

The TFM Reference Manual

TeX Font Metrics, version 1.1 patchlevel 1 "Carolingian Miniscules"

Didier Verna <didier@didierverna.net>

This manual was generated automatically by Declt 3.0 "Montgomery Scott" on Mon Mar 29 09:54:34 2021 GMT+1.

Copyright © 2018-2021 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 |
| 1 Introduction | 3 |
| 2 Systems | 5 |
| 2.1 net.didierverna.tfm | 5 |
| 2.2 net.didierverna.tfm.core | 5 |
| 2.3 net.didierverna.tfm.setup | 6 |
| 3 Modules | 7 |
| 3.1 net.didierverna.tfm.core/src | 7 |
| 3.2 net.didierverna.tfm.setup/src | 7 |
| 4 Files | 9 |
| 4.1 Lisp | 9 |
| 4.1.1 net.didierverna.tfm.asd | 9 |
| 4.1.2 net.didierverna.tfm.core.asd | 9 |
| 4.1.3 net.didierverna.tfm.setup.asd | 9 |
| 4.1.4 net.didierverna.tfm.core/package.lisp | 9 |
| 4.1.5 net.didierverna.tfm.core/src/util.lisp | 9 |
| 4.1.6 net.didierverna.tfm.core/src/intermediate.lisp | 10 |
| 4.1.7 net.didierverna.tfm.core/src/character.lisp | 11 |
| 4.1.8 net.didierverna.tfm.core/src/font.lisp | 12 |
| 4.1.9 net.didierverna.tfm.core/src/file.lisp | 16 |
| 4.1.10 net.didierverna.tfm.setup/package.lisp | 18 |
| 4.1.11 net.didierverna.tfm.setup/src/version.lisp | 18 |
| 4.1.12 net.didierverna.tfm.setup/src/configuration.lisp | 19 |
| 4.1.13 net.didierverna.tfm.setup/src/readtable.lisp | 19 |
| 5 Packages | 21 |
| 5.1 net.didierverna.tfm | 21 |
| 5.2 net.didierverna.tfm.setup | 33 |
| 6 Definitions | 35 |
| 6.1 Exported definitions | 35 |
| 6.1.1 Special variables | 35 |
| 6.1.2 Functions | 35 |
| 6.1.3 Generic functions | 38 |
| 6.1.4 Conditions | 59 |
| 6.1.5 Classes | 73 |
| 6.2 Internal definitions | 86 |
| 6.2.1 Constants | 86 |
| 6.2.2 Special variables | 86 |
| 6.2.3 Macros | 86 |
| 6.2.4 Functions | 88 |

| | | |
|---------------------------------|-------------------------|------------|
| 6.2.5 | Generic functions | 95 |
| 6.2.6 | Conditions | 97 |
| 6.2.7 | Structures | 98 |
| 6.2.8 | Classes | 100 |
| Appendix A Indexes | | 103 |
| A.1 | Concepts | 103 |
| A.2 | Functions | 104 |
| A.3 | Variables | 111 |
| A.4 | Data types | 114 |

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

TFM is a Common-Lisp interface to the T_EX Font Metrics format. It decodes the contents of `tfm` files, and returns an abstract data structure storing the corresponding font information in an easily accessible way.

This is the TFM Reference Manual. See *The TFM User Manual*, for a more human-readable guide to using TFM.

2 Systems

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

2.1 net.didierverna.tfm

Long Name

TeX Font Metrics

Author Didier Verna

Contact didier@didierverna.net

Home Page

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

Source Control

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

License BSD

Description

A Common Lisp interface to the TeX Font Metrics format

Long Description

TFM (for TeX Font Metrics) is the standard font description format used by TeX. The TFM library parses and decodes TFM files into an abstract data structure, providing easy access to the corresponding font information in Common Lisp.

Version 1.1.1

Dependency

[net.didierverna.tfm.core], page 5, (system)

Source [net.didierverna.tfm.asd], page 9, (file)

2.2 net.didierverna.tfm.core

Long Name

TFM Core

Author Didier Verna

Contact didier@didierverna.net

Home Page

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

Source Control

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

License BSD

Description

TeX Font Metrics, core library

Long Description

The TeX Font Metrics core library provides the main functionality of TFM. For a more complete description of TFM, see the 'net.didierverna.tfm' system.

Version 1.1.1

Dependency

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

Source

[`net.didierverna.tfm.core.asd`], page 9, (file)

Components

- [`package.lisp`], page 9, (file)
- [`src`], page 7, (module)

2.3 `net.didierverna.tfm.setup`

Long Name

TeX Font Metrics, setup library

Author

Didier Verna

Contact

`didier@didierverna.net`

Home Page

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

Source Control

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

License

BSD

Description

TFM's preload setup library

Long Description

The TFM setup library provides support for various preload configuration parameters and meta-utilities. For a more complete description of TFM, see the '`net.didierverna.tfm`' system.

Version

1.1.1

Dependency

`named-readtables`

Source

[`net.didierverna.tfm.setup.asd`], page 9, (file)

Components

- [`package.lisp`], page 18, (file)
- [`src`], page 7, (module)

3 Modules

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

3.1 net.didierverna.tfm.core/src

Dependency

[package.lisp], page 9, (file)

Parent [net.didierverna.tfm.core], page 5, (system)

Location core/src/

Components

- [util.lisp], page 9, (file)
- [intermediate.lisp], page 10, (file)
- [character.lisp], page 11, (file)
- [font.lisp], page 12, (file)
- [file.lisp], page 16, (file)

3.2 net.didierverna.tfm.setup/src

Dependency

[package.lisp], page 18, (file)

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

Location setup/src/

Components

- [version.lisp], page 18, (file)
- [configuration.lisp], page 19, (file)
- [readtable.lisp], page 19, (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.tfm.asd

Location net.didierverna.tfm.asd

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

4.1.2 net.didierverna.tfm.core.asd

Location core/net.didierverna.tfm.core.asd

Systems [net.didierverna.tfm.core], page 5, (system)

4.1.3 net.didierverna.tfm.setup.asd

Location setup/net.didierverna.tfm.setup.asd

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

4.1.4 net.didierverna.tfm.core/package.lisp

Parent [net.didierverna.tfm.core], page 5, (system)

Location core/package.lisp

Packages [net.didierverna.tfm], page 21,

Exported Definitions

[nickname-package], page 37, (function)

4.1.5 net.didierverna.tfm.core/src/util.lisp

Parent [src], page 7, (module)

Location core/src/util.lisp

Exported Definitions

- [fix-word-overflow], page 61, (condition)
- [invalid-bcpl-string], page 61, (condition)
- [invalid-string-length], page 66, (condition)
- [padding], page 53, (method)
- [(setf padding)], page 53, (method)
- [tfm], page 70, (condition)
- [tfm-compliance], page 70, (condition)
- [tfm-compliance-error], page 70, (condition)
- [tfm-compliance-warning], page 71, (condition)
- [tfm-error], page 71, (condition)
- [tfm-usage], page 71, (condition)
- [tfm-usage-error], page 71, (condition)
- [tfm-usage-warning], page 72, (condition)
- [tfm-warning], page 72, (condition)

- [u16-overflow], page 72, (condition)
- [value], page 58, (method)
- [(setf value)], page 58, (method)
- [value], page 58, (method)
- [(setf value)], page 58, (method)
- [value], page 58, (method)
- [(setf value)], page 58, (method)
- [value], page 58, (method)
- [(setf value)], page 58, (method)

Internal Definitions

- [*stream*], page 86, (special variable)
- [define-constant], page 87, (macro)
- [map-accessors], page 87, (macro)
- [read-fix-word], page 93, (function)
- [read-padded-string], page 93, (function)
- [read-u16], page 94, (function)
- [read-u32], page 94, (function)
- [report], page 94, (function)

4.1.6 net.didierverna.tfm.core/src/intermediate.lisp

Dependency

[util.lisp], page 9, (file)

Parent

[src], page 7, (module)

Location

core/src/intermediate.lisp

Internal Definitions

- [bot], page 88, (function)
- [(setf bot)], page 88, (function)
- [char-info], page 98, (structure)
- [char-info-p], page 88, (function)
- [copy-char-info], page 89, (function)
- [copy-exten], page 89, (function)
- [copy-lig/kern], page 89, (function)
- [decode-char-info], page 89, (function)
- [decode-exten], page 89, (function)
- [decode-lig/kern], page 89, (function)
- [depth-index], page 89, (function)
- [(setf depth-index)], page 89, (function)
- [exten], page 99, (structure)
- [exten-index], page 90, (function)
- [(setf exten-index)], page 90, (function)
- [exten-p], page 90, (function)
- [height-index], page 90, (function)
- [(setf height-index)], page 90, (function)

- [italic-index], page 90, (function)
- [(setf italic-index)], page 90, (function)
- [lig/kern], page 100, (structure)
- [lig/kern-index], page 90, (function)
- [(setf lig/kern-index)], page 90, (function)
- [lig/kern-p], page 90, (function)
- [make-char-info], page 91, (function)
- [make-exten], page 91, (function)
- [make-lig/kern], page 92, (function)
- [mid], page 92, (function)
- [(setf mid)], page 92, (function)
- [next], page 92, (function)
- [(setf next)], page 92, (function)
- [next-char], page 92, (function)
- [(setf next-char)], page 92, (function)
- [op], page 92, (function)
- [(setf op)], page 92, (function)
- [remainder], page 94, (function)
- [(setf remainder)], page 94, (function)
- [rep], page 94, (function)
- [(setf rep)], page 94, (function)
- [skip], page 94, (function)
- [(setf skip)], page 94, (function)
- [top], page 95, (function)
- [(setf top)], page 95, (function)
- [width-index], page 95, (function)
- [(setf width-index)], page 95, (function)

4.1.7 net.didierverna.tfm.core/src/character.lisp

Dependency

[intermediate.lisp], page 10, (file)

Parent

[src], page 7, (module)

Location core/src/character.lisp

Exported Definitions

- [bottom-character], page 40, (method)
- [bottom-character], page 40, (method)
- [(setf bottom-character)], page 40, (method)
- [character-metrics], page 73, (class)
- [code], page 42, (method)
- [depth], page 44, (method)
- [(setf depth)], page 44, (method)
- [extensiblep], page 36, (function)
- [font], page 47, (method)

- [height], page 47, (method)
- [(setf height)], page 47, (method)
- [italic-correction], page 48, (method)
- [(setf italic-correction)], page 48, (method)
- [middle-character], page 49, (method)
- [middle-character], page 49, (method)
- [(setf middle-character)], page 49, (method)
- [next-character], page 50, (method)
- [(setf next-character)], page 50, (method)
- [not-extensible], page 69, (condition)
- [repeated-character], page 53, (method)
- [repeated-character], page 54, (method)
- [(setf repeated-character)], page 54, (method)
- [top-character], page 56, (method)
- [top-character], page 56, (method)
- [(setf top-character)], page 56, (method)
- [value], page 58, (method)
- [(setf value)], page 58, (method)
- [width], page 58, (method)
- [(setf width)], page 58, (method)

Internal Definitions

- [+character-metrics-dimension-accessors+], page 86, (constant)
- [define-extension-recipe-pseudo-accessor], page 87, (macro)
- [extension-recipe], page 95, (method)
- [(setf extension-recipe)], page 95, (method)
- [extension-recipe], page 100, (class)
- [make-character-metrics], page 91, (function)
- [make-extension-recipe], page 91, (function)
- [map-character-metrics-dimension-accessors], page 87, (macro)

4.1.8 net.didierverna.tfm.core/src/font.lisp

Dependency

[character.lisp], page 11, (file)

Parent [src], page 7, (module)

Location core/src/font.lisp

Exported Definitions

- [7bits-safe], page 38, (method)
- [(setf 7bits-safe)], page 38, (method)
- [axis-height], page 38, (method)
- [(setf axis-height)], page 38, (method)
- [big-op-spacing1], page 39, (method)
- [(setf big-op-spacing1)], page 39, (method)
- [big-op-spacing2], page 39, (method)

- [(setf big-op-spacing2)], page 39, (method)
- [big-op-spacing3], page 39, (method)
- [(setf big-op-spacing3)], page 39, (method)
- [big-op-spacing4], page 39, (method)
- [(setf big-op-spacing4)], page 39, (method)
- [big-op-spacing5], page 40, (method)
- [(setf big-op-spacing5)], page 40, (method)
- [boundary-character], page 40, (method)
- [(setf boundary-character)], page 40, (method)
- [character-count], page 40, (method)
- [(setf character-count)], page 40, (method)
- [character1], page 41, (method)
- [(setf character1)], page 41, (method)
- [character2], page 41, (method)
- [(setf character2)], page 41, (method)
- [characters], page 41, (method)
- [(setf characters)], page 41, (method)
- [checksum], page 41, (method)
- [(setf checksum)], page 41, (method)
- [composite], page 42, (method)
- [default-rule-thickness], page 42, (method)
- [(setf default-rule-thickness)], page 42, (method)
- [delete-after], page 42, (method)
- [delete-before], page 42, (method)
- [delim1], page 43, (method)
- [(setf delim1)], page 43, (method)
- [delim2], page 43, (method)
- [(setf delim2)], page 43, (method)
- [denom1], page 43, (method)
- [(setf denom1)], page 43, (method)
- [denom2], page 43, (method)
- [(setf denom2)], page 43, (method)
- [design-size], page 44, (method)
- [(setf design-size)], page 44, (method)
- [(setf design-size)], page 44, (method)
- [(setf design-size)], page 44, (method)
- [(setf design-size)], page 44, (method)
- [different-fonts], page 59, (condition)
- [em], page 45, (method)
- [(setf em)], page 45, (method)
- [encoding], page 45, (method)
- [(setf encoding)], page 45, (method)
- [ex], page 45, (method)

- [(setf ex)], page 45, (method)
- [expansion], page 45, (method)
- [(setf expansion)], page 45, (method)
- [extra-space], page 45, (method)
- [(setf extra-space)], page 45, (method)
- [face-code], page 46, (method)
- [(setf face-code)], page 46, (method)
- [face-number], page 46, (method)
- [(setf face-number)], page 46, (method)
- [family], page 46, (method)
- [(setf family)], page 46, (method)
- [file], page 46, (method)
- [(setf file)], page 46, (method)
- [font], page 74, (class)
- [freeze], page 36, (function)
- [frozen], page 47, (method)
- [(setf frozen)], page 47, (method)
- [get-character], page 36, (function)
- [interword-shrink], page 47, (method)
- [(setf interword-shrink)], page 47, (method)
- [interword-space], page 47, (method)
- [(setf interword-space)], page 47, (method)
- [interword-stretch], page 48, (method)
- [(setf interword-stretch)], page 48, (method)
- [invalid-character-code], page 62, (condition)
- [kerning], page 36, (function)
- [(setf kerning)], page 36, (function)
- [ligature], page 36, (function)
- [(setf ligature)], page 37, (function)
- [ligature], page 80, (class)
- [math-extension-font], page 81, (class)
- [math-symbols-font], page 82, (class)
- [max-code], page 49, (method)
- [(setf max-code)], page 49, (method)
- [min-code], page 49, (method)
- [(setf min-code)], page 49, (method)
- [name], page 50, (method)
- [(setf name)], page 50, (method)
- [num1], page 52, (method)
- [(setf num1)], page 52, (method)
- [num2], page 52, (method)
- [(setf num2)], page 52, (method)
- [num3], page 52, (method)

- [(setf num3)], page 52, (method)
- [original-design-size], page 53, (method)
- [(setf original-design-size)], page 53, (method)
- [parameters], page 53, (method)
- [(setf parameters)], page 53, (method)
- [pass-over], page 53, (method)
- [slant], page 54, (method)
- [(setf slant)], page 54, (method)
- [slope], page 54, (method)
- [(setf slope)], page 54, (method)
- [sub1], page 54, (method)
- [(setf sub1)], page 54, (method)
- [sub2], page 55, (method)
- [(setf sub2)], page 55, (method)
- [subdrop], page 55, (method)
- [(setf subdrop)], page 55, (method)
- [sup1], page 55, (method)
- [(setf sup1)], page 55, (method)
- [sup2], page 55, (method)
- [(setf sup2)], page 55, (method)
- [sup3], page 56, (method)
- [(setf sup3)], page 56, (method)
- [supdrop], page 56, (method)
- [(setf supdrop)], page 56, (method)
- [unfreeze], page 37, (function)
- [value], page 57, (method)
- [(setf value)], page 57, (method)
- [weight], page 58, (method)
- [(setf weight)], page 58, (method)

Internal Definitions

- [+font-dimension-accessors+], page 86, (constant)
- [+math-extension-font-dimension-accessors+], page 86, (constant)
- [+math-symbols-font-dimension-accessors+], page 86, (constant)
- [anonymous-font], page 97, (condition)
- [apply-ligature], page 88, (function)
- [code-character], page 89, (function)
- [(setf code-character)], page 89, (function)
- [kernings], page 95, (method)
- [(setf kernings)], page 95, (method)
- [ligatures], page 96, (method)
- [(setf ligatures)], page 96, (method)
- [make-font], page 91, (function)
- [make-ligature], page 92, (function)

- [map-font-dimension-accessors], page 87, (macro)
- [map-math-extension-font-dimension-accessors], page 87, (macro)
- [map-math-symbols-font-dimension-accessors], page 87, (macro)
- [scale], page 96, (generic function)
- [scale], page 96, (method)
- [scale], page 96, (method)
- [scale], page 96, (method)

4.1.9 net.didierverna.tfm.core/src/file.lisp

Dependency

[font.lisp], page 12, (file)

Parent

[src], page 7, (module)

Location

core/src/file.lisp

Exported Definitions

- [actual-size], page 38, (method)
- [(setf actual-size)], page 38, (method)
- [bc], page 38, (method)
- [(setf bc)], page 38, (method)
- [character-list-cycle], page 59, (condition)
- [characters], page 41, (method)
- [(setf characters)], page 41, (method)
- [declared-size], page 42, (method)
- [(setf declared-size)], page 42, (method)
- [ec], page 44, (method)
- [(setf ec)], page 44, (method)
- [extended-tfm], page 60, (condition)
- [file], page 46, (method)
- [(setf file)], page 46, (method)
- [file-overflow], page 60, (condition)
- [file-underflow], page 60, (condition)
- [invalid-char-info], page 61, (condition)
- [invalid-character-range], page 62, (condition)
- [invalid-design-size], page 63, (condition)
- [invalid-header-length], page 63, (condition)
- [invalid-ligature-opcode], page 63, (condition)
- [invalid-section-lengths], page 64, (condition)
- [invalid-table-index], page 67, (condition)
- [invalid-table-length], page 67, (condition)
- [invalid-table-start], page 68, (condition)
- [largest], page 48, (method)
- [(setf largest)], page 48, (method)
- [largest], page 48, (method)
- [(setf largest)], page 48, (method)

- [value], page 57, (method)
- [(setf value)], page 57, (method)
- [value], page 57, (method)
- [(setf value)], page 57, (method)
- [value], page 57, (method)
- [(setf value)], page 57, (method)
- [value], page 57, (method)
- [(setf value)], page 57, (method)

Internal Definitions

- [%run-ligature/kerning-program], page 88, (function)
- [file-size-mixin], page 97, (condition)
- [font-extension-recipe], page 90, (function)
- [invalid-original-design-size], page 97, (condition)
- [load-tfm-font], page 90, (function)
- [parse-character-information], page 92, (function)
- [parse-header], page 93, (function)
- [parse-parameters], page 96, (generic function)
- [parse-parameters], page 96, (method)
- [parse-parameters], page 96, (method)
- [parse-parameters], page 96, (method)
- [parse-parameters], page 96, (method)
- [run-ligature/kerning-program], page 94, (function)
- [table-aref], page 95, (function)
- [tfm-table-error], page 98, (condition)
- [tref], page 87, (macro)

4.1.10 net.didierverna.tfm.setup/package.lisp

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

Location setup/package.lisp

Packages [net.didierverna.tfm.setup], page 33,

4.1.11 net.didierverna.tfm.setup/src/version.lisp

Parent [src], page 7, (module)

Location setup/src/version.lisp

Exported Definitions

- [*copyright-years*], page 35, (special variable)
- [*release-major-level*], page 35, (special variable)
- [*release-minor-level*], page 35, (special variable)
- [*release-name*], page 35, (special variable)
- [*release-status*], page 35, (special variable)
- [*release-status-level*], page 35, (special variable)
- [version], page 37, (function)

Internal Definitions

- [%version], page 88, (function)
- [release-status-number], page 94, (function)

4.1.12 net.didierverna.tfm.setup/src/configuration.lisp

Parent [src], page 7, (module)

Location setup/src/configuration.lisp

Exported Definitions

- [configuration], page 35, (function)
- [configure], page 36, (function)

Internal Definitions

[*configuration*], page 86, (special variable)

4.1.13 net.didierverna.tfm.setup/src/readtable.lisp

Dependency

[configuration.lisp], page 19, (file)

Parent [src], page 7, (module)

Location setup/src/readtable.lisp

Internal Definitions

- [clindent], page 88, (function)
- [defindent], page 86, (macro)
- [i-reader], page 90, (function)
- [~-reader], page 95, (function)

5 Packages

Packages are listed by definition order.

5.1 net.didierverna.tfm

The TeX Font Metrics package.

