

The L^AT_EX dtxdescribe Package

v1.06 — 2022/12/07

© 2016–2022 Brian Dunn
bd@BDTechConcepts.com

Describe additional object types in dtx or tex source files.

Abstract

The `doc` package includes tools for describing macros and environments in L^AT_EX source `.dtx` format. The `dtxdescribe` package adds additional tools for describing booleans, lengths, counters, hooks, keys, packages, classes, options, files, commands, arguments, and other objects. `dtxdescribe` also works with the regular document classes, for those who do not wish to use the `ltxdoc` class and `.dtx` files.

Each described item is given a margin tag similar to `\DescribeEnv`, and is listed in the index by itself and also by category. Each item may be sorted further by an optional class. All index entries except code lines are hyperlinked.

The `dtxexample` environment is provided for typesetting example code and its results. Contents are displayed verbatim along with a caption and cross-referencing. They are then `\input` and executed, and the result is shown.

Environments are also provided for displaying verbatim or formatted source code, user-interface displays, and sidebars with titles.

Macros are provided for formatting the names of inline L^AT_EX objects such as packages and booleans, as well as program and file names, file types, internet objects, the names of certain programs, a number of logos, and inline dashes and slashes.

`dtxdescribe` works with the `ltxdoc` class, but also works with the standard classes as well, except that the macro and environment environments are not supported. Either `makeidx` or `splitidx` may be loaded by the user. `makeidx` will be used by default.

`dtxdescribe` works with pdfL^AT_EX, X_YL^AT_EX, and LuaL^AT_EX, and perhaps other engines as well.

The `doc` package has changed, and `dtxdescribe` is not yet fully compatible. When `dtxdescribe` is used with `doc`, the optional object “class” is currently ignored for macros, and not all classes are indexed.

Also, see change in setup of `hyperref` and `cleveref`: Section 2 on page 7.

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 6 |
| 2 | Using dtxdescribe | 7 |
| 3 | The macros, and the dtxexample environment | 9 |
| 3.1 | Macros and environments | 9 |
| 3.2 | Arguments | 9 |
| 3.3 | Booleans, lengths, counters, hooks, keys | 10 |
| 3.4 | Packages, classes, options | 10 |
| 3.5 | Files, programs, commands | 11 |
| 3.6 | Other source objects | 11 |
| 3.7 | In a description environment | 11 |
| 3.8 | Defaults | 12 |
| 3.9 | Nesting | 12 |
| 3.10 | \margintag, \watchout | 13 |
| 3.11 | dtxexample environment | 13 |
| 3.12 | noindmacro and noindenvironment environments | 13 |
| 3.13 | sourceverb, sourcedisplay, UIDisplay, docsidebar | 14 |
| 3.14 | Formatted objects | 15 |
| 3.14.1 | L ^A T _E X objects | 15 |
| 3.14.2 | Programs and commands | 15 |
| 3.14.3 | File types | 16 |
| 3.14.4 | Internet | 16 |
| 3.14.5 | Specific programs | 16 |
| 3.14.6 | Acronyms, brand names, trademarks | 17 |
| 3.15 | Logos | 17 |
| 3.16 | Dashes and slashes | 17 |

| | | |
|----------|---|-----------|
| 4 | Examples | 19 |
| 5 | Usage notes | 32 |
| 6 | Code | 33 |
| 6.1 | Required packages | 33 |
| 6.2 | Warning sign | 35 |
| 6.3 | Special character handling | 35 |
| 6.4 | Patching hypdoc | 35 |
| 6.5 | Gobbling comment characters | 36 |
| 6.6 | Vertical spacing | 37 |
| 6.7 | ltxdoc emulation | 37 |
| 6.8 | doc emulation | 37 |
| 6.9 | Support macros | 38 |
| 6.10 | Key handling for object classes | 42 |
| 6.11 | Handling <code>\marginpar</code> inside a float | 43 |
| 6.12 | Handling glossary | 43 |
| 6.13 | Catcode handling to support underscores | 43 |
| 6.14 | <code>\DescribeMacro</code> and <code>\DescribeEnvironment</code> | 47 |
| 6.15 | New <code>\Describe. . . macros</code> | 48 |
| 6.16 | <code>\DescribeDefault</code> | 57 |
| 6.17 | <code>\ItemDescribeMacro</code> , etc. | 57 |
| 6.18 | <code>\margintag</code> , <code>\watchout</code> | 60 |
| 6.19 | Nesting | 61 |
| 6.20 | The <code>dtxexample</code> environment | 61 |
| 6.21 | <code>noindmacro</code> and <code>noindenvironment</code> | 64 |
| 6.22 | <code>sourcedisplay</code> , <code>UIDisplay</code> , <code>docsidebar</code> | 64 |
| 6.23 | Formatted objects | 66 |
| 6.23.1 | L ^A T _E X objects | 66 |

| | |
|--|---------------|
| 6.23.2 Programs and commands | 67 |
| 6.23.3 File types | 68 |
| 6.23.4 Internet | 69 |
| 6.23.5 Specific programs | 70 |
| 6.23.6 Acronyms, brand names, trademarks | 70 |
| 6.24 Logos | 71 |
| 6.25 Dashes and slashes | 72 |
| 7 Compiling dtxdescribe | 74 |
| Change History and Index | 75 |

List of Examples

| | |
|--|----|
| 1 Macros | 19 |
| 2 Environment | 20 |
| 3 Second Environment | 20 |
| 4 Booleans and Counters | 21 |
| 5 Lengths | 21 |
| 6 Packages, Classes, and Options | 22 |
| 7 Files, Commands, and Programs | 22 |
| 8 Keys | 23 |
| 9 Arguments | 24 |
| 10 Object | 25 |
| 11 Other | 25 |
| 12 Description environments | 26 |
| 13 Nesting | 27 |
| 14 dtxexample | 28 |
| 15 fsourceverb | 29 |

| | |
|-------------|---|
| dtxdescribe | 5 |
|-------------|---|

| | | |
|----|----------------------------|----|
| 16 | <code>sourcedisplay</code> | 29 |
| 17 | <code>Udisplay</code> | 30 |
| 18 | <code>docsidebar</code> | 31 |

List of Figures

| | | |
|---|----------|----|
| 1 | A Figure | 28 |
|---|----------|----|

1 Introduction

The `doc` package provides `\DescribeMacro` and `\DescribeEnv` to help document new macros and environments. Each generates a heading in the documentation, to which `\marg`, `\oarg`, and `\parg` may be added to identify arguments to be passed to the new object. Their names are added to the margin, and index entries are added, as well as group of entries for environments.

`dtxdescribe` extends this concept to include a number of additional objects, such as booleans and keys. To help identify what is being described in the margin, small tags are added to the name, such as “Env”, “Bool”, or “Key”. These new objects are also listed in the index with the same tag shown after their names, and also by group. Optional classes may be used to further categories index entries.

Modifications have been made to interact with `hyperref` to provide hyper links for regular index entries as well as the new `\Describe` entries.

Additional macros are provided to generate colored margin tags and warnings, and a new `dtxexample` environment demonstrates code examples.

This documentation and its index show examples of these macros in use.

While the index may appear to be overkill for a small package, keep in mind that it includes a number of fictional entries from the examples. Extensive cross-referencing can be useful for larger works. And, of course, you need not cross-reference everything!

2 Using dtxdescribe

To use dtxdescribe with the ltxdoc classa and .dtx files:

```
%<*driver>
\documentclass{ltxdoc}
...
\usepackage{lmodern}
...
\usepackage{dtxdescribe}
...
\usepackage{packagename} % the name of your new package
...

% hyperref now automatically loaded by \pkg{ltxdoc}.

\AtBeginDocument{
  \hypersetup{...}%           If needed.
  \pdfstringdefDisableCommands{ ... }%  If needed.
}

\AddToHook{begindocument/before}{% Before .aux file is loaded.
  \usepackage{cleveref}%           If needed.
  \crefname{somename}{name}{names}%  If needed.
}

...
%</driver>
```

To use dtxdescribe with the regular classes, such as article, use:

```
\documentclass{article}

\usepackage{dtxdescribe}

\usepackage{hyperref}%           If needed.
  \hypersetup{...}%           If needed.
  \pdfstringdefDisableCommands{ ... }%  If needed.

\usepackage{cleveref}%           If needed.
  \crefname{somename}{name}{names}%  If needed.


\begin{document}
...
\end{document}
```

Various objects inside the dtx or tex file may be described with new macros such as `\DescribeBoolean`, `\DescribeLength`, `\DescribeCounter`, similar to the already-familiar `\DescribeMacro` and `\DescribeEnv`.

Optional “classes” may be assigned to the objects being described, including the new versions of `\DescribeMacro` and `\DescribeEnv`. These classes are printed in the margin tag and index entry for each item, and also generate additional index entries sorted by class. This is especially useful for key/value sets, where several sets may appear in the same document.

`inside a float` The margin tag is not printed if the `\Describe` macros are used inside a float such as a table, but the index entries are still made.

`\margintag{text}` `\margintag{text}` may be used to place a colored tag in the margin to summarize paragraph contents or draw attention to an index destination.

 `\watchout[text]` `\watchout[optional text]` may be used to place a red warning sign in the margin, along with optional text.

The `dtxexample` environment may be used to typeset and execute small pieces of \LaTeX code as examples of its use. Optional cross-referencing notes may be used to refer to any example float being generated.

3 The macros, and the dtxexample environment

3.1 Macros and environments

macro (*env*) These are only provided by the `ltxdoc` class and `doc` package to document a `.dtx` file, where comments are used by `docstrip` to disable these environments in the resulting `.sty` file. When using the regular document classes, the macro and environment environments would localize any definitions, and `\DescribeMacro` and `\DescribeEnv` should be used instead.

⚠ **.dtx only**

`\DescribeMacro` [*class*] {*\name*}

The preexisting macro from the `doc` package is redefined to create hyperlinked index entries, and include an optional class. A margin tag is created and an index entry is made. When the optional class is used, it is displayed in front of the margin tag, and is used to group an index entry by macro name and another index entry by class. An example would be to describe the float creation and caption setup for a new class of float, such as the `dtxexample` float and the example “photograph” float both found in the index for this document. See example 1 on page 19 for examples.

`\DescribeEnv` [*class*] {*environment name*}

The preexisting macro from the `doc` package is redefined to create hyperlinked index entries, include an optional class, and also to place an ‘Env’ tag in front of the name in the margin. See example 2 on page 20.

3.2 Arguments

The `\Describe. . .` macros may be followed by `\marg`, `\oarg`, and `\parg` to describe arguments passed to the macros.

`\marg` {*text*}

Shows a mandatory argument for a macro or environment.

The results looks like {*mandatory*}.

`\oarg` {*text*}

Shows an optional argument for a macro or environment.

The results looks like [*optional*].

`\parg` {*text*}

Used for “picture” arguments, such as coordinates.

The result looks like (*coordinate*).

`\DescribeArgument` [*class*] {*argument*}

May be used to describe actions taken when given certain macro arguments. These will be given an ‘Arg’ margin tag and will appear in the index. The class may be used to categorize arguments by their macro or environment name. See example 9 on page 24.

3.3 Booleans, lengths, counters, hooks, keys

See example 4 on page 21.

`\DescribeBoolean` [*⟨class⟩*] {*⟨name⟩*}

Describes a boolean. Given a ‘Bool’ tag in the margin and index.

`\DescribeLength` [*⟨class⟩*] {*⟨name⟩*}

Describes a length. Given a ‘Len’ tag in the margin and index.

`\DescribeCounter` [*⟨class⟩*] {*⟨name⟩*}

Describes a counter. Given a ‘Ctr’ tag in the margin and index.

`\DescribeHook` [*⟨class⟩*] {*⟨name⟩*}

Describes a hook. Given a ‘Hook’ tag in the margin and index. The class may be used to categorize hooks by package. Example:

```
\DescribeHook[LaTeX]{para/begin}
```

`\DescribeKey` [*⟨class⟩*] {*⟨name⟩*}

Describes a key. Given a ‘Key’ tag in the margin and index. The class may be used to categorize keys by their key/value group. See example 8 on page 23.

3.4 Packages, classes, options

`\DescribePackage` [*⟨class⟩*] {*⟨name⟩*}

Describes a package. Given a ‘Pkg’ tag in the margin and index.

`\DescribeClass` [*⟨class⟩*] {*⟨name⟩*}

Describes a L^AT_EX class. Given a ‘Cls’ tag in the margin and index.

`\DescribeOption` [*⟨class⟩*] {*⟨name⟩*}

Describes a L^AT_EX package or class option. Given an ‘Opt’ tag in the margin and index.

3.5 Files, programs, commands

`\DescribeFile` [*class*] {*name*}

Describes an operating-system file. Given a ‘File’ tag in the margin and index. The filename may have underscores.

`\DescribeProgram` [*class*] {*name*}

Describes an operating-system program. Given a ‘Prog’ tag in the margin and index. The program name may have underscores.

`\DescribeCommand` [*class*] {*name*}

Describes an operating-system command. Given a ‘Cmd’ tag in the margin and index. The command name may have underscores.

3.6 Other source objects

`\DescribeObject` [*class*] {*name*}

Describes an arbitrary programming object, such as a color definition or caption setup. A margin tag and index entry are created with `\ttfamily` type. When a class is used, it is pre-pended to the margin tag, appended to the index entry, and a second index entry is created grouped by class. If a macro name is to be described, use `\DescribeMacro` instead. See example 10 on page 25.

`\DescribeOther` [*class*] {*name*}

Describes an arbitrary non-programming object, such as a license agreement or credits. A margin tag and index entry are created in roman type. When a class is used, it is pre-pended to the margin tag, appended to the index entry, and a second index entry is created grouped by class. See example 11 on page 25.

3.7 In a description environment

To describe an object using a description environment, use the following. See example 12 on page 26.

`\ItemDescribeMacro` [*class*] {\name} A description.

`\ItemDescribeEnv` [*class*] {*name*} A description.

`\ItemDescribeArgument` [*class*] {*argument*} A description.

`\ItemDescribeBoolean` [*class*] {*name*} A description.

`\ItemDescribeLength` [*class*] {\name} A description.

| | | |
|-----------------------------------|---|-------------------|
| <code>\ItemDescribeCounter</code> | <code>[\langle class \rangle] {\langle name \rangle}</code> | A description. |
| <code>\ItemDescribeHook</code> | <code>[\langle class \rangle] {\langle name \rangle}</code> | A description. |
| <code>\ItemDescribeKey</code> | <code>[\langle class \rangle] {\langle name \rangle}</code> | A description. |
| <code>\ItemDescribePackage</code> | <code>[\langle class \rangle] {\langle package_name \rangle}</code> | With underscores. |
| <code>\ItemDescribeClass</code> | <code>[\langle class \rangle] {\langle class_name \rangle}</code> | With underscores. |
| <code>\ItemDescribeOption</code> | <code>[\langle class \rangle] {\langle name \rangle}</code> | A description. |
| <code>\ItemDescribeFile</code> | <code>[\langle class \rangle] {\langle file_name \rangle}</code> | With underscores. |
| <code>\ItemDescribeProgram</code> | <code>[\langle class \rangle] {\langle program_name \rangle}</code> | With underscores. |
| <code>\ItemDescribeCommand</code> | <code>[\langle class \rangle] {\langle command_name \rangle}</code> | With underscores. |
| <code>\ItemDescribeObject</code> | <code>[\langle class \rangle] {\langle name \rangle}</code> | A description. |
| <code>\ItemDescribeOther</code> | <code>[\langle class \rangle] {\langle name \rangle}</code> | A description. |

3.8 Defaults

| | | |
|--------------------------------------|--------------------------------------|---|
| <code>\DescribeDefault</code> | <code>{\langle value \rangle}</code> | |
| <code>Default: value</code> | | Shows the default value of a <code>\Describe. . .</code> item, such as displayed here. Place this macro immediately after the <code>\Describe. . .</code> macro and any arguments, but before the text description. |
| <code>\DescribeDefaultcolor</code> | | The color of the margin tag used to show the default value. This is used by <code>\textcolor</code> to create the margin tag. |
| <code>Default: green!50!black</code> | | |

3.9 Nesting

`\shownesting` * `[\langle fraction of \linewidth \rangle] {\langle container name \rangle} {\langle contents \rangle}`

It may be useful to show which objects contain which other objects. `\shownesting` shows a box enclosing a name for the container, and the container's contents. `\shownesting` be nested, showing boxes inside other boxes, which displays how each environment and macro is fit together inside each other.

