focus], page 17,  
**Source** [util], page 12, (Lisp file)

- release-status-number** *RELEASE-STATUS* [Function]  
**Package** [net.didierverna.focus.setup], page 19,  
**Source** [setup], page 14, (Lisp file)
- restrict-because** *REASON* [Function]  
 Put FoCus in restricted mode because of REASON.  
**Package** [net.didierverna.focus.setup], page 19,  
**Source** [setup], page 14, (Lisp file)
- standard-closing-directive-p** *OBJECT* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- standard-delimiting-directive-p** *OBJECT* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- standard-directive** *STRING POSITION TABLE &aux  
 BODY-POSITION DIRECTIVE* [Function]  
 Translate a STRING directive into a standard one.  
 - POSITION is the position of the ~ character.  
 - The translation is done according to format TABLE.
- Return two values:  
 - the translated directive as a string,  
 - the STRING's remainder index.
- Note that the directive arguments are copied as-is. Only the directive's body actually involves a translation.
- Package** [net.didierverna.focus], page 17,  
**Source** [string], page 13, (Lisp file)
- standard-directive-p** *OBJECT* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- standard-format-string** *FORMAT-CONTROL &optional TABLE* [Function]  
 Return the translation of FORMAT-CONTROL into a standard one.  
 When FORMAT-CONTROL is a string, the translation is done according to format TABLE (the current table by default). Otherwise, FORMAT-CONTROL is returned as-is.
- Package** [net.didierverna.focus], page 17,  
**Source** [string], page 13, (Lisp file)
- standard-opening-directive-p** *OBJECT* [Function]  
**Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)



<code>table-mappings</code>	<i>INSTANCE</i>	[Function]
<code>(setf table-mappings)</code>	<i>VALUE INSTANCE</i>	[Function]
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,	
<b>Source</b>	[ <code>table</code> ], page 13, (Lisp file)	
<code>table-name</code>	<i>TABLE</i>	[Function]
	Find <i>TABLE</i> 's name if registered, return nil otherwise.	
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,	
<b>Source</b>	[ <code>table</code> ], page 13, (Lisp file)	
<code>tilde-reader</code>	<i>STREAM CHAR</i>	[Function]
	Read a series of ~"string" to be concatenated together.	
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,	
<b>Source</b>	[ <code>meta</code> ], page 11, (Lisp file)	

### 6.2.4 Generic functions

<code>delimiter</code>	<i>CONDITION</i>	[Generic Function]
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,	
<b>Methods</b>		
	<code>delimiter</code> ( <i>CONDITION</i> <code>missing-delimiter</code> )	[Method]
	<b>Source</b> [string], page 13, (Lisp file)	
<code>format-string</code>	<i>CONDITION</i>	[Generic Function]
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,	
<b>Methods</b>		
	<code>format-string</code> ( <i>CONDITION</i> <code>format-string-error</code> )	[Method]
	<b>Source</b> [string], page 13, (Lisp file)	
<code>modifier</code>	<i>CONDITION</i>	[Generic Function]
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,	
<b>Methods</b>		
	<code>modifier</code> ( <i>CONDITION</i> <code>spurious-modifier</code> )	[Method]
	<b>Source</b> [string], page 13, (Lisp file)	
<code>name</code>	<i>CONDITION</i>	[Generic Function]
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,	
<b>Methods</b>		
	<code>name</code> ( <i>CONDITION</i> <code>table-lookup-error</code> )	[Method]
	<b>Source</b> [ <code>table</code> ], page 13, (Lisp file)	
<code>nonstandard-directive-character</code>	<i>CONDITION</i>	[Generic Function]
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,	
<b>Methods</b>		

