The LaTeX dtxdescribe Package

v1.00 — 2019/01/11

 Brian Dunn
bd@BDTechConcepts.com

Describe additional object types in `dtx` source files.

Abstract

The doc package includes tools for describing macros and environments in LaTeX source `dtx` format. The dtxdescribe package adds additional tools for describing booleans, lengths, counters, keys, packages, classes, options, files, commands, arguments, and other objects.

Each 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 LaTeX 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.

Contents

1 Introduction **5**

2 Using dtxdescribe **6**

3 The macros, and the dtxexample environment **7**

3.1 Macros and environments 7

3.2 Arguments 7

3.3 Booleans, lengths, counters, keys 8

3.4 Packages, classes, options 8

3.5 Files, programs, commands 8

3.6 Other source objects 9

3.7 In a description environment 9

3.8 Defaults 10

3.9 \margintag, \watchout 10

3.10 dtxexample environment 11

3.11 noindmacro and noindenvironment environments 11

3.12 sourceverb, sourcedisplay, UIdisplay, docsidebar 12

3.13 Formatted objects 12

3.13.1 LaTeX objects 12

3.13.2 Programs and commands 13

3.13.3 File types 13

3.13.4 Internet 14

3.13.5 Specific programs 14

3.13.6 Acronyms, brand names, trademarks 14

3.14 Logos 14

3.15 Dashes and slashes 15

4 Examples **16**

5 Usage notes **28**

6 Code **29**

6.1 Required packages 29

6.2 Vertical spacing 30

6.3 Support macros 30

6.4 `\DescribeMacro` and `\DescribeEnvironment` 34

6.5 New `\Describe...` macros 35

6.6 `\DescribeDefault` 40

6.7 `\ItemDescribeMacro`, etc. 40

6.8 `\margintag`, `\watchout` 44

6.9 The `dtxexample` environment 44

6.10 `noindmacro` and `noindenvironment` 47

6.11 `sourcedisplay`, `UIdisplay`, `docsidebar` 47

6.12 Formatted objects 49

6.12.1 LaTeX objects 50

6.12.2 Programs and commands 50

6.12.3 File types 51

6.12.4 Internet 52

6.12.5 Specific programs 53

6.12.6 Acronyms, brand names, trademarks 53

6.13 Logos 54

6.14 Dashes and slashes 55

Change History and Index **57**

List of Examples

1 Macros . . . 16
2 Environment . . . 17
3 Second Environment . . . 17
4 Booleans and Counters . . . 18
5 Lengths . . . 18
6 Packages, Classes, and Options . . . 19
7 Files, Commands, and Programs . . . 19
8 Keys . . . 20
9 Arguments . . . 21
10 Object . . . 22
11 Other . . . 22
12 Description environments . . . 23
13 `dtxexample` . . . 24
14 `fsourceverb` . . . 25
15 `sourcedisplay` . . . 25
16 `UIdisplay` . . . 26
17 `docsidebar` . . . 27

List of Figures

1 A Figure . . . 24

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

Place `\usepackage{dtxdescribe}` in the `.dtx` file's driver section:

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

Various objects inside the `dtx` file may be described with `\DescribeBoolean`, `\DescribeLength`, `\DescribeCounter`, and related macros, 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

`\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 16 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 17.

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 21.

3.3 Booleans, lengths, counters, keys

See example 4 on page 18.

`\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.

`\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 kev/value group. See example 8 on page 20.

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 LaTeX class. Given a 'Cls' tag in the margin and index.

`\DescribeOption` `[⟨class⟩]{⟨name⟩}`

Describes a LaTeX 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 22.

`\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 22.

3.7 In a `description` environment

To describe an object using a `description` environment, use the following. See example 12 on page 23.

`\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.

`\ItemDescribeCounter` `[`⟨*class*⟩`]{`⟨*name*⟩`}` A description.

`\ItemDescribeKey` `[`⟨*class*⟩`]{`⟨*name*⟩`}` A description.

`\ItemDescribePackage` `[`⟨*class*⟩`]{`⟨*package_name*⟩`}` With underscores.

`\ItemDescribeClass` `[`⟨*class*⟩`]{`⟨*class_name*⟩`}` With underscores.