The optional argument is the fraction of `\linewidth` to use for the box, from `[0]` to `[1]`. The default is `[1]`. Each `\shownesting` starts its own paragraph, unless the star `*` is used, in which case the `\shownesting*` appears inline with previous text. To place two `\shownesting` boxes side-by-side, use optional arguments to specify less than full `\linewidth` for each box, and use `\shownesting*` for the second box to place it inline.

See example 13 on page 27 for an example.

3.10 `\margintag`, `\watchout`

`\margintag` $\{\langle text \rangle\}$

`\margintag{example}`

Creates a colored margin tag. May be used to identify the topic of a paragraph or the destination of an arbitrary index entry.

`\margintagcolor` The color of the `\margintag`.

Default: `blue!70!black`

`\watchout` $[\langle text \rangle]$

 `\watchout{example}`

Creates a red margin tag with a warning sign and optional text. May be used to warn the reader of special instructions, etc. Without the optional text the warning sign is displayed by itself.

`\watchoutcolor` The color of the `\watchout`.

Default: `red!50!black`

3.11 `dtxexample` environment

`dtxexample (env)` * $[\langle Notes/cross-references \rangle] \{\langle caption \& label \rangle\}$

The `dtxexample` environment is useful for demonstrating a piece of \LaTeX code. The example is a simulated float with its own caption and optional label, along with optional notes and/or cross-referencing commands. The contents of the `dtxexample` environment are printed verbatim, then loaded and executed as \LaTeX code, showing the results just below the printed code. In the case of float commands, the floats are generated as expected somewhere nearby, and should be given their own labels. References to the float's labels may be placed in the optional argument to the `dtxexample` environment, and will be printed below the code.

The unstarred version places the code inside a `minipage`, forbidding a page break in the middle of the code listing. The starred version does not use a `minipage`. This is required when the code is too large to fit on a single page.

See example 14 for a demonstration of how `dtxexample` works.

`\dtxexamplecodename` The text name of the code section.

Default: `Code:`

`\dtxexampleresultname` The text name of the result section.

Default: `Result:`

3.12 `noindmacro` and `noindenvironment` environments

 `.dtx only`

These are like `macro` and `environment`, but not indexed. These only make sense if using the `ltxdoc` class and `doc` package to document a `.dtx` file, where comments are used by `docstrip` to disable these environments in the resulting `.sty` file. When using the regular document classes, `noindmacro` and `noindenvironment` environments should not be used, as they would localize any definitions. `\DescribeMacro` and `\DescribeEnv` should be used instead.

`noindmacro (env)` `{\name}` To document macros which should not be included in the index.

`noindenvironment (env)` `{\name}` To document environments which should not be included in the index.

Replace

```
\begin{macro}{\macroname} \oarg{optional} \marg{mandatory}
...
\end{macro}
```

with

```
\begin{noindmacro}{\macroname} \oarg{optional} \marg{mandatory}
...
\end{noindmacro}
```

and similarly for `noindenvironment`.

3.13 sourceverb, sourcedisplay, UIDisplay, docsidebar

`sourceverb (env)` [`{key/values}`] Display source code verbatim. Uses optional `fancyvrb` keys. Includes `gobble=2` to absorb the leading % and space character of a dtx file source format. Because this is a verbatim environment, it *cannot* be used inside a macro.

Default: `gobble=2,`
`tabsize=4,`
`xleftmargin=2em`

`fsourceverb (env)` [`{key/values}`] Display source code verbatim inside a frame. A label may be included using the `label` key. Because this is a verbatim environment, it *cannot* be used inside a macro. See example 15 on page 29.

Default: `gobble=2,`
`tabsize=4,`
`xleftmargin=2em,frame=lines`

`sourcedisplay (env)` Display source code with manual formatting. This is not a verbatim environment. `\textcolor`, `\textbf`, and `\emph` may be used to highlight text. Macros must be escaped with `\cs`, characters such as `{` must be produced with `\{`, etc. `\` must be used to force a new line. `\fquad`, `\fqquad`, and `\fqqqquad` may be used to force indenting. Because this is *not* a verbatim environment, it *can* be used inside a macro. See example 16 on page 29.

`\fquad` Single-level indent inside a `sourcedisplay`.

`\fqquad` Double-level indent inside a `sourcedisplay`.

`\fqqqquad` Triple-level indent inside a `sourcedisplay`.

`UIDisplay (env)` Displays a user interface, such as a dialog box entry or a menu selection. See example 17 on page 30. Also see the `\UI` macro..

`\userentry` `{\text to enter}` Typeset something for the user to enter. Also see the `\cmds` macro.

`\userentryname` Text to tell the user to enter the following item. Change with `\renewcommand`.
 Default: `Enter` ⇒

`docsidebar (env)` [$\langle title \rangle$] Creates a sidebar within the document. See example 18 on page 31.

3.14 Formatted objects

Macros to format references to various kinds of objects.

This dtxdescribe package documentation uses `erewhon`, `cabin`, and `inconsolata`, along with `metalogo`, to demonstrate the following font effects.

3.14.1 L^AT_EX objects

`\pkg` [$\langle packagename \rangle$] Prints as `packagename`. Also for a `classname`.

`\cs` [$\langle csname \rangle$] Prints as `\csname`.

`\env` [$\langle environment \rangle$] Prints as `environment`.

`\marg` [$\langle argument \rangle$] Prints $\langle arg \rangle$. Mandatory argument.

`\oarg` [$\langle argument \rangle$] Prints [$\langle arg \rangle$]. Optional argument.

`\parg` [$\langle argument \rangle$] Prints ($\langle arg \rangle$). Picture-mode argument.

`\ctr` [$\langle counter \rangle$] Prints as `counter`.

`\bool` [$\langle boolean \rangle$] Prints as `boolean`.

`\optn` [$\langle option \rangle$] Prints as `option`, for example to a macro, package, class.

`\TOC` TOC: Table of contents.

`\LOF` LOF: List of figures.

`\LOT` LOT: List of tables.

3.14.2 Programs and commands

`\progcode` Prints as inline program code: Escape underscores and other special characters such as `{`, `%`, `$`.

`\prog` Prints as *grep*, *make*: A program name. Underscores allowed.

`\filenm` Prints as `file_name`: Underscores allowed.

`\UI` Prints as **General user-interface text**. What the user sees on the display. Also see the `UIDisplay` environment.

`\cmds` Prints as **Commands to be entered**: What the user enters. Escape underscores and other special characters such as `{`, `%`, `$`. Also see the `\userentry` macro.

3.14.3 File types

| | |
|-------------------|--|
| <code>\ODT</code> | ODT OpenDocument Format word processing document |
| <code>\SVG</code> | SVG image format |
| <code>\PNG</code> | PNG image format |
| <code>\GIF</code> | GIF image format |
| <code>\JPG</code> | JPG image format |
| <code>\EPS</code> | EPS image format |
| <code>\PDF</code> | PDF image format |
| <code>\DVI</code> | DVI image format |

3.14.4 Internet

| | |
|-------------------------|--|
| <code>\UTF</code> | UTF: Unicode |
| <code>\URL</code> | URL: Uniform Resource Locator |
| <code>\element</code> | $\{\langle element name \rangle\}$ Prints as <code><element></code> , an HTML / CSS element |
| <code>\attribute</code> | $\{\langle attribute name \rangle\}$ Prints as <code>attribute</code> , an HTML / CSS attribute. pdfTeX and XeTeX only. Not for LuaTeX. |
| <code>\attrib</code> | $\{\langle attribute name \rangle\}$ Prints as <code>attribute</code> , an HTML / CSS attribute. pdfTeX, XeTeX, or LuaTeX. |
| <code>\HTML</code> | HTML: Hypertext Markup Language |
| <code>\HTMLfive</code> | HTML5: Old-style figure if font supports |
| <code>\CSS</code> | css: Cascading Style Sheet |
| <code>\CSSthree</code> | css3: Old-style figure if font supports |
| <code>\EPUB</code> | EPUB: E-book file format |

3.14.5 Specific programs

| | |
|----------------------|---|
| <code>\TikZ</code> | TikZ: Package logo |
| <code>\CTAN</code> | CTAN: Comprehensive TeX Archive Network |
| <code>\TDS</code> | TDS: TeX Directory Structure |
| <code>\MathML</code> | MathML: Mathematical Markup Language |

`\MathJax` MATHJAX: Math on the web.

3.14.6 Acronyms, brand names, trademarks

`\brand` $\{\langle name \rangle\}$ BRANDNAME, COMPANY NAME

`\acro` $\{\langle acronym \rangle\}$ ACRO: Acronym

`\supregistered` Superscript trademark symbol[®]

3.15 Logos

Several additional logos are provided.

Also see the `metologo` and `metalogox` packages.

`\dviTeX` DVI T_EX

`\dviLaTeX` DVI L^AT_EX

`\pdfTeX` PDF T_EX

`\pdfLaTeX` PDF L^AT_EX

`\LuaTeX` LuaT_EX

`\LuaLaTeX` LuaL^AT_EX

`\XeTeX` X_ET_EX, with reversed E if `graphics` is loaded.

`\XeLaTeX` X_EL^AT_EX, with reversed E if `graphics` is loaded.

`\AmS` $\mathcal{A}\mathcal{M}\mathcal{S}$

`\LyX` LyX

`\BibTeX` B_IB_TE_X

`\MakeIndex` *MakeIndex*

`\ConTeXt` ConT_EXt

`\MiKTeX` MiK_TE_X

3.16 Dashes and slashes

`\thinspace` A breakable thin skip.

`\endash` An endash: –

- `\emdash` An emdash: —
- `\thinbrspace` A thin space which allows a line break.
- `\thinthinbrspace` A very thin space which allows a line break.
- `\Dash` An unbreakable thin space, emdash, and breakable thin space: A—B
- `\dash` An unbreakable thin space, endash, and breakable thin space: A–B
- `\Slash` An unbreakable very thin space, a slash, and a breakable very thin space:

| Command | Result | |
|-------------------------|--------|---------------------------------------|
| | A--B | A–B (not breakable) |
| A <code>\dash</code> B | A – B | (only breakable before the B) |
| A <code>--</code> B | A – B | (breakable before or after the dash) |
| | A---B | A—B (not breakable) |
| A <code>\Dash</code> B | A — B | (only breakable before the B) |
| A <code>---</code> B | A — B | (breakable before or after the dash) |
| | A/B | A/B (not breakable) |
| A <code>\Slash</code> B | A / B | (only breakable before the B) |
| A <code>/</code> B | A / B | (breakable before or after the slash) |
| A <code>~/~</code> B | A / B | (not breakable) |

4 Examples

Example 1: Macros

Code:

```
\DescribeMacro{\mymacro} \oarg{optional} \marg{mandatory}
  A typical macro definition.
```

```
\DescribeMacro[photograph]{\DeclareFloatingPhoto}
Create a photograph float environment.
```

```
\DescribeMacro[c=photograph]{\photocaptionsetup}
Caption settings for a photograph float.
```

```
\DescribeMacro[photograph]{\cphotonameref}
\pkg{cleveref} name for the photograph float.
```

Result:

| | |
|--|---|
| <code>\mymacro</code> | <code>[<i><optional></i>] {<i><mandatory></i>}</code> A typical macro definition. |
| <code>\DeclareFloatingPhoto</code> | Create a photograph float environment. |
| <code>[photograph]</code> | |
| <code>\photocaptionsetup</code> | Caption settings for a photograph float. |
| <code>[photograph]</code> | |
| <code>\cphotonameref [photograph]</code> | <code>cleveref</code> name for the photograph float. |

The optional class is used to label and group tags and index entries. See this document's index entries for examples of this "photograph" class and the `dtxexample` class of macros.

The re-defined `\DescribeMacro`, `\DescribeEnv`, and all the following macros create [hyperlinks](#) hyperlinked index entries, along with regular uses of `\index`.

Example 2: Environment

Code:

```
\DescribeEnv{myenvironment} \marg{argument} Short description.
```

Result:

```
myenvironment (env.)  {(argument)} Short description.
```

add'l tags

The re-defined `\DescribeEnv` adds an 'Env' tag to the margin, and adds "(environment)" to its own index entry. Note that environments and all the other new objects defined by this package each receives two index entries, one by name, and one grouped with others of its kind.

index groups**too much text**

Example 2 shows descriptive text on the same line as the `\DescribeEnvironment`. For macros and environments with many arguments after the name, it may be better to place any additional text in a following paragraph.

Example 3: Second Environment

Code:

```
\DescribeEnv[kindofenvironment]{otherenvironment}
  \oarg{opt args} \parg{coordinates} A description.
```

Result:

```
otherenvironment (env.)  [(opt args)] ((coordinates)) A description.
[kindofenvironment]
```

The `otherenvironment` will be indexed by itself and also with `myenvironment` under the index entry "environments", and also under the class `kindofenvironment`.

Example 4: Booleans and Counters

Code:

```
\DescribeBoolean[examples]{sampleboolean} Some description.
```

```
\DescribeCounter[examples]{samplecounter} Some description.
```

Result:

sampleboolean (*bool*) [examples] Some description.

samplecounter (*Ctr*) [examples] Some description.

Most of the new `\Describe_____` macros behave like the new `\DescribeEnv`, placing a tag in the margin, an index entry by name, and another index entry by group.

Example 5: Lengths

Code:

```
\DescribeLength[photograph]{\photowidth} Some description.
```

Result:

`\photowidth` (*Len*) [photograph] Some description.

Lengths have a leading backslash, but are otherwise described the same as the rest of the objects.

Example 6: Packages, Classes, and Options

Code:

```
\DescribePackage[examples]{samplepackage}
  About a \LaTeX\ package.

\DescribeClass[examples]{sample_class}
  About a \LaTeX\ class.

\DescribeOption[examples]{sampleoption}
  About an option for a package or class.
```

Result:

| | |
|---|--|
| samplepackage (<i>Pkg</i>) [examples] | About a L ^A T _E X package. |
| sample_class (<i>Cls</i>) [examples] | About a L ^A T _E X class. |
| sampleoption (<i>Opt</i>) [examples] | About an option for a package or class. |

Example 7: Files, Commands, and Programs

Code:

```
\DescribeFile[bigfiles]{really_big_file.txt} Some description.

\DescribeFile[bigfiles]{another_big_file.txt} Some description.

\DescribeFile{lone_file.txt} Some description.

\DescribeCommand{OS_command} An operating-system command.

\DescribeProgram{program_name} An operating-system program.
```

Result:

| | |
|--------------------------------------|------------------------------|
| really_big_file.txt (<i>file</i>) | Some description. |
| [bigfiles] | |
| another_big_file.txt (<i>file</i>) | Some description. |
| [bigfiles] | |
| lone_file.txt (<i>file</i>) | Some description. |
| OS_command (<i>Cmd</i>) | An operating-system command. |
| program_name (<i>Prog</i>) | An operating-system program. |

Filenames, program names, and command names may have underscores, such as tested here. A class is used to group “bigfiles” together in the index.