- nonstandard-directive-character** (*CONDITION* nonstandard-directive-character) [Method]  
**Source** [directive], page 12, (Lisp file)
- standard-directive-body** *STRING POSITION DIRECTIVE* [Generic Function]  
 Translate a *STRING* directive's body into a standard one.  
 - *POSITION* is the position of the directive's body (i.e., past the ~ character and the potential arguments).  
 - The translation is done according to *DIRECTIVE*.
- Return two values:  
 - the translated directive's body as a string,  
 - the *STRING*'s remainder index.
- Package** [net.didierverna.focus], page 17,  
**Source** [string], page 13, (Lisp file)  
**Methods**
- standard-directive-body** *STRING POSITION* (DIRECTIVE standard-directive) [Method]  
 Method for one-character and grouping standard directives.
- standard-directive-body** *STRING POSITION* (DIRECTIVE standard-delimiting-directive) [Method]  
 Method for the / standard directive.
- standard-directive-body** *STRING POSITION* (DIRECTIVE function-directive) [Method]  
 Method for function directives.
- string-position** *CONDITION* [Generic Function]  
**Package** [net.didierverna.focus], page 17,  
**Methods**
- string-position** (*CONDITION* format-string-error) [Method]  
**Source** [string], page 13, (Lisp file)
- table** *CONDITION* [Generic Function]  
**Package** [net.didierverna.focus], page 17,  
**Methods**
- table** (*CONDITION* missing-table-directive) [Method]  
**Source** [string], page 13, (Lisp file)
- table** (*CONDITION* format-table-error) [Method]  
**Source** [table], page 13, (Lisp file)
- table-character** *CONDITION* [Generic Function]  
**Package** [net.didierverna.focus], page 17,  
**Methods**
- table-character** (*CONDITION* table-directive-error) [Method]  
**Source** [table], page 13, (Lisp file)

## 6.2.5 Conditions

**focus-error** [Condition]

A FoCuS error.

**Package** [net.didierverna.focus], page 17,

**Source** [util], page 12, (Lisp file)

**Direct superclasses**

error

**Direct subclasses**

- [standard-directive-lookup-error], page 34, (condition)
- [format-table-error], page 31, (condition)
- [table-lookup-error], page 35, (condition)
- [format-string-error], page 31, (condition)

**format-string-error** [Condition]

A format string error.

**Package** [net.didierverna.focus], page 17,

**Source** [string], page 13, (Lisp file)

**Direct superclasses**

[focus-error], page 31, (condition)

**Direct subclasses**

- [missing-string-directive], page 32, (condition)
- [spurious-parameter], page 34, (condition)
- [spurious-modifier], page 33, (condition)
- [missing-delimiter], page 32, (condition)
- [missing-table-directive], page 33, (condition)

**Direct methods**

- [string-position], page 30, (method)
- [format-string], page 29, (method)

**Direct slots**

**string** [Slot]

The format string.

**Initargs** :string

**Readers** [format-string], page 29, (generic function)

**position** [Slot]

The position at which the error occurs.

**Initargs** :position

**Initform** 'nil

**Readers** [string-position], page 30, (generic function)

**format-table-error** [Condition]

A format table error.

**Package** [net.didierverna.focus], page 17,

- Source** [table], page 13, (Lisp file)
- Direct superclasses**  
[focus-error], page 31, (condition)
- Direct subclasses**  
[table-directive-error], page 34, (condition)
- Direct methods**  
[table], page 30, (method)
- Direct slots**
- |   |        |
|---|--------|
| <b>table</b>  | [Slot] |
| The format table.                                   |        |
| <b>Initargs</b> :table                              |        |
| <b>Readers</b> [table], page 30, (generic function) |        |
- missing-delimiter** [Condition]
- A missing delimiter error.
- Package** [net.didierverna.focus], page 17,
- Source** [string], page 13, (Lisp file)
- Direct superclasses**  
[format-string-error], page 31, (condition)
- Direct methods**  
[delimiter], page 29, (method)
- Direct slots**
- |   |        |
|---|--------|
| <b>delimiter</b>  | [Slot] |
| The delimiter character.                                |        |
| <b>Initargs</b> :delimiter                              |        |
| <b>Readers</b> [delimiter], page 29, (generic function) |        |
- missing-string-directive** [Condition]
- A missing directive error.
- Package** [net.didierverna.focus], page 17,
- Source** [string], page 13, (Lisp file)
- Direct superclasses**  
[format-string-error], page 31, (condition)
- missing-table** [Condition]
- A missing format table error.
- Package** [net.didierverna.focus], page 17,
- Source** [table], page 13, (Lisp file)
- Direct superclasses**  
[table-lookup-error], page 35, (condition)