`\ItemDescribeOption` `[`⟨*class*⟩`]{`⟨*name*⟩`}` A description.

`\ItemDescribeFile` `[`⟨*class*⟩`]{`⟨*file_name*⟩`}` With underscores.

`\ItemDescribeProgram` `[`⟨*class*⟩`]{`⟨*program_name*⟩`}` With underscores.

`\ItemDescribeCommand` `[`⟨*class*⟩`]{`⟨*command_name*⟩`}` With underscores.

`\ItemDescribeObject` `[`⟨*class*⟩`]{`⟨*name*⟩`}` A description.

`\ItemDescribeOther` `[`⟨*class*⟩`]{`⟨*name*⟩`}` A description.

3.8 Defaults

`\DescribeDefault` `{`⟨*value*⟩`}`
Default: `value`

Shows the default value of a `\Describe...` item, such as displayed here. Place this macro immediately after the `\Describe...` macro and any arguments, but before the text description.

`\DescribeDefaultcolor`
Default: `green!50!black`

The color of the margin tag used to show the default value. This is used by `\textcolor` to create the margin tag.

3.9 `\margintag`, `\watchout`

`\margintag` `{`⟨*text*⟩`}`

`\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`
Default: `blue!70!black`

The color of the `\margintag`.

`\watchout` `[`⟨*text*⟩`]`

⚠ `\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`
Default: `red!50!black`

The color of the `\watchout`.

3.10 `dtxexample` environment

Env `dtxexample` * [⟨*Notes/cross-references*⟩] {⟨*caption & label*⟩}

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 13 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:`

3.11 `noindmacro` and `noindenvironment` environments

Env `noindmacro` {⟨\name⟩} To document macros which should not be included in the index.

Env `noindenvironment` {⟨*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`.

3.12 sourceverb, sourcedisplay, UIdisplay, docsidebar

Env sourceverb
Default: gobble=2, tabsize=4, xleftmargin=2em

[⟨*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.

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

[⟨*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 14 on page page 25.

Env sourcedisplay

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 \fqqquad may be used to force indenting. Because this is *not* a verbatim environment, it *can* be used inside a macro. See example 15 on page 25.

\fquad Single-level indent inside a sourcedisplay.

\fqquad Double-level indent inside a sourcedisplay.

\fqqquad Triple-level indent inside a sourcedisplay.

Env UIdisplay Displays a user interface, such as a dialog box entry or a menu selection. See example 16 on page 26. Also see the \UI macro..

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

\userentryname
Default: Enter ⇒

Text to tell the user to enter the following item. Change with \renewcommand.

Env docsidebar [⟨*title*⟩] Creates a sidebar within the document. See example 17 on page 27.

3.13 Formatted objects

Macros to format references to various kinds of objects.

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

3.13.1 LaTeX objects

\pkg packagename, also for a classname

\env `environment`

\ctr `counter`

\bool `boolean`

\optn `option`: to a macro, package, class

\TOC TOC: Table of contents.

\LOF LOF: List of figures.

\LOT LOT: List of tables.

3.13.2 Programs and commands

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

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

\filenm `file_name`: Underscores allowed.

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

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

3.13.3 File types

\ODT ODT OpenDocument Format word processing document

\SVG SVG image format

\PNG PNG image format

\GIF GIF image format

\JPG JPG image format

\EPS EPS image format

\PDF PDF image format

\DVI DVI image format

3.13.4 Internet

`\UTF`	UTF: Unicode
`\URL`	URL: Uniform Resource Locator
`\element`	`<element>`: HTML / CSS element
`\attribute`	`attribute`: HTML / CSS attribute
`\HTML`	HTML: Hypertext Markup Language
`\HTMLfive`	HTML5: Old-style figure if font supports
`\CSS`	CSS: Cascading Style Sheet
`\CSSthree`	CSS3: Old-style figure if font supports
`\EPUB`	EPUB: E-book file format

3.13.5 Specific programs

`\tikz`	Ti*k*z: Package logo
`\MathML`	MathML: Mathematical Markup Language
`\CTAN`	CTAN: Comprehensive TEX Archive Network
`\TDS`	TDS: TEX Directory Structure