Source [package.lisp], page 9, (file)

Nickname tfm

Use List

- [net.didierverna.tfm.setup], page 33,
- common-lisp

Exported Definitions

- [7bits-safe], page 38, (generic function)
- [7bits-safe], page 38, (method)
- [(setf 7bits-safe)], page 38, (method)
- [(setf 7bits-safe)], page 38, (generic function)
- [actual-size], page 38, (generic function)
- [actual-size], page 38, (method)
- [(setf actual-size)], page 38, (method)
- [(setf actual-size)], page 38, (generic function)
- [axis-height], page 38, (generic function)
- [axis-height], page 38, (method)
- [(setf axis-height)], page 38, (method)
- [(setf axis-height)], page 38, (generic function)
- [bc], page 38, (generic function)
- [bc], page 38, (method)
- [(setf bc)], page 38, (method)
- [(setf bc)], page 38, (generic function)
- [big-op-spacing1], page 38, (generic function)
- [big-op-spacing1], page 39, (method)
- [(setf big-op-spacing1)], page 39, (method)
- [(setf big-op-spacing1)], page 38, (generic function)
- [big-op-spacing2], page 39, (generic function)
- [big-op-spacing2], page 39, (method)
- [(setf big-op-spacing2)], page 39, (method)
- [(setf big-op-spacing2)], page 39, (generic function)
- [big-op-spacing3], page 39, (generic function)
- [big-op-spacing3], page 39, (method)
- [(setf big-op-spacing3)], page 39, (method)
- [(setf big-op-spacing3)], page 39, (generic function)
- [big-op-spacing4], page 39, (generic function)
- [big-op-spacing4], page 39, (method)

- [(setf big-op-spacing4)], page 39, (method)
- [(setf big-op-spacing4)], page 39, (generic function)
- [big-op-spacing5], page 39, (generic function)
- [big-op-spacing5], page 40, (method)
- [(setf big-op-spacing5)], page 40, (method)
- [(setf big-op-spacing5)], page 39, (generic function)
- [bottom-character], page 40, (generic function)
- [bottom-character], page 40, (method)
- [bottom-character], page 40, (method)
- [(setf bottom-character)], page 40, (method)
- [(setf bottom-character)], page 40, (generic function)
- [boundary-character], page 40, (generic function)
- [boundary-character], page 40, (method)
- [(setf boundary-character)], page 40, (method)
- [(setf boundary-character)], page 40, (generic function)
- [character-count], page 40, (generic function)
- [character-count], page 40, (method)
- [(setf character-count)], page 40, (method)
- [(setf character-count)], page 40, (generic function)
- [character-list-cycle], page 59, (condition)
- [character-metrics], page 73, (class)
- [character1], page 41, (generic function)
- [character1], page 41, (method)
- [(setf character1)], page 41, (method)
- [(setf character1)], page 41, (generic function)
- [character2], page 41, (generic function)
- [character2], page 41, (method)
- [(setf character2)], page 41, (method)
- [(setf character2)], page 41, (generic function)
- [characters], page 41, (generic function)
- [characters], page 41, (method)
- [(setf characters)], page 41, (method)
- [characters], page 41, (method)
- [(setf characters)], page 41, (method)
- [(setf characters)], page 41, (generic function)
- [checksum], page 41, (generic function)
- [checksum], page 41, (method)
- [(setf checksum)], page 41, (method)
- [(setf checksum)], page 41, (generic function)
- [code], page 41, (generic function)
- [code], page 42, (method)
- [composite], page 42, (generic function)
- [composite], page 42, (method)

- [declared-size], page 42, (generic function)
- [declared-size], page 42, (method)
- [(setf declared-size)], page 42, (method)
- [(setf declared-size)], page 42, (generic function)
- [default-rule-thickness], page 42, (generic function)
- [default-rule-thickness], page 42, (method)
- [(setf default-rule-thickness)], page 42, (method)
- [(setf default-rule-thickness)], page 42, (generic function)
- [delete-after], page 42, (generic function)
- [delete-after], page 42, (method)
- [delete-before], page 42, (generic function)
- [delete-before], page 42, (method)
- [delim1], page 43, (generic function)
- [delim1], page 43, (method)
- [(setf delim1)], page 43, (method)
- [(setf delim1)], page 43, (generic function)
- [delim2], page 43, (generic function)
- [delim2], page 43, (method)
- [(setf delim2)], page 43, (method)
- [(setf delim2)], page 43, (generic function)
- [denom1], page 43, (generic function)
- [denom1], page 43, (method)
- [(setf denom1)], page 43, (method)
- [(setf denom1)], page 43, (generic function)
- [denom2], page 43, (generic function)
- [denom2], page 43, (method)
- [(setf denom2)], page 43, (method)
- [(setf denom2)], page 43, (generic function)
- [depth], page 44, (generic function)
- [depth], page 44, (method)
- [(setf depth)], page 44, (method)
- [(setf depth)], page 44, (generic function)
- [design-size], page 44, (generic function)
- [design-size], page 44, (method)
- [(setf design-size)], page 44, (method)
- [(setf design-size)], page 44, (generic function)
- [(setf design-size)], page 44, (method)
- [(setf design-size)], page 44, (method)
- [(setf design-size)], page 44, (method)
- [different-fonts], page 59, (condition)
- [ec], page 44, (generic function)
- [ec], page 44, (method)
- [(setf ec)], page 44, (method)

- [(setf ec)], page 44, (generic function)
- [em], page 44, (generic function)
- [em], page 45, (method)
- [(setf em)], page 45, (method)
- [(setf em)], page 44, (generic function)
- [encoding], page 45, (generic function)
- [encoding], page 45, (method)
- [(setf encoding)], page 45, (method)
- [(setf encoding)], page 45, (generic function)
- [ex], page 45, (generic function)
- [ex], page 45, (method)
- [(setf ex)], page 45, (method)
- [(setf ex)], page 45, (generic function)
- [expansion], page 45, (generic function)
- [expansion], page 45, (method)
- [(setf expansion)], page 45, (method)
- [(setf expansion)], page 45, (generic function)
- [extended-tfm], page 60, (condition)
- [extensiblep], page 36, (function)
- [extra-space], page 45, (generic function)
- [extra-space], page 45, (method)
- [(setf extra-space)], page 45, (method)
- [(setf extra-space)], page 45, (generic function)
- [face-code], page 46, (generic function)
- [face-code], page 46, (method)
- [(setf face-code)], page 46, (method)
- [(setf face-code)], page 46, (generic function)
- [face-number], page 46, (generic function)
- [face-number], page 46, (method)
- [(setf face-number)], page 46, (method)
- [(setf face-number)], page 46, (generic function)
- [family], page 46, (generic function)
- [family], page 46, (method)
- [(setf family)], page 46, (method)
- [(setf family)], page 46, (generic function)
- [file], page 46, (generic function)
- [file], page 46, (method)
- [(setf file)], page 46, (method)
- [file], page 46, (method)
- [(setf file)], page 46, (method)
- [(setf file)], page 46, (generic function)
- [file-overflow], page 60, (condition)
- [file-underflow], page 60, (condition)

- [fix-word-overflow], page 61, (condition)
- [font], page 46, (generic function)
- [font], page 47, (method)
- [font], page 74, (class)
- [freeze], page 36, (function)
- [frozen], page 47, (generic function)
- [frozen], page 47, (method)
- [(setf frozen)], page 47, (method)
- [(setf frozen)], page 47, (generic function)
- [get-character], page 36, (function)
- [height], page 47, (generic function)
- [height], page 47, (method)
- [(setf height)], page 47, (method)
- [(setf height)], page 47, (generic function)
- [interword-shrink], page 47, (generic function)
- [interword-shrink], page 47, (method)
- [(setf interword-shrink)], page 47, (method)
- [(setf interword-shrink)], page 47, (generic function)
- [interword-space], page 47, (generic function)
- [interword-space], page 47, (method)
- [(setf interword-space)], page 47, (method)
- [(setf interword-space)], page 47, (generic function)
- [interword-stretch], page 48, (generic function)
- [interword-stretch], page 48, (method)
- [(setf interword-stretch)], page 48, (method)
- [(setf interword-stretch)], page 48, (generic function)
- [invalid-bcpl-string], page 61, (condition)
- [invalid-char-info], page 61, (condition)
- [invalid-character-code], page 62, (condition)
- [invalid-character-range], page 62, (condition)
- [invalid-design-size], page 63, (condition)
- [invalid-header-length], page 63, (condition)
- [invalid-ligature-opcode], page 63, (condition)
- [invalid-section-lengths], page 64, (condition)
- [invalid-string-length], page 66, (condition)
- [invalid-table-index], page 67, (condition)
- [invalid-table-length], page 67, (condition)
- [invalid-table-start], page 68, (condition)
- [italic-correction], page 48, (generic function)
- [italic-correction], page 48, (method)
- [(setf italic-correction)], page 48, (method)
- [(setf italic-correction)], page 48, (generic function)
- [kerning], page 36, (function)

- [(setf kerning)], page 36, (function)
- [largest], page 48, (generic function)
- [largest], page 48, (method)
- [(setf largest)], page 48, (method)
- [largest], page 48, (method)
- [(setf largest)], page 48, (method)
- [(setf largest)], page 48, (generic function)
- [lf], page 48, (generic function)
- [lf], page 48, (method)
- [(setf lf)], page 48, (method)
- [(setf lf)], page 48, (generic function)
- [lh], page 49, (generic function)
- [lh], page 49, (method)
- [(setf lh)], page 49, (method)
- [(setf lh)], page 49, (generic function)
- [ligature], page 36, (function)
- [(setf ligature)], page 37, (function)
- [ligature], page 80, (class)
- [ligature-cycle], page 68, (condition)
- [load-font], page 37, (function)
- [math-extension-font], page 81, (class)
- [math-symbols-font], page 82, (class)
- [max-code], page 49, (generic function)
- [max-code], page 49, (method)
- [(setf max-code)], page 49, (method)
- [(setf max-code)], page 49, (generic function)
- [middle-character], page 49, (generic function)
- [middle-character], page 49, (method)
- [middle-character], page 49, (method)
- [(setf middle-character)], page 49, (method)
- [(setf middle-character)], page 49, (generic function)
- [min-code], page 49, (generic function)
- [min-code], page 49, (method)
- [(setf min-code)], page 49, (method)
- [(setf min-code)], page 49, (generic function)
- [name], page 49, (generic function)
- [name], page 50, (method)
- [(setf name)], page 50, (method)
- [name], page 50, (method)
- [(setf name)], page 50, (method)
- [(setf name)], page 49, (generic function)
- [nc], page 50, (generic function)
- [nc], page 50, (method)

- `[(setf nc)]`, page 50, (method)
- `[(setf nc)]`, page 50, (generic function)
- `[nd]`, page 50, (generic function)
- `[nd]`, page 50, (method)
- `[(setf nd)]`, page 50, (method)
- `[(setf nd)]`, page 50, (generic function)
- `[ne]`, page 50, (generic function)
- `[ne]`, page 50, (method)
- `[(setf ne)]`, page 50, (method)
- `[(setf ne)]`, page 50, (generic function)
- `[next-character]`, page 50, (generic function)
- `[next-character]`, page 50, (method)
- `[(setf next-character)]`, page 50, (method)
- `[(setf next-character)]`, page 50, (generic function)
- `[nh]`, page 51, (generic function)
- `[nh]`, page 51, (method)
- `[(setf nh)]`, page 51, (method)
- `[(setf nh)]`, page 51, (generic function)
- `[ni]`, page 51, (generic function)
- `[ni]`, page 51, (method)
- `[(setf ni)]`, page 51, (method)
- `[(setf ni)]`, page 51, (generic function)
- `[nickname-package]`, page 37, (function)
- `[nk]`, page 51, (generic function)
- `[nk]`, page 51, (method)
- `[(setf nk)]`, page 51, (method)
- `[(setf nk)]`, page 51, (generic function)
- `[nl]`, page 51, (generic function)
- `[nl]`, page 51, (method)
- `[(setf nl)]`, page 51, (method)
- `[(setf nl)]`, page 51, (generic function)
- `[no-boundary-character]`, page 69, (condition)
- `[not-extensible]`, page 69, (condition)
- `[np]`, page 51, (generic function)
- `[np]`, page 52, (method)
- `[(setf np)]`, page 52, (method)
- `[(setf np)]`, page 51, (generic function)
- `[num1]`, page 52, (generic function)
- `[num1]`, page 52, (method)
- `[(setf num1)]`, page 52, (method)
- `[(setf num1)]`, page 52, (generic function)
- `[num2]`, page 52, (generic function)
- `[num2]`, page 52, (method)

- [(setf num2)], page 52, (method)
- [(setf num2)], page 52, (generic function)
- [num3], page 52, (generic function)
- [num3], page 52, (method)
- [(setf num3)], page 52, (method)
- [(setf num3)], page 52, (generic function)
- [nw], page 52, (generic function)
- [nw], page 52, (method)
- [(setf nw)], page 52, (method)
- [(setf nw)], page 52, (generic function)
- [original-design-size], page 53, (generic function)
- [original-design-size], page 53, (method)
- [(setf original-design-size)], page 53, (method)
- [(setf original-design-size)], page 53, (generic function)
- [padding], page 53, (generic function)
- [padding], page 53, (method)
- [(setf padding)], page 53, (method)
- [(setf padding)], page 53, (generic function)
- [parameters], page 53, (generic function)
- [parameters], page 53, (method)
- [(setf parameters)], page 53, (method)
- [(setf parameters)], page 53, (generic function)
- [pass-over], page 53, (generic function)
- [pass-over], page 53, (method)
- [repeated-character], page 53, (generic function)
- [repeated-character], page 53, (method)
- [repeated-character], page 54, (method)
- [(setf repeated-character)], page 54, (method)
- [(setf repeated-character)], page 53, (generic function)
- [slant], page 54, (generic function)
- [slant], page 54, (method)
- [(setf slant)], page 54, (method)
- [(setf slant)], page 54, (generic function)
- [slope], page 54, (generic function)
- [slope], page 54, (method)
- [(setf slope)], page 54, (method)
- [(setf slope)], page 54, (generic function)
- [smallest], page 54, (generic function)
- [smallest], page 54, (method)
- [(setf smallest)], page 54, (method)
- [(setf smallest)], page 54, (generic function)
- [sub1], page 54, (generic function)
- [sub1], page 54, (method)

- `[(setf sub1)]`, page 54, (method)
- `[(setf sub1)]`, page 54, (generic function)
- `[sub2]`, page 55, (generic function)
- `[sub2]`, page 55, (method)
- `[(setf sub2)]`, page 55, (method)
- `[(setf sub2)]`, page 55, (generic function)
- `[subdrop]`, page 55, (generic function)
- `[subdrop]`, page 55, (method)
- `[(setf subdrop)]`, page 55, (method)
- `[(setf subdrop)]`, page 55, (generic function)
- `[sup1]`, page 55, (generic function)
- `[sup1]`, page 55, (method)
- `[(setf sup1)]`, page 55, (method)
- `[(setf sup1)]`, page 55, (generic function)
- `[sup2]`, page 55, (generic function)
- `[sup2]`, page 55, (method)
- `[(setf sup2)]`, page 55, (method)
- `[(setf sup2)]`, page 55, (generic function)
- `[sup3]`, page 56, (generic function)
- `[sup3]`, page 56, (method)
- `[(setf sup3)]`, page 56, (method)
- `[(setf sup3)]`, page 56, (generic function)
- `[supdrop]`, page 56, (generic function)
- `[supdrop]`, page 56, (method)
- `[(setf supdrop)]`, page 56, (method)
- `[(setf supdrop)]`, page 56, (generic function)
- `[tfm]`, page 70, (condition)
- `[tfm-compliance]`, page 70, (condition)
- `[tfm-compliance-error]`, page 70, (condition)
- `[tfm-compliance-warning]`, page 71, (condition)
- `[tfm-error]`, page 71, (condition)
- `[tfm-usage]`, page 71, (condition)
- `[tfm-usage-error]`, page 71, (condition)
- `[tfm-usage-warning]`, page 72, (condition)
- `[tfm-warning]`, page 72, (condition)
- `[top-character]`, page 56, (generic function)
- `[top-character]`, page 56, (method)
- `[top-character]`, page 56, (method)
- `[(setf top-character)]`, page 56, (method)
- `[(setf top-character)]`, page 56, (generic function)
- `[u16-overflow]`, page 72, (condition)
- `[unfreeze]`, page 37, (function)
- `[value]`, page 56, (generic function)

Internal Definitions

- [%run-ligature/kerning-program], page 88, (function)
- [*stream*], page 86, (special variable)
- [+character-metrics-dimension-accessors+], page 86, (constant)
- [+font-dimension-accessors+], page 86, (constant)
- [+math-extension-font-dimension-accessors+], page 86, (constant)
- [+math-symbols-font-dimension-accessors+], page 86, (constant)
- [anonymous-font], page 97, (condition)
- [apply-ligature], page 88, (function)
- [bot], page 88, (function)
- [(setf bot)], page 88, (function)
- [char-info], page 98, (structure)
- [char-info-p], page 88, (function)
- [code-character], page 89, (function)
- [(setf code-character)], page 89, (function)
- [copy-char-info], page 89, (function)
- [copy-exten], page 89, (function)
- [copy-lig/kern], page 89, (function)
- [decode-char-info], page 89, (function)
- [decode-exten], page 89, (function)
- [decode-lig/kern], page 89, (function)
- [define-constant], page 87, (macro)
- [define-extension-recipe-pseudo-accessor], page 87, (macro)
- [depth-index], page 89, (function)
- [(setf depth-index)], page 89, (function)
- [exten], page 99, (structure)
- [exten-index], page 90, (function)
- [(setf exten-index)], page 90, (function)
- [exten-p], page 90, (function)
- [extension-recipe], page 95, (generic function)
- [extension-recipe], page 95, (method)
- [(setf extension-recipe)], page 95, (method)
- [(setf extension-recipe)], page 95, (generic function)
- [extension-recipe], page 100, (class)
- [file-size-mixin], page 97, (condition)
- [font-extension-recipe], page 90, (function)
- [height-index], page 90, (function)
- [(setf height-index)], page 90, (function)
- [invalid-original-design-size], page 97, (condition)
- [italic-index], page 90, (function)
- [(setf italic-index)], page 90, (function)
- [kernings], page 95, (generic function)
- [kernings], page 95, (method)

- [(setf kernings)], page 95, (method)
- [(setf kernings)], page 95, (generic function)
- [lig/kern], page 100, (structure)
- [lig/kern-index], page 90, (function)
- [(setf lig/kern-index)], page 90, (function)
- [lig/kern-p], page 90, (function)
- [ligatures], page 96, (generic function)
- [ligatures], page 96, (method)
- [(setf ligatures)], page 96, (method)
- [(setf ligatures)], page 96, (generic function)
- [load-tfm-font], page 90, (function)
- [make-char-info], page 91, (function)
- [make-character-metrics], page 91, (function)
- [make-exten], page 91, (function)
- [make-extension-recipe], page 91, (function)
- [make-font], page 91, (function)
- [make-lig/kern], page 92, (function)
- [make-ligature], page 92, (function)
- [map-accessors], page 87, (macro)
- [map-character-metrics-dimension-accessors], page 87, (macro)
- [map-font-dimension-accessors], page 87, (macro)
- [map-math-extension-font-dimension-accessors], page 87, (macro)
- [map-math-symbols-font-dimension-accessors], page 87, (macro)
- [mid], page 92, (function)
- [(setf mid)], page 92, (function)
- [next], page 92, (function)
- [(setf next)], page 92, (function)
- [next-char], page 92, (function)
- [(setf next-char)], page 92, (function)
- [op], page 92, (function)
- [(setf op)], page 92, (function)
- [parse-character-information], page 92, (function)
- [parse-header], page 93, (function)
- [parse-parameters], page 96, (generic function)
- [parse-parameters], page 96, (method)
- [parse-parameters], page 96, (method)
- [parse-parameters], page 96, (method)
- [parse-parameters], page 96, (method)
- [read-fix-word], page 93, (function)
- [read-padded-string], page 93, (function)
- [read-u16], page 94, (function)
- [read-u32], page 94, (function)
- [remainder], page 94, (function)

- [(setf remainder)], page 94, (function)
- [rep], page 94, (function)
- [(setf rep)], page 94, (function)
- [report], page 94, (function)
- [run-ligature/kerning-program], page 94, (function)
- [scale], page 96, (generic function)
- [scale], page 96, (method)
- [scale], page 96, (method)
- [scale], page 96, (method)
- [skip], page 94, (function)
- [(setf skip)], page 94, (function)
- [table-aref], page 95, (function)
- [tfm-table-error], page 98, (condition)
- [top], page 95, (function)
- [(setf top)], page 95, (function)
- [tref], page 87, (macro)
- [width-index], page 95, (function)
- [(setf width-index)], page 95, (function)

5.2 net.didierverna.tfm.setup

The TFM setup library's package.

Source [package.lisp], page 18, (file)

Use List common-lisp

Used By List

[net.didierverna.tfm], page 21,

Exported Definitions

- [*copyright-years*], page 35, (special variable)
- [*release-major-level*], page 35, (special variable)
- [*release-minor-level*], page 35, (special variable)
- [*release-name*], page 35, (special variable)
- [*release-status*], page 35, (special variable)
- [*release-status-level*], page 35, (special variable)
- [configuration], page 35, (function)
- [configure], page 36, (function)
- [version], page 37, (function)

Internal Definitions

- [%version], page 88, (function)
- [*configuration*], page 86, (special variable)
- [clindent], page 88, (function)
- [defindent], page 86, (macro)
- [i-reader], page 90, (function)
- [release-status-number], page 94, (function)
- [~-reader], page 95, (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

copyright-years [Special Variable]

A string denoting the copyright years for the whole project.

Package [net.didierverna.tfm.setup], page 33,

Source [version.lisp], page 18, (file)

release-major-level [Special Variable]

The major level of this release.

Package [net.didierverna.tfm.setup], page 33,

Source [version.lisp], page 18, (file)

release-minor-level [Special Variable]

The minor level of this release.

Package [net.didierverna.tfm.setup], page 33,

Source [version.lisp], page 18, (file)

release-name [Special Variable]

The name of this release.

The general naming theme for TFM is "Uncial Fonts", from the LaTeX Font Catalogue (<https://www.tug.org/FontCatalogue/uncialfonts.html>).

Package [net.didierverna.tfm.setup], page 33,

Source [version.lisp], page 18, (file)

release-status [Special Variable]

The status of this release.

Package [net.didierverna.tfm.setup], page 33,

Source [version.lisp], page 18, (file)

release-status-level [Special Variable]

The status level of this release.

Package [net.didierverna.tfm.setup], page 33,

Source [version.lisp], page 18, (file)

6.1.2 Functions

configuration *KEY* [Function]

Return *KEY*'s value in the current TFM configuration.

Package [net.didierverna.tfm.setup], page 33,

Source [configuration.lisp], page 19, (file)

- configure** *KEY VALUE* [Function]
 Set KEY to VALUE in the current TFM configuration.
- Package** [net.didierverna.tfm.setup], page 33,
Source [configuration.lisp], page 19, (file)
- extensiblep** *CHARACTER* [Function]
 Return T if CHARACTER has an extension recipe.
- Package** [net.didierverna.tfm], page 21,
Source [character.lisp], page 11, (file)
- freeze** *FONT* [Function]
 Freeze FONT.
 Freezing a font means that all dimensions normally expressed in design size units are multiplied by it, so as to lead values in TeX point units.
 If FONT is already frozen, this function does nothing and returns NIL. Otherwise, it returns T.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
- get-character** *CODE FONT* [Function]
 Return FONT's CODE character, or NIL.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
- kerning** *CHARACTER1 CHARACTER2* [Function]
 Return kerning for CHARACTER1 and CHARACTER2, or NIL.
 If CHARACTER1 and CHARACTER2 don't belong to the same font, signal a DIFFERENT-FONTS error.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
Writer [(setf kerning)], page 36, (function)
- (setf kerning)** *KERNING CHARACTER1 CHARACTER2* [Function]
 Set KERNING for CHARACTER1 and CHARACTER2.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
Reader [kerning], page 36, (function)
- ligature** *CHARACTER1 CHARACTER2* [Function]
 Return ligature for CHARACTER1 and CHARACTER2, or NIL.
 If CHARACTER1 and CHARACTER2 don't belong to the same font, signal a DIFFERENT-FONTS error.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
Writer [(setf ligature)], page 37, (function)

(setf ligature) *LIGATURE CHARACTER1 CHARACTER2* [Function]
Set LIGATURE for CHARACTER1 and CHARACTER2.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

Reader [ligature], page 36, (function)

load-font *FILE &rest ARGUMENTS &key DESIGN-SIZE FREEZE* [Function]
Load FILE into a new font, and return it.

- If provided, DESIGN-SIZE overrides the font's original value. It must be a real greater or equal to 1.

- When FREEZE (NIL by default), freeze the font immediately after creation. See the eponymous function for more information.

Only actual TFM data is currently supported. If OFM or JFM data is detected, this function signals an EXTENDED-TFM warning and returns NIL.

While loading TFM data, any signalled condition is restartable with CANCEL-LOADING, in which case this function simply returns NIL.