Example 8: Keys

Code:

```
\DescribeKey[groupofkeys]{firstkey} About the first key
    of the |groupofkeys| set.

\DescribeKey[groupofkeys]{secondkey} About the second key
    of |groupofkeys|.

\DescribeKey[examples]{samplekey} About some key of |otherkeys|.

\DescribeKey[examples]{sampletwokey} About another key of |otherkeys|.

\DescribeKey{lonekey} A key without a class.
```

Result:

```
firstkey (Key) [groupofkeys]  About the first key of the groupofkeys set.
secondkey (Key) [groupofkeys] About the second key of groupofkeys.
  samplekey (Key) [examples]  About some key of otherkeys.
sampletwokey (Key) [examples] About another key of otherkeys.
  lonekey (Key)                A key without a class.
```

See the index key groups.

Example 9: Arguments

Code:

```
\DescribeArgument[figure]{[H]}
What happens when a figure is [H]ere.
```

```
\DescribeArgument[figure]{[M]}
What happens when a figure is in the [M]argin.
```

```
\DescribeArgument[\mymacro]{bold}
What happens when \cs{mymacro} is given the |bold| argument.
```


Result:

[H] (*Arg*) [figure] What happens when a figure is [H]ere.

[M] (*Arg*) [figure] What happens when a figure is in the [M]argin.

bold (*Arg*) [\mymacro] What happens when \mymacro is given the bold argument.

Arguments behave like keys, and may have an optional class to identify their macro or environment, and group their entries in the index.

 **macro names** Note you may need to use `\cs{mymacro}` for the macro's name.

Example 10: Object

Code:

```

\DescribeObject[color]{somecolor}
    The color of something.

\DescribeObject[color]{othercolor}
    The other color.

\DescribeObject{randomobject} About some random object.

```

Result:

| | |
|--------------------|---------------------------|
| somecolor [color] | The color of something. |
| othercolor [color] | The other color. |
| randomobject | About some random object. |

Describes an arbitrary programming object, using `\ttfamily` text.

Example 11: Other

Code:

```

\DescribeOther{license agreement}
The following is the fictional license agreement:

\DescribeOther{Before myenvironment}
    Actions to be done \cs{BeforeBeginEnvironment}.

\DescribeOther[otherclass]{Other Item} About the other item.

\DescribeOther[otherclass]{Additional Item} About the add'l item.

```

Result:

| | |
|------------------------------|---|
| license agreement | The following is the fictional license agreement: |
| Before myenvironment | Actions to be done \BeforeBeginEnvironment. |
| Other Item [otherclass] | About the other item. |
| Additional Item [otherclass] | About the add'l item. |

Describes an arbitrary non-programming object, using roman text.

Example 12: Description environments

Code:

```
\begin{description}
\ItemDescribeMacro[descexamples]{\macroname} Describe the macro.
\ItemDescribeBoolean[descexamples]{booleanname} Describe the boolean.
\ItemDescribeLength[descexamples]{\lengthname} Describe the length.
\ItemDescribeKey[descexamples]{keyname} Describe the key.
\ItemDescribePackage[descexamples]{package_name} Describe the package.
\ItemDescribeClass[descexamples]{class_name} Describe the class.
\ItemDescribeFile[descexamples]{file_name} Describe the file.
\ItemDescribeProgram[descexamples]{program_name} Describe the program.
\ItemDescribeCommand[descexamples]{command_name} Describe the class.
\end{description}
```

Result:

| | |
|---|--|
| <code>\macroname</code> [descexamples] | \macroname: Describe the macro. |
| <code>booleanname</code> (<i>bool</i>) [descexamples] | booleanname: Describe the boolean. |
| <code>\lengthname</code> (<i>Len</i>) [descexamples] | \lengthname: Describe the length. |
| <code>keyname</code> (<i>Key</i>) [descexamples] | keyname: Describe the key. |
| <code>package_name</code> (<i>Pkg</i>) [descexamples] | package_name: Describe the package. |
| <code>class_name</code> (<i>Cls</i>) [descexamples] | class_name: Describe the class. |
| <code>file_name</code> (<i>file</i>) [descexamples] | file_name: Describe the file. |
| <code>program_name</code> (<i>Prog</i>) [descexamples] | program_name: Describe the program. |
| <code>command_name</code> (<i>Cmd</i>) [descexamples] | command_name: Describe the class. |

Uses a description environment to describe objects.

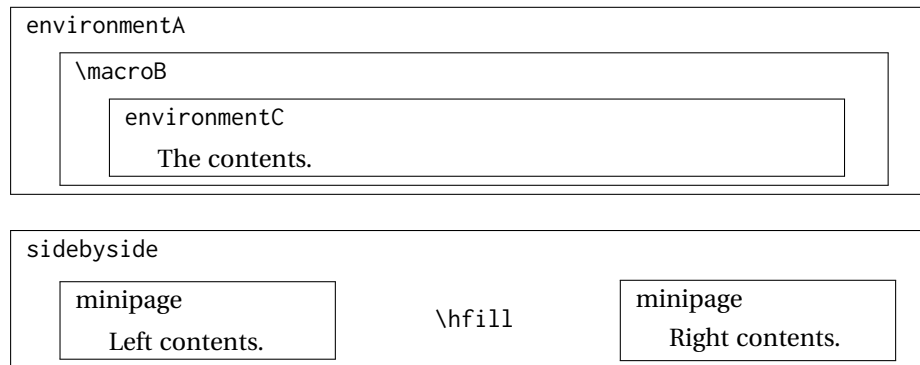
Example 13: Nesting*Code:*

```

\shownesting{\env{environmentA}}{
  \shownesting{\cs{macroB}}{
    \shownesting{\env{environmentC}}{
      The contents.
    }
  }
}

\shownesting{\env{sidebyside}}{
  \shownesting[.35]{minipage}{
    Left contents.
  }
  \hfill \cs{hfill} \hfill
  \shownesting* [.35]{minipage}{
    Right contents.
  }
}

```

Result:

Note the use of the optional arguments to select less than full `\linewidth`, and the starred form for the second box to place it inline with the `\hfill` text.

Contents of the figure.

Figure 1: A Figure

Example 14: dtxexample

Code:

```
\begin{figure}
  \centering\fbbox{Contents of the figure.}
  \caption{A Figure}\label{fig:afigure}
\end{figure}
```

Result:

See *fig. 1*

Example 14, typeset above, was created with the following code:

```
\begin{dtxexample}[See \cref{fig:afigure}]
  {\env{dtxexample}\label{ex:dtxexample}}
\begin{figure}
  \centering\fbbox{Contents of the figure.}
  \caption{A Figure}\label{fig:afigure}
\end{figure}
\end{dtxexample}
```

When the example was created:

1. The “float” of type `dtxexamplefloat` was created, with the caption “dtxexample” and the label `ex:dtxexample`, which points to example 14.
2. The code was displayed verbatim.
3. The code was written to the file `dtxexample_cut.tex`.
4. The code was `\input` from `dtxexample_cut.tex`.
5. Executing the code created the figure with caption “A Figure” and label `fig:afigure`, which points to *fig. 1*.
6. The cross-reference to the figure was shown on the optional display line by the optional argument to `dtxexample`.
7. The starred form of `dtxexample` was used to create the closing rule below the code, since a float was being generated and nothing followed the code inline. An unstarred version would have created an extra rule.

Example 15: fsourceverb*Code:*

```
% \begin{fsourceverb}[label=An fsourceverb example]
% \newcommand{fdosomething}[1][whattodo]{
%   doing #1
% }
% \end{fsourceverb}
```

Result:

An fsourceverb example

```
\newcommand{fdosomething}[1][whattodo]{
  doing #1
}
```

(The leading % characters would be present in the dtx source.)

Example 16: sourcedisplay*Code:*

```
\begin{sourcedisplay}
\cs{newcommand}\{dosomething\}[1][\textcolor{red}{whattodo}]{\{\}
\quad \textcolor{blue}{doing \textcolor{red}{\#1}}\}
\}
\end{sourcedisplay}
```

Result:

```
\newcommand{dosomething}[1][whattodo]{
  doing #1
}
```

Example 17: UIdisplay

Code:

```
Select:
\begin{UIdisplay}
  \textsf{Preferences $\to$ Plugins $\to$ Files $\to$ HTML}
\end{UIdisplay}
For the field
\begin{UIdisplay}
Title heading:
\end{UIdisplay}
\userentry{H1}
```

Result:

Select:

Preferences → Plugins → Files → HTML

For the field

Title heading:

Enter ⇒ **H1**

Example 18: docsidebar

Code:

Main text.

More main text.

```
\begin{docsidebar}[A title]
An aside, which may help explain something
incidental to the main text.
\end{docsidebar}
```

Additional main text.

Result:

Main text.

More main text.

A title

An aside, which may help explain something incidental to the main text.

Additional main text.

5 Usage notes

Placement of `\Describe` macros: Typically L^AT_EX macro and environment definitions are enclosed in macro and environment environments at their place in the source code. `\DescribeMacro` and `\DescribeEnv` would be used elsewhere in the manual to describe how to use the code. `\DescribeBoolean` and such might be at their place in the source code, unless they are worthy of discussion for the end-user, in which case they should be in the “User’s Manual” section of the document.¹ It may be useful to use `\DeclareBoolean` and friends both at the code location and also in the User’s Manual section.

Extra spaces: When placing multiple uses of `\Describe`, `\index`, `\margintag`, and `\watchout` macros together, care must be taken to avoid extra space in the printed text where these macros occur. A trailing percent character may be used to avoid the extra space:

```
text text text% <-- avoids extra space
\margintag{A comment.}
\index{An entry}
\index{Another entry}
more inline text
```

Unwanted vertical space: Other environments nested inside a docsidebar may produce excessive vertical space. It may be required to insert

```
\vspace*{-\baselineskip}
```

`\margintag` placement: To have the margin tag appear next to the first line of a paragraph, place the `\margintag` or `\watchout` somewhere after the first few words in the paragraph. The `\margintag` may be on its own line, and the rest of the paragraph may follow on the next line. If too many words are printed before the `\margintag`, the words may wrap to the next line before the tag occurs.

Margin tag overlap: To keep margin tags in proper alignment, use a new paragraph or multiple lines between `\margintag`, `\watchout`, or `\Declare` macros

[missing tags](#) **`\Describe` inside floats:** When these macros are used inside a float, the margin tag is suppressed (there is no margin in a float), but the index entries are still created.

¹Future versions may include `\DeclareBoolean` for use at the point where the boolean is defined, creating an index entry with a code line number, and `\DescribeBoolean` with a page number index entry for the related discussion in the User’s Manual portion of the document.

6 Code

6.1 Required packages

`makeidx` (*Pkg*) One of several index programs must be provided. One of several index programs must be provided.

`splitidx` (*Pkg*)

```
1 \AtBeginDocument{
2   \ifpackageloaded{makeidx}{}{
3     \ifpackageloaded{splitidx}{}{
4       \RequirePackage{makeidx}
5       \makeindex
6     }}
7 }
```

`etoolbox` (*Pkg*) v2.6 or later for `\BeforeBeginEnvironment`, `\AfterEndEnvironment`

```
8 \RequirePackage{etoolbox}[2011/01/03]%
```

`xparse` (*Pkg*) Used for the examples.

```
9 \RequirePackage{xparse}
```

`calc` (*Pkg*) Used for `\shownesting`.

```
10 \RequirePackage{calc}
```

`xcolor` (*Pkg*) Used for the examples.

```
11 \RequirePackage{xcolor}
12 \definecolor{myurlcolor}{rgb}{0,0,.7}
13 \definecolor{mylinkcolor}{rgb}{.7,0,0}
```

`caption` (*Pkg*) Used for the examples.

```
14 \RequirePackage{caption}
```

`newfloat` (*Pkg*) Used for the examples.

```
15 \RequirePackage{newfloat}
```

`fancyvrb` (*Pkg*) Used for the examples.

```
16 \RequirePackage{fancyvrb}
```

`xstring` (*Pkg*) Used for `\StrSubstitute` for `\DescribeFile`.

```
17 \RequirePackage{xstring}
```

`hyperref` (*Pkg*) If `hyperref` is loaded, disable some macros in PDF bookmarks:
[PDF bookmarks](#)

```
18 \AtBeginDocument{
19   \ifpackageloaded{hyperref}{
20     \pdfstringdefDisableCommands{%
21       \def\quad{ }%
22       \def\{\ }%
23       \def\pkg#1{#1}%
24       \def\ctr#1{#1}%
25       \def\bool#1{#1}%
26       \def\optn#1{#1}%
27       \def\env#1{#1}%
28       \def\cs#1{\textbackslash#1}%
29       \def\,{ }%
30       \def\LuaLaTeX{LuaLaTeX}%
31       \def\XeLaTeX{XeLaTeX}%
32       \def\TeX{TeX}%
33       \def\LaTeX{LaTeX}%
34       \def\LaTeXe{LaTeX2e}%
35       \def\LuaTeX{LuaTeX}%
36       \def\LuaLaTeX{LuaLaTeX}%
37       \def\XeTeX{XeTeX}%
38       \def\AmS{AMS}%
39       \def\Dash{ --- }%
40       \def\dash{ -- }%
41       \def\Slash{/}%
42       \def\prog#1{\detokenize{#1}}%
43       \def\progcode#1{#1}%
44       \def\filenm#1{\detokenize{#1}}%
45       \def\brand#1{#1}%
46       \def\acro#1{#1}%
47       \def\ODT{ODT}%
48       \def\SVG{SVG}%
49       \def\PNG{PNG}%
50       \def\GIF{GIF}%
51       \def\JPG{JPG}%
52       \def\EPS{EPS}%
53       \def\PDF{PDF}%
54       \def\DVI{DVI}%
55       \def\UTF{UTF}%
56       \def\URL{URL}%
57       \def\element#1{#1}%
58       \def\attribute#1{#1}%
59       \def\attrib#1{#1}%
60       \def\HTML{HTML}%
61       \def\HTMLfive{HTML5}%
62       \def\CSS{CSS}%
63       \def\CSSthree{CSS3}%
64       \def\E PUB{EPUB}%
65       \def\TOC{TOC}%
66       \def\LOF{LOF}%
67       \def\LOT{LOT}%
68     }
69   }% yes hyperref
```


If `hyperref` is not loaded, emulate `\hyperpage` here.

```
70   {% no hyperref
71     \newcommand*\hyperpage}[1]{#1}
72   }
73 }
```

`pict2e` (*Pkg*)

```
74 \RequirePackage{pict2e}
75 \setlength{\unitlength}{1pt}
```

6.2 Warning sign

`\warningsign` Prints an exclamation point inside a triangle. Displays as: 