3.13.6 Acronyms, brand names, trademarks

`\brand`	Brandname, Company Name
`\acro`	ACRO: Acronym
`\supregistered`	Superscript trademark symbol®

3.14 Logos

Several additional logos are provided:

`\LuaTeX`	LuaTEX
`\LuaLaTeX`	LuaLATEX

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

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

`\AmS` $\mathcal{AMS}$

`\LyX` LyX

`\BibTeX` BibTeX

`\MakeIndex` *MakeIndex*

`\ConTeXt` ConTeXt

`\MiKTeX` MiKTeX

3.15 Dashes and slashes

`\thinskip` 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 unbreakeable thin space, emdash, and breakable thin space: A — B

`\dash` An unbreakeable 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 \dash B`	A – B	(only breakable before the B)
`A -- B`	A – B	(breakable before or after the dash)
`A---B`	A—B	(not breakable)
`A \Dash B`	A — B	(only breakable before the B)
`A --- B`	A — B	(breakable before or after the dash)
`A/B`	A/B	(not breakable)
`A \Slash B`	A/B	(only breakable before the B)
`A / B`	A / B	(breakable before or after the slash)
`A~/~B`	A / B	(not breakable)

4 Examples

Example 1: Macros

Code:

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

\DescribeMacro[photograph]{\DeclareFloatingEnvironment}
Create a photograph float. \bigskip

\DescribeMacro[photograph]{\captionsetup}
Caption settings for a photograph float.

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

Result:

`\mymacro` `[`⟨*optional*⟩`]` `{`⟨*mandatory*⟩`}` A typical macro definition.

photograph `\DeclareFloatingEnvironment` Create a photograph float.

photograph `\captionsetup` Caption settings for a photograph float.

photograph `\cnameref` cleveref 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.

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

Example 2: Environment

Code:

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

Result:

Env myenvironment {⟨*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

index groups

defined by this package each receives two index entries, one by name, and one grouped with others of its kind.

⚠ 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:

Env kindofenvironment otherenvironment [⟨*opt args*⟩] (⟨*coordinates*⟩) A description.

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:

Bool examples `sampleboolean` Some description.

Ctr examples `samplecounter` 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:

Len photograph `\photowidth` 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:

Pkg examples `samplepackage` About a LaTeX package.

Cls examples `sample_class` About a LaTeX class.

Opt examples `sampleoption` 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:

File bigfiles `really_big_file.txt` Some description.

File bigfiles `another_big_file.txt` Some description.

File `lone_file.txt` Some description.

Cmd `OS_command` An operating-system command.

Prog `program_name` 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:

Key groupofkeys `firstkey` About the first key of the `groupofkeys` set.

Key groupofkeys `secondkey` About the second key of `groupofkeys`.

Key examples `samplekey` About some key of `otherkeys`.

Key examples `sampletwokey` About another key of `otherkeys`.

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

Result:

Arg figure `[H]` What happens when a figure is [H]ere.

Arg figure `[M]` What happens when a figure is in the [M]argin.

Arg `\mymacro` `bold` 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 the 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:

color somecolor — The color of something.

color othercolor — 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 \env{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.

otherclass Other Item — About the other item.

otherclass Additional Item — 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:

descexamples `\macroname` **`\macroname:`** Describe the macro.

Bool descexamples `booleanname` **`booleanname:`** Describe the boolean.

Len descexamples `\lengthname` **`\lengthname:`** Describe the length.

Key descexamples `keyname` **`keyname:`** Describe the key.

Pkg descexamples `package_name` **`package_name:`** Describe the package.

Cls descexamples `class_name` **`class_name:`** Describe the class.

File descexamples `file_name` **`file_name:`** Describe the file.

Prog descexamples `program_name` **`program_name:`** Describe the program.

Cmd descexamples `command_name` **`command_name:`** Describe the class.

Uses a `description` environment to describe objects.

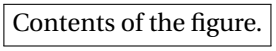

Figure 1: A Figure

Example 13: `dtxexample`

Code:

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

Result:

See fig. 1

Example 13, typeset above, was created with the following code:

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

When the example was created:

1. The "float" of type `example` was created, with the caption "`dtxexample`" and the label `ex:dtxexample`, which points to example 13.
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 14: `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 15: `sourcedisplay`

Code:

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

Result:

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

Example 16: `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 17: `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 LaTeX 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 `\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 supressed (there is no margin in a float), but the index entries are still created.