Package [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

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

Package [net.didierverna.tfm], page 21,

Source [package.lisp], page 9, (file)

unfreeze *FONT* [Function]
Unfreeze FONT.

Unfreezing means performing the inverse of what FREEZE does.

If FONT is not frozen, this function does nothing and returns NIL. Otherwise, it returns T.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

version &optional TYPE [Function]
Return the current version of TFM.

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 "Artificial Uncial".

As for the short version, a patchlevel of 0 is ignored in the output.

Package [net.didierverna.tfm.setup], page 33,

Source [version.lisp], page 18, (file)

6.1.3 Generic functions

7bits-safe *OBJECT* [Generic Function]

(setf 7bits-safe) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

7bits-safe (*FONT font*) [Method]

(setf 7bits-safe) *NEW-VALUE (FONT font)* [Method]

Whether the font is 7 bits safe (0 or 1), or NIL.

When 1, it means that no character of code lesser than 128 can lead to a character of code greater than 128 by ways of ligatures or extensible recipes.

Source [font.lisp], page 12, (file)

actual-size *CONDITION* [Generic Function]

(setf actual-size) *NEW-VALUE CONDITION* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

actual-size (*CONDITION file-size-mixin*) [Method]

(setf actual-size) *NEW-VALUE (CONDITION file-size-mixin)* [Method]

Source [file.lisp], page 16, (file)

axis-height *OBJECT* [Generic Function]

(setf axis-height) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

axis-height (*MATH-SYMBOLS-FONT math-symbols-font*) [Method]

(setf axis-height) *NEW-VALUE (MATH-SYMBOLS-FONT math-symbols-font)* [Method]

The font's AXIS-HEIGHT parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

bc *CONDITION* [Generic Function]

(setf bc) *NEW-VALUE CONDITION* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

bc (*CONDITION invalid-character-range*) [Method]

(setf bc) *NEW-VALUE (CONDITION invalid-character-range)* [Method]

Source [file.lisp], page 16, (file)

big-op-spacing1 *OBJECT* [Generic Function]

(setf big-op-spacing1) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`big-op-spacing1` (*MATH-EXTENSION-FONT* `math-extension-font`) [Method]

`(setf big-op-spacing1) NEW-VALUE` [Method]
 (*MATH-EXTENSION-FONT* `math-extension-font`)

The font's BIG-OP-SPACING1 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`big-op-spacing2` *OBJECT* [Generic Function]

`(setf big-op-spacing2) NEW-VALUE OBJECT` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`big-op-spacing2` (*MATH-EXTENSION-FONT* `math-extension-font`) [Method]

`(setf big-op-spacing2) NEW-VALUE` [Method]
 (*MATH-EXTENSION-FONT* `math-extension-font`)

The font's BIG-OP-SPACING2 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`big-op-spacing3` *OBJECT* [Generic Function]

`(setf big-op-spacing3) NEW-VALUE OBJECT` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`big-op-spacing3` (*MATH-EXTENSION-FONT* `math-extension-font`) [Method]

`(setf big-op-spacing3) NEW-VALUE` [Method]
 (*MATH-EXTENSION-FONT* `math-extension-font`)

The font's BIG-OP-SPACING3 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`big-op-spacing4` *OBJECT* [Generic Function]

`(setf big-op-spacing4) NEW-VALUE OBJECT` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`big-op-spacing4` (*MATH-EXTENSION-FONT* `math-extension-font`) [Method]

`(setf big-op-spacing4) NEW-VALUE` [Method]
 (*MATH-EXTENSION-FONT* `math-extension-font`)

The font's BIG-OP-SPACING4 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`big-op-spacing5` *OBJECT* [Generic Function]

`(setf big-op-spacing5) NEW-VALUE OBJECT` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`big-op-spacing5` (*MATH-EXTENSION-FONT* `math-extension-font`) [Method]

`(setf big-op-spacing5) NEW-VALUE` [Method]
 (*MATH-EXTENSION-FONT* `math-extension-font`)

The font's BIG-OP-SPACING5 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`bottom-character` *OBJECT* [Generic Function]

`(setf bottom-character) NEW-VALUE OBJECT` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`bottom-character` (*CHARACTER* `character-metrics`) [Method]

Return extensible CHARACTER's BOTTOM-CHARACTER.

If CHARACTER is not extensible, signal a NOT-EXTENSIBLE error.

Source [character.lisp], page 11, (file)

`bottom-character` (*EXTENSION-RECIPE* `extension-recipe`) [Method]

`(setf bottom-character) NEW-VALUE` [Method]
 (*EXTENSION-RECIPE* `extension-recipe`)

The recipe's bottom character, or NIL.

Source [character.lisp], page 11, (file)

`boundary-character` *OBJECT* [Generic Function]

`(setf boundary-character) NEW-VALUE OBJECT` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`boundary-character` (*FONT* `font`) [Method]

`(setf boundary-character) NEW-VALUE` (*FONT* `font`) [Method]

The font's boundary character, or NIL.

This character is also accessible by code, like normal ones. However, it is the only character the code of which may be outside [MIN-CODE,MAX-CODE] (see TeX: the Program [545]). Finally, this character is not included in the character count, unless it exists for real in the font (has non-zero metrics).

Source [font.lisp], page 12, (file)

`character-count` *OBJECT* [Generic Function]

`(setf character-count) NEW-VALUE OBJECT` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`character-count` (*FONT* `font`) [Method]

`(setf character-count) NEW-VALUE` (*FONT* `font`) [Method]

The font's number of characters.

The character count does not include the boundary character, unless that character really exists in the font (has non-zero metrics).

Source [font.lisp], page 12, (file)

- character1** *CONDITION* [Generic Function]
 (setf character1) *NEW-VALUE CONDITION* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- character1 (*CONDITION* different-fonts) [Method]
 (setf character1) *NEW-VALUE (CONDITION*
different-fonts) [Method]
Source [font.lisp], page 12, (file)
- character2** *CONDITION* [Generic Function]
 (setf character2) *NEW-VALUE CONDITION* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- character2 (*CONDITION* different-fonts) [Method]
 (setf character2) *NEW-VALUE (CONDITION*
different-fonts) [Method]
Source [font.lisp], page 12, (file)
- characters** *OBJECT* [Generic Function]
 (setf characters) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- characters (*CONDITION* ligature-cycle) [Method]
 (setf characters) *NEW-VALUE (CONDITION*
ligature-cycle) [Method]
Source [file.lisp], page 16, (file)
- characters (*FONT* font) [Method]
 (setf characters) *NEW-VALUE (FONT* font) [Method]
 The font's characters.
 This is a hash table associating character codes with characters.
Source [font.lisp], page 12, (file)
- checksum** *OBJECT* [Generic Function]
 (setf checksum) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- checksum (*FONT* font) [Method]
 (setf checksum) *NEW-VALUE (FONT* font) [Method]
 The font's checksum, as provided by Metafont.
Source [font.lisp], page 12, (file)
- code** *OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods

- `code` (*CHARACTER-METRICS* character-metrics) [Method]
 The character's numerical code.
Source [character.lisp], page 11, (file)
- `composite` *OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `composite` (*LIGATURE* ligature) [Method]
 The character to insert between the two original ones.
Source [font.lisp], page 12, (file)
- `declared-size` *CONDITION* [Generic Function]
`(setf declared-size) NEW-VALUE CONDITION` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `declared-size` (*CONDITION* file-size-mixin) [Method]
`(setf declared-size) NEW-VALUE (CONDITION` [Method]
`file-size-mixin)`
Source [file.lisp], page 16, (file)
- `default-rule-thickness` *OBJECT* [Generic Function]
`(setf default-rule-thickness) NEW-VALUE OBJECT` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `default-rule-thickness` (*MATH-EXTENSION-FONT* [Method]
`math-extension-font)`
`(setf default-rule-thickness) NEW-VALUE` [Method]
`(MATH-EXTENSION-FONT math-extension-font)`
 The font's default rule thickness.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- `delete-after` *OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `delete-after` (*LIGATURE* ligature) [Method]
 Whether to delete the character after the ligature.
Source [font.lisp], page 12, (file)
- `delete-before` *OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `delete-before` (*LIGATURE* ligature) [Method]
 Whether to delete the character before the ligature.
Source [font.lisp], page 12, (file)

`delim1` *OBJECT* [Generic Function]
 (`setf delim1`) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`delim1` (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

(`setf delim1`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

The font's DELIM1 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`delim2` *OBJECT* [Generic Function]
 (`setf delim2`) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`delim2` (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

(`setf delim2`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

The font's DELIM2 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`denom1` *OBJECT* [Generic Function]
 (`setf denom1`) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`denom1` (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

(`setf denom1`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

The font's DENOM1 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`denom2` *OBJECT* [Generic Function]
 (`setf denom2`) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`denom2` (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

(`setf denom2`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

The font's DENOM2 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`depth` *OBJECT* [Generic Function]
`(setf depth) NEW-VALUE OBJECT` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`depth (CHARACTER-METRICS character-metrics)` [Method]
`(setf depth) NEW-VALUE (CHARACTER-METRICS` [Method]
`character-metrics)`

The character's depth.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [character.lisp], page 11, (file)

`design-size` *OBJECT* [Generic Function]
`(setf design-size) NEW-VALUE OBJECT` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`design-size (FONT font)` [Method]
`(setf design-size) NEW-VALUE (FONT font)` [Method]

The font's design size, in units of TeX points.

Source [font.lisp], page 12, (file)

`(setf design-size) DESIGN-SIZE FONT around` [Method]
 Check that DESIGN-SIZE is a real greater or equal to 1.

Source [font.lisp], page 12, (file)

`(setf design-size) DESIGN-SIZE FONT after` [Method]
 Rescale FONT if frozen.

Source [font.lisp], page 12, (file)

`(setf design-size) DESIGN-SIZE FONT before` [Method]
 Unscale FONT if frozen.

Source [font.lisp], page 12, (file)

`ec` *CONDITION* [Generic Function]
`(setf ec) NEW-VALUE CONDITION` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`ec (CONDITION invalid-character-range)` [Method]
`(setf ec) NEW-VALUE (CONDITION` [Method]
`invalid-character-range)`

Source [file.lisp], page 16, (file)

`em` *OBJECT* [Generic Function]
`(setf em) NEW-VALUE OBJECT` [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

- `em` (*FONT font*) [Method]
 (`setf em`) *NEW-VALUE (FONT font)* [Method]
 The font's em size.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- `encoding` *OBJECT* [Generic Function]
 (`setf encoding`) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `encoding` (*FONT font*) [Method]
 (`setf encoding`) *NEW-VALUE (FONT font)* [Method]
 The font's character coding scheme (a BCPL string), or NIL.
Source [font.lisp], page 12, (file)
- `ex` *OBJECT* [Generic Function]
 (`setf ex`) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `ex` (*FONT font*) [Method]
 (`setf ex`) *NEW-VALUE (FONT font)* [Method]
 The font's ex size.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- `expansion` *OBJECT* [Generic Function]
 (`setf expansion`) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `expansion` (*FONT font*) [Method]
 (`setf expansion`) *NEW-VALUE (FONT font)* [Method]
 The font's expansion (:regular, :condensed, :extended), or NIL. When available, it is decoded from the font face number.
Source [font.lisp], page 12, (file)
- `extra-space` *OBJECT* [Generic Function]
 (`setf extra-space`) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `extra-space` (*FONT font*) [Method]
 (`setf extra-space`) *NEW-VALUE (FONT font)* [Method]
 The font's extra space.
 It is expressed in design size units, or in TeX point units if the font is frozen.
 This is the additional space to put at the end of sentences.
Source [font.lisp], page 12, (file)

- face-code *OBJECT* [Generic Function]
 (setf face-code) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- face-code (*FONT font*) [Method]
 (setf face-code) *NEW-VALUE (FONT font)* [Method]
 The font's 3-letters face code, or NIL.
 When available, it is the concatenation of the upcased first letters of the font's weight, slope, and expansion.
Source [font.lisp], page 12, (file)
- face-number *OBJECT* [Generic Function]
 (setf face-number) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- face-number (*FONT font*) [Method]
 (setf face-number) *NEW-VALUE (FONT font)* [Method]
 The font's face number, or NIL.
Source [font.lisp], page 12, (file)
- family *OBJECT* [Generic Function]
 (setf family) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- family (*FONT font*) [Method]
 (setf family) *NEW-VALUE (FONT font)* [Method]
 The font's family (a BCPL string), or NIL.
Source [font.lisp], page 12, (file)
- file *OBJECT* [Generic Function]
 (setf file) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- file (*CONDITION extended-tfm*) [Method]
 (setf file) *NEW-VALUE (CONDITION extended-tfm)* [Method]
Source [file.lisp], page 16, (file)
- file (*FONT font*) [Method]
 (setf file) *NEW-VALUE (FONT font)* [Method]
 The file from which the font was loaded, or NIL.
Source [font.lisp], page 12, (file)
- font *OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods

- font** (*CHARACTER-METRICS* character-metrics) [Method]
 The character's font.
Source [character.lisp], page 11, (file)
- frozen** *OBJECT* [Generic Function]
 (setf frozen) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- frozen** (*FONT* font) [Method]
 (setf frozen) *NEW-VALUE (FONT font)* [Method]
 Whether the font is frozen.
Source [font.lisp], page 12, (file)
- height** *OBJECT* [Generic Function]
 (setf height) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- height** (*CHARACTER-METRICS* character-metrics) [Method]
 (setf height) *NEW-VALUE (CHARACTER-METRICS*
character-metrics) [Method]
 The character's height.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [character.lisp], page 11, (file)
- interword-shrink** *OBJECT* [Generic Function]
 (setf interword-shrink) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- interword-shrink** (*FONT* font) [Method]
 (setf interword-shrink) *NEW-VALUE (FONT font)* [Method]
 The font's interword shrinkability.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- interword-space** *OBJECT* [Generic Function]
 (setf interword-space) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- interword-space** (*FONT* font) [Method]
 (setf interword-space) *NEW-VALUE (FONT font)* [Method]
 The font's normal interword space.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)

- `interword-stretch` *OBJECT* [Generic Function]
`(setf interword-stretch) NEW-VALUE OBJECT` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `interword-stretch` (*FONT font*) [Method]
`(setf interword-stretch) NEW-VALUE (FONT font)` [Method]
 The font's interword stretchability.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- `italic-correction` *OBJECT* [Generic Function]
`(setf italic-correction) NEW-VALUE OBJECT` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `italic-correction` (*CHARACTER-METRICS character-metrics*) [Method]
`(setf italic-correction) NEW-VALUE (CHARACTER-METRICS character-metrics)` [Method]
 The character's italic correction.
 TeX uses this value for regular characters followed by the command /, and also in math mode for superscript placement. It is expressed in design size units, or in TeX point units if the font is frozen.
Source [character.lisp], page 11, (file)
- `largest` *CONDITION* [Generic Function]
`(setf largest) NEW-VALUE CONDITION` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `largest` (*CONDITION invalid-table-length*) [Method]
`(setf largest) NEW-VALUE (CONDITION invalid-table-length)` [Method]
Source [file.lisp], page 16, (file)
- `largest` (*CONDITION invalid-table-index*) [Method]
`(setf largest) NEW-VALUE (CONDITION invalid-table-index)` [Method]
Source [file.lisp], page 16, (file)
- `lf` *CONDITION* [Generic Function]
`(setf lf) NEW-VALUE CONDITION` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `lf` (*CONDITION invalid-section-lengths*) [Method]
`(setf lf) NEW-VALUE (CONDITION invalid-section-lengths)` [Method]
Source [file.lisp], page 16, (file)

- lh** *CONDITION* [Generic Function]
 (setf lh) *NEW-VALUE CONDITION* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- lh (*CONDITION* invalid-section-lengths) [Method]
 (setf lh) *NEW-VALUE (CONDITION*
 invalid-section-lengths) [Method]
Source [file.lisp], page 16, (file)
- max-code *OBJECT* [Generic Function]
 (setf max-code) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- max-code (*FONT* font) [Method]
 (setf max-code) *NEW-VALUE (FONT* font) [Method]
 The font's largest character code.
Source [font.lisp], page 12, (file)
- middle-character *OBJECT* [Generic Function]
 (setf middle-character) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- middle-character (*CHARACTER* character-metrics) [Method]
 Return extensible CHARACTER's MIDDLE-CHARACTER.
 If CHARACTER is not extensible, signal a NOT-EXTENSIBLE error.
Source [character.lisp], page 11, (file)
- middle-character (*EXTENSION-RECIPE*
 extension-recipe) [Method]
 (setf middle-character) *NEW-VALUE* [Method]
 (*EXTENSION-RECIPE* extension-recipe)
 The recipe's middle character, or NIL.
Source [character.lisp], page 11, (file)
- min-code *OBJECT* [Generic Function]
 (setf min-code) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- min-code (*FONT* font) [Method]
 (setf min-code) *NEW-VALUE (FONT* font) [Method]
 The font's smallest character code.
Source [font.lisp], page 12, (file)
- name *OBJECT* [Generic Function]
 (setf name) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods

- name** (*CONDITION* *tfm-table-error*) [Method]
 (setf name) *NEW-VALUE* (*CONDITION* *tfm-table-error*) [Method]
Source [file.lisp], page 16, (file)
- name** (*FONT* font) [Method]
 (setf name) *NEW-VALUE* (*FONT* font) [Method]
 The font's name.
 When the font is loaded from a file, it defaults to the file's base name, along with potential scaling information.
Source [font.lisp], page 12, (file)
- nc** *CONDITION* [Generic Function]
 (setf nc) *NEW-VALUE* *CONDITION* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- nc** (*CONDITION* *invalid-section-lengths*) [Method]
 (setf nc) *NEW-VALUE* (*CONDITION* *invalid-section-lengths*) [Method]
Source [file.lisp], page 16, (file)
- nd** *CONDITION* [Generic Function]
 (setf nd) *NEW-VALUE* *CONDITION* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- nd** (*CONDITION* *invalid-section-lengths*) [Method]
 (setf nd) *NEW-VALUE* (*CONDITION* *invalid-section-lengths*) [Method]
Source [file.lisp], page 16, (file)
- ne** *CONDITION* [Generic Function]
 (setf ne) *NEW-VALUE* *CONDITION* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- ne** (*CONDITION* *invalid-section-lengths*) [Method]
 (setf ne) *NEW-VALUE* (*CONDITION* *invalid-section-lengths*) [Method]
Source [file.lisp], page 16, (file)
- next-character** *OBJECT* [Generic Function]
 (setf next-character) *NEW-VALUE* *OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- next-character** (*CHARACTER-METRICS* character-metrics) [Method]
 (setf next-character) *NEW-VALUE* (*CHARACTER-METRICS* character-metrics) [Method]
 The next character in a character list.
 This slot is non-null only if the character is part of a chain of characters

of ascending size, and not the last one (see TeX: the Program [544]). It is mutually exclusive with the EXTENSION-RECIPE slot, and also with the existence of a ligature or kerning program for this character.

Source [character.lisp], page 11, (file)

nh *CONDITION* [Generic Function]
 (setf nh) *NEW-VALUE CONDITION* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

nh (*CONDITION* invalid-section-lengths) [Method]

(setf nh) *NEW-VALUE (CONDITION* [Method]

invalid-section-lengths)

Source [file.lisp], page 16, (file)

ni *CONDITION* [Generic Function]
 (setf ni) *NEW-VALUE CONDITION* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

ni (*CONDITION* invalid-section-lengths) [Method]

(setf ni) *NEW-VALUE (CONDITION* [Method]

invalid-section-lengths)

Source [file.lisp], page 16, (file)

nk *CONDITION* [Generic Function]
 (setf nk) *NEW-VALUE CONDITION* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

nk (*CONDITION* invalid-section-lengths) [Method]

(setf nk) *NEW-VALUE (CONDITION* [Method]

invalid-section-lengths)

Source [file.lisp], page 16, (file)

nl *CONDITION* [Generic Function]
 (setf nl) *NEW-VALUE CONDITION* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

nl (*CONDITION* invalid-section-lengths) [Method]

(setf nl) *NEW-VALUE (CONDITION* [Method]

invalid-section-lengths)

Source [file.lisp], page 16, (file)

np *CONDITION* [Generic Function]
 (setf np) *NEW-VALUE CONDITION* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

- `np` (*CONDITION* *invalid-section-lengths*) [Method]
 (`setf np`) *NEW-VALUE* (*CONDITION* *invalid-section-lengths*) [Method]
Source [file.lisp], page 16, (file)
- `num1` *OBJECT* [Generic Function]
 (`setf num1`) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `num1` (*MATH-SYMBOLS-FONT* *math-symbols-font*) [Method]
 (`setf num1`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* *math-symbols-font*) [Method]
 The font's NUM1 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- `num2` *OBJECT* [Generic Function]
 (`setf num2`) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `num2` (*MATH-SYMBOLS-FONT* *math-symbols-font*) [Method]
 (`setf num2`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* *math-symbols-font*) [Method]
 The font's NUM2 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- `num3` *OBJECT* [Generic Function]
 (`setf num3`) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `num3` (*MATH-SYMBOLS-FONT* *math-symbols-font*) [Method]
 (`setf num3`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* *math-symbols-font*) [Method]
 The font's NUM2 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- `nw` *CONDITION* [Generic Function]
 (`setf nw`) *NEW-VALUE CONDITION* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `nw` (*CONDITION* *invalid-section-lengths*) [Method]
 (`setf nw`) *NEW-VALUE* (*CONDITION* *invalid-section-lengths*) [Method]
Source [file.lisp], page 16, (file)

- `original-design-size` *OBJECT* [Generic Function]
`(setf original-design-size) NEW-VALUE OBJECT` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `original-design-size` (*FONT* font) [Method]
`(setf original-design-size) NEW-VALUE (FONT font)` [Method]
 The font's original design size, in units of TeX points.
Source [font.lisp], page 12, (file)
- `padding` *CONDITION* [Generic Function]
`(setf padding) NEW-VALUE CONDITION` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `padding` (*CONDITION* invalid-string-length) [Method]
`(setf padding) NEW-VALUE (CONDITION invalid-string-length)` [Method]
Source [util.lisp], page 9, (file)
- `parameters` *OBJECT* [Generic Function]
`(setf parameters) NEW-VALUE OBJECT` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `parameters` (*FONT* font) [Method]
`(setf parameters) NEW-VALUE (FONT font)` [Method]
 The font's additional parameters array, or NIL.
 Parameters are expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- `pass-over` *OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `pass-over` (*LIGATURE* ligature) [Method]
 The number of characters to skip for reaching the next character.
Source [font.lisp], page 12, (file)
- `repeated-character` *OBJECT* [Generic Function]
`(setf repeated-character) NEW-VALUE OBJECT` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `repeated-character` (*CHARACTER* character-metrics) [Method]
 Return extensible CHARACTER's REPEATED-CHARACTER.
 If CHARACTER is not extensible, signal a NOT-EXTENSIBLE error.
Source [character.lisp], page 11, (file)