Creates a warning sign without relying on the presence of the fourier font. During copy/paste, this shows up as a simple exclamation point.

```
76 \newcommand*\warningsign){%
77 \begin{picture}(10,9)
78 \put(4,1){\scriptsize!}
79 \put(0,0){\line(500,866){5}}
80 \put(10,0){\line(-500,866){5}}
81 \put(0,0){\line(1,0){10}}
82 \end{picture}
83 }
```

6.3 Special character handling

The literal backslash character:

```
84 \begingroup
85 \catcode'\|=0
86 \catcode'\|=12
87 |gdef|DTXD@backslash{\}
88 |endgroup
```

6.4 Patching hypdoc

If `hyperref` is disabled (by `lwarp`) then define the missing `\hdclindex`.

```
89 \@ifpackageloaded{doc}{
90 \@ifpackageloaded{hypdoc}{
91 \AddToHook{begindocument/before}[doc/hyperref]{%
92 \ifdoc@hyperref
93 \else
94 \def\hdclindex#1#2{%
```

```

95   \ifx\@nil#2\@nil\else\csname #2\expandafter\endcsname\fi%
96   }%
97   \fi
98   }
99   }{}% hypdoc loaded
100  }{}% doc loaded

```

`splitidx` is modified to add `|hdpindex{}` to work with `hypdoc`.

```

101 \AtBeginDocument{
102 \ifpackageloaded{hypdoc}{
103 \ifpackageloaded{splitidx}{
104 \renewcommand*{\@wrsindex}[2][{}]{%
105   \ifx\relax#1\relax
106     \if@splitidx
107       \@wrsindex[idx]{#2}%
108     \else
109       \def\@tempa{#2}%
110       \if@verbindex\@onelevel@sanitize\@tempa\fi
111       \@wrsindex{\@tempa}%
112     \fi
113   \else
114   %
115   %   \def\@tempa{#2}%
116   %   \def\@tempa{#2\encapchar hdpindex{}}%           dtxdescribe
117   %
118   %   \csname index@#1@hook\endcsname
119   %   \expandafter\ifx\csname @wrsindex\endcsname\relax
120   %     \@wrsindex{#1}{\@tempa}{\thepage}%
121   %   \else
122   %     \def\@tempb{\@wrsindex{#1}}%
123   %     \expandafter\@tempb\@tempa||\%
124   %   \fi
125   %   \endgroup
126   %   \@esphack
127   %   \fi
128   % }
129 }{}% splitidx loaded
130 }{}% hypdoc loaded
131 }{}% AtBeginDocument

```

6.5 Gobbling comment characters

`DTXD@gobble` The `.dtx` format uses leading percent characters for code to be in the documentation only. Other classes do not.

```

132 \ifpackageloaded{doc}{
133   \newcommand*{\DTXD@gobble}{2}
134 }{}
135   \newcommand*{\DTXD@gobble}{0}
136 }

```

6.6 Vertical spacing

```

137 \setlength{\marginparsep}{1em}
138 \setlength{\marginparpush}{.7ex}
139
140 \setlength{\parindent}{0em}
141 \setlength{\parskip}{2ex}

```

`\IndexMin` (*Len*) From `ltxdoc`.

```

142 \ifdef{\IndexMin}
143   {\setlength{\IndexMin}{40ex}}
144   {\newlength{\IndexMin}}

```

6.7 ltxdoc emulation

If the `ltxdoc` class is not used, some of its macros are replicated here.

```

145 \@ifclassloaded{ltxdoc}{}{
146   \def\cmd#1{\cs{\expandafter\cmd@to@cs\string#1}}
147   \def\cmd@to@cs#1#2{\char\number'#2\relax}
148   \DeclareRobustCommand\cs[1]{\texttt{\char'\#1}}
149   \providecommand\marg[1]{%
150     {\ttfamily\char'\{} \meta{#1} {\ttfamily\char'\}}}
151   \providecommand\oarg[1]{%
152     {\ttfamily[] \meta{#1} {\ttfamily}}}
153   \providecommand\parg[1]{%
154     {\ttfamily() \meta{#1} {\ttfamily}}}
155   \providecommand\url{\texttt}
156 }

```

6.8 doc emulation

If the `doc` class is not used, some of its macros are replicated here.

```

157 \AtBeginDocument{
158   \ifpackageloaded{doc}{}{
159     \newenvironment*{macro}[1]{%
160       \PackageError{dtxdescribe}
161         {The 'macro' environment is only\MessageBreak
162           available when using the doc package\MessageBreak
163           with a .dtx source file}
164       {This environment only makes sense for .dtx source.}
165     }{}
166     \newenvironment*{environment}[1]{%
167       \PackageError{dtxdescribe}
168         {The 'environment' environment is only\MessageBreak
169           available when using the doc package\MessageBreak
170           with a .dtx source file}
171       {This environment only makes sense for .dtx source.}
172     }{}

```

```

173     \def\MacroFont{\fontencoding\encodingdefault
174                 \fontfamily\ttdefault
175                 \fontseries\mddefault
176                 \fontshape\updefault
177                 \small}%
178     \@ifundefined{actualchar}{\def\actualchar{@}}{}
179     \@ifundefined{quotechar}{\def\quotechar{"}}{}
180     \@ifundefined{levelchar}{\def\levelchar{!}}{}
181     \@ifundefined{encapchar}{\def\encapchar{|}}{}
182     \@ifundefined{verbatimchar}{\def\verbatimchar{+}}{}
183     \setlength\marginparpush{0pt} \setlength\marginparwidth{8pc}
184     \reversemarginpar
185     \DeclareRobustCommand\meta[1]{%
186         \ensuremath\langle
187         \ifmmode \expandafter \nfss@text \fi
188         {%
189             \meta@font@select
190             \edef\meta@hyphen@restore
191                 {\hyphenchar\the\font\the\hyphenchar\font}%
192             \hyphenchar\font\m@ne
193             \language\l@nohyphenation
194             #1\/%
195             \meta@hyphen@restore
196             }\ensuremath\rangle
197     }
198     \def\meta@font@select{\itshape}
199 }% doc loaded
200 }% AtBeginDocument

```

6.9 Support macros

`\PrintEnvName` $\langle name \rangle$ Prints an environment name.

```

201 \providecommand*\PrintEnvName{}
202 \renewcommand*\PrintEnvName}[1]
203 {\strut{\scriptsize}Env}\quad\MacroFont#1\ }

```

`\DTXD@printtype` $\langle text \rangle$

Used to print the object class in the margin:

```

204 \newcommand*\DTXD@printtype}[1]
205 {\raggedleft\strut{\scriptsize\sffamily#1}\quad\MacroFont}

```

`\usage` $\langle text \rangle$

Allow hyperlinks in the “usage” index entries:

```

206 \IfPackageLoadedTF{doc}{% not doc package
207

```

```

208 \providecommand{\usage}{}
209 \renewcommand{\usage}[1]{\textit{\hyperpage{#1}}}
210
211 }% not doc package

```

`\DTXD@origwindex` Used to bypass `hyperref` index modifications.

```
212 \let\DTXD@origwindex\@windex
```

`\DTXD@margin` `{\langle class \rangle}{\langle name \rangle}{\langle margin tag \rangle}`

Creates the margin tag for the object being described.

The `class` is used to sub-categorize keys into their key/value groups.

```

213 \newcommand*\DTXD@margin[3]{%
214   \ifundefined{@capttype}{% not float?
215     \leavevmode%
216     \marginpar{%
217       {%
218         \hbadness=10000%
219         \hfuzz=5em%
220         \DTXD@printtype{%
221           #3% margin tag
222           \ifblank{#1}{\{ [#1]\}% class
223         }%
224         \texttt{#2}% name
225       }%
226     }% marginpar
227   }\}% not float?
228 }

```

`\DTXD@index` `{\langle class \rangle}{\langle name \rangle}{\langle margin tag \rangle}{\langle index tag \rangle}{\langle main/usage \rangle}`

Creates the index entries for the object being described, where `name` has no backslash or underscore.

The `class` is used to sub-categories keys into their key/value groups. `main` prints code lines in the index, and `usage` prints page numbers.

```
229 \newcommand*\DTXD@index[5]{%
```

The `makeindex` program allows each index entry to call a macro by appending a vertical bar and a macro name to each entry. `hyperref` adds a call by `\hyperpage` to each index entry, by appending the phrase `|hyperpage` to the entry in the `.idx` file. The `doc` package uses the same mechanism to distinguish between code line entries (`|main`) and references to the use of a macro (`|usage`). The problem is that `makeindex` can only handle one macro call, but `hyperref` tries to append its `|hyperpage` to the already-existing `|usage` or `|main`.

The solution used for `dtxdescribe` is to allow `hyperref` to modify all regular index entries, but use the original definition of `\@windex` for the `\Describe_____` macros,

before `hyperref` modified it. Then, the `\usage` macro, defined above, manually adds the hyperlink.

Below, `\@bsphack` and `\@esphack` seem to be required for `\@wrindex` to work. `\ignorespaces` is used in addition because `\Declare` and `\index` entries often come in groups.

```
230 \@bsphack%
231 \beginingroup%
232 \DTXD@origwrindex{%
```

Index by name:

Write the name, the formatted name, the index tag, and the class:

```
233 #2\actualchar{\protect\ttfamily#2} % name
234 (#4)% index tag
235 \ifblank{#1}{}[#1]%
236 \encapchar #5}%
```

Index by tag and class:

Write the tag and class as a group, under which is the name and the formatted name.

```
237 \beginingroup%
238 \DTXD@origwrindex{%
239 #4:\levelchar% index tag
240 \ifblank{#1}{}[#1:\levelchar}%
241 #2\actualchar{\protect\ttfamily#2}% name
242 \encapchar #5}%
```

Possibly index by class and name:

```
243 \ifblank{#1}{}[% class given
244 \beginingroup%
245 \DTXD@origwrindex{%
246 #1\actualchar[#1]:\levelchar% class
247 #2\actualchar{\protect\ttfamily#2} % name
248 (#4)% index tag
249 \encapchar #5}%
250 }% class given
251 % \@esphack%
252 \@esphack%
253 \ignorespaces%
254 }
```

```
\DTXD@marginindex {<class>} {<name>} {<margin tag>} {<index tag>} {<main/usage>}
```

Creates the margin tag and the index entries. The `class` is used to sub-categories keys into their key/value groups.


```
255 \newcommand*{\DTXD@marginindex}[5]{%
256 % \@bsphack%
```

The margin tag and the name:

```
257 \DTXD@marginindex{#1}{#2}{#3}%
```

The index entries:

```
258 \DTXD@index{#1}{#2}{#3}{#4}{#5}%
259 }
```

```
\DTXD@macroname {<control sequence>}
```

Given a control sequence such as `\name`, prints its name without the backslash.

From: <http://tex.stackexchange.com/questions/42318/removing-a-backslash-from-a-character-sequence>

```
260 \begingroup\lccode'\|='\
261 \lowercase{\endgroup\def\removebs#1{\if#1|\else#1\fi}}
262 \newcommand*{\DTXD@macroname}[1]{\expandafter\removebs\string#1}
```

```
\DTXD@verbatimcmd {<\name>}
```

While printing to the index file, prints the `\name` verbatim. From `\SpecialIndex` in the `doc` package.

```
263 \newcommand*{\DTXD@verbatimcmd}[1]{%
264 \string\verb\quotechar*\verbatimchar\string#1\verbatimchar%
265 }
```

```
\DTXD@cmdmarginindex {<class>} {<name>} {<margin tag>} {<index tag>} {<main/usage>}
```

Creates the margin tag and index entries where `name` is a `\macro`.

```
266 \newcommand*{\DTXD@cmdmarginindex}[5]{%
267 \@bsphack%
```

Create a margin tag with the name of the macro:

```
268 \@ifundefined{@capttype}{% not float?
269 \leavevmode%
270 \marginpar{%
271     {%
272         \hbadness=10000%
273         \hfuzz=5em%
274         \DTXD@printtype{%
275             #3% margin tag
276             \ifblank{#1}{}{ [ #1]}% class
277         }%

```

```

278     \cmd{#2}% name
279   }%
280 }% marginpar
281 }{}% not float?

```

Create an index entry sorted by the name without its leading backslash, followed by the macro name with the backslash, and the tag. Prepend with the class if given.

Write (class):>name=csname (indextag)|usage

```

282 \begingroup%
283 \DTXD@origwindex{%
284 \ifblank{#1}{}{#1\actualchar[#1]:\levelchar}% class
285 \DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2} % name
286 (#4)% index tag
287 \encapchar #5}%

```

Create an index entry grouped by the tag, then printed and sorted by the macro name with the backslash, and the tag.

Write indextag:>(class):>csname|usage

```

288 \begingroup%
289 \DTXD@origwindex{%
290 #4:\levelchar% index tag
291 \ifblank{#1}{}{[#1]:\levelchar}% class
292 \DTXD@verbatimcmd{#2}% name
293 \encapchar #5}%
294 \@esphack%
295 \ignorespaces%
296 }

```

6.10 Key handling for object classes

If using `doc`, the optional key/value argument may also include an object class. This is supported by assigning any unknown key to be the class.

```

297 \ExplSyntaxOn
298 \IfPackageLoadedTF{doc}{
299
300 \newcommand*{\DTXD@category}{}
301
302 \keys_define:nn {doc}
303 {
304   c .cs_set:Np = \DTXD@category,
305   unknown .code:n = {%
306     \renewcommand*{\DTXD@category}{\l_keys_key_str}%
307   },
308 }
309
310 }{}% doc loaded
311 \ExplSyntaxOff

```

6.11 Handling `\marginpar` inside a float

To avoid a floats lost error, do not print margin tags if inside a float.

```

312 \def\@doc@describe#1#2{%                               dtxdescribe
313   \ifdoc@noprint\else
314     \ifundefined{@capttype}{% not float? dtxdescribe
315       \marginpar{\raggedleft
316                 \strut
317                 \doc@providetarget
318                 \@nameuse{PrintDescribe#1}{#2}%
319                 \ifdefvoid{\DTXD@category}{}% dtxdescribe
320                 \space{\footnotesize[\mbox{\DTXD@category}]]%
321                 }%
322       }
323     }{%
324   \fi
325   \ifdoc@noindex\else
326     \@nameuse{Special#1Index}{#2}%
327   \fi
328   \@esphack
329   \endgroup
330   \ignorespaces}

```

6.12 Handling glossary

Fix for `\StopEventually` in some conditions:

```

331 \AtBeginDocument{
332 \def\HD@guestoclevel#1{1}
333 }

```

6.13 Catcode handling to support underscores

File names and such may include underscores, so they must be neutralized before processing.

```

334 \IfPackageLoadedTF{doc}{%
335
336 \def\@NewDocElement#1#2#3{%
337   \doc@macrolikefalse
338   \doc@topleveltrue
339   \def\doc@idxtype{#3}%
340   \def\doc@idxgroup{#3s}%
341   \let\doc@printtype\@empty
342   \csname keys_set:nn\endcsname{doc}{#1}%
343   \ifx\doc@printtype\@empty
344     \@temptokena{}%
345   \else