- missing-table-directive** [Condition]  
 A missing table directive error.
- Package** [net.didierverna.focus], page 17,  
**Source** [string], page 13, (Lisp file)
- Direct superclasses**  
 [format-string-error], page 31, (condition)
- Direct methods**  
 [table], page 30, (method)
- Direct slots**
- table** [Slot]  
 The format table.
- Initargs** :table  
**Readers** [table], page 30, (generic function)
- nonstandard-directive-character** [Condition]  
 A nonstandard directive character error.
- Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- Direct superclasses**  
 [standard-directive-lookup-error], page 34, (condition)
- Direct methods**  
 [nonstandard-directive-character], page 30, (method)
- Direct slots**
- character** [Slot]  
 The character.
- Initargs** :character  
**Readers** [nonstandard-directive-character], page 29, (generic function)
- spurious-modifier** [Condition]  
 A spurious modifier error.
- Package** [net.didierverna.focus], page 17,  
**Source** [string], page 13, (Lisp file)
- Direct superclasses**  
 [format-string-error], page 31, (condition)
- Direct methods**  
 [modifier], page 29, (method)
- Direct slots**
- modifier** [Slot]  
 The modifier character.
- Initargs** :modifier  
**Readers** [modifier], page 29, (generic function)

- spurious-parameter** [Condition]  
 A spurious parameter error.
- Package** [net.didierverna.focus], page 17,  
**Source** [string], page 13, (Lisp file)
- Direct superclasses**  
 [format-string-error], page 31, (condition)
- standard-directive-lookup-error** [Condition]  
 A standard directive lookup error.
- Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- Direct superclasses**  
 [focus-error], page 31, (condition)
- Direct subclasses**  
 [nonstandard-directive-character], page 33, (condition)
- table-collision** [Condition]  
 A table collision error.
- Package** [net.didierverna.focus], page 17,  
**Source** [table], page 13, (Lisp file)
- Direct superclasses**  
 [table-lookup-error], page 35, (condition)
- table-directive-collision** [Condition]  
 A table directive collision error.
- Package** [net.didierverna.focus], page 17,  
**Source** [table], page 13, (Lisp file)
- Direct superclasses**  
 [table-directive-error], page 34, (condition)
- table-directive-error** [Condition]  
 A table directive error.
- Package** [net.didierverna.focus], page 17,  
**Source** [table], page 13, (Lisp file)
- Direct superclasses**  
 [format-table-error], page 31, (condition)
- Direct subclasses**  
 [table-directive-collision], page 34, (condition)
- Direct methods**  
 [table-character], page 30, (method)
- Direct slots**
- character** [Slot]  
 The directive character.
- Initargs** :character  
**Readers** [table-character], page 30, (generic function)

<code>table-lookup-error</code>	[Condition]
A format table lookup error.	
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,
<b>Source</b>	[ <code>table</code> ], page 13, (Lisp file)
<b>Direct superclasses</b>	[ <code>focus-error</code> ], page 31, (condition)
<b>Direct subclasses</b>	<ul style="list-style-type: none"> <li>• [<code>missing-table</code>], page 32, (condition)</li> <li>• [<code>table-collision</code>], page 34, (condition)</li> </ul>
<b>Direct methods</b>	[ <code>name</code> ], page 29, (method)
<b>Direct slots</b>	
<b>name</b>	[Slot]
The looked up name.	
<b>Initargs</b>	: <code>name</code>
<b>Readers</b>	[ <code>name</code> ], page 29, (generic function)

### 6.2.6 Structures

<code>format-directive</code>	[Structure]
Base structure for format directives.	
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,
<b>Source</b>	[ <code>directive</code> ], page 12, (Lisp file)
<b>Direct superclasses</b>	<code>structure-object</code>
<b>Direct subclasses</b>	<ul style="list-style-type: none"> <li>• [<code>standard-directive</code>], page 36, (structure)</li> <li>• [<code>function-directive</code>], page 36, (structure)</li> </ul>
<code>format-table</code>	[Structure]
Structure for format tables.	
This structure holds the MAPPINGS from characters to directives.	
<b>Package</b>	[ <code>net.didierverna.focus</code> ], page 17,
<b>Source</b>	[ <code>table</code> ], page 13, (Lisp file)
<b>Direct superclasses</b>	<code>structure-object</code>
<b>Direct methods</b>	<code>print-object</code>
<b>Direct slots</b>	
<b>mappings</b>	[Slot]
<b>Type</b>	<code>hash-table</code>
<b>Initform</b>	( <code>make-hash-table</code> )
<b>Readers</b>	[ <code>table-mappings</code> ], page 29, (function)
<b>Writers</b>	[ <code>(setf table-mappings)</code> ], page 29, (function)