- repeated-character (*EXTENSION-RECIPE* extension-recipe) [Method]
 (setf repeated-character) *NEW-VALUE* (*EXTENSION-RECIPE* extension-recipe) [Method]
 The recipe's repeated character.
Source [character.lisp], page 11, (file)
- slant *OBJECT* [Generic Function]
 (setf slant) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- slant (*FONT* font) [Method]
 (setf slant) *NEW-VALUE (FONT* font) [Method]
 The font's slant (a scalar ratio).
Source [font.lisp], page 12, (file)
- slope *OBJECT* [Generic Function]
 (setf slope) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- slope (*FONT* font) [Method]
 (setf slope) *NEW-VALUE (FONT* font) [Method]
 The font's slope (:roman, :italic), or NIL.
 When available, it is decoded from the font face number.
Source [font.lisp], page 12, (file)
- smallest *CONDITION* [Generic Function]
 (setf smallest) *NEW-VALUE CONDITION* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- smallest (*CONDITION* invalid-table-length) [Method]
 (setf smallest) *NEW-VALUE (CONDITION* invalid-table-length) [Method]
Source [file.lisp], page 16, (file)
- sub1 *OBJECT* [Generic Function]
 (setf sub1) *NEW-VALUE OBJECT* [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- sub1 (*MATH-SYMBOLS-FONT* math-symbols-font) [Method]
 (setf sub1) *NEW-VALUE (MATH-SYMBOLS-FONT* math-symbols-font) [Method]
 The font's SUB1 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)

`sub2` *OBJECT* [Generic Function]
 (`setf sub2`) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`sub2` (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

(`setf sub2`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

The font's SUB2 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`subdrop` *OBJECT* [Generic Function]
 (`setf subdrop`) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`subdrop` (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

(`setf subdrop`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

The font's SUBDROP parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`sup1` *OBJECT* [Generic Function]
 (`setf sup1`) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`sup1` (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

(`setf sup1`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

The font's SUP1 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

`sup2` *OBJECT* [Generic Function]
 (`setf sup2`) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

`sup2` (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

(`setf sup2`) *NEW-VALUE* (*MATH-SYMBOLS-FONT* `math-symbols-font`) [Method]

The font's SUP2 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [font.lisp], page 12, (file)

- `sup3` *OBJECT* [Generic Function]
`(setf sup3) NEW-VALUE OBJECT` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `sup3` (*MATH-SYMBOLS-FONT* math-symbols-font) [Method]
`(setf sup3) NEW-VALUE (MATH-SYMBOLS-FONT`
`math-symbols-font)` [Method]
The font's SUP2 parameter.
It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- `supdrop` *OBJECT* [Generic Function]
`(setf supdrop) NEW-VALUE OBJECT` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `supdrop` (*MATH-SYMBOLS-FONT* math-symbols-font) [Method]
`(setf supdrop) NEW-VALUE (MATH-SYMBOLS-FONT`
`math-symbols-font)` [Method]
The font's SUPDROP parameter.
It is expressed in design size units, or in TeX point units if the font is frozen.
Source [font.lisp], page 12, (file)
- `top-character` *OBJECT* [Generic Function]
`(setf top-character) NEW-VALUE OBJECT` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `top-character` (*CHARACTER* character-metrics) [Method]
Return extensible CHARACTER's TOP-CHARACTER.
If CHARACTER is not extensible, signal a NOT-EXTENSIBLE error.
Source [character.lisp], page 11, (file)
- `top-character` (*EXTENSION-RECIPE*
extension-recipe) [Method]
`(setf top-character) NEW-VALUE` [Method]
`(EXTENSION-RECIPE extension-recipe)`
The recipe's top character, or NIL.
Source [character.lisp], page 11, (file)
- `value` *CONDITION* [Generic Function]
`(setf value) NEW-VALUE CONDITION` [Generic Function]
Package [net.didierverna.tfm], page 21,
Methods
- `value` (*CONDITION* extended-tfm) [Method]
`(setf value) NEW-VALUE (CONDITION` extended-tfm) [Method]
Source [file.lisp], page 16, (file)

| | |
|---|----------|
| value (<i>CONDITION</i> <i>invalid-table-length</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>invalid-table-length</i>) | [Method] |
| Source [file.lisp], page 16, (file) | |
| value (<i>CONDITION</i> <i>invalid-header-length</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>invalid-header-length</i>) | [Method] |
| Source [file.lisp], page 16, (file) | |
| value (<i>CONDITION</i> <i>ligature-cycle</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>ligature-cycle</i>) | [Method] |
| Source [file.lisp], page 16, (file) | |
| value (<i>CONDITION</i> <i>character-list-cycle</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>character-list-cycle</i>) | [Method] |
| Source [file.lisp], page 16, (file) | |
| value (<i>CONDITION</i> <i>invalid-table-start</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>invalid-table-start</i>) | [Method] |
| Source [file.lisp], page 16, (file) | |
| value (<i>CONDITION</i> <i>invalid-char-info</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>invalid-char-info</i>) | [Method] |
| Source [file.lisp], page 16, (file) | |
| value (<i>CONDITION</i> <i>invalid-ligature-opcode</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>invalid-ligature-opcode</i>) | [Method] |
| Source [file.lisp], page 16, (file) | |
| value (<i>CONDITION</i> <i>invalid-table-index</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>invalid-table-index</i>) | [Method] |
| Source [file.lisp], page 16, (file) | |
| value (<i>CONDITION</i> <i>invalid-original-design-size</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>invalid-original-design-size</i>) | [Method] |
| Source [file.lisp], page 16, (file) | |
| value (<i>CONDITION</i> <i>invalid-design-size</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>invalid-design-size</i>) | [Method] |
| Source [file.lisp], page 16, (file) | |
| value (<i>CONDITION</i> <i>invalid-character-code</i>) | [Method] |
| (setf value) <i>NEW-VALUE</i> (<i>CONDITION</i> <i>invalid-character-code</i>) | [Method] |
| Source [font.lisp], page 12, (file) | |

value (*CONDITION* not-extensible) [Method]
 (setf value) *NEW-VALUE* (*CONDITION*
 not-extensible) [Method]

Source [character.lisp], page 11, (file)

value (*CONDITION* invalid-bcpl-string) [Method]
 (setf value) *NEW-VALUE* (*CONDITION*
 invalid-bcpl-string) [Method]

Source [util.lisp], page 9, (file)

value (*CONDITION* invalid-string-length) [Method]
 (setf value) *NEW-VALUE* (*CONDITION*
 invalid-string-length) [Method]

Source [util.lisp], page 9, (file)

value (*CONDITION* fix-word-overflow) [Method]
 (setf value) *NEW-VALUE* (*CONDITION*
 fix-word-overflow) [Method]

Source [util.lisp], page 9, (file)

value (*CONDITION* u16-overflow) [Method]
 (setf value) *NEW-VALUE* (*CONDITION* u16-overflow) [Method]

Source [util.lisp], page 9, (file)

weight *OBJECT* [Generic Function]
 (setf weight) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

weight (*FONT* font) [Method]
 (setf weight) *NEW-VALUE* (*FONT* font) [Method]

The font's weight (:medium, :bold, :light), or NIL. When available, it is decoded from the font face number.

Source [font.lisp], page 12, (file)

width *OBJECT* [Generic Function]
 (setf width) *NEW-VALUE OBJECT* [Generic Function]

Package [net.didierverna.tfm], page 21,

Methods

width (*CHARACTER-METRICS* character-metrics) [Method]
 (setf width) *NEW-VALUE* (*CHARACTER-METRICS*
 character-metrics) [Method]

The character's width.

It is expressed in design size units, or in TeX point units if the font is frozen.

Source [character.lisp], page 11, (file)

6.1.4 Conditions

`character-list-cycle ()` [Condition]

The Character List Cycle compliance error.

It signals that a cycle was found in a list of ascending character sizes.

Package [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

Direct superclasses

[tfm-compliance-error], page 70, (condition)

Direct methods

- value (method)
- [value], page 57, (method)

Direct slots

value [Slot]

The cyclic character list.

Initargs :value

Readers [value], page 56, (generic function)

Writers [(setf value)], page 56, (generic function)

`different-fonts ()` [Condition]

The Different Fonts usage error.

It signals an attempt at retrieving a ligature or kerning for two characters from different fonts.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

Direct superclasses

[tfm-usage-error], page 71, (condition)

Direct methods

- character2 (method)
- [character2], page 41, (method)
- character1 (method)
- [character1], page 41, (method)

Direct slots

character1 [Slot]

The first character.

Initargs :character1

Readers [character1], page 41, (generic function)

Writers [(setf character1)], page 41, (generic function)

character2 [Slot]

The second character.

Initargs :character2

Readers [character2], page 41, (generic function)

Writers [(setf character2)], page 41, (generic function)

- extended-tfm ()** [Condition]
 The Extended TFM warning.
 It signals that a file contains extended TFM data (OFM or JFM) rather than plain TFM data.
- Package** [net.didierverna.tfm], page 21,
Source [file.lisp], page 16, (file)
- Direct superclasses**
 [tfm-warning], page 72, (condition)
- Direct methods**
- file (method)
 - [file], page 46, (method)
 - value (method)
 - [value], page 56, (method)
- Direct slots**
- value** [Slot]
 The TFM extension.
- Initargs** :value
Readers [value], page 56, (generic function)
Writers [(setf value)], page 56, (generic function)
- file** [Slot]
 The extended TFM file.
- Initargs** :file
Readers [file], page 46, (generic function)
Writers [(setf file)], page 46, (generic function)
- file-overflow ()** [Condition]
 The File Overflow compliance warning.
 It signals that the file size is longer than expected.
- Package** [net.didierverna.tfm], page 21,
Source [file.lisp], page 16, (file)
- Direct superclasses**
- [tfm-compliance-warning], page 71, (condition)
 - [file-size-mixin], page 97, (condition)
- file-underflow ()** [Condition]
 The File Underflow compliance error.
 It signals that the file size is shorter than expected.
- Package** [net.didierverna.tfm], page 21,
Source [file.lisp], page 16, (file)
- Direct superclasses**
- [tfm-compliance-error], page 70, (condition)
 - [file-size-mixin], page 97, (condition)

- fix-word-overflow ()** [Condition]
- The Fix Word Overflow compliance error.
It signals that a fix word is outside `]-16,+16[`.
- Package** `[net.didierverna.tfm]`, page 21,
Source `[util.lisp]`, page 9, (file)
- Direct superclasses**
`[tfm-compliance-error]`, page 70, (condition)
- Direct methods**
- `value` (method)
 - `[value]`, page 58, (method)
- Direct slots**
- value** [Slot]
The faulty value.
- Initargs** `:value`
- Readers** `[value]`, page 56, (generic function)
- Writers** `[(setf value)]`, page 56, (generic function)
- invalid-bcpl-string ()** [Condition]
- The Invalid BCPL String compliance error.
It signals that a BCPL string contains parentheses or non-ASCII characters.
- Package** `[net.didierverna.tfm]`, page 21,
Source `[util.lisp]`, page 9, (file)
- Direct superclasses**
`[tfm-compliance-error]`, page 70, (condition)
- Direct methods**
- `value` (method)
 - `[value]`, page 58, (method)
- Direct slots**
- value** [Slot]
The invalid string.
- Initargs** `:value`
- Readers** `[value]`, page 56, (generic function)
- Writers** `[(setf value)]`, page 56, (generic function)
- invalid-char-info ()** [Condition]
- The Invalid Char Info compliance error.
It signals that a char-info with a width-index of 0 is not completely zero'ed out.
- Package** `[net.didierverna.tfm]`, page 21,
Source `[file.lisp]`, page 16, (file)
- Direct superclasses**
`[tfm-compliance-error]`, page 70, (condition)
- Direct methods**
- `value` (method)

- [value], page 57, (method)

Direct slots

value [Slot]
 The invalid char-info structure.

Initargs :value

Readers [value], page 56, (generic function)

Writers [(setf value)], page 56, (generic function)

invalid-character-code () [Condition]

The Invalid Character Code compliance error.

It signals a reference to a character code which does not exist in the font being loaded.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

Direct superclasses

[tfm-compliance-error], page 70, (condition)

Direct methods

- value (method)
- [value], page 57, (method)

Direct slots

value [Slot]
 The invalid character code.

Initargs :value

Readers [value], page 56, (generic function)

Writers [(setf value)], page 56, (generic function)

invalid-character-range () [Condition]

The Invalid Character Range compliance error. It signals that $BC-1 > EC$, or that $EC > 255$.

Package [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

Direct superclasses

[tfm-compliance-error], page 70, (condition)

Direct methods

- ec (method)
- [ec], page 44, (method)
- bc (method)
- [bc], page 38, (method)

Direct slots

bc [Slot]
 The smallest character code.

Initargs :bc

Readers [bc], page 38, (generic function)

Writers [(setf bc)], page 38, (generic function)

- ec** [Slot]
 The largest character code.
- Initargs** :ec
Readers [ec], page 44, (generic function)
Writers [(setf ec)], page 44, (generic function)
- invalid-design-size ()** [Condition]
 The Invalid Design Size compliance error.
 It signals that a design size is too small (< 1pt).
- Package** [net.didierverna.tfm], page 21,
Source [file.lisp], page 16, (file)
- Direct superclasses**
 [tfm-compliance-error], page 70, (condition)
- Direct methods**
- value (method)
 - [value], page 57, (method)
- Direct slots**
- value** [Slot]
 The invalid design size.
- Initargs** :value
Readers [value], page 56, (generic function)
Writers [(setf value)], page 56, (generic function)
- invalid-header-length ()** [Condition]
 The Invalid Header Length compliance error.
 It signals that a header length is too small (< 2 words).
- Package** [net.didierverna.tfm], page 21,
Source [file.lisp], page 16, (file)
- Direct superclasses**
 [tfm-compliance-error], page 70, (condition)
- Direct methods**
- value (method)
 - [value], page 57, (method)
- Direct slots**
- value** [Slot]
 The invalid header length.
- Initargs** :value
Readers [value], page 56, (generic function)
Writers [(setf value)], page 56, (generic function)
- invalid-ligature-opcode ()** [Condition]
 The Invalid Ligature Opcode compliance error. It signals that a ligature opcode is invalid.
- Package** [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

Direct superclasses

[tfm-compliance-error], page 70, (condition)

Direct methods

- value (method)
- [value], page 57, (method)

Direct slots

value [Slot]

The invalid ligature opcode.

Initargs :value

Readers [value], page 56, (generic function)

Writers [(setf value)], page 56, (generic function)

invalid-section-lengths () [Condition]

The Section Lengths compliance error.

It signals that $LF \neq 6 + LH + NC + NW + NH + ND + NI + NL + NK + NE + NP$.

Package [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

Direct superclasses

[tfm-compliance-error], page 70, (condition)

Direct methods

- np (method)
- [np], page 52, (method)
- ne (method)
- [ne], page 50, (method)
- nk (method)
- [nk], page 51, (method)
- nl (method)
- [nl], page 51, (method)
- ni (method)
- [ni], page 51, (method)
- nd (method)
- [nd], page 50, (method)
- nh (method)
- [nh], page 51, (method)
- nw (method)
- [nw], page 52, (method)
- nc (method)
- [nc], page 50, (method)
- lh (method)
- [lh], page 49, (method)
- lf (method)

- [lf], page 48, (method)

Direct slots

| | | |
|-----------------|---|--------|
| lf | | [Slot] |
| | The declared length of the file. | |
| Initargs | :lf | |
| Readers | [lf], page 48, (generic function) | |
| Writers | [(setf lf)], page 48, (generic function) | |
| lh | | [Slot] |
| | The declared length of the file header. | |
| Initargs | :lh | |
| Readers | [lh], page 49, (generic function) | |
| Writers | [(setf lh)], page 49, (generic function) | |
| nc | | [Slot] |
| | $EC - BC + 1$. | |
| Initargs | :nc | |
| Readers | [nc], page 50, (generic function) | |
| Writers | [(setf nc)], page 50, (generic function) | |
| nw | | [Slot] |
| | The declared length of the width table. | |
| Initargs | :nw | |
| Readers | [nw], page 52, (generic function) | |
| Writers | [(setf nw)], page 52, (generic function) | |
| nh | | [Slot] |
| | The declared length of the height table. | |
| Initargs | :nh | |
| Readers | [nh], page 51, (generic function) | |
| Writers | [(setf nh)], page 51, (generic function) | |
| nd | | [Slot] |
| | The declared length of the depth table. | |
| Initargs | :nd | |
| Readers | [nd], page 50, (generic function) | |
| Writers | [(setf nd)], page 50, (generic function) | |
| ni | | [Slot] |
| | The declared length of the italic correction table. | |
| Initargs | :ni | |
| Readers | [ni], page 51, (generic function) | |
| Writers | [(setf ni)], page 51, (generic function) | |

| | | |
|---------------------------------|--|-------------|
| nl | | [Slot] |
| | The declared length of the lig/kern table. | |
| Initargs | :nl | |
| Readers | [nl], page 51, (generic function) | |
| Writers | [(setf nl)], page 51, (generic function) | |
| nk | | [Slot] |
| | The declared length of the kern table. | |
| Initargs | :nk | |
| Readers | [nk], page 51, (generic function) | |
| Writers | [(setf nk)], page 51, (generic function) | |
| ne | | [Slot] |
| | The declared length of the extensible character table. | |
| Initargs | :ne | |
| Readers | [ne], page 50, (generic function) | |
| Writers | [(setf ne)], page 50, (generic function) | |
| np | | [Slot] |
| | The declared length of the parameters section. | |
| Initargs | :np | |
| Readers | [np], page 51, (generic function) | |
| Writers | [(setf np)], page 51, (generic function) | |
| invalid-string-length () | | [Condition] |
| | The Invalid String Length compliance error. | |
| | It signals that the declared length of a padded string is greater than its maximum. | |
| Package | [net.didierverna.tfm], page 21, | |
| Source | [util.lisp], page 9, (file) | |
| Direct superclasses | [tfm-compliance-error], page 70, (condition) | |
| Direct methods | | |
| | <ul style="list-style-type: none"> padding (method) [padding], page 53, (method) value (method) [value], page 58, (method) | |
| Direct slots | | |
| value | | [Slot] |
| | The invalid length. | |
| Initargs | :value | |
| Readers | [value], page 56, (generic function) | |
| Writers | [(setf value)], page 56, (generic function) | |

- padding** [Slot]
 The maximum length.
- Initargs** :padding
- Readers** [padding], page 53, (generic function)
- Writers** [(setf padding)], page 53, (generic function)
- invalid-table-index ()** [Condition]
 The Invalid Table Index compliance error.
 It signals that a table index is greater than its largest value.
- Package** [net.didierverna.tfm], page 21,
- Source** [file.lisp], page 16, (file)
- Direct superclasses**
 [tfm-table-error], page 98, (condition)
- Direct methods**
- largest (method)
 - [largest], page 48, (method)
 - value (method)
 - [value], page 57, (method)
- Direct slots**
- value** [Slot]
 The invalid index.
- Initargs** :value
- Readers** [value], page 56, (generic function)
- Writers** [(setf value)], page 56, (generic function)
- largest** [Slot]
 The largest index.
- Initargs** :largest
- Readers** [largest], page 48, (generic function)
- Writers** [(setf largest)], page 48, (generic function)
- invalid-table-length ()** [Condition]
 The Invalid Table Length compliance error.
 It signals that a declared TFM table's length is out of range.
- Package** [net.didierverna.tfm], page 21,
- Source** [file.lisp], page 16, (file)
- Direct superclasses**
 [tfm-table-error], page 98, (condition)
- Direct methods**
- largest (method)
 - [largest], page 48, (method)
 - smallest (method)
 - [smallest], page 54, (method)

- `value` (method)
- `[value]`, page 57, (method)

Direct slots

`value` [Slot]
The invalid table length.

Initargs :`value`

Readers `[value]`, page 56, (generic function)

Writers `[(setf value)]`, page 56, (generic function)

`smallest` [Slot]
The smallest table length.

Initargs :`smallest`

Readers `[smallest]`, page 54, (generic function)

Writers `[(setf smallest)]`, page 54, (generic function)

`largest` [Slot]
The largest table length.

Initargs :`largest`

Readers `[largest]`, page 48, (generic function)

Writers `[(setf largest)]`, page 48, (generic function)

`invalid-table-start ()` [Condition]

The Invalid Table Start compliance error.
It signals that the first value in a TFM table is not 0.

Package `[net.didierverna.tfm]`, page 21,

Source `[file.lisp]`, page 16, (file)

Direct superclasses

`[tfm-table-error]`, page 98, (condition)

Direct methods

- `value` (method)
- `[value]`, page 57, (method)

Direct slots

`value` [Slot]
The invalid first table value.

Initargs :`value`

Readers `[value]`, page 56, (generic function)

Writers `[(setf value)]`, page 56, (generic function)

`ligature-cycle ()` [Condition]

The Ligature Cycle compliance error.
It signals that a ligature introduces a cycle for a cons of characters.

Package `[net.didierverna.tfm]`, page 21,

Source `[file.lisp]`, page 16, (file)

Direct superclasses

[`tfm-compliance-error`], page 70, (condition)

Direct methods

- `characters` (method)
- [`characters`], page 41, (method)
- `value` (method)
- [`value`], page 57, (method)

Direct slots

`value` [Slot]

The ligature introducing a cycle.

Initargs :`value`

Readers [`value`], page 56, (generic function)

Writers [(`setf value`)], page 56, (generic function)

`characters` [Slot]

The cons of characters involved in the ligature.

Initargs :`characters`

Readers [`characters`], page 41, (generic function)

Writers [(`setf characters`)], page 41, (generic function)

`no-boundary-character` () [Condition]

The No Boundary Character compliance error.

It signals that a boundary character ligature/kerning program was found, without a boundary character being defined.

Package [`net.didierverna.tfm`], page 21,

Source [`file.lisp`], page 16, (file)

Direct superclasses

[`tfm-compliance-error`], page 70, (condition)

`not-extensible` () [Condition]

The Not Extensible usage error.

It signals an attempt at accessing the extension recipe of a non extensible character.

Package [`net.didierverna.tfm`], page 21,

Source [`character.lisp`], page 11, (file)

Direct superclasses

[`tfm-usage-error`], page 71, (condition)

Direct methods

- `value` (method)
- [`value`], page 58, (method)

Direct slots

`value` [Slot]

The non extensible character.

Initargs :`value`

Readers [`value`], page 56, (generic function)

Writers [(`setf value`)], page 56, (generic function)

- tfm ()** [Condition]
 The TFM root condition.
- Package** [net.didierverna.tfm], page 21,
Source [util.lisp], page 9, (file)
- Direct superclasses**
 condition (condition)
- Direct subclasses**
- [tfm-warning], page 72, (condition)
 - [tfm-error], page 71, (condition)
 - [tfm-compliance], page 70, (condition)
 - [tfm-usage], page 71, (condition)
- tfm-compliance ()** [Condition]
 The TFM Compliance root condition.
 This is the mixin for conditions related to TFM compliance.
- Package** [net.didierverna.tfm], page 21,
Source [util.lisp], page 9, (file)
- Direct superclasses**
 [tfm], page 70, (condition)
- Direct subclasses**
- [tfm-compliance-warning], page 71, (condition)
 - [tfm-compliance-error], page 70, (condition)
- tfm-compliance-error ()** [Condition]
 The TFM compliance errors root condition.
 This is the root condition for errors related to TFM compliance.
- Package** [net.didierverna.tfm], page 21,
Source [util.lisp], page 9, (file)
- Direct superclasses**
- [tfm-compliance], page 70, (condition)
 - [tfm-error], page 71, (condition)
- Direct subclasses**
- [u16-overflow], page 72, (condition)
 - [fix-word-overflow], page 61, (condition)
 - [invalid-string-length], page 66, (condition)
 - [invalid-bcpl-string], page 61, (condition)
 - [invalid-character-code], page 62, (condition)
 - [invalid-design-size], page 63, (condition)
 - [tfm-table-error], page 98, (condition)
 - [invalid-ligature-opcode], page 63, (condition)
 - [invalid-char-info], page 61, (condition)
 - [no-boundary-character], page 69, (condition)
 - [character-list-cycle], page 59, (condition)
 - [ligature-cycle], page 68, (condition)

- [file-underflow], page 60, (condition)
- [invalid-header-length], page 63, (condition)
- [invalid-character-range], page 62, (condition)
- [invalid-section-lengths], page 64, (condition)

tfm-compliance-warning () [Condition]

The TFM compliance warnings root condition.