[1] Future 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

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

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

Pkg xparse Used for the examples.

```
\RequirePackage{xparse}
```

Pkg xifthen Used for the examples.

```
\RequirePackage{xifthen}
```

Pkg xcolor Used for the examples.

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

Pkg caption Used for the examples.

```
\RequirePackage{caption}
```

Pkg newfloat Used for the examples.

```
\RequirePackage{newfloat}
```

Pkg fancyvrb Used for the examples.

```
\RequirePackage{fancyvrb}
```

Pkg xstring Used for `\StrSubstitute` for `\DescribeFile`.

```
\RequirePackage{xstring}
```

Pkg pict2e

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

`\warningsign` Prints an exclamation point inside a triangle.

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

```
\newcommand*{\warningsign}{%
\begin{picture}(10,9)
\put(4,1){\scriptsize!}
\put(0,0){\line(500,866){5}}
\put(10,0){\line(-500,866){5}}
\put(0,0){\line(1,0){10}}
\end{picture}
}
```

6.2 Vertical spacing

```
\setlength{\marginparsep}{1em}
\setlength{\marginparpush}{.7ex}

\setlength{\parindent}{0em}
\setlength{\parskip}{2ex}

\setlength{\IndexMin}{40ex}
```

6.3 Support macros

```
\renewcommand*{\PrintEnvName}[1]
{\strut{\scriptsize{}Env}\quad\MacroFont#1\ }
```

`\DTXD@printtype` {⟨*text*⟩}

Used to print the object class in the margin:

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

`\usage` {⟨*text*⟩}

Allow hyperlinks in the “usage” index entries:

```
\renewcommand{\usage}[1]{\textit{\hyperpage{#1}}}
```

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

```
\let\DTXD@origwrindex\@wrindex
```

`\DTXD@margintag` {⟨*class*⟩} {⟨*name*⟩} {⟨*margin tag*⟩}

Creates the margin tag for the object being described.

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

```
\newcommand*{\DTXD@margintag}[3]{%
\@ifundefined{@captype}{% not float?
\leavevmode%
\marginpar{%
\DTXD@printtype{%
#3% margintag
\ifblank{#1}{}{ #1}% class
}% Desc@Type
\texttt{#2}% name
}% marginpar
}{}% not float?
}
```

`\DTXD@index` {⟨*class*⟩} {⟨*name*⟩} {⟨*margin tag*⟩} {⟨*index tag*⟩} {⟨*main/usage*⟩}

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.

```
\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 `\@wrindex` 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.

```
\@bsphack%
\begingroup%
\DTXD@origwrindex{%
```

Index by name:

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

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

Index by tag and class:

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

```
\begingroup%
\DTXD@origwrindex{%
#4:\levelchar% index tag
\ifblank{#1}{}{[#1]:\levelchar}% class
#2\actualchar{\protect\ttfamily#2}% name
\encapchar #5}%
```

Possibly index by class and name:

```
\ifblank{#1}{}{% class given
\begingroup%
\DTXD@origwrindex{%
#1\actualchar[#1]:\levelchar% class
#2\actualchar{\protect\ttfamily#2} % name
(#4)% index tag
\encapchar #5}%
}% class given
% \@esphack%
\@esphack%
\ignorespaces%
}
```

`\DTXD@margintagindex` {⟨*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.

```
\newcommand*{\DTXD@margintagindex}[5]{%
% \@bsphack%
```

The margin tag and the name:

```
\DTXD@margintag{#1}{#2}{#3}%
```

The index entries:

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

`\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

```
\begingroup\lccode`\|=`\\
\lowercase{\endgroup\def\removebs#1{\if#1|\else#1\fi}}
\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.