- function-directive** [Structure]  
 Structure for user-defined function directives.  
 This structure holds the associated function name.
- Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- Direct superclasses**  
 [format-directive], page 35, (structure)
- Direct methods**  
 [standard-directive-body], page 30, (method)
- Direct slots**
- function-name** [Slot]
- Type** symbol
- Readers** [directive-function-name], page 26, (function)
- Writers** [(setf directive-function-name)], page 26, (function)
- standard-closing-directive** [Structure]  
 Structure for standard closing directives.  
 This structure holds the corresponding (upcase) opening character.
- Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- Direct superclasses**  
 [standard-directive], page 36, (structure)
- Direct slots**
- opening-character** [Slot]
- Type** character
- Readers** [directive-opening-character], page 26, (function)
- Writers** (setf directive-opening-character)
- standard-delimiting-directive** [Structure]  
 Structure for standard delimiting directives.
- Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)
- Direct superclasses**  
 [standard-directive], page 36, (structure)
- Direct methods**  
 [standard-directive-body], page 30, (method)
- standard-directive** [Structure]  
 Base structure for standard directives.  
 This structure holds the (upcase) directive character.
- Package** [net.didierverna.focus], page 17,  
**Source** [directive], page 12, (Lisp file)



**Direct superclasses**

[format-directive], page 35, (structure)

**Direct subclasses**

- [standard-delimiting-directive], page 36, (structure)
- [standard-opening-directive], page 37, (structure)
- [standard-closing-directive], page 36, (structure)

**Direct methods**

[standard-directive-body], page 30, (method)

**Direct slots**

character [Slot]

**Type** character

**Readers** [directive-character], page 26, (function)

**Writers** (setf directive-character)

standard-opening-directive [Structure]

Structure for standard opening directives.

This structure holds the corresponding (upcase) closing character.

**Package** [net.didierverna.focus], page 17,

**Source** [directive], page 12, (Lisp file)

**Direct superclasses**

[standard-directive], page 36, (structure)

**Direct slots**

closing-character [Slot]

**Type** character

**Readers** [directive-closing-character], page 26, (function)

**Writers** (setf directive-closing-character)



## Appendix A Indexes

### A.1 Concepts

#### C

core/meta.lisp.....	11
core/src/.....	9
core/src/directive.lisp.....	12
core/src/string.lisp.....	13
core/src/table.lisp.....	13
core/src/util.lisp.....	12
core/src/wrapper.lisp.....	14

#### F

File, Lisp, core/meta.lisp.....	11
File, Lisp, core/src/directive.lisp.....	12
File, Lisp, core/src/string.lisp.....	13
File, Lisp, core/src/table.lisp.....	13
File, Lisp, core/src/util.lisp.....	12
File, Lisp, core/src/wrapper.lisp.....	14
File, Lisp, flv/flv.lisp.....	11
File, Lisp, net.didierverna.focus.asd.....	11
File, Lisp, net.didierverna.focus.core.asd.....	11
File, Lisp, net.didierverna.focus.flv.asd.....	11
File, Lisp, net.didierverna.focus.setup.asd.....	11
File, Lisp, setup/setup.lisp.....	14
flv/flv.lisp.....	11

#### L

Lisp File, core/meta.lisp.....	11
Lisp File, core/src/directive.lisp.....	12
Lisp File, core/src/string.lisp.....	13
Lisp File, core/src/table.lisp.....	13
Lisp File, core/src/util.lisp.....	12
Lisp File, core/src/wrapper.lisp.....	14
Lisp File, flv/flv.lisp.....	11
Lisp File, net.didierverna.focus.asd.....	11
Lisp File, net.didierverna.focus.core.asd.....	11
Lisp File, net.didierverna.focus.flv.asd.....	11
Lisp File, net.didierverna.focus.setup.asd.....	11
Lisp File, setup/setup.lisp.....	14

#### M

Module, core/src/.....	9
------------------------	---

#### N

net.didierverna.focus.asd.....	11
net.didierverna.focus.core.asd.....	11
net.didierverna.focus.flv.asd.....	11
net.didierverna.focus.setup.asd.....	11

#### S

setup/setup.lisp.....	14
-----------------------	----

## A.2 Functions

### %

%make-format-table .....	25
%version .....	25

### (

(setf directive-function-name) .....	26
(setf table-mappings) .....	29

### C

clindent .....	25
configuration .....	22
configure .....	22
copy-format-directive .....	25
copy-format-table .....	25
copy-function-directive .....	25
copy-standard-closing-directive .....	25
copy-standard-delimiting-directive .....	25
copy-standard-directive .....	25
copy-standard-opening-directive .....	25

### D

defindent .....	24
delimiter .....	29
directive-body-position .....	26
directive-character .....	26
directive-closing-character .....	26
directive-function-name .....	26
directive-opening-character .....	26