This is the root condition for warnings related to TFM compliance.

Package [net.didierverna.tfm], page 21,

Source [util.lisp], page 9, (file)

Direct superclasses

- [tfm-compliance], page 70, (condition)
- [tfm-warning], page 72, (condition)

Direct subclasses

- [invalid-original-design-size], page 97, (condition)
- [file-overflow], page 60, (condition)

tfm-error () [Condition]

The TFM errors root condition.

Package [net.didierverna.tfm], page 21,

Source [util.lisp], page 9, (file)

Direct superclasses

- error (condition)
- [tfm], page 70, (condition)

Direct subclasses

- [tfm-compliance-error], page 70, (condition)
- [tfm-usage-error], page 71, (condition)

tfm-usage () [Condition]

The TFM Usage root condition.

This is the mixin for conditions related to the use of the library.

Package [net.didierverna.tfm], page 21,

Source [util.lisp], page 9, (file)

Direct superclasses

[tfm], page 70, (condition)

Direct subclasses

- [tfm-usage-warning], page 72, (condition)
- [tfm-usage-error], page 71, (condition)

tfm-usage-error () [Condition]

The TFM usage errors root condition.

This is the root condition for errors related to the use of the library.

Package [net.didierverna.tfm], page 21,

Source [util.lisp], page 9, (file)

Direct superclasses

- [tfm-usage], page 71, (condition)
- [tfm-error], page 71, (condition)

Direct subclasses

- [not-extensible], page 69, (condition)
- [anonymous-font], page 97, (condition)
- [different-fonts], page 59, (condition)

tfm-usage-warning () [Condition]

The TFM usage warnings root condition.

This is the root condition for warnings related to the use of the library.

Package [net.didierverna.tfm], page 21,

Source [util.lisp], page 9, (file)

Direct superclasses

- [tfm-usage], page 71, (condition)
- [tfm-warning], page 72, (condition)

tfm-warning () [Condition]

The TFM warnings root condition.

Package [net.didierverna.tfm], page 21,

Source [util.lisp], page 9, (file)

Direct superclasses

- warning (condition)
- [tfm], page 70, (condition)

Direct subclasses

- [tfm-compliance-warning], page 71, (condition)
- [tfm-usage-warning], page 72, (condition)
- [extended-tfm], page 60, (condition)

u16-overflow () [Condition]

The U16 Overflow compliance error.

It signals that an unsigned 16 bits integer is greater than 2^{15} .

Package [net.didierverna.tfm], page 21,

Source [util.lisp], page 9, (file)

Direct superclasses

[tfm-compliance-error], page 70, (condition)

Direct methods

- value (method)
- [value], page 58, (method)

Direct slots

value [Slot]

The faulty value.

Initargs :value

Readers [value], page 56, (generic function)

Writers [(setf value)], page 56, (generic function)

6.1.5 Classes

`character-metrics` () [Class]

The Character Metrics class.

This class represents decoded character information. Within the context of this library, the term "character" denotes an instance of this class.

Package `[net.didierverna.tfm]`, page 21,

Source `[character.lisp]`, page 11, (file)

Direct superclasses

`standard-object` (class)

Direct methods

- `[repeated-character]`, page 53, (method)
- `[bottom-character]`, page 40, (method)
- `[middle-character]`, page 49, (method)
- `[top-character]`, page 56, (method)
- `print-object` (method)
- `extension-recipe` (method)
- `[extension-recipe]`, page 95, (method)
- `next-character` (method)
- `[next-character]`, page 50, (method)
- `italic-correction` (method)
- `[italic-correction]`, page 48, (method)
- `depth` (method)
- `[depth]`, page 44, (method)
- `height` (method)
- `[height]`, page 47, (method)
- `width` (method)
- `[width]`, page 58, (method)
- `[font]`, page 47, (method)
- `[code]`, page 42, (method)

Direct slots

`code` [Slot]

The character's numerical code.

Initargs `:code`

Readers `[code]`, page 41, (generic function)

`font` [Slot]

The character's font.

Initargs `:font`

Readers `[font]`, page 46, (generic function)

`width` [Slot]

The character's width.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initargs `:width`

Readers [width], page 58, (generic function)
Writers [(setf width)], page 58, (generic function)

height [Slot]

The character's height.
 It is expressed in design size units, or in TeX point units if the font is frozen.

Initargs :height
Readers [height], page 47, (generic function)
Writers [(setf height)], page 47, (generic function)

depth [Slot]

The character's depth.
 It is expressed in design size units, or in TeX point units if the font is frozen.

Initargs :depth
Readers [depth], page 44, (generic function)
Writers [(setf depth)], page 44, (generic function)

italic-correction [Slot]

The character's italic correction.
 TeX uses this value for regular characters followed by the command /, and also in math mode for superscript placement. It is expressed in design size units, or in TeX point units if the font is frozen.

Initargs :italic-correction
Readers [italic-correction], page 48, (generic function)
Writers [(setf italic-correction)], page 48, (generic function)

next-character [Slot]

The next character in a character list.
 This slot is non-null only if the character is part of a chain of characters of ascending size, and not the last one (see TeX: the Program [544]). It is mutually exclusive with the EXTENSION-RECIPE slot, and also with the existence of a ligature or kerning program for this character.

Readers [next-character], page 50, (generic function)
Writers [(setf next-character)], page 50, (generic function)

extension-recipe [Slot]

The character's extension recipe, or NIL.
 This slot is non-null only if this character is extensible (see TeX: the Program [544]). It is mutually exclusive with the NEXT-CHARACTER slot, and also with the existence of a ligature or kerning program for this character.

Readers [extension-recipe], page 95, (generic function)
Writers [(setf extension-recipe)], page 95, (generic function)

font () [Class]

The TeX Font Metrics class.
 This class represents decoded font information. Within the context of this library, the term "font" denotes an instance of this class, or of one of its subclasses.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

Direct superclasses

standard-object (class)

Direct subclasses

- [math-symbols-font], page 82, (class)
- [math-extension-font], page 81, (class)

Direct methods

- initialize-instance (method)
- print-object (method)
- boundary-character (method)
- [boundary-character], page 40, (method)
- kernings (method)
- [kernings], page 95, (method)
- ligatures (method)
- [ligatures], page 96, (method)
- character-count (method)
- [character-count], page 40, (method)
- characters (method)
- [characters], page 41, (method)
- max-code (method)
- [max-code], page 49, (method)
- min-code (method)
- [min-code], page 49, (method)
- parameters (method)
- [parameters], page 53, (method)
- extra-space (method)
- [extra-space], page 45, (method)
- em (method)
- [em], page 45, (method)
- ex (method)
- [ex], page 45, (method)
- interword-shrink (method)
- [interword-shrink], page 47, (method)
- interword-stretch (method)
- [interword-stretch], page 48, (method)
- interword-space (method)
- [interword-space], page 47, (method)
- slant (method)
- [slant], page 54, (method)
- face-code (method)
- [face-code], page 46, (method)
- expansion (method)
- [expansion], page 45, (method)

- `slope` (method)
- `[slope]`, page 54, (method)
- `weight` (method)
- `[weight]`, page 58, (method)
- `face-number` (method)
- `[face-number]`, page 46, (method)
- `7bits-safe` (method)
- `[7bits-safe]`, page 38, (method)
- `family` (method)
- `[family]`, page 46, (method)
- `encoding` (method)
- `[encoding]`, page 45, (method)
- `original-design-size` (method)
- `[original-design-size]`, page 53, (method)
- `design-size` (method)
- `[design-size]`, page 44, (method)
- `frozen` (method)
- `[frozen]`, page 47, (method)
- `checksum` (method)
- `[checksum]`, page 41, (method)
- `file` (method)
- `[file]`, page 46, (method)
- `name` (method)
- `[name]`, page 50, (method)

Direct slots

- name** [Slot]
- The font's name.
- When the font is loaded from a file, it defaults to the file's base name, along with potential scaling information.
- Initargs** : `name`
- Readers** `[name]`, page 49, (generic function)
- Writers** `[(setf name)]`, page 49, (generic function)
- file** [Slot]
- The file from which the font was loaded, or NIL.
- Initargs** : `file`
- Readers** `[file]`, page 46, (generic function)
- Writers** `[(setf file)]`, page 46, (generic function)
- checksum** [Slot]
- The font's checksum, as provided by Metafont.
- Readers** `[checksum]`, page 41, (generic function)
- Writers** `[(setf checksum)]`, page 41, (generic function)

| | |
|---|--|
| frozen | [Slot] |
| Whether the font is frozen. | |
| Readers | [frozen], page 47, (generic function) |
| Writers | [(setf frozen)], page 47, (generic function) |
| design-size | [Slot] |
| The font's design size, in units of TeX points. | |
| Readers | [design-size], page 44, (generic function) |
| Writers | [(setf design-size)], page 44, (generic function) |
| original-design-size | [Slot] |
| The font's original design size, in units of TeX points. | |
| Readers | [original-design-size], page 53, (generic function) |
| Writers | [(setf original-design-size)], page 53, (generic function) |
| encoding | [Slot] |
| The font's character coding scheme (a BCPL string), or NIL. | |
| Readers | [encoding], page 45, (generic function) |
| Writers | [(setf encoding)], page 45, (generic function) |
| family | [Slot] |
| The font's family (a BCPL string), or NIL. | |
| Readers | [family], page 46, (generic function) |
| Writers | [(setf family)], page 46, (generic function) |
| 7bits-safe | [Slot] |
| Whether the font is 7 bits safe (0 or 1), or NIL. | |
| When 1, it means that no character of code lesser than 128 can lead to a character of code greater than 128 by ways of ligatures or extensible recipes. | |
| Readers | [7bits-safe], page 38, (generic function) |
| Writers | [(setf 7bits-safe)], page 38, (generic function) |
| face-number | [Slot] |
| The font's face number, or NIL. | |
| Readers | [face-number], page 46, (generic function) |
| Writers | [(setf face-number)], page 46, (generic function) |
| weight | [Slot] |
| The font's weight (:medium, :bold, :light), or NIL. When available, it is decoded from the font face number. | |
| Readers | [weight], page 58, (generic function) |
| Writers | [(setf weight)], page 58, (generic function) |
| slope | [Slot] |
| The font's slope (:roman, :italic), or NIL. | |
| When available, it is decoded from the font face number. | |
| Readers | [slope], page 54, (generic function) |
| Writers | [(setf slope)], page 54, (generic function) |

- expansion** [Slot]
 The font's expansion (:regular, :condensed, :extended), or NIL. When available, it is decoded from the font face number.
Readers [expansion], page 45, (generic function)
Writers [(setf expansion)], page 45, (generic function)
- face-code** [Slot]
 The font's 3-letters face code, or NIL.
 When available, it is the concatenation of the upcased first letters of the font's weight, slope, and expansion.
Readers [face-code], page 46, (generic function)
Writers [(setf face-code)], page 46, (generic function)
- slant** [Slot]
 The font's slant (a scalar ratio).
Initform 0
Readers [slant], page 54, (generic function)
Writers [(setf slant)], page 54, (generic function)
- interword-space** [Slot]
 The font's normal interword space.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Initform 0
Readers [interword-space], page 47, (generic function)
Writers [(setf interword-space)], page 47, (generic function)
- interword-stretch** [Slot]
 The font's interword stretchability.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Initform 0
Readers [interword-stretch], page 48, (generic function)
Writers [(setf interword-stretch)], page 48, (generic function)
- interword-shrink** [Slot]
 The font's interword shrinkability.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Initform 0
Readers [interword-shrink], page 47, (generic function)
Writers [(setf interword-shrink)], page 47, (generic function)
- ex** [Slot]
 The font's ex size.
 It is expressed in design size units, or in TeX point units if the font is frozen.
Initform 0
Readers [ex], page 45, (generic function)
Writers [(setf ex)], page 45, (generic function)

- em** [Slot]
 The font's em size.
 It is expressed in design size units, or in TeX point units if the font is frozen.
- Initform** 0
Readers [em], page 44, (generic function)
Writers [(setf em)], page 44, (generic function)
- extra-space** [Slot]
 The font's extra space.
 It is expressed in design size units, or in TeX point units if the font is frozen.
- This is the additional space to put at the end of sentences.
- Initform** 0
Readers [extra-space], page 45, (generic function)
Writers [(setf extra-space)], page 45, (generic function)
- parameters** [Slot]
 The font's additional parameters array, or NIL.
 Parameters are expressed in design size units, or in TeX point units if the font is frozen.
- Readers** [parameters], page 53, (generic function)
Writers [(setf parameters)], page 53, (generic function)
- min-code** [Slot]
 The font's smallest character code.
- Readers** [min-code], page 49, (generic function)
Writers [(setf min-code)], page 49, (generic function)
- max-code** [Slot]
 The font's largest character code.
- Readers** [max-code], page 49, (generic function)
Writers [(setf max-code)], page 49, (generic function)
- characters** [Slot]
 The font's characters.
 This is a hash table associating character codes with characters.
- Initform** (make-hash-table :test (function eq))
Readers [characters], page 41, (generic function)
Writers [(setf characters)], page 41, (generic function)
- character-count** [Slot]
 The font's number of characters.
 The character count does not include the boundary character, unless that character really exists in the font (has non-zero metrics).
- Readers** [character-count], page 40, (generic function)
Writers [(setf character-count)], page 40, (generic function)

ligatures [Slot]

The font's ligatures.

This is a hash table associating conseqs of characters with the corresponding ligature.

Initform (make-hash-table :test (function equal))

Readers [ligatures], page 96, (generic function)

Writers [(setf ligatures)], page 96, (generic function)

kernings [Slot]

The font's kernings.

This is a hash table associating conseqs of characters with the corresponding kerning. They are expressed in design size units, or in TeX point units if the font is frozen.

Initform (make-hash-table :test (function equal))

Readers [kernings], page 95, (generic function)

Writers [(setf kernings)], page 95, (generic function)

boundary-character [Slot]

The font's boundary character, or NIL.

This character is also accessible by code, like normal ones. However, it is the only character the code of which may be outside [MIN-CODE,MAX-CODE] (see TeX: the Program [545]). Finally, this character is not included in the character count, unless it exists for real in the font (has non-zero metrics).

Readers [boundary-character], page 40, (generic function)

Writers [(setf boundary-character)], page 40, (generic function)

ligature () [Class]

The Ligature class.

This class represents a decoded ligature program. Within the context of this library, the term "ligature" denotes an instance of this class.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

Direct superclasses

standard-object (class)

Direct methods

- [pass-over], page 53, (method)
- [delete-after], page 42, (method)
- [delete-before], page 42, (method)
- [composite], page 42, (method)

Direct slots**composite** [Slot]

The character to insert between the two original ones.

Initargs :composite

Readers [composite], page 42, (generic function)

| | |
|--|---|
| <code>delete-before</code> | [Slot] |
| Whether to delete the character before the ligature. | |
| Initargs | <code>:delete-before</code> |
| Readers | <code>[delete-before]</code> , page 42, (generic function) |
| <code>delete-after</code> | [Slot] |
| Whether to delete the character after the ligature. | |
| Initargs | <code>:delete-after</code> |
| Readers | <code>[delete-after]</code> , page 42, (generic function) |
| <code>pass-over</code> | [Slot] |
| The number of characters to skip for reaching the next character. | |
| Initargs | <code>:pass-over</code> |
| Readers | <code>[pass-over]</code> , page 53, (generic function) |
| <code>math-extension-font ()</code> | [Class] |
| The Math Extension Font class. | |
| This class represents fonts with the "TeX math extension" character coding scheme. | |
| Package | <code>[net.didierverna.tfm]</code> , page 21, |
| Source | <code>[font.lisp]</code> , page 12, (file) |
| Direct superclasses | |
| | <code>[font]</code> , page 74, (class) |
| Direct methods | |
| | <ul style="list-style-type: none"> • <code>[parse-parameters]</code>, page 96, (method) • <code>[scale]</code>, page 96, (method) • <code>big-op-spacing5</code> (method) • <code>[big-op-spacing5]</code>, page 40, (method) • <code>big-op-spacing4</code> (method) • <code>[big-op-spacing4]</code>, page 39, (method) • <code>big-op-spacing3</code> (method) • <code>[big-op-spacing3]</code>, page 39, (method) • <code>big-op-spacing2</code> (method) • <code>[big-op-spacing2]</code>, page 39, (method) • <code>big-op-spacing1</code> (method) • <code>[big-op-spacing1]</code>, page 39, (method) • <code>default-rule-thickness</code> (method) • <code>[default-rule-thickness]</code>, page 42, (method) |
| Direct slots | |
| <code>default-rule-thickness</code> | [Slot] |
| The font's default rule thickness. | |
| It is expressed in design size units, or in TeX point units if the font is frozen. | |
| Initform | 0 |
| Readers | <code>[default-rule-thickness]</code> , page 42, (generic function) |
| Writers | <code>[(setf default-rule-thickness)]</code> , page 42, (generic function) |

big-op-spacing1 [Slot]

The font's BIG-OP-SPACING1 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [big-op-spacing1], page 38, (generic function)

Writers [(setf big-op-spacing1)], page 38, (generic function)

big-op-spacing2 [Slot]

The font's BIG-OP-SPACING2 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [big-op-spacing2], page 39, (generic function)

Writers [(setf big-op-spacing2)], page 39, (generic function)

big-op-spacing3 [Slot]

The font's BIG-OP-SPACING3 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [big-op-spacing3], page 39, (generic function)

Writers [(setf big-op-spacing3)], page 39, (generic function)

big-op-spacing4 [Slot]

The font's BIG-OP-SPACING4 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [big-op-spacing4], page 39, (generic function)

Writers [(setf big-op-spacing4)], page 39, (generic function)

big-op-spacing5 [Slot]

The font's BIG-OP-SPACING5 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [big-op-spacing5], page 39, (generic function)

Writers [(setf big-op-spacing5)], page 39, (generic function)

math-symbols-font () [Class]

The Math Symbols Font class.

This class represents fonts with the "TeX math symbols" character coding scheme.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

Direct superclasses

[font], page 74, (class)

Direct methods

- [parse-parameters], page 96, (method)
- [scale], page 96, (method)

- `axis-height` (method)
- `[axis-height]`, page 38, (method)
- `delim2` (method)
- `[delim2]`, page 43, (method)
- `delim1` (method)
- `[delim1]`, page 43, (method)
- `subdrop` (method)
- `[subdrop]`, page 55, (method)
- `supdrop` (method)
- `[supdrop]`, page 56, (method)
- `sub2` (method)
- `[sub2]`, page 55, (method)
- `sub1` (method)
- `[sub1]`, page 54, (method)
- `sup3` (method)
- `[sup3]`, page 56, (method)
- `sup2` (method)
- `[sup2]`, page 55, (method)
- `sup1` (method)
- `[sup1]`, page 55, (method)
- `denom2` (method)
- `[denom2]`, page 43, (method)
- `denom1` (method)
- `[denom1]`, page 43, (method)
- `num3` (method)
- `[num3]`, page 52, (method)
- `num2` (method)
- `[num2]`, page 52, (method)
- `num1` (method)
- `[num1]`, page 52, (method)

Direct slots

- `num1` [Slot]
- The font's NUM1 parameter.
It is expressed in design size units, or in TeX point units if the font is frozen.
- Initform** 0
- Readers** `[num1]`, page 52, (generic function)
- Writers** `[(setf num1)]`, page 52, (generic function)
- `num2` [Slot]
- The font's NUM2 parameter.
It is expressed in design size units, or in TeX point units if the font is frozen.
- Initform** 0
- Readers** `[num2]`, page 52, (generic function)
- Writers** `[(setf num2)]`, page 52, (generic function)

- num3** [Slot]
 The font's NUM2 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
- Initform** 0
Readers [num3], page 52, (generic function)
Writers [(setf num3)], page 52, (generic function)
- denom1** [Slot]
 The font's DENOM1 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
- Initform** 0
Readers [denom1], page 43, (generic function)
Writers [(setf denom1)], page 43, (generic function)
- denom2** [Slot]
 The font's DENOM2 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
- Initform** 0
Readers [denom2], page 43, (generic function)
Writers [(setf denom2)], page 43, (generic function)
- sup1** [Slot]
 The font's SUP1 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
- Initform** 0
Readers [sup1], page 55, (generic function)
Writers [(setf sup1)], page 55, (generic function)
- sup2** [Slot]
 The font's SUP2 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
- Initform** 0
Readers [sup2], page 55, (generic function)
Writers [(setf sup2)], page 55, (generic function)
- sup3** [Slot]
 The font's SUP2 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
- Initform** 0
Readers [sup3], page 56, (generic function)
Writers [(setf sup3)], page 56, (generic function)
- sub1** [Slot]
 The font's SUB1 parameter.
 It is expressed in design size units, or in TeX point units if the font is frozen.
- Initform** 0

Readers [sub1], page 54, (generic function)

Writers [(setf sub1)], page 54, (generic function)

sub2 [Slot]

The font's SUB2 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [sub2], page 55, (generic function)

Writers [(setf sub2)], page 55, (generic function)

supdrop [Slot]

The font's SUPDROP parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [supdrop], page 56, (generic function)

Writers [(setf supdrop)], page 56, (generic function)

subdrop [Slot]

The font's SUBDROP parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [subdrop], page 55, (generic function)

Writers [(setf subdrop)], page 55, (generic function)

delim1 [Slot]

The font's DELIM1 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [delim1], page 43, (generic function)

Writers [(setf delim1)], page 43, (generic function)

delim2 [Slot]

The font's DELIM2 parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [delim2], page 43, (generic function)

Writers [(setf delim2)], page 43, (generic function)

axis-height [Slot]

The font's AXIS-HEIGHT parameter.

It is expressed in design size units, or in TeX point units if the font is frozen.

Initform 0

Readers [axis-height], page 38, (generic function)

Writers [(setf axis-height)], page 38, (generic function)

6.2 Internal definitions

6.2.1 Constants

+character-metrics-dimension-accessors+ [Constant]
The list of dimension accessor names in the CHARACTER-METRICS class.

Package [net.didierverna.tfm], page 21,

Source [character.lisp], page 11, (file)

+font-dimension-accessors+ [Constant]
The list of dimension accessor names in the FONT class.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

+math-extension-font-dimension-accessors+ [Constant]
The list of dimension accessor names in the MATH-EXTENSION-FONT class.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

+math-symbols-font-dimension-accessors+ [Constant]
The list of dimension accessor names in the MATH-SYMBOLS-FONT class.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

6.2.2 Special variables

configuration [Special Variable]
The TFM configuration settings.
This variable contains a property list of configuration options. Current options are:
- :swank-eval-in-emacs (Boolean)

See Chapter 5 of the user manual for more information.