```

```

346   \@temptokena\expandafter{\expandafter
347     \textnormal\expandafter{\expandafter
348       \space\expandafter
349         (\doc@printtype)}}%
350 \fi
351 \@nameedef{Print#2Name}##1{%
352   {\noexpand\MacroFont
353     \ifdoc@macrolike
354       \noexpand\string##1%           dtxdescribe
355     \else
356       \noexpand\detokenize\expandafter{##1}%       dtxdescribe
357     \fi
358     \the\@temptokena
359   }}%
360 \expandafter\let\csname PrintDescribe#2\expandafter\endcsname
361     \csname Print#2Name\endcsname
362 \edef\doc@expr{%
363   \ifdoc@macrolike
364     \noexpand\doc@createspecialmacrolikeindexes
365   \else
366     \noexpand\doc@createspecialindexes
367   \fi
368   {#2}%
369 }%
370 \expandafter\expandafter\expandafter
371 \doc@expr
372 \expandafter\expandafter\expandafter
373 {\expandafter\doc@idxtype\expandafter}\expandafter
374 {\doc@idxgroup}%
375 \doc@createdescribe{#2}%
376 \ifdoc@macrolike
377   \doc@createenv{TT}{#2}{#3}%
378 \else
379   \doc@createenv{TF}{#2}{#3}%
380 \fi
381 }
382
383 \newcommand*{\DTXD@maybecategory}{%
384   \ifdefvoid{\DTXD@category}%
385     {}%
386     {\space[%
387       \string\verb%
388       \verbatimchar\DTXD@category\verbatimchar%
389     ]}%
390 }
391
392 \newcommand*{\DTXD@macrotoctype}[1]{%
393   \ifcat\relax\noexpand#1%
394     \expandafter\expandafter\expandafter\@gobble\expandafter\string
395     \fi
396     #1%
397 }
398
399 \newcommand*{\DTXD@categorylevelname}

```

```

400
401 \newcommand*{\DTXD@maybelevel}[1]{%
402   \ifvoid{\DTXD@categorylevelname}%
403     {}
404     {%
405       \index{%
406         \DTXD@categorylevelname%
407         \noexpand\actualchar%
408         \string\verb*% %
409         \noexpand\verbatimchar
410         [\DTXD@category]:%
411         \verbatimchar%
412         \noexpand\levelchar%
413 \@gtempa\noexpand\actualchar%
414 \string\verb*% to fool emacs highlighting
415 \noexpand\quotechar%
416 %*
417 \noexpand\verbatimchar%
418 %\noexpand\bslash
419 \@gtempa\noexpand\verbatimchar%
420   \ifx\@nil#1\@nil\else \the\@temptokena \fi
421     \noexpand\doc@handleencap{usage}
422     }%
423   }%
424 }
425
426
427 \newcommand*{\DTXD@findcategorylevelname}{%
428   \edef\DTXD@categorylevelname{\DTXD@category}%
429   \edef\DTXD@categorylevelname{\detokenize\expandafter{\DTXD@categorylevelname}}%
430   \IfBeginWith{\DTXD@categorylevelname}{\DTXD@backslash}
431     {\StrGobbleLeft{\DTXD@categorylevelname}{1}[\DTXD@categorylevelname]}
432     {}
433 }
434
435 \def\doc@createspecialindexes#1#2#3{%
436   \@temptokena{\space (#2)}%
437   \@temptokenb{#3:}%
438   \@nameedef{SpecialMain#1Index}##1{%
439     \noexpand\@SpecialIndexHelper@##1\noexpand\@nil%           dtxdescribe
440     \noexpand\@bsphack
441     \noexpand\DTXD@findcategorylevelname%                     dtxdescribe
442     \ifdoc@toplevel
443 %     \noexpand\special@index{##1\noexpand\actualchar
444 %
445 %
446     \noexpand\special@index{\noexpand\@gtempa\noexpand\actualchar%
447 %     {\string\ttfamily\space##1}%
448 \string\verb*% to fool emacs highlighting%                   dtxdescribe
449 \noexpand\quotechar%
450 %*
451 \noexpand\verbatimchar%                                       dtxdescribe
452 %\noexpand\bslash
453 \noexpand\@gtempa\noexpand\verbatimchar%                       dtxdescribe

```

```

454 \ifx\@nil#2\@nil\else \the\@temptokena \fi
455 \noexpand\DTXD@maybecategory% dtxdescribe
456 \noexpand\encapchar main}%
457 \noexpand\DTXD@maybecategorylevel{#2}% dtxdescribe
458 \fi
459 %
460 %
461 \ifx\@nil#3\@nil\else
462 \noexpand\special@index{\the\@temptokenb\noexpand\levelchar%
463 % ##1\noexpand\actualchar{\string\ttfamily\space##1}%
464 \noexpand\@gtempa\noexpand\actualchar% dtxdescribe
465 \string\verb% % to fool emacs highlighting% dtxdescribe
466 \noexpand\quotechar%
467 %*
468 \noexpand\verbatimchar% dtxdescribe
469 %\noexpand\bslash
470 \noexpand\@gtempa\noexpand\verbatimchar% dtxdescribe
471 \noexpand\DTXD@maybecategory% dtxdescribe
472 \noexpand\encapchar main}%
473 \noexpand\DTXD@maybecategorylevel{#2}% dtxdescribe
474 \fi
475 \noexpand\@esphack}%
476 %
477 %
478 \@nameedef{Special#1Index}##1{%
479 \noexpand\@SpecialIndexHelper@##1\noexpand\@nil% dtxdescribe
480 \noexpand\@bsphack
481 \noexpand\DTXD@findcategorylevelname% dtxdescribe
482 %
483 %
484 \ifdoc@toplevel
485 \noexpand\doc@providetarget
486 % \noexpand\index{##1\noexpand\actualchar{\string\ttfamily\space##1}%
487 % \noexpand\index{\noexpand\@gtempa\noexpand\actualchar% dtxdescribe
488 \string\verb% % to fool emacs highlighting% dtxdescribe
489 \noexpand\quotechar%
490 %*
491 \noexpand\verbatimchar% dtxdescribe
492 %\noexpand\bslash
493 \noexpand\@gtempa\noexpand\verbatimchar% dtxdescribe
494 \ifx\@nil#2\@nil\else \the\@temptokena \fi
495 \noexpand\DTXD@maybecategory% dtxdescribe
496 \noexpand\doc@handleencap{usage}}%
497 \noexpand\DTXD@maybecategorylevel{#2}% dtxdescribe
498 \fi
499 %
500 %
501 \ifx\@nil#3\@nil\else
502 \noexpand\index{\the\@temptokenb\noexpand\levelchar%
503 % ##1\noexpand\actualchar{\string\ttfamily\space##1}%
504 \noexpand\@gtempa\noexpand\actualchar% dtxdescribe
505 \string\verb% % to fool emacs highlighting% dtxdescribe
506 \noexpand\quotechar
507 %*

```

```

508 \noexpand\verbatimchar%                               dtxdescribe
509 %\noexpand\bslash
510 \noexpand\@gtempa\noexpand\verbatimchar%             dtxdescribe
511   \noexpand\DTXD@maybecategory%                       dtxdescribe
512     \noexpand\doc@handleencap{usage}}%
513   \noexpand\DTXD@maybecategorylevel{#2}%              dtxdescribe
514   \fi
515 %
516 %
517   \noexpand\@esphack}}
518
519 }{}% doc loaded

```

6.14 \DescribeMacro and \DescribeEnvironment

`\DescribeMacro` [*class*] {`\name`}

Redefined to allow hyperlinked index entries and an optional class:

```

520 \IfPackageLoadedTF{doc}{
521 }{% not doc
522
523 \providecommand*\DescribeMacro{}
524 \renewcommand*\DescribeMacro}[2][{}]{%
525 \@bsphack%

```

Create the margin tag with the macro's name:

```

526 \@ifundefined{@capttype}{% not float?
527 \leavevmode%
528 \marginpar{%
529   {%
530     \hbadness=10000%
531     \hfuzz=5em%
532     \raggedleft%
533     \ifblank{#1}{}{\scriptsize\textsf{[#1]}} }% class
534     \cmd{#2}% name
535   }%
536 }% marginpar
537 }{}% not float?

```

Write the index sorted by the name without the backslash, followed by the actual name with the backslash. Append the class if given.

Write `name=csname>(class)|usage`

```

538 \begingroup%
539 \DTXD@origwindex{%
540   \DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2}% name
541   \ifblank{#1}{}{\levelchar[#1]}% class
542   \encapchar usage%
543 }%

```

Only if a class was given:

```

544 \ifblank{#1}%
545 {}% no class
546 {% class given
547 % Again, and prepend the class:
548 %
549 % Write class=(class):>name=csname\verb+|usage+
550 %   \begin{macrocode}
551   \begingroup%
552   \DTXD@origwindex{%
553   #1\actualchar[#1]:\levelchar%
554   \DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2}%
555   \encapchar usage}%
556 }% class given
557 \@esphack%
558 \ignorespaces%
559 }
560
561 }% not doc

```

`\DescribeEnv` [*class*] [*environment name*]

Redefined to allow hyperlinked index entries:

```

562 \IfPackageLoadedTF{doc}{}{% not doc
563
564 \providecommand*\DescribeEnv{}
565 \renewcommand*\DescribeEnv[2]{}
566 {\DTXD@marginindex{#1}{#2}{Env}{environment}{usage}}
567
568 }% not doc

```

6.15 New `\Describe. . . macros`

`\DTXD@filename` Stores the filename with a sanitized underscore.

```

569 \newcommand*\DTXD@filename{}

```

`\DTXD@filemarginparindex` [*class*] [*name*] [*margin tag*] [*index tag*] [*main/usage*]

The name may have underscores.

```

570 \newcommand*\DTXD@filemarginparindex[5]{%

```

Create a detokenized version of the filename...

```

571 \renewcommand*\DTXD@filename{\detokenize{#2}}%

```


... then replace any underscores with a detokenized `_`, which will print as an underscore when read back from the index file:

```
572 \StrSubstitute{\DTXD@filename}{\detokenize{\_}}{\detokenize{\_}}[\DTXD@filename]%
```

The original filename is printed in the margin. Any underscore characters have already been disabled by the `\catcode` change.

```
573 \DTXD@margin tag{#1}{#2}{#3}%
```

The detokenized and sanitized version is sent to the index file:

```
574 \DTXD@index{#1}{\DTXD@filename}{#3}{#4}{#5}%
```

End the group with the disabled underscore, and clean up the extra space from the `\catcode` command:

```
575 \endgroup%
576 \ignorespaces%
577 }
```

```
\DTXD@DescribeFile [class] {name}
```

The name may have underscores.

```
578 \newcommand*{\DTXD@DescribeFile}[2][]{%
579 \DTXD@filemarginparindex{#1}{#2}{File}{file}{usage}%
580 }
```

```
\DescribeFile {name}
```

The underscore character is temporarily disabled, then the name is passed directly to `\DTXD@DescribeFile`.

```
581 \IfPackageLoadedTF{doc}{% doc
582
583 \NewDocElement[
584   macrolike=false,
585   toplevel=false,
586   idxtype=file,
587   idxgroup=Files,
588   printtype=\textit{file}
589 ]{File}{file}
590
591 }{% not doc
592
593 \newcommand*{\DescribeFile}{%
594   \begingroup\catcode'\_ =12 \DTXD@DescribeFile%
595 }
596
597 }{% not doc
```

`\DTXD@DescribeProgram` [*⟨class⟩*] {*⟨name⟩*}

The name may have underscores.

```
598 \newcommand*{\DTXD@DescribeProgram}[2][]{%
599 \DTXD@filemarginparindex{#1}{#2}{Prog}{program}{usage}%
600 }
```

`\DescribeProgram` {*⟨name⟩*}

The underscore character is temporarily disabled, then the name is passed directly to `\DTXD@DescribeProgram`.

```
601 \IfPackageLoadedTF{doc}{% doc
602
603 \NewDocElement[
604   macrolike=false,
605   toplevel=false,
606   idxtype=program,
607   idxgroup=Programs,
608   printtype=\textit{Prog}
609 ]{Program}{program}
610
611 }{% not doc
612
613 \newcommand*{\DescribeProgram}{%
614 \begingroup\catcode'\_ =12 \DTXD@DescribeProgram%
615 }
616 }{% not doc
617 }
```

`\DTXD@DescribeCommand` [*⟨class⟩*] {*⟨name⟩*}

The name may have underscores.

```
618 \newcommand*{\DTXD@DescribeCommand}[2][]{%
619 \DTXD@filemarginparindex{#1}{#2}{Cmd}{command}{usage}%
620 }
```

`\DescribeCommand` {*⟨name⟩*}

The underscore character is temporarily disabled, then the name is passed directly to `\DTXD@DescribeCommand`.

```
621 \IfPackageLoadedTF{doc}{% doc
622
623 \NewDocElement[
624   macrolike=false,
625   toplevel=false,
626   idxtype=command,
627   idxgroup=Commands,
628   printtype=\textit{Cmd}

```

```

629 ]{Command}{command}
630
631 }{% not doc
632
633 \newcommand*\DescribeCommand}{%
634 \begingroup\catcode'\_ =12 \DTXD@DescribeCommand%
635 }
636
637 }{% not doc

```

`\DTXD@DescribePackage` [*⟨class⟩*] {*⟨name⟩*} The name may have underscores.

```

638 \newcommand*\DTXD@DescribePackage}[2][{%
639 \DTXD@filemarginparindex{#1}{#2}{Pkg}{package}{usage}%
640 }

```

`\DescribePackage` {*⟨name⟩*}

The underscore character is temporarily disabled, then the name is passed directly to `\DTXD@DescribePackage`.

```

641 \IfPackageLoadedTF{doc}{% doc
642
643 \NewDocElement[
644   macrolike=false,
645   toplevel=false,
646   idxtype=package,
647   idxgroup=Packages,
648   printtype=\textit{Pkg}
649 ]{Package}{package}
650
651 }{% not doc
652
653 \newcommand*\DescribePackage}{%
654 \begingroup\catcode'\_ =12 \DTXD@DescribePackage%
655 }
656
657 }{% not doc
658

```

`\DTXD@DescribeClass` [*⟨class⟩*] {*⟨name⟩*}

The name may have underscores.

```

659 \newcommand*\DTXD@DescribeClass}[2][{%
660 \DTXD@filemarginparindex{#1}{#2}{Cls}{class}{usage}%
661 }

```

`\DescribeClass` {*⟨name⟩*}

The underscore character is temporarily disabled, then the name is passed directly to `\DTXD@DescribeClass`.

```

662 \IfPackageLoadedTF{doc}{% doc
663
664 \NewDocElement[
665     macrolike=false,
666     toplevel=false,
667     idxtype=class,
668     idxgroup=Classes,
669     printtype=\textit{Cls}
670 ]{Class}{class}
671
672 }{% not doc
673
674 \newcommand*{\DescribeClass}{%
675 \begingroup\catcode'\_ =12 \DTXD@DescribeClass%
676 }
677
678 }{% not doc

```

`\DescribeOption` [*class*] {*name*}

```

679 \IfPackageLoadedTF{doc}{% doc
680
681 \NewDocElement[
682     macrolike=false,
683     toplevel=false,
684     idxtype=option,
685     idxgroup=Options,
686     printtype=\textit{Opt}
687 ]{Option}{option}
688
689 }{% not doc
690
691 \newcommand*{\DescribeOption}[2][
692 {\DTXD@margintagindex{#1}{#2}{Opt}{option}{usage}}
693
694 }{% not doc