### F

find-standard-directive .....	26
find-table .....	26
format .....	22
format-directive-p .....	26
format-string .....	29
format-table-p .....	26
formatter .....	22
Function, %make-format-table .....	25
Function, %version .....	25
Function, (setf directive-function-name) .....	26
Function, (setf table-mappings) .....	29
Function, clindent .....	25
Function, configuration .....	22
Function, configure .....	22
Function, copy-format-directive .....	25
Function, copy-format-table .....	25
Function, copy-function-directive .....	25
Function, copy-standard-closing-directive .....	25
Function,	
copy-standard-delimiting-directive .....	25
Function, copy-standard-directive .....	25
Function, copy-standard-opening-directive .....	25
Function, directive-body-position .....	26
Function, directive-character .....	26
Function, directive-closing-character .....	26
Function, directive-function-name .....	26
Function, directive-opening-character .....	26

Function, find-standard-directive .....	26
Function, find-table .....	26
Function, format-directive-p .....	26
Function, format-table-p .....	26
Function, function-directive-p .....	26
Function, i-reader .....	27
Function, lookup-table .....	27
Function, make-format-directive .....	27
Function, make-format-table .....	22
Function, make-function-directive .....	27
Function, make-standard-closing-directive .....	27
Function,	
make-standard-delimiting-directive .....	27
Function, make-standard-directive .....	27
Function, make-standard-opening-directive .....	27
Function, next-directive-position .....	27
Function, nickname-package .....	23
Function, other-case .....	27
Function, register-format-table .....	23
Function, release-status-number .....	28
Function, restrict-because .....	28
Function, set-format-directive .....	23
Function, setup-flv .....	23
Function, standard-closing-directive-p .....	28
Function, standard-delimiting-directive-p .....	28
Function, standard-directive .....	28
Function, standard-directive-p .....	28
Function, standard-format-string .....	28
Function, standard-opening-directive-p .....	28
Function, table-mappings .....	29
Function, table-name .....	29
Function, tilde-reader .....	29
Function, unregister-format-table .....	23
Function, version .....	23
function-directive-p .....	26

### G

Generic Function, delimiter .....	29
Generic Function, format-string .....	29
Generic Function, modifier .....	29
Generic Function, name .....	29
Generic Function,	
nonstandard-directive-character .....	29
Generic Function, standard-directive-body .....	30
Generic Function, string-position .....	30
Generic Function, table .....	30
Generic Function, table-character .....	30

### I

i-reader .....	27
in-format-table .....	22
in-readtable .....	24

### L

lookup-table .....	27
--------------------	----

**M**

Macro, defindent .....	24
Macro, format .....	22
Macro, formatter .....	22
Macro, in-format-table .....	22
Macro, in-readtable .....	24
Macro, with-format-table .....	22
make-format-directive .....	27
make-format-table .....	22
make-function-directive .....	27
make-standard-closing-directive .....	27
make-standard-delimiting-directive .....	27
make-standard-directive .....	27
make-standard-opening-directive .....	27
Method, delimiter .....	29
Method, format-string .....	29
Method, modifier .....	29
Method, name .....	29
Method, nonstandard-directive-character .....	30
Method, standard-directive-body .....	30
Method, string-position .....	30
Method, table .....	30
Method, table-character .....	30
modifier .....	29

**N**

name .....	29
next-directive-position .....	27
nickname-package .....	23
nonstandard-directive-character .....	29, 30

**O**

other-case .....	27
------------------	----

**R**

register-format-table .....	23
release-status-number .....	28
restrict-because .....	28

**S**

set-format-directive .....	23
setup-flv .....	23
standard-closing-directive-p .....	28
standard-delimiting-directive-p .....	28
standard-directive .....	28
standard-directive-body .....	30
standard-directive-p .....	28
standard-format-string .....	28
standard-opening-directive-p .....	28
string-position .....	30

**T**

table .....	30
table-character .....	30
table-mappings .....	29
table-name .....	29
tilde-reader .....	29

**U**

unregister-format-table .....	23
-------------------------------	----

**V**

version .....	23
---------------	----

**W**

with-format-table .....	22
-------------------------	----

## A.3 Variables

### \*

<code>*compile*</code> .....	21
<code>*configuration*</code> .....	24
<code>*format-table*</code> .....	21
<code>*readtable*</code> .....	24
<code>*release-major-level*</code> .....	21
<code>*release-minor-level*</code> .....	21
<code>*release-name*</code> .....	21
<code>*release-status*</code> .....	21
<code>*release-status-level*</code> .....	21
<code>*standard-directives*</code> .....	24