Package [net.didierverna.tfm.setup], page 33,

Source [configuration.lisp], page 19, (file)

stream [Special Variable]
The stream being read.

Package [net.didierverna.tfm], page 21,

Source [util.lisp], page 9, (file)

6.2.3 Macros

defindent *SYMBOL INDENT* [Macro]
Wrapper around ‘clindent’ to avoid quoting SYMBOL and INDENT.

Package [net.didierverna.tfm.setup], page 33,

Source [readtable.lisp], page 19, (file)

- define-constant** *NAME VALUE &optional DOCUMENTATION* [Macro]
Like DEFCONSTANT, but reuse existing value if any.
- Package** [net.didierverna.tfm], page 21,
Source [util.lisp], page 9, (file)
- define-extension-recipe-pseudo-accessor** *NAME* [Macro]
Package [net.didierverna.tfm], page 21,
Source [character.lisp], page 11, (file)
- map-accessors** *VAR OBJECT ACCESSORS &body BODY* [Macro]
Map BODY on OBJECT ACCESSORS, each in turn available as VAR.
- Package** [net.didierverna.tfm], page 21,
Source [util.lisp], page 9, (file)
- map-character-metrics-dimension-accessors** *VAR CHARACTER &body BODY* [Macro]
Map BODY on CHARACTER metrics dimension accessors available as VAR.
- Package** [net.didierverna.tfm], page 21,
Source [character.lisp], page 11, (file)
- map-font-dimension-accessors** *VAR FONT &body BODY* [Macro]
Map BODY on FONT dimension accessors available as VAR.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
- map-math-extension-font-dimension-accessors** *VAR FONT &body BODY* [Macro]
Map BODY on math extension FONT dimension accessors available as VAR.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
- map-math-symbols-font-dimension-accessors** *VAR FONT &body BODY* [Macro]
Map BODY on FONT dimension accessors available as VAR.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
- tref** *TABLE INDEX* [Macro]
Call TABLE-AREF, computing the table name from TABLE.
- Package** [net.didierverna.tfm], page 21,
Source [file.lisp], page 16, (file)

6.2.4 Functions

`%run-ligature/kerning-program` *CHARACTER INDEX LIG/KERNS* [Function]
KERNS &aux FONT

Run a ligature/kerning program for CHARACTER.

The program starts at LIG/KERNS[INDEX] and uses the KERNS array. Running the program eventually creates ligatures or kernings for CHARACTER and some other character.

If an invalid index into LIG/KERNS is encountered, signal an INVALID-TABLE-INDEX error. This error is immediately restartable with ABORT-LIG/KERN-PROGRAM.

If an invalid ligature opcode is encountered, signal an INVALID-LIGATURE-OPCODE error. This error is immediately restartable with DISCARD-LIGATURE.

If an invalid index into KERNS is encountered, signal an INVALID-TABLE-INDEX error. This error is immediately restartable with DISCARD-KERNING.

Finally, if an invalid character code is encountered, signal an INVALID-CHARACTER-CODE error. Depending on the context, this error is immediately restartable with DISCARD-LIGATURE or DISCARD-KERNING.

Package [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

`%version` *TYPE MAJOR MINOR STATUS LEVEL NAME* [Function]

Package [net.didierverna.tfm.setup], page 33,

Source [version.lisp], page 18, (file)

`apply-ligature` *LIGATURE STATE &aux NEW-STATE* [Function]

Apply LIGATURE to STATE and return the new state.

STATE is a list of characters, the first two being subject to LIGATURE.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

`bot` *INSTANCE* [Function]

`(setf bot)` *VALUE INSTANCE* [Function]

Package [net.didierverna.tfm], page 21,

Source [intermediate.lisp], page 10, (file)

`char-info-p` *OBJECT* [Function]

Package [net.didierverna.tfm], page 21,

Source [intermediate.lisp], page 10, (file)

`clindent` *SYMBOL INDENT* [Function]

Send SYMBOL's INDENTation information to Emacs.

Emacs will set the 'common-lisp-indent-function property.

If INDENT is a symbol, use its indentation definition. Otherwise, INDENT is considered as an indentation definition.

Package [net.didierverna.tfm.setup], page 33,

Source [readtable.lisp], page 19, (file)

- code-character** *CODE FONT &optional ERRORP* [Function]
 Return FONT's CODE character.
 If ERRORP (the default), check that the character exists, or signal an INVALID-CHARACTER-CODE error. Note that a fake boundary character may be retrieved by this function.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
Writer [(setf code-character)], page 89, (function)
- (setf code-character)** *CHARACTER FONT* [Function]
 Make FONT's CHARACTER accessible by its code.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
Reader [code-character], page 89, (function)
- copy-char-info** *INSTANCE* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- copy-exten** *INSTANCE* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- copy-lig/kern** *INSTANCE* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- decode-char-info** *WORD* [Function]
 Decode char-info WORD into a new CHAR-INFO instance, and return it.
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- decode-exten** *WORD* [Function]
 Decode exten WORD into a new EXTEN instance, and return it.
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- decode-lig/kern** *WORD* [Function]
 Decode lig/kern WORD into a new LIG/KERN instance, and return it.
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- depth-index** *INSTANCE* [Function]
(setf depth-index) *VALUE INSTANCE* [Function]
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)

- `exten-index` *INSTANCE* [Function]
 (`setf exten-index`) *VALUE INSTANCE* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- `exten-p` *OBJECT* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- `font-extension-recipe` *EXTEN FONT &aux INITARGS* [Function]
 Make an extension recipe based on EXTEN with FONT's characters.
Package [net.didierverna.tfm], page 21,
Source [file.lisp], page 16, (file)
- `height-index` *INSTANCE* [Function]
 (`setf height-index`) *VALUE INSTANCE* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- `i-reader` *STREAM SUBCHAR ARG* [Function]
 Construct a call to 'defindent' by reading an argument list from STREAM. This dispatch macro character function is installed on #i in the NET.DIDIERVERNA.TFM named readtable.
Package [net.didierverna.tfm.setup], page 33,
Source [readtable.lisp], page 19, (file)
- `italic-index` *INSTANCE* [Function]
 (`setf italic-index`) *VALUE INSTANCE* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- `lig/kern-index` *INSTANCE* [Function]
 (`setf lig/kern-index`) *VALUE INSTANCE* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- `lig/kern-p` *OBJECT* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- `load-tfm-font` *LF &key FILE NAME DESIGN-SIZE FREEZE &aux FONT* [Function]
 Parse *STREAM* of declared length LF into a new font, and return it.
 - FILE defaults to *STREAM*'s associated file if any.
 - NAME defaults to the FILE's base name, if any.
 - If DESIGN-SIZE is provided and not a real greater or equal to 1, signal a type error. Otherwise, override the original design size with it.
 - When FREEZE (NIL by default), freeze the font immediately after creation.

If **STREAM** is shorter than expected, signal a FILE-UNDERFLOW error.
 If **STREAM** is longer than expected, signal a FILE-OVERFLOW warning.

If the declared header length is less than 2, signal an INVALID-HEADER-LENGTH error.

If BC and EC don't make sense, signal an INVALID-CHARACTER-RANGE error.

If the widths, heights, depths, italic corrections, or extens tables lengths are not within the expected range, signal an INVALID-TABLE-LENGTH error.

Finally, if the declared sections lengths don't add up to the declared file length, signal an INVALID-SECTION-LENGTHS error.

Package [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

make-char-info **&key** (*WIDTH-INDEX* **WIDTH-INDEX**) [Function]
 (*HEIGHT-INDEX* **HEIGHT-INDEX**) (*DEPTH-INDEX* **DEPTH-INDEX**)
 (*ITALIC-INDEX* **ITALIC-INDEX**) (*LIG/KERN-INDEX*
LIG/KERN-INDEX) (*NEXT-CHAR* **NEXT-CHAR**) (*EXTEN-INDEX*
EXTEN-INDEX)

Package [net.didierverna.tfm], page 21,

Source [intermediate.lisp], page 10, (file)

make-character-metrics *CODE* *FONT* *WIDTH* *HEIGHT* *DEPTH* [Function]
ITALIC-CORRECTION

Make a new CHARACTER-METRICS instance, and return it.

Initialize the character's CODE, FONT, WIDTH, HEIGHT, DEPTH, and ITALIC-CORRECTION appropriately. The two remaining slots (NEXT-CHARACTER and EXTENSION-RECIPE) will be initialized later if needed, when all character metrics instances are created.

Package [net.didierverna.tfm], page 21,

Source [character.lisp], page 11, (file)

make-exten **&key** (*TOP* **TOP**) (*MID* **MID**) (*BOT* **BOT**) (*REP* **REP**) [Function]

Package [net.didierverna.tfm], page 21,

Source [intermediate.lisp], page 10, (file)

make-extension-recipe *REPEATED-CHARACTER* **&rest** *INITARGS* [Function]
&key *TOP-CHARACTER* *MIDDLE-CHARACTER*
BOTTOM-CHARACTER

Make a new EXTENSION-RECIPE with REPEATED-CHARACTER and return it. The recipe may also have a TOP-, MIDDLE-, and BOTTOM-CHARACTER.

Package [net.didierverna.tfm], page 21,

Source [character.lisp], page 11, (file)

make-font *NAME* **&rest** *INITARGS* [Function]

Make a new NAMED FONT instance, and return it.

If INITARGS are provided, pass them as-is to MAKE-INSTANCE.

Package [net.didierverna.tfm], page 21,

- Source** [font.lisp], page 12, (file)
- make-lig/kern &key** (*SKIP SKIP*) (*NEXT NEXT*) (*OP OP*) [Function]
(REMAINDER REMAINDER)
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- make-ligature** *COMPOSITE DELETE-BEFORE DELETE-AFTER* [Function]
PASS-OVER
 Make a new LIGATURE instance, and return it.
- Package** [net.didierverna.tfm], page 21,
Source [font.lisp], page 12, (file)
- mid** *INSTANCE* [Function]
(setf mid) VALUE INSTANCE [Function]
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- next** *INSTANCE* [Function]
(setf next) VALUE INSTANCE [Function]
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- next-char** *INSTANCE* [Function]
(setf next-char) VALUE INSTANCE [Function]
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- op** *INSTANCE* [Function]
(setf op) VALUE INSTANCE [Function]
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- parse-character-information** *NC NW NH ND NI NL NK NE FONT* [Function]
 Parse the 8 character information tables from *STREAM* into FONT.
 NC (EC - BC + 1), NW, NH, ND, NI, NL, NK, and NE are the declared lengths of the 8 tables, that is, the char infos, widths, heights, depths, italic corrections, lig/kern instructions, kerns, and extens respectively.

If a char info structure with a width index of 0 is not completely zero'ed out, signal an INVALID-CHAR-INFO error. This error is immediately restartable with SET-TO-ZERO.

If the first entry in the widths, heights, depths, or italic corrections table is not 0, signal an INVALID-TABLE-START error. This error is immediately restartable with SET-TO-ZERO.

If an index into the widths, heights, depths, or italic corrections tables is invalid, signal an INVALID-TABLE-INDEX error. This error is immediately restartable with SET-TO-ZERO.

If a lig/kern program is found for a boundary character, but there is no such character in the font, signal a NO-BOUNDARY-CHARACTER error. This error is immediately restartable

with ABORT-LIG/KERN-PROGRAM.

If an index into the extens table is invalid, signal an INVALID-TABLE-INDEX error. This error is immediately restartable with DISCARD-EXTENSION-RECIPE.

If a cycle is found in a list of characters of ascending size, signal a CHARACTER-LIST-CYCLE error. This error is immediately restartable with DISCARD-NEXT-CHARACTER.

If a ligature is found to be cyclic, signal a LIGATURE-CYCLE error. This error is immediately restartable with DISCARD-LIGATURE.

Finally, if an invalid character code is encountered, signal an INVALID-CHARACTER-CODE error. Depending on the context, this error is immediately restartable with DISCARD-NEXT-CHARACTER, or DISCARD-EXTENSION-RECIPE.

Package [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

parse-header *LENGTH FONT* [Function]

Parse a header of LENGTH words from *STREAM* into FONT.

If FONT's design size is less than 1pt, signal an INVALID-DESIGN-SIZE error. This error is immediately restartable with SET-TO-TEN.

However, if FONT's design size was explicitly overridden, only signal an INVALID-ORIGINAL-DESIGN-SIZE warning.

Package [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

read-fix-word *&optional LIMIT* [Function]

Read a fix word from *STREAM* and return it.

If LIMIT (the default), check that the number lies within]-16,+16[, or signal a FIX-WORD-OVERFLOW error. This error is immediately restartable with SET-TO-ZERO.

Package [net.didierverna.tfm], page 21,

Source [util.lisp], page 9, (file)

read-padded-string *PADDING &aux LENGTH STRING* [Function]

Read a BCPL string out of PADDING bytes from *STREAM*.

The first byte in *STREAM* indicates the actual length of the string. The remaining bytes are ignored.

If the declared string length is too large, signal an INVALID-STRING-LENGTH error. This error is immediately restartable with READ-MAXIMUM-LENGTH or DISCARD-STRING.

If the string is not in BCPL format (it contains parentheses or non plain ASCII characters, signal an INVALID-BCPL-STRING error. This error is immediately restartable with KEEP-STRING, FIX-STRING (replacing parentheses with slashes, and non plain ASCII characters with question marks), or DISCARD-STRING.

Package [net.didierverna.tfm], page 21,

Source [util.lisp], page 9, (file)

- read-u16** () [Function]
 Read an unsigned 16 bits Big Endian integer from *STREAM* and return it. If $\geq 2^{15}$, signal a U16-OVERFLOW error.
- Package** [net.didierverna.tfm], page 21,
Source [util.lisp], page 9, (file)
- read-u32** () [Function]
 Read an unsigned 32 bits Big Endian integer from *STREAM*.
- Package** [net.didierverna.tfm], page 21,
Source [util.lisp], page 9, (file)
- release-status-number** *RELEASE-STATUS* [Function]
Package [net.didierverna.tfm.setup], page 33,
Source [version.lisp], page 18, (file)
- remainder** *INSTANCE* [Function]
 (setf remainder) *VALUE INSTANCE* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- rep** *INSTANCE* [Function]
 (setf rep) *VALUE INSTANCE* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- report** *STREAM FORMAT-STRING &rest FORMAT-ARGUMENTS* [Function]
 Like FORMAT, but if *STREAM* is bound, report that we're reading from it.
- Package** [net.didierverna.tfm], page 21,
Source [util.lisp], page 9, (file)
- run-ligature/kerning-program** *CHARACTER INDEX LIG/KERNS* [Function]
KERNS &aux LIG/KERN
 Find the real start of a ligature/kerning program and run it.
 See %run-ligature/kerning-program for more information.
- If INDEX is invalid, signal an INVALID-TABLE-INDEX error. This error is immediately restartable with ABORT-LIG/KERN-PROGRAM.
- Package** [net.didierverna.tfm], page 21,
Source [file.lisp], page 16, (file)
- skip** *INSTANCE* [Function]
 (setf skip) *VALUE INSTANCE* [Function]
Package [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)

- `table-aref` *NAME TABLE INDEX* [Function]
 Access NAMED *TABLE* at *INDEX*.
 If *INDEX* is out of bounds, signal an `INVALID-TABLE-INDEX` error.
- Package** [net.didierverna.tfm], page 21,
Source [file.lisp], page 16, (file)
- `top` *INSTANCE* [Function]
`(setf top)` *VALUE INSTANCE* [Function]
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- `width-index` *INSTANCE* [Function]
`(setf width-index)` *VALUE INSTANCE* [Function]
- Package** [net.didierverna.tfm], page 21,
Source [intermediate.lisp], page 10, (file)
- `~-reader` *STREAM CHAR* [Function]
 Read a series of ~"string" to be concatenated together.
- Package** [net.didierverna.tfm.setup], page 33,
Source [readtable.lisp], page 19, (file)

6.2.5 Generic functions

- `extension-recipe` *OBJECT* [Generic Function]
`(setf extension-recipe)` *NEW-VALUE OBJECT* [Generic Function]
- Package** [net.didierverna.tfm], page 21,
Methods
- `extension-recipe` (*CHARACTER-METRICS* character-metrics) [Method]
`(setf extension-recipe)` *NEW-VALUE* (*CHARACTER-METRICS* character-metrics) [Method]
- The character's extension recipe, or NIL.
 This slot is non-null only if this character is extensible (see TeX: the Program [544]). It is mutually exclusive with the `NEXT-CHARACTER` slot, and also with the existence of a ligature or kerning program for this character.
- Source** [character.lisp], page 11, (file)
- `kernings` *OBJECT* [Generic Function]
`(setf kernings)` *NEW-VALUE OBJECT* [Generic Function]
- Package** [net.didierverna.tfm], page 21,
Methods
- `kernings` (*FONT* font) [Method]
`(setf kernings)` *NEW-VALUE* (*FONT* font) [Method]
- The font's kernings.
 This is a hash table associating conseqs of characters with the corresponding kerning. They are expressed in design size units, or in TeX point units if the font is frozen.
- Source** [font.lisp], page 12, (file)

- ligatures** *OBJECT* [Generic Function]
(setf ligatures) *NEW-VALUE OBJECT* [Generic Function]
- Package** [net.didierverna.tfm], page 21,
- Methods**
- ligatures** (*FONT font*) [Method]
(setf ligatures) *NEW-VALUE (FONT font)* [Method]
 The font's ligatures.
 This is a hash table associating conses of characters with the corresponding ligature.
- Source** [font.lisp], page 12, (file)
- parse-parameters** *LENGTH FONT* [Generic Function]
 Parse a parameters section of LENGTH words from *STREAM* into FONT. Return remaining LENGTH.
- Package** [net.didierverna.tfm], page 21,
- Source** [file.lisp], page 16, (file)
- Methods**
- parse-parameters** *LENGTH FONT* [Method]
 Parse the 7 regular FONT parameters. Return remaining LENGTH.
- parse-parameters** *LENGTH (FONT math-symbols-font)* [Method]
 Parse the 15 additional TeX math symbols FONT parameters.
- parse-parameters** *LENGTH (FONT math-extension-font)* [Method]
 Parse the 6 additional TeX math extension FONT parameters.
- parse-parameters** *LENGTH FONT around* [Method]
 Read remaining parameters into a parameters array.
- scale** *FONT FACTOR* [Generic Function]
 Scale all FONT dimensions by FACTOR.
- Package** [net.didierverna.tfm], page 21,
- Source** [font.lisp], page 12, (file)
- Methods**
- scale** (*FONT math-extension-font*) *FACTOR around* [Method]
 Scaling method for MATH-EXTENSION-FONTS.
- scale** (*FONT math-symbols-font*) *FACTOR around* [Method]
 Scaling method for MATH-SYMBOL-FONTS.
- scale** *FONT FACTOR* [Method]
 Scaling method for regular FONTS.

6.2.6 Conditions

anonymous-font () [Condition]

The Anonymous Font usage error.

It signals an attempt at creating a font with no name.

Package [net.didierverna.tfm], page 21,

Source [font.lisp], page 12, (file)

Direct superclasses

[tfm-usage-error], page 71, (condition)

file-size-mixin () [Condition]

The File Size Mixin condition.

It is used in both errors and warnings to report different declared and actual file sizes.

Package [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

Direct superclasses

condition (condition)

Direct subclasses

- [file-underflow], page 60, (condition)
- [file-overflow], page 60, (condition)

Direct methods

- actual-size (method)
- [actual-size], page 38, (method)
- declared-size (method)
- [declared-size], page 42, (method)

Direct slots

declared-size [Slot]

The declared file size.

Initargs :declared-size

Readers [declared-size], page 42, (generic function)

Writers [(setf declared-size)], page 42, (generic function)

actual-size [Slot]

The actual file size.

Initargs :actual-size

Readers [actual-size], page 38, (generic function)

Writers [(setf actual-size)], page 38, (generic function)

invalid-original-design-size () [Condition]

The Invalid Original Design Size compliance warning.

It signals that, although overridden explicitly, an original design size was too small (< 1pt).

Package [net.didierverna.tfm], page 21,

Source [file.lisp], page 16, (file)

Direct superclasses

[`tfm-compliance-warning`], page 71, (condition)

Direct methods

- `value` (method)
- [`value`], page 57, (method)

Direct slots

`value` [Slot]

The invalid original design size.

Initargs :`value`

Readers [`value`], page 56, (generic function)

Writers [(`setf value`)], page 56, (generic function)

`tfm-table-error` () [Condition]

The TFM table errors root condition.

This is the root condition for errors related to TFM tables.

Package [`net.didierverna.tfm`], page 21,

Source [`file.lisp`], page 16, (file)

Direct superclasses

[`tfm-compliance-error`], page 70, (condition)

Direct subclasses

- [`invalid-table-index`], page 67, (condition)
- [`invalid-table-start`], page 68, (condition)
- [`invalid-table-length`], page 67, (condition)

Direct methods

- `name` (method)
- [`name`], page 50, (method)

Direct slots

`name` [Slot]

The table's name.

Initargs :`name`

Readers [`name`], page 49, (generic function)

Writers [(`setf name`)], page 49, (generic function)

6.2.7 Structures

`char-info` () [Structure]

The Char Info structure.

This structure is used to store decoded information from the `char-info` table (see TeX: the Program [543]). Only one of `LIG/KERN-INDEX`, `NEXT-CHAR`, and `EXTEN-INDEX` may be non-null at a time (see TeX: the Program [544]).

Package [`net.didierverna.tfm`], page 21,

Source [`intermediate.lisp`], page 10, (file)

Direct superclasses

`structure-object` (structure)

Direct slots

| | | |
|-----------------------------|---|--------|
| <code>width-index</code> | | [Slot] |
| Readers | [<code>width-index</code>], page 95, (function) | |
| Writers | [<code>(setf width-index)</code>], page 95, (function) | |
| <code>height-index</code> | | [Slot] |
| Readers | [<code>height-index</code>], page 90, (function) | |
| Writers | [<code>(setf height-index)</code>], page 90, (function) | |
| <code>depth-index</code> | | [Slot] |
| Readers | [<code>depth-index</code>], page 89, (function) | |
| Writers | [<code>(setf depth-index)</code>], page 89, (function) | |
| <code>italic-index</code> | | [Slot] |
| Readers | [<code>italic-index</code>], page 90, (function) | |
| Writers | [<code>(setf italic-index)</code>], page 90, (function) | |
| <code>lig/kern-index</code> | | [Slot] |
| Readers | [<code>lig/kern-index</code>], page 90, (function) | |
| Writers | [<code>(setf lig/kern-index)</code>], page 90, (function) | |
| <code>next-char</code> | | [Slot] |
| Readers | [<code>next-char</code>], page 92, (function) | |
| Writers | [<code>(setf next-char)</code>], page 92, (function) | |
| <code>exten-index</code> | | [Slot] |
| Readers | [<code>exten-index</code>], page 90, (function) | |
| Writers | [<code>(setf exten-index)</code>], page 90, (function) | |

`exten ()` [Structure]

The Exten structure.

This structure is used to store decoded information from the exten table (see TeX: the Program [546]).