```

`\DescribeArgument` [*class*] {*name*}

The class may be used to categorize arguments by their macro or environment name.

```

695 \IfPackageLoadedTF{doc}{% doc
696
697 \NewDocElement[
698     macrolike=false,
699     toplevel=false,
700     idxtype=argument,
701     idxgroup=Arguments,

```

```

702   printtype=\textit{Arg}
703 ]{Argument}{argument}
704
705 }{% not doc
706
707 \newcommand*{\DescribeArgument}[2][[]
708 {\DTXD@margintagindex{#1}{#2}{Arg}{argument}{usage}}
709
710 }{% not doc

```

`\DescribeBoolean` [*class*] {*name*}

```

711 \IfPackageLoadedTF{doc}{% doc
712
713 \NewDocElement[
714   macrolike=false,
715   toplevel=false,
716   idxtype=boolean,
717   idxgroup=Booleans,
718   printtype=\textit{bool}
719 ]{Boolean}{boolenv}
720
721 }{% not doc
722
723 \newcommand*{\DescribeBoolean}[2][[]
724 {\DTXD@margintagindex{#1}{#2}{Bool}{boolean}{usage}}
725
726 }{% not doc

```

`\DescribeLength` [*class*] {*name*}

```

727 \IfPackageLoadedTF{doc}{% doc
728
729 \NewDocElement[
730   macrolike=true,
731   toplevel=false,
732   idxtype=length,
733   idxgroup=Lengths,
734   printtype=\textit{Len}
735 ]{Length}{length}
736
737 }{% not doc
738
739 \newcommand*{\DescribeLength}[2][[]
740 {\DTXD@cmdmargintagindex{#1}{#2}{Len}{length}{usage}}
741
742 }{% not doc

```

`\DescribeCounter` [*class*] {*name*}

```

743 \IfPackageLoadedTF{doc}{% doc
744

```

```

745 \NewDocElement[
746   macrolike=false,
747   toplevel=false,
748   idxtype=counter,
749   idxgroup=Counters,
750   printtype=\textit{Ctr}
751 ]{Counter}{counter}
752
753 }{% not doc
754
755 \newcommand*\DescribeCounter[2][
756 {\DTXD@margintagindex{#1}{#2}{Ctr}{counter}{usage}}
757
758 }{% not doc

```

`\DescribeHook [class] {name}`

```

759 \IfPackageLoadedTF{doc}{% doc
760
761 \NewDocElement[
762   macrolike=false,
763   toplevel=false,
764   idxtype=hook,
765   idxgroup=Hooks,
766   printtype=\textit{Hook}
767 ]{Hook}{hook}
768
769 }{% not doc
770
771 \newcommand*\DescribeHook[2][
772 {\DTXD@margintagindex{#1}{#2}{Hook}{hook}{usage}}
773
774 }{% not doc

```

`\DescribeKey [class] {name}`

The class may be used to categorize keys by their kev/value group.

```

775 \IfPackageLoadedTF{doc}{% doc
776
777 \NewDocElement[
778   macrolike=false,
779   toplevel=false,
780   idxtype=key,
781   idxgroup=Keys,
782   printtype=\textit{Key}
783 ]{Key}{key}
784
785 }{% not doc
786
787 \newcommand*\DescribeKey[2][
788 {\DTXD@margintagindex{#1}{#2}{Key}{key}{usage}}
789

```

790 }% not doc

`\DescribeObject` [*<class>*] {*<name>*}

May be used to describe an arbitrary piece of code. Creates a margin tag and index entries with `\ttfamily`.

```

791 \IfPackageLoadedTF{doc}{% doc
792
793 \NewDocElement[
794     macrolike=false,
795     toplevel=false,
796     idxtype=object,
797     idxgroup=Objects,
798     printtype=,
799 ]{Object}{object}
800
801 }{% not doc
802
803 \newcommand*{\DescribeObject}[2][[%
804 \ifundefined{@capttype}{% not float?
805     \@bsphack%
806     \leavevmode%
807     \marginpar{%
808         \hbadness=10000%
809         \hfuzz=5em%
810         \raggedleft%
811         \ifblank{#1}{\raggedleft{\scriptsize[#1]} }
812         \texttt{#2}%
813     }%
814 }]{% not float?
815 \ifblank{#1}%
816 {%
817     \begingroup%
818     \DTXD@origwindex{%
819         #2\actualchar{\protect\ttfamily#2}%
820         \encapchar usage%
821     }%
822 }%
823 {%
824     \begingroup%
825     \DTXD@origwindex{%
826         #2\actualchar{\protect\ttfamily#2} [#1]%
827         \encapchar usage%
828     }%
829     \begingroup%
830     \DTXD@origwindex{%
831         #1\actualchar[#1]:\levelchar#2\actualchar{\protect\ttfamily#2}%
832         \encapchar usage%
833     }%
834 }%
835 \@esphack%
836 \ignorespaces%

```

```
837 }
838
839 }% not doc
```

`\DescribeOther` [*class*] {*name*}

May be used to describe an arbitrary non-programming object. Creates a margin tag and index entries with roman type.

```
840 \IfPackageLoadedTF{doc}{% doc
841
842 \NewDocElement[
843   macrolike=false,
844   toplevel=false,
845   idxtype=other,
846   idxgroup=Other,
847   printtype=,
848 ]{Other}{other}
849
850 }% not doc
851
852 \newcommand*{\DescribeOther}[2][[%
853 \ifundefined{@capttype}{% not float?
854   \@bspack%
855   \leavevmode%
856   \marginpar{%
857     \hbadness=10000%
858     \hfuzz=5em%
859     \raggedleft%
860     \ifblank{#1}{\raggedleft{\scriptsize[#1]} }%
861     #2%
862   }%
863 }]{% not float?
864 \ifblank{#1}%
865 {%
866   \begingroup%
867   \DTXD@origwindex{#2\encapchar usage}%
868 }%
869 {%
870   \begingroup%
871   \DTXD@origwindex{#2 [#1]\encapchar usage}%
872   \begingroup%
873   \DTXD@origwindex{#1\actualchar[#1]:\levelchar#2\encapchar usage}%
874 }%
875 \@espack%
876 \ignorespaces%
877 }
878
879 }% not doc
```


6.16 `\DescribeDefault`

`\DescribeDefaultcolor` The color of the margin tag used to show the default value.

```
880 \newcommand*{\DescribeDefaultcolor}{green!50!black}
```

`\DescribeDefault` $\langle value \rangle$

Creates a colored margin tag showing the booleandefault value.

```
881 \newcommand{\DescribeDefault}[1]{%
882   \margintag{%
883     \footnotesize%
884     \textcolor{\DescribeDefaultcolor}{%
885       Default: \texttt{#1}%
886     }%
887   }%
888 }
```

6.17 `\ItemDescribeMacro`, etc.

The following are for use inside a description.

`\ItemDescribeMacro` [$\langle class \rangle$] $\langle \backslash name \rangle$

```
889 \newcommand{\ItemDescribeMacro}[2][[]]{%
890   \item[\cmd{#2}:]%
891   \setlength{\parskip}{1.5ex}%
892   \DescribeMacro[#1]{#2}%
893 }
```

`\ItemDescribeEnv` [$\langle class \rangle$] $\langle name \rangle$

```
894 \newcommand{\ItemDescribeEnv}[2][[]]{%
895   \item[\env{#2}:]%
896   \setlength{\parskip}{1.5ex}%
897   \DescribeEnv[#1]{#2}%
898 }
```

`\ItemDescribeArgument` [$\langle class \rangle$] $\langle argument \rangle$

```
899 \newcommand{\ItemDescribeArgument}[2][[]]{%
900   \item[\texttt{#2}:]%
901   \setlength{\parskip}{1.5ex}%
902   \DescribeArgument[#1]{#2}%
903 }
```

`\ItemDescribeBoolean` [*class*] {*name*}

```
904 \newcommand{\ItemDescribeBoolean}[2][]{%
905   \item[\texttt{#2}:]%
906   \setlength{\parskip}{1.5ex}%
907   \DescribeBoolean[#1]{#2}%
908 }
```

`\ItemDescribeLength` [*class*] {*name*}

```
909 \newcommand{\ItemDescribeLength}[2][]{%
910   \item[\cmd{#2}:]%
911   \setlength{\parskip}{1.5ex}%
912   \DescribeLength[#1]{#2}%
913 }
```

`\ItemDescribeCounter` [*class*] {*name*}

```
914 \newcommand{\ItemDescribeCounter}[2][]{%
915   \item[\texttt{#2}:]%
916   \setlength{\parskip}{1.5ex}%
917   \DescribeCounter[#1]{#2}%
918 }
```

`\ItemDescribeHook` [*class*] {*name*}

```
919 \newcommand{\ItemDescribeHook}[2][]{%
920   \item[\texttt{#2}:]%
921   \setlength{\parskip}{1.5ex}%
922   \DescribeHook[#1]{#2}%
923 }
```

`\ItemDescribeKey` [*class*] {*name*}

```
924 \newcommand{\ItemDescribeKey}[2][]{%
925   \item[\texttt{#2}:]%
926   \setlength{\parskip}{1.5ex}%
927   \DescribeKey[#1]{#2}%
928 }
```

`\ItemDescribePackage` [*class*] {*name*}

```
929 \newcommand{\DTXD@ItemDescribePackage}[2][]{%
930   \item[\texttt{#2}:]%
931   \setlength{\parskip}{1.5ex}%
932   \DescribePackage[#1]{#2}%
933   \endgroup%
934 }
935
936 \newcommand{\ItemDescribePackage}{%
```

```

937 \begingroup\catcode'\_ =12 \DTXD@ItemDescribePackage%
938 }

```

`\ItemDescribeClass` [*class*] {*name*}

```

939 \newcommand{\DTXD@ItemDescribeClass}[2][]{%
940 \item[\texttt{#2}:]%
941 \setlength{\parskip}{1.5ex}%
942 \DescribeClass[#1]{#2}%
943 \endgroup%
944 }
945
946 \newcommand{\ItemDescribeClass}{%
947 \begingroup\catcode'\_ =12 \DTXD@ItemDescribeClass%
948 }

```

`\ItemDescribeOption` [*class*] {*name*}

```

949 \newcommand{\ItemDescribeOption}[2][]{%
950 \item[\texttt{#2}:]%
951 \setlength{\parskip}{1.5ex}%
952 \DescribeOption[#1]{#2}%
953 }

```

`\ItemDescribeFile` [*class*] {*name*}

```

954 \newcommand{\DTXD@ItemDescribeFile}[2][]{%
955 \item[\texttt{#2}:]%
956 \setlength{\parskip}{1.5ex}%
957 \DescribeFile[#1]{#2}%
958 \endgroup%
959 }
960
961 \newcommand{\ItemDescribeFile}{%
962 \begingroup\catcode'\_ =12 \DTXD@ItemDescribeFile%
963 }

```

`\ItemDescribeProgram` [*class*] {*name*}

```

964 \newcommand{\DTXD@ItemDescribeProgram}[2][]{%
965 \item[\texttt{#2}:]%
966 \setlength{\parskip}{1.5ex}%
967 \DescribeProgram[#1]{#2}%
968 \endgroup%
969 }
970
971 \newcommand{\ItemDescribeProgram}{%
972 \begingroup\catcode'\_ =12 \DTXD@ItemDescribeProgram%
973 }

```

`\ItemDescribeCommand` [*class*] {*name*}

```

974 \newcommand{\DTXD@ItemDescribeCommand}[2][]{%
975   \item[\texttt{#2}:]%
976   \setlength{\parskip}{1.5ex}%
977   \DescribeCommand[#1]{#2}%
978   \endgroup%
979 }
980
981 \newcommand{\ItemDescribeCommand}{%
982   \begingroup\catcode'\_ =12 \DTXD@ItemDescribeCommand%
983 }
```

`\ItemDescribeObject` [*class*] {*name*}

```

984 \newcommand{\ItemDescribeObject}[2][]{%
985   \item[\texttt{#2}:]%
986   \setlength{\parskip}{1.5ex}%
987   \DescribeObject[#1]{#2}%
988 }
```

`\ItemDescribeOther` [*class*] {*name*}

```

989 \newcommand{\ItemDescribeOther}[2][]{%
990   \item[\texttt{#2}:]%
991   \setlength{\parskip}{1.5ex}%
992   \DescribeOther[#1]{#2}%
993 }
```

6.18 `\margintag`, `\watchout`

`\margintagcolor` The color of the `\margintag`.

```
994 \newcommand*\margintagcolor{blue!70!black}
```

`\margintag` {*text*}

Prints a colored margin tag.

```

995 \newcommand{\margintag}[1]{%
996 \@ifundefined{@capttype}{% not float?
997 \marginpar{\raggedleft\textcolor{\margintagcolor}{#1}}%
998 \ignorespaces%
999 }{ }% not float?
1000 }
```

`\watchoutcolor` The color of the `\watchout`.

```
1001 \newcommand*\watchoutcolor{red!50!black}
```

`\watchout` [*text*]

Prints a warning sign and optional text.

```

1002 \newcommand{\watchout}[1][]{%
1003 \@ifundefined{@capttype}{% not float?
1004     \marginpar{%
1005         \raggedleft%
1006         \textcolor{\watchoutcolor}{\warningsign\normalsize\quad#1}%
1007     }%
1008     \ignorespaces%
1009 }{ }% not float?
1010 }
```

6.19 Nesting

Shows a box enclosing a label for the container, and the container's contents. May be nested.

`\shownesting` [*fraction of \linewidth*] {*container*} {*contents*}

```

1011 \NewDocumentCommand{\shownesting}{s O{1} m m}{
1012     \IfBooleanF{#1}{
1013         \par\smallskip
1014     }
1015     \fbox{
1016         \begin{minipage}{#2\linewidth-2em}
1017             \hbadness=10000\relax%
1018             #3\par\smallskip
1019             \hspace{1em}
1020             \begin{minipage}{\linewidth-1.5em}
1021                 #4
1022             \end{minipage}
1023         \end{minipage}
1024     }
1025 }
```

6.20 The dtxexample environment

Also see example 14 on page 28.

`dtxexample_cut.tex` (*file*) Used to store the `\input` example code.