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

`\DTXD@cmdmargintagindex` {⟨*class*⟩} {⟨*name*⟩} {⟨*margin tag*⟩} {⟨*index tag*⟩} {⟨*main/usage*⟩}

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

```
\newcommand*{\DTXD@cmdmargintagindex}[5]{%
\@bsphack%
```

Create a margin tag with the name of the macro:

```
\@ifundefined{@captype}{% not float?
\leavevmode%
\marginpar{%
\DTXD@printtype{%
#3% margin tag
\ifblank{#1}{}{ #1}% class
}% Desc@Type
\cmd{#2}% name
}% marginpar
}{}% 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

```
\begingroup%
\DTXD@origwrindex{%
\ifblank{#1}{}{#1\actualchar[#1]:\levelchar}% class
```

```
\DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2} % name
(#4)% index tag
\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

```
\begingroup%
\DTXD@origwrindex{%
#4:\levelchar% index tag
\ifblank{#1}{}{[#1]:\levelchar}% class
\DTXD@verbatimcmd{#2}% name
\encapchar #5}%
\@esphack%
\ignorespaces%
}
```

6.4 `\DescribeMacro` and `\DescribeEnvironment`

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

Redefined to allow hyperlinked index entries and an optional class:

```
\renewcommand*{\DescribeMacro}[2][]{%
\@bsphack%
```

Create the margin tag with the macro's name:

```
\@ifundefined{@captype}{% not float?
\leavevmode%
\marginpar{%
\raggedleft%
\ifblank{#1}{}{{\scriptsize\textsf{#1}} }% class
\cmd{#2}% name
}% marginpar
}{}% 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

```
\begingroup%
\DTXD@origwrindex{%
```

```
\DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2}% name
\ifblank{#1}{}{\levelchar[#1]}% class
\encapchar usage}%
```

Only if a class was given:

```
\ifblank{#1}%
{}% no class
{% class given
% Again, and prepend the class:
%
% Write class=(class):>name=csname\verb+|usage+
%    \begin{macrocode}
\begingroup%
\DTXD@origwrindex{%
#1\actualchar[#1]:\levelchar%
\DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2}%
\encapchar usage}%
}% class given
\@esphack%
\ignorespaces%
}
```

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

Redefined to allow hyperlinked index entries:

```
\renewcommand*{\DescribeEnv}[2][]
{\DTXD@margintagindex{#1}{#2}{Env}{environment}{usage}}
```

6.5 New \Describe... macros

\DTX@filename Stores the filename with a sanitized underscore.

```
\newcommand*{\DTXD@filename}{}
```

\DTXD@filemarginparindex `{`⟨*class*⟩`}` `{`⟨*name*⟩`}` `{`⟨*margin tag*⟩`}` `{`⟨*index tag*⟩`}` `{`⟨*main/usage*⟩`}`

The name may have underscores.

```
\newcommand*{\DTXD@filemarginparindex}[5]{%
```

Create a detokenized version of the filename…

```
\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:

```
\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.

```
\DTXD@margintag{#1}{#2}{#3}%
```

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

```
\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:

```
\endgroup%
\ignorespaces%
}
```

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

The name may have underscores.

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

\DescribeFile `{`⟨*name*⟩`}`

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

```
\newcommand*{\DescribeFile}{%
\begingroup\catcode`\_=12 \DTXD@DescribeFile%
}
```

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

The name may have underscores.

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

\DescribeProgram {⟨*name*⟩}

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

```
\newcommand*{\DescribeProgram}{%
\begingroup\catcode`\_=12 \DTXD@DescribeProgram%
}
```

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

The name may have underscores.

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

\DescribeCommand {⟨*name*⟩}

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

```
\newcommand*{\DescribeCommand}{%
\begingroup\catcode`\_=12 \DTXD@DescribeCommand%
}
```

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

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

\DescribePackage {⟨*name*⟩}

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

```
\newcommand*{\DescribePackage}{%
\begingroup\catcode`\_=12 \DTXD@DescribePackage%
}
```

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

The name may have underscores.

```
\newcommand*{\DTXD@DescribeClass}[2][]{%
```

```
\DTXD@filemarginparindex{#1}{#2}{Cls}{class}{usage}%
}
```

`\DescribeClass` {⟨*name*⟩}

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

```
\newcommand*{\DescribeClass}{%
\begingroup\catcode`\_=12 \DTXD@DescribeClass%
}
```

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

```
\newcommand*{\DescribeOption}[2][]
{\DTXD@margintagindex{#1}{#2}{Opt}{option}{usage}}
```