Package [net.didierverna.tfm], page 21,

Source [intermediate.lisp], page 10, (file)

Direct superclasses

structure-object (structure)

Direct slots

| | | |
|------------------|--|--------|
| <code>top</code> | | [Slot] |
| Readers | [<code>top</code>], page 95, (function) | |
| Writers | [<code>(setf top)</code>], page 95, (function) | |
| <code>mid</code> | | [Slot] |
| Readers | [<code>mid</code>], page 92, (function) | |
| Writers | [<code>(setf mid)</code>], page 92, (function) | |

| | | |
|---|--|-------------|
| <code>bot</code> | | [Slot] |
| Readers | [<code>bot</code>], page 88, (function) | |
| Writers | [<code>(setf bot)</code>], page 88, (function) | |
| <code>rep</code> | | [Slot] |
| Readers | [<code>rep</code>], page 94, (function) | |
| Writers | [<code>(setf rep)</code>], page 94, (function) | |
| <code>lig/kern ()</code> | | [Structure] |
| The Lig/Kern structure. | | |
| This structure is used to store decoded information from the lig/kern table (see TeX: the Program [545]). | | |
| Package | [<code>net.didierverna.tfm</code>], page 21, | |
| Source | [<code>intermediate.lisp</code>], page 10, (file) | |
| Direct superclasses | <code>structure-object</code> (structure) | |
| Direct slots | | |
| <code>skip</code> | | [Slot] |
| Readers | [<code>skip</code>], page 94, (function) | |
| Writers | [<code>(setf skip)</code>], page 94, (function) | |
| <code>next</code> | | [Slot] |
| Readers | [<code>next</code>], page 92, (function) | |
| Writers | [<code>(setf next)</code>], page 92, (function) | |
| <code>op</code> | | [Slot] |
| Readers | [<code>op</code>], page 92, (function) | |
| Writers | [<code>(setf op)</code>], page 92, (function) | |
| <code>remainder</code> | | [Slot] |
| Readers | [<code>remainder</code>], page 94, (function) | |
| Writers | [<code>(setf remainder)</code>], page 94, (function) | |

6.2.8 Classes

| | | |
|---|--|---------|
| <code>extension-recipe ()</code> | | [Class] |
| The Extension Recipe class. | | |
| This class represents decoded information for extensible characters. Within the context of this library, the expression "extension recipe" denotes an instance of this class. | | |
| Package | [<code>net.didierverna.tfm</code>], page 21, | |
| Source | [<code>character.lisp</code>], page 11, (file) | |
| Direct superclasses | <code>standard-object</code> (class) | |
| Direct methods | | |
| | <ul style="list-style-type: none"> • <code>print-object</code> (method) • <code>repeated-character</code> (method) | |

- [repeated-character], page 54, (method)
- bottom-character (method)
- [bottom-character], page 40, (method)
- middle-character (method)
- [middle-character], page 49, (method)
- top-character (method)
- [top-character], page 56, (method)

Direct slots

| | |
|--|--|
| top-character | [Slot] |
| The recipe's top character, or NIL. | |
| Initargs | :top-character |
| Readers | [top-character], page 56, (generic function) |
| Writers | [(setf top-character)], page 56, (generic function) |
| middle-character | [Slot] |
| The recipe's middle character, or NIL. | |
| Initargs | :middle-character |
| Readers | [middle-character], page 49, (generic function) |
| Writers | [(setf middle-character)], page 49, (generic function) |
| bottom-character | [Slot] |
| The recipe's bottom character, or NIL. | |
| Initargs | :bottom-character |
| Readers | [bottom-character], page 40, (generic function) |
| Writers | [(setf bottom-character)], page 40, (generic function) |
| repeated-character | [Slot] |
| The recipe's repeated character. | |
| Initargs | :repeated-character |
| Readers | [repeated-character], page 53, (generic function) |
| Writers | [(setf repeated-character)], page 53, (generic function) |

Appendix A Indexes

A.1 Concepts

F

File, Lisp, `net.didierverna.tfm.asd` 9
 File, Lisp, `net.didierverna.tfm.core.asd` 9
 File, Lisp,
 `net.didierverna.tfm.core/package.lisp` 9
 File, Lisp,
 `net.didierverna.tfm.core/src/character.lisp`..11■
 File, Lisp,
 `net.didierverna.tfm.core/src/file.lisp`.... 16
 File, Lisp,
 `net.didierverna.tfm.core/src/font.lisp`.... 12
 File, Lisp,
 `net.didierverna.tfm.core/src/intermediate.lisp`..10■
 File, Lisp,
 `net.didierverna.tfm.core/src/util.lisp`..... 9
 File, Lisp, `net.didierverna.tfm.setup.asd` 9
 File, Lisp,
 `net.didierverna.tfm.setup/package.lisp`.... 18
 File, Lisp,
 `net.didierverna.tfm.setup/src/configuration.lisp`..19■
 File, Lisp,
 `net.didierverna.tfm.setup/src/readtable.lisp`..19■
 File, Lisp,
 `net.didierverna.tfm.setup/src/version.lisp`..18■

L

Lisp File, `net.didierverna.tfm.asd` 9
 Lisp File, `net.didierverna.tfm.core.asd` 9
 Lisp File,
 `net.didierverna.tfm.core/package.lisp` 9
 Lisp File,
 `net.didierverna.tfm.core/src/character.lisp`..11■
 Lisp File,
 `net.didierverna.tfm.core/src/file.lisp`.... 16
 Lisp File,
 `net.didierverna.tfm.core/src/font.lisp`.... 12

Lisp File,

`net.didierverna.tfm.core/src/intermediate.lisp`..10■

Lisp File,

`net.didierverna.tfm.core/src/util.lisp`..... 9

Lisp File, `net.didierverna.tfm.setup.asd` 9

Lisp File,

`net.didierverna.tfm.setup/package.lisp`.... 18

Lisp File,

`net.didierverna.tfm.setup/src/configuration.lisp`..19■

Lisp File,

`net.didierverna.tfm.setup/src/readtable.lisp`..19■

Lisp File,

`net.didierverna.tfm.setup/src/version.lisp`..18■

M

Module, `net.didierverna.tfm.core/src` 7

Module, `net.didierverna.tfm.setup/src` 7

N

`net.didierverna.tfm.asd` 9

`net.didierverna.tfm.core.asd` 9

`net.didierverna.tfm.core/package.lisp` 9

`net.didierverna.tfm.core/src` 7

`net.didierverna.tfm.core/src/character.lisp`..11■

`net.didierverna.tfm.core/src/file.lisp` 16

`net.didierverna.tfm.core/src/font.lisp` 12

`net.didierverna.tfm.core/src/intermediate.lisp`..10■

`net.didierverna.tfm.core/src/util.lisp` 9

`net.didierverna.tfm.setup.asd` 9

`net.didierverna.tfm.setup/package.lisp` 18

`net.didierverna.tfm.setup/src` 7

`net.didierverna.tfm.setup/src/configuration.lisp`..19■

`net.didierverna.tfm.setup/src/readtable.lisp`..19■

`net.didierverna.tfm.setup/src/version.lisp`..18■

A.2 Functions

%

%run-ligature/kerning-program..... 88
%version..... 88

(

(setf 7bits-safe)..... 38
(setf actual-size)..... 38
(setf axis-height)..... 38
(setf bc)..... 38
(setf big-op-spacing1)..... 38, 39
(setf big-op-spacing2)..... 39
(setf big-op-spacing3)..... 39
(setf big-op-spacing4)..... 39
(setf big-op-spacing5)..... 39, 40
(setf bot)..... 88
(setf bottom-character)..... 40
(setf boundary-character)..... 40
(setf character-count)..... 40
(setf character1)..... 41
(setf character2)..... 41
(setf characters)..... 41
(setf checksum)..... 41
(setf code-character)..... 89
(setf declared-size)..... 42
(setf default-rule-thickness)..... 42
(setf delim1)..... 43
(setf delim2)..... 43
(setf denom1)..... 43
(setf denom2)..... 43
(setf depth)..... 44
(setf depth-index)..... 89
(setf design-size)..... 44
(setf ec)..... 44
(setf em)..... 44, 45
(setf encoding)..... 45
(setf ex)..... 45
(setf expansion)..... 45
(setf exten-index)..... 90
(setf extension-recipe)..... 95
(setf extra-space)..... 45
(setf face-code)..... 46
(setf face-number)..... 46
(setf family)..... 46
(setf file)..... 46
(setf frozen)..... 47
(setf height)..... 47
(setf height-index)..... 90
(setf interword-shrink)..... 47
(setf interword-space)..... 47
(setf interword-stretch)..... 48
(setf italic-correction)..... 48
(setf italic-index)..... 90
(setf kerning)..... 36
(setf kernings)..... 95
(setf largest)..... 48
(setf lf)..... 48
(setf lh)..... 49
(setf lig/kern-index)..... 90
(setf ligature)..... 37
(setf ligatures)..... 96

(setf max-code)..... 49
(setf mid)..... 92
(setf middle-character)..... 49
(setf min-code)..... 49
(setf name)..... 49, 50
(setf nc)..... 50
(setf nd)..... 50
(setf ne)..... 50
(setf next)..... 92
(setf next-char)..... 92
(setf next-character)..... 50
(setf nh)..... 51
(setf ni)..... 51
(setf nk)..... 51
(setf nl)..... 51
(setf np)..... 51, 52
(setf num1)..... 52
(setf num2)..... 52
(setf num3)..... 52
(setf nw)..... 52
(setf op)..... 92
(setf original-design-size)..... 53
(setf padding)..... 53
(setf parameters)..... 53
(setf remainder)..... 94
(setf rep)..... 94
(setf repeated-character)..... 53, 54
(setf skip)..... 94
(setf slant)..... 54
(setf slope)..... 54
(setf smallest)..... 54
(setf sub1)..... 54
(setf sub2)..... 55
(setf subdrop)..... 55
(setf sup1)..... 55
(setf sup2)..... 55
(setf sup3)..... 56
(setf supdrop)..... 56
(setf top)..... 95
(setf top-character)..... 56
(setf value)..... 56, 57, 58
(setf weight)..... 58
(setf width)..... 58
(setf width-index)..... 95

~

~-reader..... 95

7

7bits-safe..... 38

A

actual-size..... 38
apply-ligature..... 88
axis-height..... 38

B

| | |
|--------------------|--------|
| bc | 38 |
| big-op-spacing1 | 38, 39 |
| big-op-spacing2 | 39 |
| big-op-spacing3 | 39 |
| big-op-spacing4 | 39 |
| big-op-spacing5 | 39, 40 |
| bot | 88 |
| bottom-character | 40 |
| boundary-character | 40 |

C

| | |
|-----------------|--------|
| char-info-p | 88 |
| character-count | 40 |
| character1 | 41 |
| character2 | 41 |
| characters | 41 |
| checksum | 41 |
| clindent | 88 |
| code | 41, 42 |
| code-character | 89 |
| composite | 42 |
| configuration | 35 |
| configure | 36 |
| copy-char-info | 89 |
| copy-exten | 89 |
| copy-lig/kern | 89 |

D

| | |
|---|----|
| declared-size | 42 |
| decode-char-info | 89 |
| decode-exten | 89 |
| decode-lig/kern | 89 |
| default-rule-thickness | 42 |
| defindent | 86 |
| define-constant | 87 |
| define-extension-recipe-pseudo-accessor | 87 |
| delete-after | 42 |
| delete-before | 42 |
| delim1 | 43 |
| delim2 | 43 |
| denom1 | 43 |
| denom2 | 43 |
| depth | 44 |
| depth-index | 89 |
| design-size | 44 |

E

| | |
|------------------|--------|
| ec | 44 |
| em | 44, 45 |
| encoding | 45 |
| ex | 45 |
| expansion | 45 |
| exten-index | 90 |
| exten-p | 90 |
| extensiblep | 36 |
| extension-recipe | 95 |
| extra-space | 45 |

F

| | |
|---|--------|
| face-code | 46 |
| face-number | 46 |
| family | 46 |
| file | 46 |
| font | 46, 47 |
| font-extension-recipe | 90 |
| freeze | 36 |
| frozen | 47 |
| Function, %run-ligature/kerling-program | 88 |
| Function, %version | 88 |
| Function, (setf bot) | 88 |
| Function, (setf code-character) | 89 |
| Function, (setf depth-index) | 89 |
| Function, (setf exten-index) | 90 |
| Function, (setf height-index) | 90 |
| Function, (setf italic-index) | 90 |
| Function, (setf kerning) | 36 |
| Function, (setf lig/kern-index) | 90 |
| Function, (setf ligature) | 37 |
| Function, (setf mid) | 92 |
| Function, (setf next) | 92 |
| Function, (setf next-char) | 92 |
| Function, (setf op) | 92 |
| Function, (setf remainder) | 94 |
| Function, (setf rep) | 94 |
| Function, (setf skip) | 94 |
| Function, (setf top) | 95 |
| Function, (setf width-index) | 95 |
| Function, ~-reader | 95 |
| Function, apply-ligature | 88 |
| Function, bot | 88 |
| Function, char-info-p | 88 |
| Function, clindent | 88 |
| Function, code-character | 89 |
| Function, configuration | 35 |
| Function, configure | 36 |
| Function, copy-char-info | 89 |
| Function, copy-exten | 89 |
| Function, copy-lig/kern | 89 |
| Function, decode-char-info | 89 |
| Function, decode-exten | 89 |
| Function, decode-lig/kern | 89 |
| Function, depth-index | 89 |
| Function, exten-index | 90 |
| Function, exten-p | 90 |
| Function, extensiblep | 36 |
| Function, font-extension-recipe | 90 |
| Function, freeze | 36 |
| Function, get-character | 36 |
| Function, height-index | 90 |
| Function, i-reader | 90 |
| Function, italic-index | 90 |
| Function, kerning | 36 |
| Function, lig/kern-index | 90 |
| Function, lig/kern-p | 90 |
| Function, ligature | 36 |
| Function, load-font | 37 |
| Function, load-tfm-font | 90 |
| Function, make-char-info | 91 |
| Function, make-character-metrics | 91 |
| Function, make-exten | 91 |
| Function, make-extension-recipe | 91 |
| Function, make-font | 91 |

| | |
|---|----|
| Function, <code>make-lig/kern</code> | 92 |
| Function, <code>make-ligature</code> | 92 |
| Function, <code>mid</code> | 92 |
| Function, <code>next</code> | 92 |
| Function, <code>next-char</code> | 92 |
| Function, <code>nickname-package</code> | 37 |
| Function, <code>op</code> | 92 |
| Function, <code>parse-character-information</code> | 92 |
| Function, <code>parse-header</code> | 93 |
| Function, <code>read-fix-word</code> | 93 |
| Function, <code>read-padded-string</code> | 93 |
| Function, <code>read-u16</code> | 94 |
| Function, <code>read-u32</code> | 94 |
| Function, <code>release-status-number</code> | 94 |
| Function, <code>remainder</code> | 94 |
| Function, <code>rep</code> | 94 |
| Function, <code>report</code> | 94 |
| Function, <code>run-ligature/kerning-program</code> | 94 |
| Function, <code>skip</code> | 94 |
| Function, <code>table-aref</code> | 95 |
| Function, <code>top</code> | 95 |
| Function, <code>unfreeze</code> | 37 |
| Function, <code>version</code> | 37 |
| Function, <code>width-index</code> | 95 |

G

| | |
|--|----|
| Generic Function, <code>(setf 7bits-safe)</code> | 38 |
| Generic Function, <code>(setf actual-size)</code> | 38 |
| Generic Function, <code>(setf axis-height)</code> | 38 |
| Generic Function, <code>(setf bc)</code> | 38 |
| Generic Function, <code>(setf big-op-spacing1)</code> | 38 |
| Generic Function, <code>(setf big-op-spacing2)</code> | 39 |
| Generic Function, <code>(setf big-op-spacing3)</code> | 39 |
| Generic Function, <code>(setf big-op-spacing4)</code> | 39 |
| Generic Function, <code>(setf big-op-spacing5)</code> | 39 |
| Generic Function, <code>(setf bottom-character)</code> | 40 |
| Generic Function, <code>(setf boundary-character)</code> | 40 |
| Generic Function, <code>(setf character-count)</code> | 40 |
| Generic Function, <code>(setf character1)</code> | 41 |
| Generic Function, <code>(setf character2)</code> | 41 |
| Generic Function, <code>(setf characters)</code> | 41 |
| Generic Function, <code>(setf checksum)</code> | 41 |
| Generic Function, <code>(setf declared-size)</code> | 42 |
| Generic Function, <code>(setf</code> <code>default-rule-thickness)</code> | 42 |
| Generic Function, <code>(setf delim1)</code> | 43 |
| Generic Function, <code>(setf delim2)</code> | 43 |
| Generic Function, <code>(setf denom1)</code> | 43 |
| Generic Function, <code>(setf denom2)</code> | 43 |
| Generic Function, <code>(setf depth)</code> | 44 |
| Generic Function, <code>(setf design-size)</code> | 44 |
| Generic Function, <code>(setf ec)</code> | 44 |
| Generic Function, <code>(setf em)</code> | 44 |
| Generic Function, <code>(setf encoding)</code> | 45 |
| Generic Function, <code>(setf ex)</code> | 45 |
| Generic Function, <code>(setf expansion)</code> | 45 |
| Generic Function, <code>(setf extension-recipe)</code> | 95 |
| Generic Function, <code>(setf extra-space)</code> | 45 |
| Generic Function, <code>(setf face-code)</code> | 46 |
| Generic Function, <code>(setf face-number)</code> | 46 |
| Generic Function, <code>(setf family)</code> | 46 |
| Generic Function, <code>(setf file)</code> | 46 |
| Generic Function, <code>(setf frozen)</code> | 47 |
| Generic Function, <code>(setf height)</code> | 47 |
| Generic Function, <code>(setf interword-shrink)</code> | 47 |
| Generic Function, <code>(setf interword-space)</code> | 47 |
| Generic Function, <code>(setf interword-stretch)</code> | 48 |
| Generic Function, <code>(setf italic-correction)</code> | 48 |
| Generic Function, <code>(setf kernings)</code> | 95 |
| Generic Function, <code>(setf largest)</code> | 48 |
| Generic Function, <code>(setf lf)</code> | 48 |
| Generic Function, <code>(setf lh)</code> | 49 |
| Generic Function, <code>(setf ligatures)</code> | 96 |
| Generic Function, <code>(setf max-code)</code> | 49 |
| Generic Function, <code>(setf middle-character)</code> | 49 |
| Generic Function, <code>(setf min-code)</code> | 49 |
| Generic Function, <code>(setf name)</code> | 49 |
| Generic Function, <code>(setf nc)</code> | 50 |
| Generic Function, <code>(setf nd)</code> | 50 |
| Generic Function, <code>(setf ne)</code> | 50 |
| Generic Function, <code>(setf next-character)</code> | 50 |
| Generic Function, <code>(setf nh)</code> | 51 |
| Generic Function, <code>(setf ni)</code> | 51 |
| Generic Function, <code>(setf nk)</code> | 51 |
| Generic Function, <code>(setf nl)</code> | 51 |
| Generic Function, <code>(setf np)</code> | 51 |
| Generic Function, <code>(setf num1)</code> | 52 |
| Generic Function, <code>(setf num2)</code> | 52 |
| Generic Function, <code>(setf num3)</code> | 52 |
| Generic Function, <code>(setf nw)</code> | 52 |
| Generic Function, <code>(setf</code> <code>original-design-size)</code> | 53 |
| Generic Function, <code>(setf padding)</code> | 53 |
| Generic Function, <code>(setf parameters)</code> | 53 |
| Generic Function, <code>(setf repeated-character)</code> | 53 |
| Generic Function, <code>(setf slant)</code> | 54 |
| Generic Function, <code>(setf slope)</code> | 54 |
| Generic Function, <code>(setf smallest)</code> | 54 |
| Generic Function, <code>(setf sub1)</code> | 54 |
| Generic Function, <code>(setf sub2)</code> | 55 |
| Generic Function, <code>(setf subdrop)</code> | 55 |
| Generic Function, <code>(setf sup1)</code> | 55 |
| Generic Function, <code>(setf sup2)</code> | 55 |
| Generic Function, <code>(setf sup3)</code> | 56 |
| Generic Function, <code>(setf supdrop)</code> | 56 |
| Generic Function, <code>(setf top-character)</code> | 56 |
| Generic Function, <code>(setf value)</code> | 56 |
| Generic Function, <code>(setf weight)</code> | 58 |
| Generic Function, <code>(setf width)</code> | 58 |
| Generic Function, <code>7bits-safe</code> | 38 |
| Generic Function, <code>actual-size</code> | 38 |
| Generic Function, <code>axis-height</code> | 38 |
| Generic Function, <code>bc</code> | 38 |
| Generic Function, <code>big-op-spacing1</code> | 38 |
| Generic Function, <code>big-op-spacing2</code> | 39 |
| Generic Function, <code>big-op-spacing3</code> | 39 |
| Generic Function, <code>big-op-spacing4</code> | 39 |
| Generic Function, <code>big-op-spacing5</code> | 39 |
| Generic Function, <code>bottom-character</code> | 40 |
| Generic Function, <code>boundary-character</code> | 40 |
| Generic Function, <code>character-count</code> | 40 |
| Generic Function, <code>character1</code> | 41 |
| Generic Function, <code>character2</code> | 41 |
| Generic Function, <code>characters</code> | 41 |
| Generic Function, <code>checksum</code> | 41 |
| Generic Function, <code>code</code> | 41 |
| Generic Function, <code>composite</code> | 42 |
| Generic Function, <code>declared-size</code> | 42 |