`DTXD@examplerulecolor` [*color*] The color of the middle rule in the `dtxexample`.

```

1026 \definecolor{DTXD@examplerulecolor}{rgb}{.9,.9,.9}
```

`\dtxexamplecodename` The text name of the code section.

```

1027 \newcommand*{\dtxexamplecodename}{Code: }
```

`\dtxexampleresultname` The text name of the result section.

```
1028 \newcommand*{\dtxexampleresultname}{Result:}
```

```
dtxexample (env.) * [<notes/cross-references>] [<caption & label>]
```

Reads the code listing as a verbatim input using the `fancybox` package, then displays the code listing as a verbatim output, and also executes the code and displays the result. A title caption is specified, along with optional cross-referencing commands or notes to refer to the results. The unstarred version places the code inside a minipage, forbidding a page break in the middle of the code listing. The starred version does not use a minipage. This is required when the code is too large to fit on a single page.

```
1029 \NewDocumentEnvironment{dtxexample}{s +0{} m}
1030 {% start dtxexample
```

Copy the environment's contents to the file `dtxexample_cut.tex`:

```
1031 \VerbatimOut[gobble=\DTXD@gobble,tabsize=4]{dtxexample_cut.tex}%
1032 }% start dtxexample
```

When the environment closes:

```
1033 {% end dtxexample
```

Finish the verbatim output:

```
1034 \endVerbatimOut
1035 \par
1036 \addvspace{\bigskipamount}
```

If unstarred, typeset the example in a minipage, else use a float:

```
1037 \IfBooleanTF{#1}%
1038   {% minipage
1039     \minipage{\linewidth}%
1040     \captionsetup{type=dtxexamplefloat}%
1041   }%
1042   {% float
1043     \begin{dtxexamplefloat}%
1044   }%
```

```
1045 \hrule\medskip
1046 \caption{#3}
```

Typeset the contents as verbatim:

```
1047 \textcolor{DTXD@examplerulecolor}{\smallskip\hrule}
1048 \smallskip
1049 {\scriptsize\itshape\dtxexamplecodename}
1050 \VerbatimInput[tabsize=4]{dtxexample_cut.tex}
```

```

1051 \unskip
1052 \textcolor{DTXD@examplerulecolor}{\hrule}
1053 \smallskip
1054 {\scriptsize\itshape\dtxexampleresultname}
1055

```

Possible add the optional cross-references or notes:

```

1056 \ifstrempty{#2}
1057 {}
1058 {{\itshape\small #2}}

```

If unstarred, close the float or `\minipage`.

```

1059 \IfBooleanTF{#1}%
1060   {\endminipage}%
1061   {\end{dtxexamplefloat}}%
1062 } % end dtxexample

```

Outside of the environment's scope, input the example to generate its output and labels:

```

1063 \AfterEndEnvironment{dtxexample}
1064 {%

```

Execute the code:

```

1065 \par\unskip\input{dtxexample_cut.tex}%

```

Closing rule::

```

1066 \medskip\hrule%
1067 }

```

`\DeclareFloatingEnvironment [dtxexamplefloat]` A new float type for the examples.

```

1068 \DeclareFloatingEnvironment[
1069 fileext=lox,
1070 listname={List of Examples},
1071 name=Example,
1072 placement=hbp
1073 ]{dtxexamplefloat}

```

`\captionsetup [dtxexamplefloat]` Caption setup for the examples.

```

1074 \captionsetup*[dtxexamplefloat]{
1075 format=hang,
1076 font=bf,
1077 justification=raggedright,
1078 singlelinecheck=false,
1079 skip=0pt,
1080 position=top,

```

1081 }

`\crefname [dtxexamplefloat]` Name for `cleveref`. `\crefname` here is required for documents not using the doc class:

```
1082 \AddToHook{begindocument/before}{% Before .aux file is loaded.
1083     \ifdef{\crefname}{
1084         \crefname{dtxexamplefloat}{example}{examples}
1085     }{}
1086 }
```

6.21 noindmacro and noindenvironment

Similar to `macro` and `environment`, but not indexed.

`noindmacro (env.)` $\{\langle name \rangle\}$

```
1087 \newenvironment{noindmacro}[1]
1088 {
1089     \setlength{\parskip}{\marginparpush}
1090     \leavevmode\par\DTXD@margin tag{\cmd{#1}}{}
1091 }
1092 {\unskip}
```

`noindenvironment (env.)` $\{\langle name \rangle\}$

```
1093 \newenvironment{noindenvironment}[1]
1094 {
1095     \setlength{\parskip}{\marginparpush}
1096     \leavevmode\par\DTXD@margin tag{#1}{Env}
1097 }
1098 {\unskip}
```

6.22 sourcedisplay, UIdisplay, docsidebar

For use in a `sourcedisplay`:

`\fquad` Forces a quad indent.

```
1099 \newcommand*\fquad{\hspace*{1em}}
```

`\fqquad` Forces a double-quad indent.

```
1100 \newcommand*\fqquad{\hspace*{2em}}
```


`\fqquad` Forces a triple-quad indent.

```
1101 \newcommand*\fqquad{\hspace*{3em}}
```

`sourceverb` (*env.*) To typeset a block of source code, verbatim.

```
1102 \DefineVerbatimEnvironment{sourceverb}{Verbatim}
1103     {gobble=\DTXD@gobble,tabsize=4,xleftmargin=2em}
1104 \BeforeBeginEnvironment{sourceverb}{\vspace*{-.5\parskip}}
```

`fsourceverb` (*env.*) To typeset a framed block of source code, verbatim.

```
1105     \DefineVerbatimEnvironment{fsourceverb}{Verbatim}
1106         {gobble=\DTXD@gobble,tabsize=4,xleftmargin=2em,frame=lines}
1107 \BeforeBeginEnvironment{fsourceverb}{\vspace*{-.5\parskip}}
```

`sourcedisplay` (*env.*) To typeset a block of source code, allowing direct formatting.

```
1108 \newenvironment{sourcedisplay}
1109 {
1110     \leavevmode
1111     \par
1112     \fqquad\minipage{\linewidth-4em}
1113     \ttfamily
1114 }
1115 {%
1116     \endminipage
1117     \par
1118 }
```

`UIdisplay` (*env.*) To typeset a user interface display.

```
1119 \newenvironment{UIdisplay}
1120 {
1121     \leavevmode
1122     \par
1123     \fqquad\minipage{\linewidth-4em}
1124     \sffamily\bfseries
1125 }
1126 {
1127     \endminipage
1128     \par
1129 }
```

`\userentryname` Text to tell the user to enter the following item.

```
1130 \newcommand*\userentryname{Enter~$\Rightarrow$}
```

`\userentry` {*<text to enter>*}

Typesets text to be entered by the users.

```

1131 \newcommand{\userentry}[1]{%
1132 \par
1133 \fqquad%
1134 \begin{minipage}{\linewidth-2em}
1135   {\footnotesize \userentryname}\quad\cmds{#1}
1136 \end{minipage}
1137 \par
1138 }

```

`docsidebar` (*env.*) To typeset a sidebar in the documentation.

```

1139 \newenvironment{docsidebar}[1][]
1140 {%
1141   \quote\unskip\medskip
1142   \setlength{\parskip}{1.5ex}%
1143   \ifblank{#1}{}{\textit{#1}\newline}%
1144   \rule[.5\bigskipamount]{\linewidth}{.4pt}%
1145   \newline%
1146 }
1147 {%
1148   \leavevmode\par
1149   \rule[\bigskipamount]{\linewidth}{.4pt}
1150   \endquote\unskip
1151 }

```

6.23 Formatted objects

Macros to format references to various kinds of objects.

6.23.1 L^AT_EX objects

`\pkg` $\langle name \rangle$ Also useable for class names.

```
1152 \provideroobustcmd*{\pkg}[1]{\mbox{\textsf{#1}}}
```

`\cs` $\langle csname \rangle$ From `ltxdoc`.

```
1153 \provideroobustcmd*{\cs}[1]{\texttt{\char'\#1}}
```

`\env` $\langle name \rangle$

```
1154 \provideroobustcmd*{\env}[1]{\mbox{\texttt{#1}}}
```

`\marg` $\langle argument \rangle$ From `ltxdoc`.

```
1155 \providecommand\marg[1]{%
1156   {\ttfamily\char'\{\}\meta{#1}{\ttfamily\char'\}}}
```

`\oarg` [*⟨argument⟩*] From `ltxdoc`.

```
1157 \providecommand\oarg[1]{%
1158   {\ttfamily[]\meta{#1}{\ttfamily[]}}}
```

`\parg` (*⟨argument⟩*) From `ltxdoc`.

```
1159 \providecommand\parg[1]{%
1160   {\ttfamily()\meta{#1}{\ttfamily()}}}
```

`\ctr` {*⟨name⟩*}

```
1161 \providerobustcmd*\ctr}[1]{\mbox{\texttt{#1}}}
```

`\bool` {*⟨name⟩*}

```
1162 \providerobustcmd*\bool}[1]{\mbox{\texttt{#1}}}
```

`\optn` {*⟨name⟩*}

```
1163 \providerobustcmd*\optn}[1]{\mbox{\texttt{#1}}}
```

`\TOC`

```
1164 \providerobustcmd*\TOC}{\acro{TOC}}
```

`\LOF`

```
1165 \providerobustcmd*\LOF}{\acro{LOF}}
```

`\LOT`

```
1166 \providerobustcmd*\LOT}{\acro{LOT}}
```

6.23.2 Programs and commands

`\cmds` {*⟨commands to print⟩*} No processing is provided for special characters.

```
1167 \providerobustcmd*\cmds}[1]{\mbox{\textbf{\texttt{#1}}}}
```

`\progcode` {*⟨code to print⟩*} No processing is provided for special characters.

```
1168 \providerobustcmd*\progcode}[1]{\mbox{\texttt{#1}}}
```

`\prog` \langle *program name* \rangle Underscores are allowed.

```

1169 \newcommand*\DTXD@prog}[1]{%
1170   \mbox{\textsf{\textsl{\detokenize{#1}}}}%
1171   \endgroup%
1172 }
1173
1174 \providerobustcmd*\prog}{%
1175   \begingroup%
1176   \catcode'\_ =12%
1177   \DTXD@prog%
1178 }
```

`\filenm` \langle *file name* \rangle Underscores are allowed.

```

1179 \newcommand*\DTXD@filenm}[1]{%
1180   \mbox{\texttt{\detokenize{#1}}}%
1181   \endgroup%
1182 }
1183
1184 \providerobustcmd*\filenm}{%
1185   \begingroup%
1186   \catcode'\_ =12%
1187   \DTXD@filenm%
1188 }
```

`\UI` General user-interface text.

```

1189 \providerobustcmd*\UI}[1]{\textbf{\textsf{#1}}}
```

6.23.3 File types

`\ODT`

```

1190 \providerobustcmd*\ODT}{\acro{ODT}}
```

`\SVG`

```

1191 \providerobustcmd*\SVG}{\acro{SVG}}
```

`\PNG`

```

1192 \providerobustcmd*\PNG}{\acro{PNG}}
```

`\GIF`

```

1193 \providerobustcmd*\GIF}{\acro{GIF}}
```

\JPG

1194 \providerobustcmd*{\JPG}{\acro{JPG}}

\EPS

1195 \providerobustcmd*{\EPS}{\acro{EPS}}

\PDF

1196 \providerobustcmd*{\PDF}{\acro{PDF}}

\DVI

1197 \providerobustcmd*{\DVI}{\acro{DVI}}

6.23.4 Internet

\UTF

1198 \providerobustcmd*{\UTF}{\acro{UTF}}

\URL

1199 \providerobustcmd*{\URL}{\acro{URL}}

\element {<*name*>}

1200 \providerobustcmd*{\element}[1]{\texttt{<#1>}}

\attribute {<*name*>}

\attrib {<*name*>}

Each of these is “provided”, and any prior meaning will be unchanged. In particular, LuaTeX uses \attribute, so its meaning is unchanged if using LuaTeX.

1201 \providerobustcmd*{\attrib}[1]{\mbox{\texttt{#1}}}

1202

1203 \providerobustcmd*{\attribute}[1]{\mbox{\texttt{#1}}}

\HTML

1204 \providerobustcmd*{\HTML}{\acro{HTML}}

\HTMLfive

1205 \providerobustcmd*{\HTMLfive}{\HTML\textsc{5}}

\CSS

1206 \providerobustcmd*{\CSS}{\acro{CSS}}

\CSSthree

1207 \providerobustcmd*{\CSSthree}{\CSS\textsc{3}}

\EPUB

1208 \providerobustcmd*{\EPUB}{\acro{EPUB}}

6.23.5 Specific programs

\TikZ

1209 \providerobustcmd*{\TikZ}{Ti\emph{k}Z}

\CTAN

1210 \providerobustcmd*{\CTAN}{\acro{CTAN}}

\TDS

1211 \providerobustcmd*{\TDS}{\acro{TDS}}

\MathML

1212 \providerobustcmd*{\MathML}{Math\acro{ML}}

\MathJax

1213 \providerobustcmd*{\MathJax}{\brand{MathJax}}

6.23.6 Acronyms, brand names, trademarks

\brand {<name>}

1214 \providerobustcmd*{\brand}[1]{\textsc{#1}}

`\acro` $\langle \{acronym\} \rangle$

1215 `\providerobustcmd*{\acro}[1]{\textsc{\lowercase{#1}}}`

`\supregistered` Superscript trademark symbol.

1216 `\providerobustcmd*{\supregistered}{\textregistered}`

6.24 Logos

`\dviTeX` DVI T_EX

1217 `\providerobustcmd*{\dviTeX}{\mbox{\DVI\, \TeX}}`

`\dviLaTeX` DVI L^AT_EX

1218 `\providerobustcmd*{\dviLaTeX}{\mbox{\DVI\, \LaTeX}}`

`\pdfTeX` PDF T_EX

1219 `\providerobustcmd*{\pdfTeX}{\mbox{\PDF\, \TeX}}`

`\pdfLaTeX` PDF L^AT_EX

1220 `\providerobustcmd*{\pdfLaTeX}{\mbox{\PDF\, \LaTeX}}`

`\LuaTeX` LuaT_EX

1221 `\providerobustcmd*{\LuaTeX}{\mbox{\Lua\TeX}}`

`\LuaLaTeX` LuaL^AT_EX

1222 `\providerobustcmd*{\LuaLaTeX}{\mbox{\Lua\LaTeX}}`

`\XeTeX` X_ET_EX, X_EL^AT_EX

`\XeLaTeX`

1223 `\providerobustcmd*{\XeTeXrevE}`

1224 `\hspace{-.1667em}\raisebox{-.5ex}{E}\hspace{-.125em}`

1225

1226 `\AtBeginDocument{`

1227 `\ifpackageloaded{graphics}{`

1228 `\renewrobustcmd*{\XeTeXrevE}`

1229 `\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}`

1230 `}{}`

1231 `}`

1232

1233 `\providerobustcmd*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}`

1234 `\providerobustcmd*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}`

`\AmS` \mathcal{AMS}

```
1235 \providerobustcmd*{\AmS}{%
1236   \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
1237   \hbox{$\mathcal M$}\kern-.2em\mathcal S$}%
1238 }
```

`\LyX` LyX

```
1239 \providerobustcmd*{\LyX}{\textsf{LyX}}
```

`\BibTeX` BibTeX

```
1240 \providerobustcmd*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
```

`\MakeIndex` *MakeIndex*

```
1241 \providerobustcmd*{\MakeIndex}{\prog{MakeIndex}}
```

`\ConTeXt` ConTeXt

```
1242 \providerobustcmd*{\ConTeXt}{\mbox{Con\TeXt}}
```

`\MiKTeX` MiKTeX

```
1243 \providerobustcmd*{\MiKTeX}{\mbox{MiK\TeX}}
```

6.25 Dashes and slashes

`\thinspace` A breakable thin skip.

```
1244 \DeclareRobustCommand{\thinspace}{\hskip 0.16667em\relax}
```

`\endash` An endash: –

```
1245 \def\endash{-}
```

`\emdash` An emdash: —

```
1246 \def\emdash{-}
```

`\thinbrspace` A thin space which allows a line break.

```
1247 \newcommand{\thinbrspace}{%
1248   \hspace{.16667em}\penalty\exhyphenpenalty\hspace{0pt}%
1249 }
```


`\thinthinbrspace` A thin space which allows a line break.

```
1250 \newcommand{\thinthinbrspace}{%  
1251   \hspace{.08333em}\penalty\exhyphenpenalty\hspace{0pt}%  
1252 }
```

`\Dash` An unbreakable thin space, emdash, and breakable thin space.

```
1253 \newrobustcmd{\Dash}{\unskip\thinspace\emdash\thinbrspace}
```