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

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

```
\newcommand*{\DescribeArgument}[2][]
{\DTXD@margintagindex{#1}{#2}{Arg}{argument}{usage}}
```

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

```
\newcommand*{\DescribeBoolean}[2][]
{\DTXD@margintagindex{#1}{#2}{Bool}{boolean}{usage}}
```

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

```
\newcommand*{\DescribeLength}[2][]
{\DTXD@cmdmargintagindex{#1}{#2}{Len}{length}{usage}}
```

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

```
\newcommand*{\DescribeCounter}[2][]
{\DTXD@margintagindex{#1}{#2}{Ctr}{counter}{usage}}
```

`\DescribeKey` [⟨*class*⟩] {⟨*name*⟩}

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

```
\newcommand*{\DescribeKey}[2][]
{\DTXD@margintagindex{#1}{#2}{Key}{key}{usage}}
```

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

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

```
\newcommand*{\DescribeObject}[2][]{%
\@ifundefined{@captype}{% not float?
\@bsphack%
\leavevmode\marginpar{\raggedleft{\scriptsize#1} \texttt{#2}}%
}{}% not float?
\ifblank{#1}%
{\begingroup%
\DTXD@origwrindex{%
#2\actualchar{\protect\ttfamily#2}%
\encapchar usage%
}%
}%
{%
\begingroup%
\DTXD@origwrindex{%
#2\actualchar{\protect\ttfamily#2} [#1]%
\encapchar usage%
}%
\begingroup%
\DTXD@origwrindex{%
#1\actualchar[#1]:\levelchar#2\actualchar{\protect\ttfamily#2}%
\encapchar usage%
}%
}%
\@esphack%
% \ignorespaces%
}
```

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

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

```
\newcommand*{\DescribeOther}[2][]{%
\@ifundefined{@captype}{% not float?
\@bsphack%
\leavevmode\marginpar{\raggedleft{\scriptsize#1} #2}%
}{}% not float?
\ifblank{#1}%
{%
\begingroup%
\DTXD@origwrindex{#2\encapchar usage}%
}%
{%
```

```
\begingroup%
\DTXD@origwrindex{#2 [#1]\encapchar usage}%
\begingroup%
\DTXD@origwrindex{#1\actualchar[#1]:\levelchar#2\encapchar usage}%
}%
\@esphack%
% \ignorespaces%
}
```

6.6 \DescribeDefault

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

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

\DescribeDefault {⟨*value*⟩}

Creates a colored margin tag showing the booleandefault value.

```
\newcommand{\DescribeDefault}[1]{%
    \margintag{%
        \footnotesize%
        \textcolor{\DescribeDefaultcolor}{%
            Default: \texttt{#1}%
        }%
    }%
}
```

6.7 \ItemDescribeMacro, etc.

The following are for use inside a `description`.

\ItemDescribeMacro [⟨*class*⟩] {⟨\name⟩}

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

\ItemDescribeEnv [⟨*class*⟩] {⟨*name*⟩}

```
255 \newcommand{\ItemDescribeEnv}[2][]{%
256 \item[\env{#2}:]%
257 \setlength{\parskip}{1.5ex}%
258 \DescribeEnv[#1]{#2}%
259 }
```

`\ItemDescribeArgument` `[`⟨*class*⟩`]` `{`⟨*argument*⟩`}`

```
260 \newcommand{\ItemDescribeArgument}[2][]{%
261 \item[\texttt{#2}:]%
262 \setlength{\parskip}{1.5ex}%
263 \DescribeArgument[#1]{#2}%
264 }
```

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

```
265 \newcommand{\ItemDescribeBoolean}[2][]{%
266 \item[\texttt{#2}:]%
267 \setlength{\parskip}{1.5ex}%
268 \DescribeBoolean[#1]{#2}%
269 }
```

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

```
270 \newcommand{\ItemDescribeLength}[2][]{%
271 \item[\cmd{#2}:]%
272 \setlength{\parskip}{1.5ex}%
273 \DescribeLength[#1]{#2}%
274 }
```

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

```
275 \newcommand{\ItemDescribeCounter}[2][]{%
276 \item[\texttt{#2}:]%
277 \setlength{\parskip}{1.5ex}%
278 \DescribeCounter[#1]{#2}%
279 }
```