| | |
|--|----|
| Generic Function, <code>default-rule-thickness</code> | 42 |
| Generic Function, <code>delete-after</code> | 42 |
| Generic Function, <code>delete-before</code> | 42 |
| Generic Function, <code>delim1</code> | 43 |
| Generic Function, <code>delim2</code> | 43 |
| Generic Function, <code>denom1</code> | 43 |
| Generic Function, <code>denom2</code> | 43 |
| Generic Function, <code>depth</code> | 44 |
| Generic Function, <code>design-size</code> | 44 |
| Generic Function, <code>ec</code> | 44 |
| Generic Function, <code>em</code> | 44 |
| Generic Function, <code>encoding</code> | 45 |
| Generic Function, <code>ex</code> | 45 |
| Generic Function, <code>expansion</code> | 45 |
| Generic Function, <code>extension-recipe</code> | 95 |
| Generic Function, <code>extra-space</code> | 45 |
| Generic Function, <code>face-code</code> | 46 |
| Generic Function, <code>face-number</code> | 46 |
| Generic Function, <code>family</code> | 46 |
| Generic Function, <code>file</code> | 46 |
| Generic Function, <code>font</code> | 46 |
| Generic Function, <code>frozen</code> | 47 |
| Generic Function, <code>height</code> | 47 |
| Generic Function, <code>interword-shrink</code> | 47 |
| Generic Function, <code>interword-space</code> | 47 |
| Generic Function, <code>interword-stretch</code> | 48 |
| Generic Function, <code>italic-correction</code> | 48 |
| Generic Function, <code>kernings</code> | 95 |
| Generic Function, <code>largest</code> | 48 |
| Generic Function, <code>lf</code> | 48 |
| Generic Function, <code>lh</code> | 49 |
| Generic Function, <code>ligatures</code> | 96 |
| Generic Function, <code>max-code</code> | 49 |
| Generic Function, <code>middle-character</code> | 49 |
| Generic Function, <code>min-code</code> | 49 |
| Generic Function, <code>name</code> | 49 |
| Generic Function, <code>nc</code> | 50 |
| Generic Function, <code>nd</code> | 50 |
| Generic Function, <code>ne</code> | 50 |
| Generic Function, <code>next-character</code> | 50 |
| Generic Function, <code>nh</code> | 51 |
| Generic Function, <code>ni</code> | 51 |
| Generic Function, <code>nk</code> | 51 |
| Generic Function, <code>nl</code> | 51 |
| Generic Function, <code>np</code> | 51 |
| Generic Function, <code>num1</code> | 52 |
| Generic Function, <code>num2</code> | 52 |
| Generic Function, <code>num3</code> | 52 |
| Generic Function, <code>nw</code> | 52 |
| Generic Function, <code>original-design-size</code> | 53 |
| Generic Function, <code>padding</code> | 53 |
| Generic Function, <code>parameters</code> | 53 |
| Generic Function, <code>parse-parameters</code> | 96 |
| Generic Function, <code>pass-over</code> | 53 |
| Generic Function, <code>repeated-character</code> | 53 |
| Generic Function, <code>scale</code> | 96 |
| Generic Function, <code>slant</code> | 54 |
| Generic Function, <code>slope</code> | 54 |
| Generic Function, <code>smallest</code> | 54 |
| Generic Function, <code>sub1</code> | 54 |
| Generic Function, <code>sub2</code> | 55 |
| Generic Function, <code>subdrop</code> | 55 |
| Generic Function, <code>sup1</code> | 55 |
| Generic Function, <code>sup2</code> | 55 |
| Generic Function, <code>sup3</code> | 56 |
| Generic Function, <code>supdrop</code> | 56 |
| Generic Function, <code>top-character</code> | 56 |
| Generic Function, <code>value</code> | 56 |
| Generic Function, <code>weight</code> | 58 |
| Generic Function, <code>width</code> | 58 |
| <code>get-character</code> | 36 |
| H | |
| <code>height</code> | 47 |
| <code>height-index</code> | 90 |
| I | |
| <code>i-reader</code> | 90 |
| <code>interword-shrink</code> | 47 |
| <code>interword-space</code> | 47 |
| <code>interword-stretch</code> | 48 |
| <code>italic-correction</code> | 48 |
| <code>italic-index</code> | 90 |
| K | |
| <code>kerning</code> | 36 |
| <code>kernings</code> | 95 |
| L | |
| <code>largest</code> | 48 |
| <code>lf</code> | 48 |
| <code>lh</code> | 49 |
| <code>lig/kern-index</code> | 90 |
| <code>lig/kern-p</code> | 90 |
| <code>ligature</code> | 36 |
| <code>ligatures</code> | 96 |
| <code>load-font</code> | 37 |
| <code>load-tfm-font</code> | 90 |
| M | |
| Macro, <code>defindent</code> | 86 |
| Macro, <code>define-constant</code> | 87 |
| Macro, <code>define-extension-recipe-pseudo-accessor</code> ... | 87 |
| Macro, <code>map-accessors</code> | 87 |
| Macro, <code>map-character-metrics-</code> <code>dimension-accessors</code> | 87 |
| Macro, <code>map-font-dimension-accessors</code> | 87 |
| Macro, <code>map-math-extension-font-</code> <code>dimension-accessors</code> | 87 |
| Macro, <code>map-math-symbols-font-</code> <code>dimension-accessors</code> | 87 |
| Macro, <code>tref</code> | 87 |
| <code>make-char-info</code> | 91 |
| <code>make-character-metrics</code> | 91 |
| <code>make-exten</code> | 91 |
| <code>make-extension-recipe</code> | 91 |
| <code>make-font</code> | 91 |
| <code>make-lig/kern</code> | 92 |
| <code>make-ligature</code> | 92 |
| <code>map-accessors</code> | 87 |
| <code>map-character-metrics-dimension-accessors</code> ... | 87 |
| <code>map-font-dimension-accessors</code> | 87 |

| | |
|--|------------|
| map-math-extension-font- | |
| dimension-accessors | 87 |
| map-math-symbols-font-dimension-accessors... | 87 |
| max-code | 49 |
| Method, (setf 7bits-safe) | 38 |
| Method, (setf actual-size) | 38 |
| Method, (setf axis-height) | 38 |
| Method, (setf bc) | 38 |
| Method, (setf big-op-spacing1) | 39 |
| Method, (setf big-op-spacing2) | 39 |
| Method, (setf big-op-spacing3) | 39 |
| Method, (setf big-op-spacing4) | 39 |
| Method, (setf big-op-spacing5) | 40 |
| Method, (setf bottom-character) | 40 |
| Method, (setf boundary-character) | 40 |
| Method, (setf character-count) | 40 |
| Method, (setf character1) | 41 |
| Method, (setf character2) | 41 |
| Method, (setf characters) | 41 |
| Method, (setf checksum) | 41 |
| Method, (setf declared-size) | 42 |
| Method, (setf default-rule-thickness) | 42 |
| Method, (setf delim1) | 43 |
| Method, (setf delim2) | 43 |
| Method, (setf denom1) | 43 |
| Method, (setf denom2) | 43 |
| Method, (setf depth) | 44 |
| Method, (setf design-size) | 44 |
| Method, (setf ec) | 44 |
| Method, (setf em) | 45 |
| Method, (setf encoding) | 45 |
| Method, (setf ex) | 45 |
| Method, (setf expansion) | 45 |
| Method, (setf extension-recipe) | 95 |
| Method, (setf extra-space) | 45 |
| Method, (setf face-code) | 46 |
| Method, (setf face-number) | 46 |
| Method, (setf family) | 46 |
| Method, (setf file) | 46 |
| Method, (setf frozen) | 47 |
| Method, (setf height) | 47 |
| Method, (setf interword-shrink) | 47 |
| Method, (setf interword-space) | 47 |
| Method, (setf interword-stretch) | 48 |
| Method, (setf italic-correction) | 48 |
| Method, (setf kernings) | 95 |
| Method, (setf largest) | 48 |
| Method, (setf lf) | 48 |
| Method, (setf lh) | 49 |
| Method, (setf ligatures) | 96 |
| Method, (setf max-code) | 49 |
| Method, (setf middle-character) | 49 |
| Method, (setf min-code) | 49 |
| Method, (setf name) | 50 |
| Method, (setf nc) | 50 |
| Method, (setf nd) | 50 |
| Method, (setf ne) | 50 |
| Method, (setf next-character) | 50 |
| Method, (setf nh) | 51 |
| Method, (setf ni) | 51 |
| Method, (setf nk) | 51 |
| Method, (setf nl) | 51 |
| Method, (setf np) | 52 |
| Method, (setf num1) | 52 |
| Method, (setf num2) | 52 |
| Method, (setf num3) | 52 |
| Method, (setf nw) | 52 |
| Method, (setf original-design-size) | 53 |
| Method, (setf padding) | 53 |
| Method, (setf parameters) | 53 |
| Method, (setf repeated-character) | 54 |
| Method, (setf slant) | 54 |
| Method, (setf slope) | 54 |
| Method, (setf smallest) | 54 |
| Method, (setf sub1) | 54 |
| Method, (setf sub2) | 55 |
| Method, (setf subdrop) | 55 |
| Method, (setf sup1) | 55 |
| Method, (setf sup2) | 55 |
| Method, (setf sup3) | 56 |
| Method, (setf supdrop) | 56 |
| Method, (setf top-character) | 56 |
| Method, (setf value) | 56, 57, 58 |
| Method, (setf weight) | 58 |
| Method, (setf width) | 58 |
| Method, 7bits-safe | 38 |
| Method, actual-size | 38 |
| Method, axis-height | 38 |
| Method, bc | 38 |
| Method, big-op-spacing1 | 39 |
| Method, big-op-spacing2 | 39 |
| Method, big-op-spacing3 | 39 |
| Method, big-op-spacing4 | 39 |
| Method, big-op-spacing5 | 40 |
| Method, bottom-character | 40 |
| Method, boundary-character | 40 |
| Method, character-count | 40 |
| Method, character1 | 41 |
| Method, character2 | 41 |
| Method, characters | 41 |
| Method, checksum | 41 |
| Method, code | 42 |
| Method, composite | 42 |
| Method, declared-size | 42 |
| Method, default-rule-thickness | 42 |
| Method, delete-after | 42 |
| Method, delete-before | 42 |
| Method, delim1 | 43 |
| Method, delim2 | 43 |
| Method, denom1 | 43 |
| Method, denom2 | 43 |
| Method, depth | 44 |
| Method, design-size | 44 |
| Method, ec | 44 |
| Method, em | 45 |
| Method, encoding | 45 |
| Method, ex | 45 |
| Method, expansion | 45 |
| Method, extension-recipe | 95 |
| Method, extra-space | 45 |
| Method, face-code | 46 |
| Method, face-number | 46 |
| Method, family | 46 |
| Method, file | 46 |
| Method, font | 47 |
| Method, frozen | 47 |
| Method, height | 47 |
| Method, interword-shrink | 47 |
| Method, interword-space | 47 |
| Method, interword-stretch | 48 |

Method, *italic-correction* 48
 Method, *kernings* 95
 Method, *largest* 48
 Method, *lf* 48
 Method, *lh* 49
 Method, *ligatures* 96
 Method, *max-code* 49
 Method, *middle-character* 49
 Method, *min-code* 49
 Method, *name* 50
 Method, *nc* 50
 Method, *nd* 50
 Method, *ne* 50
 Method, *next-character* 50
 Method, *nh* 51
 Method, *ni* 51
 Method, *nk* 51
 Method, *nl* 51
 Method, *np* 52
 Method, *num1* 52
 Method, *num2* 52
 Method, *num3* 52
 Method, *nw* 52
 Method, *original-design-size* 53
 Method, *padding* 53
 Method, *parameters* 53
 Method, *parse-parameters* 96
 Method, *pass-over* 53
 Method, *repeated-character* 53, 54
 Method, *scale* 96
 Method, *slant* 54
 Method, *slope* 54
 Method, *smallest* 54
 Method, *sub1* 54
 Method, *sub2* 55
 Method, *subdrop* 55
 Method, *sup1* 55
 Method, *sup2* 55
 Method, *sup3* 56
 Method, *supdrop* 56
 Method, *top-character* 56
 Method, *value* 56, 57, 58
 Method, *weight* 58
 Method, *width* 58
mid 92
middle-character 49
min-code 49

N

name 49, 50
nc 50
nd 50
ne 50
next 92
next-char 92
next-character 50
nh 51
ni 51
nickname-package 37
nk 51
nl 51
np 51, 52
num1 52
num2 52

num3 52
nw 52

O

op 92
original-design-size 53

P

padding 53
parameters 53
parse-character-information 92
parse-header 93
parse-parameters 96
pass-over 53

R

read-fix-word 93
read-padded-string 93
read-u16 94
read-u32 94
release-status-number 94
remainder 94
rep 94
repeated-character 53, 54
report 94
run-ligature/kerning-program 94

S

scale 96
skip 94
slant 54
slope 54
smallest 54
sub1 54
sub2 55
subdrop 55
sup1 55
sup2 55
sup3 56
supdrop 56

T

table-aref 95
top 95
top-character 56
tref 87

U

unfreeze 37

V

value 56, 57, 58
version 37

W

| | | | |
|---------------------|----|--------------------------|----|
| weight | 58 | width | 58 |
| | | width-index | 95 |

A.3 Variables

*

| | |
|------------------------|----|
| *configuration* | 86 |
| *copyright-years* | 35 |
| *release-major-level* | 35 |
| *release-minor-level* | 35 |
| *release-name* | 35 |
| *release-status* | 35 |
| *release-status-level* | 35 |
| *stream* | 86 |

+

| | |
|---|----|
| +character-metrics-dimension-accessors+ | 86 |
| +font-dimension-accessors+ | 86 |
| +math-extension-font-dimension-accessors+ | 86 |
| +math-symbols-font-dimension-accessors+ | 86 |

7

| | |
|------------|----|
| 7bits-safe | 77 |
|------------|----|

A

| | |
|-------------|----|
| actual-size | 97 |
| axis-height | 85 |

B

| | |
|--------------------|-----|
| bc | 62 |
| big-op-spacing1 | 82 |
| big-op-spacing2 | 82 |
| big-op-spacing3 | 82 |
| big-op-spacing4 | 82 |
| big-op-spacing5 | 82 |
| bot | 100 |
| bottom-character | 101 |
| boundary-character | 80 |

C

| | |
|---|--------|
| character-count | 79 |
| character1 | 59 |
| character2 | 59 |
| characters | 69, 79 |
| checksum | 76 |
| code | 73 |
| composite | 80 |
| Constant, | |
| +character-metrics-dimension-accessors+ | 86 |
| Constant, +font-dimension-accessors+ | 86 |
| Constant, +math-extension-font- | |
| dimension-accessors+ | 86 |
| Constant, | |
| +math-symbols-font-dimension-accessors+ | 86 |

D

| | |
|------------------------|----|
| declared-size | 97 |
| default-rule-thickness | 81 |
| delete-after | 81 |
| delete-before | 81 |
| delim1 | 85 |
| delim2 | 85 |
| denom1 | 84 |
| denom2 | 84 |
| depth | 74 |
| depth-index | 99 |
| design-size | 77 |

E

| | |
|------------------|----|
| ec | 63 |
| em | 79 |
| encoding | 77 |
| ex | 78 |
| expansion | 78 |
| exten-index | 99 |
| extension-recipe | 74 |
| extra-space | 79 |

F

| | |
|-------------|--------|
| face-code | 78 |
| face-number | 77 |
| family | 77 |
| file | 60, 76 |
| font | 73 |
| frozen | 77 |

H

| | |
|--------------|----|
| height | 74 |
| height-index | 99 |

I

| | |
|-------------------|----|
| interword-shrink | 78 |
| interword-space | 78 |
| interword-stretch | 78 |
| italic-correction | 74 |
| italic-index | 99 |

K

| | |
|----------|----|
| kernings | 80 |
|----------|----|

L

| | |
|----------------|--------|
| largest | 67, 68 |
| lf | 65 |
| lh | 65 |
| lig/kern-index | 99 |
| ligatures | 80 |

M

| | |
|------------------|-----|
| max-code | 79 |
| mid | 99 |
| middle-character | 101 |
| min-code | 79 |

N

| | |
|----------------|--------|
| name | 76, 98 |
| nc | 65 |
| nd | 65 |
| ne | 66 |
| next | 100 |
| next-char | 99 |
| next-character | 74 |
| nh | 65 |
| ni | 65 |
| nk | 66 |
| nl | 66 |
| np | 66 |
| num1 | 83 |
| num2 | 83 |
| num3 | 84 |
| nw | 65 |

O

| | |
|----------------------|-----|
| op | 100 |
| original-design-size | 77 |

P

| | |
|------------|----|
| padding | 67 |
| parameters | 79 |
| pass-over | 81 |

R

| | |
|--------------------|-----|
| remainder | 100 |
| rep | 100 |
| repeated-character | 101 |

S

| | |
|--------------------------|--------|
| skip | 100 |
| slant | 78 |
| slope | 77 |
| Slot, 7bits-safe | 77 |
| Slot, actual-size | 97 |
| Slot, axis-height | 85 |
| Slot, bc | 62 |
| Slot, big-op-spacing1 | 82 |
| Slot, big-op-spacing2 | 82 |
| Slot, big-op-spacing3 | 82 |
| Slot, big-op-spacing4 | 82 |
| Slot, big-op-spacing5 | 82 |
| Slot, bot | 100 |
| Slot, bottom-character | 101 |
| Slot, boundary-character | 80 |
| Slot, character-count | 79 |
| Slot, character1 | 59 |
| Slot, character2 | 59 |
| Slot, characters | 69, 79 |
| Slot, checksum | 76 |

| | |
|------------------------------|--------|
| Slot, code | 73 |
| Slot, composite | 80 |
| Slot, declared-size | 97 |
| Slot, default-rule-thickness | 81 |
| Slot, delete-after | 81 |
| Slot, delete-before | 81 |
| Slot, delim1 | 85 |
| Slot, delim2 | 85 |
| Slot, denom1 | 84 |
| Slot, denom2 | 84 |
| Slot, depth | 74 |
| Slot, depth-index | 99 |
| Slot, design-size | 77 |
| Slot, ec | 63 |
| Slot, em | 79 |
| Slot, encoding | 77 |
| Slot, ex | 78 |
| Slot, expansion | 78 |
| Slot, exten-index | 99 |
| Slot, extension-recipe | 74 |
| Slot, extra-space | 79 |
| Slot, face-code | 78 |
| Slot, face-number | 77 |
| Slot, family | 77 |
| Slot, file | 60, 76 |
| Slot, font | 73 |
| Slot, frozen | 77 |
| Slot, height | 74 |
| Slot, height-index | 99 |
| Slot, interword-shrink | 78 |
| Slot, interword-space | 78 |
| Slot, interword-stretch | 78 |
| Slot, italic-correction | 74 |
| Slot, italic-index | 99 |
| Slot, kernings | 80 |
| Slot, largest | 67, 68 |
| Slot, lf | 65 |
| Slot, lh | 65 |
| Slot, lig/kern-index | 99 |
| Slot, ligatures | 80 |
| Slot, max-code | 79 |
| Slot, mid | 99 |
| Slot, middle-character | 101 |
| Slot, min-code | 79 |
| Slot, name | 76, 98 |
| Slot, nc | 65 |
| Slot, nd | 65 |
| Slot, ne | 66 |
| Slot, next | 100 |
| Slot, next-char | 99 |
| Slot, next-character | 74 |
| Slot, nh | 65 |
| Slot, ni | 65 |
| Slot, nk | 66 |
| Slot, nl | 66 |
| Slot, np | 66 |
| Slot, num1 | 83 |
| Slot, num2 | 83 |
| Slot, num3 | 84 |
| Slot, nw | 65 |
| Slot, op | 100 |
| Slot, original-design-size | 77 |
| Slot, padding | 67 |
| Slot, parameters | 79 |
| Slot, pass-over | 81 |

| | | | |
|--|-----|--|-----|
| Slot, remainder | 100 | Special Variable, <i>*release-status*</i> | 35 |
| Slot, rep | 100 | Special Variable, <i>*release-status-level*</i> | 35 |
| Slot, repeated-character | 101 | Special Variable, <i>*stream*</i> | 86 |
| Slot, skip | 100 | sub1 | 84 |
| Slot, slant | 78 | sub2 | 85 |
| Slot, slope | 77 | subdrop | 85 |
| Slot, smallest | 68 | sup1 | 84 |
| Slot, sub1 | 84 | sup2 | 84 |
| Slot, sub2 | 85 | sup3 | 84 |
| Slot, subdrop | 85 | supdrop | 85 |
| Slot, sup1 | 84 | | |
| Slot, sup2 | 84 | T | |
| Slot, sup3 | 84 | top | 99 |
| Slot, supdrop | 85 | top-character | 101 |
| Slot, top | 99 | | |
| Slot, top-character | 101 | V | |
| Slot, value . . . 59, 60, 61, 62, 63, 64, 66, 67, 68, 69, 72, 98 | | value 59, 60, 61, 62, 63, 64, 66, 67, 68, 69, 72, 98 | |
| Slot, weight | 77 | | |
| Slot, width | 73 | W | |
| Slot, width-index | 99 | weight | 77 |
| smallest | 68 | width | 73 |
| Special Variable, <i>*configuration*</i> | 86 | width-index | 99 |
| Special Variable, <i>*copyright-years*</i> | 35 | | |
| Special Variable, <i>*release-major-level*</i> | 35 | | |
| Special Variable, <i>*release-minor-level*</i> | 35 | | |
| Special Variable, <i>*release-name*</i> | 35 | | |

A.4 Data types

A

anonymous-font 97

C

char-info 98
 character-list-cycle 59
 character-metrics 73
 Class, character-metrics 73
 Class, extension-recipe 100
 Class, font 74
 Class, ligature 80
 Class, math-extension-font 81
 Class, math-symbols-font 82
 Condition, anonymous-font 97
 Condition, character-list-cycle 59
 Condition, different-fonts 59
 Condition, extended-tfm 60
 Condition, file-overflow 60
 Condition, file-size-mixin 97
 Condition, file-underflow 60
 Condition, fix-word-overflow 61
 Condition, invalid-bcpl-string 61
 Condition, invalid-char-info 61
 Condition, invalid-character-code 62
 Condition, invalid-character-range 62
 Condition, invalid-design-size 63
 Condition, invalid-header-length 63
 Condition, invalid-ligature-opcode 63
 Condition, invalid-original-design-size 97
 Condition, invalid-section-lengths 64
 Condition, invalid-string-length 66
 Condition, invalid-table-index 67
 Condition, invalid-table-length 67
 Condition, invalid-table-start 68
 Condition, ligature-cycle 68
 Condition, no-boundary-character 69
 Condition, not-extensible 69
 Condition, tfm 70
 Condition, tfm-compliance 70
 Condition, tfm-compliance-error 70
 Condition, tfm-compliance-warning 71
 Condition, tfm-error 71
 Condition, tfm-table-error 98
 Condition, tfm-usage 71
 Condition, tfm-usage-error 71
 Condition, tfm-usage-warning 72
 Condition, tfm-warning 72
 Condition, u16-overflow 72

D

different-fonts 59

E

exten 99
 extended-tfm 60
 extension-recipe 100

F

file-overflow 60
 file-size-mixin 97
 file-underflow 60
 fix-word-overflow 61
 font 74

I

invalid-bcpl-string 61
 invalid-char-info 61
 invalid-character-code 62
 invalid-character-range 62
 invalid-design-size 63
 invalid-header-length 63
 invalid-ligature-opcode 63
 invalid-original-design-size 97
 invalid-section-lengths 64
 invalid-string-length 66
 invalid-table-index 67
 invalid-table-length 67
 invalid-table-start 68

L

lig/kern 100
 ligature 80
 ligature-cycle 68

M

math-extension-font 81
 math-symbols-font 82

N

net.didierverna.tfm 5, 21
 net.didierverna.tfm.core 5
 net.didierverna.tfm.setup 6, 33
 no-boundary-character 69
 not-extensible 69

P

Package, net.didierverna.tfm 21
 Package, net.didierverna.tfm.setup 33

S

Structure, char-info 98
 Structure, exten 99
 Structure, lig/kern 100
 System, net.didierverna.tfm 5
 System, net.didierverna.tfm.core 5
 System, net.didierverna.tfm.setup 6

T

| | |
|-----------------------------|----|
| tfm | 70 |
| tfm-compliance..... | 70 |
| tfm-compliance-error..... | 70 |
| tfm-compliance-warning..... | 71 |
| tfm-error..... | 71 |
| tfm-table-error..... | 98 |

| | |
|------------------------|----|
| tfm-usage..... | 71 |
| tfm-usage-error..... | 71 |
| tfm-usage-warning..... | 72 |
| tfm-warning..... | 72 |

U

| | |
|-------------------|----|
| u16-overflow..... | 72 |
|-------------------|----|