`\dash` An unbreakable thin space, endash, and breakable thin space.

```
1254 \newrobustcmd{\dash}{\unskip\thinspace\endash\thinbrspace}
```

`\Slash` An unbreakable very thin space, a slash, and a breakable thin space.

```
1255 \newrobustcmd{\Slash}{\unskip\hspace{.08333em}/\thinthinbrspace}
```

7 Compiling dtxdescribe

To compile the dtxdescribe package:

```
Enter ⇒ pdflatex dtxdescribe.ins
```

To compile the dtxdescribe documentation

```
Enter ⇒ pdflatex dtxdescribe.dtx
```

(Several times)

```
Enter ⇒ makeindex -s gglo.ist -o dtxdescribe.gls dtxdescribe.glo
```

```
Enter ⇒ makeindex -s gind.ist dtxdescribe
```

```
Enter ⇒ pdflatex dtxdescribe.dtx
```

(Several times)

Change History and Index

Change History

| | | | |
|-------------------------------------|-------|-------------|----|
| v0.10 | | | |
| General: 2016/12/08 Initial ver | | 1 | |
| v0.11 | | | |
| General: 2018/03/30 | | 1 | |
| \DTXD@cmdmargintagindex: Index | | | |
| tag no longer plural. | | 42 | |
| \DTXD@index: Index tag no longer | | | |
| plural. | | 40 | |
| \watchout: Changed to \raggedleft. | | 61 | |
| v1.00 | | | |
| General: 2019/01/11 | | 1 | |
| Added formatted objects. | | 66 | |
| Added logos. | | 71 | |
| Cut file name changed to | | | |
| dtxexample_cut.tex | | 61 | |
| \DescribeClass: Fix: Allow | | | |
| underscore. | | 52 | |
| \DescribeDefault: Added. | | 57 | |
| \DescribeDefaultcolor: Added. | .. | 57 | |
| \DescribeMacro: Sans tag font. | ... | 47 | |
| \DescribePackage: Fix: Allow | | | |
| underscore. | | 51 | |
| docsidebar: Added. | | 66 | |
| \DTXD@cmdmargintagindex: Sans tag | | | |
| font. | | 41 | |
| \DTXD@filemarginparindex: Fix: File | | | |
| class. | | 49 | |
| \DTXD@printtype: Sans tag font. | .. | 38 | |
| \dtxexamplecodename: Added. | | 61 | |
| \dtxexampleresultname: Added. | .. | 62 | |
| \fqquad: Added. | | 65 | |
| \fquad: Added. | | 64 | |
| \quad: Added. | | 64 | |
| fsourceverb: Added. | | 65 | |
| \ItemDescribeArgument: Added. | .. | 57 | |
| \ItemDescribeBoolean: Added. | ... | 58 | |
| \ItemDescribeClass: Added. | | 59 | |
| \ItemDescribeCommand: Added. | ... | 60 | |
| \ItemDescribeCounter: Added. | ... | 58 | |
| \ItemDescribeEnv: Added. | | 57 | |
| \ItemDescribeFile: Added. | | 59 | |
| \ItemDescribeKey: Added. | | 58 | |
| \ItemDescribeLength: Added. | | 58 | |
| \ItemDescribeMacro: Added. | | 57 | |
| \ItemDescribeObject: Added. | | 60 | |
| \ItemDescribeOption: Added. | | 59 | |
| \ItemDescribeOther: Added. | | 60 | |
| \ItemDescribePackage: Added. | ... | 58 | |
| \ItemDescribeProgram: Added. | ... | 59 | |
| \margin tag: Uses \margin color. | | 60 | |
| \margin tag color: Added. | | 60 | |
| noindent environment: Added. | | 64 | |
| noindent macro: Added. | | 64 | |
| sourcedisplay: Added. | | 65 | |
| sourceverb: Added. | | 65 | |
| UIdisplay: Added. | | 65 | |
| \userentry: Added. | | 66 | |
| \userentryname: Added. | | 65 | |
| \watchout color: Added. | | 60 | |
| v1.01 | | | |
| General: 2019/03/22 | | 1 | |
| Sanitize PDF bookmarks. | | 33 | |
| \DescribeMacro: Put margin tag | | | |
| class in brackets. | | 47 | |
| \DescribeObject: Put margin tag | | | |
| class in brackets. | | 55 | |
| \DescribeOther: Put margin tag | | | |
| class in brackets. | | 56 | |
| \DTXD@cmdmargintagindex: Put | | | |
| margin tag class in brackets. | ... | 41 | |
| \DTXD@margintag: Put margin tag | | | |
| class in brackets. | | 39 | |
| sourcedisplay: Reduced width. | .. | 65 | |
| UIdisplay: Reduced width. | | 65 | |
| v1.02 | | | |
| General: 2019/07/16 | | 1 | |
| Fix if not doc package. | | 37 | |
| Fix if not hyperref package. | | 33 | |
| Fix if not ltxdoc class. | | 33, 37 | |
| \DescribeEnv: Fix if not ltxdoc | | class. | 48 |
| \DescribeMacro: Fix if not ltxdoc | | class. | 47 |
| \DescribeObject: \raggedleft | | margin par. | 55 |

| | | | |
|--|----|---|----------------|
| Added \ignorespaces. | 55 | \DTXD@margintag: Reduce hbox | |
| \DescribeOther: \raggedleft | | warnings. | 39 |
| margin par. | 56 | \dviLaTeX: Added. | 71 |
| Added \ignorespaces. | 56 | \dviTeX: Added. | 71 |
| dtxexample: Fix if not doc package. | 62 | \ItemDescribeHook: Added. | 58 |
| \PrintEnvName: Fix if not ltxdoc | | \marg: Provided. | 66 |
| class. | 38 | \MathJax: Added. | 70 |
| DTXD@gobble: Fix if not doc package. | 36 | \oarg: Provided. | 67 |
| \usage: Fix if not ltxdoc class. | 38 | \parg: Provided. | 67 |
| v1.03 | | \pdfLaTeX: Added. | 71 |
| General: 2022/02/01 | 1 | \pdfTeX: Added. | 71 |
| No longer requires xifthen. | 33 | \shownesting: Added. | 61 |
| \attrib: For LuaTeX. | 69 | v1.04 | |
| \cs: Provided. | 66 | General: 2022/02/01 | 1 |
| \DescribeHook: Added. | 54 | Corrected copyright date. | 1 |
| \DescribeMacro: Reduce hbox | | v1.05 | |
| warnings. | 47 | General: 2022/09/08 | 1 |
| \DescribeObject: Reduce hbox | | \TikZ: Renamed from \tikz, cap Z. | 70 |
| warnings. | 55 | v1.06 | |
| \DescribeOther: Reduce hbox | | General: 2022/12/07 | 1 |
| warnings. | 56 | Fixed for updated doc. | 35, 36, 42, 43 |
| \DTXD@cmdmargintagindex: Reduce | | | |
| hbox warnings. | 41 | | |

Index

Numbers written in *italics* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

| | | | |
|--------------------------------------|-----------------|---------------------------------------|-----------------|
| A | | C | |
| \acro | 17, <u>1215</u> | \captionsetup | 63 |
| \AmS | 17, <u>1235</u> | Classes: | |
| Arguments: | | class_name [descexamples] | 26 |
| [H] [figure] | 24 | sample_class [examples] | 22 |
| [M] [figure] | 24 | \cmds | 15, <u>1167</u> |
| bold [\mymacro] | 24 | [color]: | |
| \attrib | 16, <u>1201</u> | DTXD@examplerulecolor | 61 |
| \attribute | 16, <u>1201</u> | othercolor | 25 |
| B | | somecolor | 25 |
| \BibTeX | 17, <u>1240</u> | Commands: | |
| [bigfiles]: | | command_name [descexamples] | 26 |
| another_big_file.txt | 22 | OS_command | 22 |
| really_big_file.txt | 22 | \ConTeXt | 17, <u>1242</u> |
| \bool | 15, <u>1162</u> | Counters: | |
| Booleans: | | samplecounter [examples] | 21 |
| booleanname [descexamples] | 26 | \cphotonameref | 19 |
| sampleboolean [examples] | 21 | \crefname | 64 |
| \brand | 17, <u>1214</u> | \cs | 15, <u>1153</u> |
| | | \CSS | 16, <u>1206</u> |

| | | | |
|--|--------------------------|--|--------------------------|
| <code>\CSSthree</code> | 16, 1207 | <code>\dtxexampleresultname</code> | 13, 1028 |
| <code>\CTAN</code> | 16, 1210 | <code>\DVI</code> | 16, 1197 |
| <code>\ctr</code> | 15, 1161 | <code>\dviLaTeX</code> | 17, 1218 |
| | | <code>\dviTeX</code> | 17, 1217 |
| D | | | |
| <code>\Dash</code> | 18, 1253 | E | |
| <code>\dash</code> | 18, 1254 | <code>\element</code> | 16, 1200 |
| <code>\DeclareFloatingEnvironment</code> | 63 | <code>\emdash</code> | 18, 1246 |
| <code>\DeclareFloatingPhoto</code> | 19 | <code>\endash</code> | 17, 1245 |
| [descexamples]: | | <code>\env</code> | 15, 1154 |
| <code>booleanname</code> | 26 | environment (env.) | 9 |
| <code>class_name</code> | 26 | environments: | |
| <code>command_name</code> | 26 | <code>docsidebar</code> | 15, 1139 |
| <code>file_name</code> | 26 | <code>dtxexample</code> | 13, 1029 |
| <code>keyname</code> | 26 | <code>environment</code> | 9 |
| <code>package_name</code> | 26 | <code>fsourceverb</code> | 14, 1105 |
| <code>program_name</code> | 26 | <code>macro</code> | 9 |
| <code>\DescribeArgument</code> | 9, 695 | <code>myenvironment</code> | 20 |
| <code>\DescribeBoolean</code> | 10, 711 | <code>noindenvironment</code> | 14, 1093 |
| <code>\DescribeClass</code> | 10, 662 | <code>noindmacro</code> | 14, 1087 |
| <code>\DescribeCommand</code> | 11, 621 | <code>otherenvironment</code> | 20 |
| <code>\DescribeCounter</code> | 10, 743 | <code>sourcedisplay</code> | 14, 1108 |
| <code>\DescribeDefault</code> | 12, 881 | <code>sourceverb</code> | 14, 1102 |
| <code>\DescribeDefaultcolor</code> | 12, 880 | <code>UIDisplay</code> | 14, 1119 |
| <code>\DescribeEnv</code> | 9, 562 | <code>\EPS</code> | 16, 1195 |
| <code>\DescribeFile</code> | 11, 581 | <code>\EPUB</code> | 16, 1208 |
| <code>\DescribeHook</code> | 10, 759 | [examples]: | |
| <code>\DescribeKey</code> | 10, 775 | <code>sample_class</code> | 22 |
| <code>\DescribeLength</code> | 10, 727 | <code>sampleboolean</code> | 21 |
| <code>\DescribeMacro</code> | 9, 520 | <code>samplecounter</code> | 21 |
| <code>\DescribeObject</code> | 11, 791 | <code>samplekey</code> | 23 |
| <code>\DescribeOption</code> | 10, 679 | <code>sampleoption</code> | 22 |
| <code>\DescribeOther</code> | 11, 840 | <code>samplepackage</code> | 22 |
| <code>\DescribePackage</code> | 10, 641 | <code>sampletwokey</code> | 23 |
| <code>\DescribeProgram</code> | 11, 601 | | |
| <code>docsidebar (env.)</code> | 15, 1139 | F | |
| <code>\DTX@filename</code> | 569 | [figure]: | |
| <code>\DTXD@cmdmargintagindex</code> | 266 | [H] | 24 |
| <code>\DTXD@DescribeClass</code> | 659 | [M] | 24 |
| <code>\DTXD@DescribeCommand</code> | 618 | <code>\filenm</code> | 15, 1179 |
| <code>\DTXD@DescribeFile</code> | 578 | Files: | |
| <code>\DTXD@DescribePackage</code> | 638 | <code>another_big_file.txt [bigfiles]</code> | 22 |
| <code>\DTXD@DescribeProgram</code> | 598 | <code>dtxexample_cut.tex</code> | 61 |
| <code>\DTXD@filemarginparindex</code> | 570 | <code>file_name [descexamples]</code> | 26 |
| <code>\DTXD@gobble</code> | 132 | <code>lone_file.txt</code> | 22 |
| <code>\DTXD@index</code> | 229 | <code>really_big_file.txt [bigfiles]</code> | 22 |
| <code>\DTXD@macroname</code> | 260 | <code>\fqquad</code> | 14, 1101 |
| <code>\DTXD@margintag</code> | 213 | <code>\fquad</code> | 14, 1100 |
| <code>\DTXD@margintagindex</code> | 255 | <code>\quad</code> | 14, 1099 |
| <code>\DTXD@origwindex</code> | 212 | <code>fsourceverb (env.)</code> | 14, 1105 |
| <code>\DTXD@printtype</code> | 204 | | |
| <code>\DTXD@verbatimcmd</code> | 263 | G | |
| <code>dtxexample (env.)</code> | 13, 1029 | <code>\GIF</code> | 16, 1193 |
| <code>\dtxexamplecodename</code> | 13, 1027 | group of objects | 20 |

| | | | |
|--|--------------------------|-------------------------------------|--------------------------|
| <code>\PDF</code> | 16, 1196 | <code>\thinbrspace</code> | 18, 1247 |
| <code>\pdfLaTeX</code> | 17, 1220 | <code>\thinspace</code> | 17, 1244 |
| <code>\pdfTeX</code> | 17, 1219 | <code>\thinthinbrspace</code> | 18, 1250 |
| <code>\photocaptionsetup</code> | 19 | <code>\TikZ</code> | 16, 1209 |
| <code>\pkg</code> | 15, 1152 | <code>\TOC</code> | 15, 1164 |
| <code>\PNG</code> | 16, 1192 | | |
| <code>\PrintEnvName</code> | 201 | | |
| | | U | |
| <code>\prog</code> | 15, 1169 | <code>\UI</code> | 15, 1189 |
| <code>\progcode</code> | 15, 1168 | <code>UIDisplay (env.)</code> | 14, 1119 |
| Programs: | | <code>\URL</code> | 16, 1199 |
| <code>program_name</code> | 22 | <code>\usage</code> | 206 |
| <code>program_name [desceexamples]</code> | 26 | <code>\userentry</code> | 14, 1131 |
| | | <code>\userentryname</code> | 14, 1130 |
| | | <code>\UTF</code> | 16, 1198 |
| | | | |
| | | W | |
| S | | <code>\warningsign</code> | 76 |
| <code>\shownesting</code> | 12, 1011 | <code>\watchout</code> | 13, 1002 |
| <code>\Slash</code> | 18, 1255 | <code>\watchoutcolor</code> | 13, 1001 |
| <code>sourcedisplay (env.)</code> | 14, 1108 | | |
| <code>sourceverb (env.)</code> | 14, 1102 | | |
| <code>\supregistered</code> | 17, 1216 | | |
| <code>\SVG</code> | 16, 1191 | | |
| | | | |
| | | X | |
| | | <code>\XeLaTeX</code> | 17, 1223 |
| T | | <code>\XeTeX</code> | 17, 1223 |
| <code>\TDS</code> | 16, 1211 | | |