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

```
280 \newcommand{\ItemDescribeKey}[2][]{%
281 \item[\texttt{#2}:]%
282 \setlength{\parskip}{1.5ex}%
283 \DescribeKey[#1]{#2}%
284 }
```

\ItemDescribePackage [⟨*class*⟩] {⟨*name*⟩}

```
\newcommand{\DTXD@ItemDescribePackage}[2][]{%
\item[\texttt{#2}:]%
\setlength{\parskip}{1.5ex}%
\DescribePackage[#1]{#2}%
\endgroup
}

\newcommand{\ItemDescribePackage}{%
\begingroup\catcode`\_=12 \DTXD@ItemDescribePackage%
}
```

\ItemDescribeClass [⟨*class*⟩] {⟨*name*⟩}

```
\newcommand{\DTXD@ItemDescribeClass}[2][]{%
\item[\texttt{#2}:]%
\setlength{\parskip}{1.5ex}%
\DescribeClass[#1]{#2}%
\endgroup
}

\newcommand{\ItemDescribeClass}{%
\begingroup\catcode`\_=12 \DTXD@ItemDescribeClass%
}
```

\ItemDescribeOption [⟨*class*⟩] {⟨*name*⟩}

```
\newcommand{\ItemDescribeOption}[2][]{%
\item[\texttt{#2}:]%
\setlength{\parskip}{1.5ex}%
\DescribeOption[#1]{#2}%
}
```

\ItemDescribeFile [⟨*class*⟩] {⟨*name*⟩}

```
\newcommand{\DTXD@ItemDescribeFile}[2][]{%
\item[\texttt{#2}:]%
\setlength{\parskip}{1.5ex}%
\DescribeFile[#1]{#2}%
\endgroup
}

\newcommand{\ItemDescribeFile}{%
\begingroup\catcode`\_=12 \DTXD@ItemDescribeFile%
}
```

\ItemDescribeProgram [⟨*class*⟩] {⟨*name*⟩}

```
\newcommand{\DTXD@ItemDescribeProgram}[2][]{%
\item[\texttt{#2}:]%
\setlength{\parskip}{1.5ex}%
\DescribeProgram[#1]{#2}%
\endgroup
}

\newcommand{\ItemDescribeProgram}{%
\begingroup\catcode`\_=12 \DTXD@ItemDescribeProgram%
}
```

\ItemDescribeCommand [⟨*class*⟩] {⟨*name*⟩}

```
\newcommand{\DTXD@ItemDescribeCommand}[2][]{%
\item[\texttt{#2}:]%
\setlength{\parskip}{1.5ex}%
\DescribeCommand[#1]{#2}%
\endgroup
}

\newcommand{\ItemDescribeCommand}{%
\begingroup\catcode`\_=12 \DTXD@ItemDescribeCommand%
}
```

\ItemDescribeObject [⟨*class*⟩] {⟨*name*⟩}

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

\ItemDescribeOther [⟨*class*⟩] {⟨*name*⟩}

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

6.8 \margintag, \watchout

\margintagcolor The color of the \margintag.

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

\margintag {⟨*text*⟩}

Prints a colored margin tag.

```
\newcommand{\margintag}[1]{%
\@ifundefined{@captype}{% not float?
\marginpar{\raggedleft\textcolor{\margintagcolor}{#1}}%
\ignorespaces%
}{}% not float?
}
```

\watchoutcolor The color of the \watchout.

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

\watchout [⟨*text*⟩]

Prints a warning sign and optional text.

```
\newcommand{\watchout}[1][]{%
\@ifundefined{@captype}{% not float?
    \marginpar{%
        \raggedleft%
        \textcolor{\watchoutcolor}{\warningsign\normalsize\quad#1}%
    }%
    \ignorespaces%
}{}% not float?
}
```

6.9 The dtxexample environment

Also see example 13 on page page 24.

File dtxexample_cut.tex Used to store then \input example code.

color DTXD@examplerulecolor The color of the middle rule in the dtxexample.

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

\dtxexamplecodename The text name of the code section.

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

\dtxexampleresultname The text name of the result section.