### C

character .....	33, 34, 37
closing-character .....	37

### D

delimiter.....	32
----------------	----

### F

function-name .....	36
---------------------	----

### M

mappings.....	35
modifier.....	33

### N

name .....	35
------------	----

### O

opening-character .....	36
-------------------------	----

### P

position.....	31
---------------	----

### S

Slot, character .....	33, 34, 37
Slot, closing-character .....	37
Slot, delimiter .....	32
Slot, function-name .....	36
Slot, mappings .....	35
Slot, modifier .....	33
Slot, name .....	35
Slot, opening-character .....	36
Slot, position .....	31
Slot, string.....	31
Slot, table .....	32, 33
Special Variable, <code>*compile*</code> .....	21
Special Variable, <code>*configuration*</code> .....	24
Special Variable, <code>*format-table*</code> .....	21
Special Variable, <code>*readtable*</code> .....	24
Special Variable, <code>*release-major-level*</code> .....	21
Special Variable, <code>*release-minor-level*</code> .....	21
Special Variable, <code>*release-name*</code> .....	21
Special Variable, <code>*release-status*</code> .....	21
Special Variable, <code>*release-status-level*</code> .....	21
Special Variable, <code>*standard-directives*</code> .....	24
string .....	31

### T

table.....	32, 33
------------	--------

## A.4 Data types

### C

Condition, <code>focus-error</code> .....	31
Condition, <code>format-string-error</code> .....	31
Condition, <code>format-table-error</code> .....	31
Condition, <code>missing-delimiter</code> .....	32
Condition, <code>missing-string-directive</code> .....	32
Condition, <code>missing-table</code> .....	32
Condition, <code>missing-table-directive</code> .....	33
Condition, <code>nonstandard-directive-character</code> ...	33
Condition, <code>spurious-modifier</code> .....	33
Condition, <code>spurious-parameter</code> .....	34
Condition, <code>standard-directive-lookup-error</code> ...	34
Condition, <code>table-collision</code> .....	34
Condition, <code>table-directive-collision</code> .....	34
Condition, <code>table-directive-error</code> .....	34
Condition, <code>table-lookup-error</code> .....	35

### F

<code>focus-error</code> .....	31
<code>format-directive</code> .....	35
<code>format-string-error</code> .....	31
<code>format-table</code> .....	35
<code>format-table-error</code> .....	31
<code>function-directive</code> .....	36

### M

<code>missing-delimiter</code> .....	32
<code>missing-string-directive</code> .....	32
<code>missing-table</code> .....	32
<code>missing-table-directive</code> .....	33

### N

<code>net.didierverna.focus</code> .....	5, 17
<code>net.didierverna.focus.core</code> .....	6
<code>net.didierverna.focus.flv</code> .....	5
<code>net.didierverna.focus.setup</code> .....	6, 19
<code>net.didierverna.focus.user-functions</code> .....	19
<code>nonstandard-directive-character</code> .....	33

### P

Package, <code>net.didierverna.focus</code> .....	17
Package, <code>net.didierverna.focus.setup</code> .....	19
Package,	
<code>net.didierverna.focus.user-functions</code> .....	19

### S

<code>spurious-modifier</code> .....	33
<code>spurious-parameter</code> .....	34
<code>standard-closing-directive</code> .....	36
<code>standard-delimiting-directive</code> .....	36
<code>standard-directive</code> .....	36
<code>standard-directive-lookup-error</code> .....	34
<code>standard-opening-directive</code> .....	37
Structure, <code>format-directive</code> .....	35
Structure, <code>format-table</code> .....	35
Structure, <code>function-directive</code> .....	36
Structure, <code>standard-closing-directive</code> .....	36
Structure, <code>standard-delimiting-directive</code> .....	36
Structure, <code>standard-directive</code> .....	36
Structure, <code>standard-opening-directive</code> .....	37
System, <code>net.didierverna.focus</code> .....	5
System, <code>net.didierverna.focus.core</code> .....	6
System, <code>net.didierverna.focus.flv</code> .....	5
System, <code>net.didierverna.focus.setup</code> .....	6

### T

<code>table-collision</code> .....	34
<code>table-directive-collision</code> .....	34
<code>table-directive-error</code> .....	34
<code>table-lookup-error</code> .....	35