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

Env dtxexample * [⟨*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.

```
\NewDocumentEnvironment{dtxexample}{s +O{} m}
{% start dtxexample
```

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

```
\VerbatimOut[gobble=2,tabsize=4]{dtxexample_cut.tex}%
}% start dtxexample
```

When the environment closes:

```
{% end dtxexample
```

Finish the verbatim output:

```
\endVerbatimOut
\par
\addvspace{\bigskipamount}
```

If unstarred, typeset the example in a minipage:

```
\IfBooleanTF{#1}{\vspace{\bigskipamount}}{\minipage{\linewidth}}%
```

Emulated a float of type "example":

```
\captionsetup{type=dtxdexample}%
\hrule\medskip
\caption{#3}
```

Typeset the contents as verbatim:

```
\textcolor{DTXD@examplerulecolor}{\smallskip\hrule}
\smallskip
{\scriptsize\itshape\dtxexamplecodename}
\VerbatimInput[tabsize=4]{dtxexample_cut.tex}
\unskip
\textcolor{DTXD@examplerulecolor}{\hrule}
\smallskip
{\scriptsize\itshape\dtxexampleresultname}

```

Possible add the optional cross-references or notes:

```
\ifstrempty{#2}
{}
{{\itshape\small #2}}
```

If unstarred, close the `\minipage`.

```
\IfBooleanTF{#1}{}{\endminipage}%
} % end dtxexample
```

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

```
\AfterEndEnvironment{dtxexample}
{%
```

Execute the code:

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

Closing rule::

```
\medskip\hrule%
}
```

dtxexample A new float type for the examples.

`\DeclareFloatingEnvironment`

```
\DeclareFloatingEnvironment[
fileext=lox,
listname={List of Examples},
name=Example,
placement=hbp
]{dtxdexample}
```

dtxexample `\captionsetup` Caption setup for the examples.

```
\captionsetup*[dtxdexample]{
format=hang,
font=bf,
justification=raggedright,
singlelinecheck=false,
skip=0pt,
position=top,
}
```

dtxexample `\crefname` Name for cleveref.

```
\AtBeginDocument{
\@ifpackageloaded{cleveref}{\crefname{dtxdexample}{example}{examples}}{}
}
```

6.10 noindmacro and noindenvironment

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

Env `noindmacro` {⟨*name*⟩}

```
\newenvironment{noindmacro}[1]
{
    \setlength{\parskip}{\marginparpush}
    \leavevmode\par\DTXD@margintag{}{\cmd{#1}}{}
}
{\unskip}
```

Env `noindenvironment` {⟨*name*⟩}

```
\newenvironment{noindenvironment}[1]
{
    \setlength{\parskip}{\marginparpush}
    \leavevmode\par\DTXD@margintag{}{#1}{Env}
}
{\unskip}
```

6.11 sourcedisplay, UIdisplay, docsidebar

For use in a `sourcedisplay`:

`\fquad` Forces a quad indent.

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

`\fqquad` Forces a double-quad indent.

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

`\fqqquad` Forces a triple-quad indent.

```
\newcommand*{\fqqquad}{\hspace*{3em}}
```

Env `sourceverb` To typeset a block of source code, verbatim.

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

Env `fsourceverb` To typeset a framed block of source code, verbatim.

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

Env `sourcedisplay` To typeset a block of source code, allowing direct formatting.

```
\newenvironment{sourcedisplay}
{
    \leavevmode
    \par
    \fqquad\minipage{\linewidth-2em}
    \ttfamily
}
{%
    \endminipage
    \par
}
```

Env `UIdisplay` To typeset a user interface display.

```
\newenvironment{UIdisplay}
{
    \leavevmode
    \par
    \fqquad\minipage{\linewidth-2em}
    \sffamily\bfseries
```

```
}
{
    \endminipage
    \par
}
```

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

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

\userentry {⟨*text to enter*⟩}

Typesets text to be entered by the users.

```
\newcommand{\userentry}[1]{%
\par
\fqquad%
\begin{minipage}{\linewidth-2em}
    {\footnotesize \userentryname}\quad\cmds{#1}
\end{minipage}
\par
}
```

Env docsidebar To typeset a sidebar in the documentation.

```
\newenvironment{docsidebar}[1][]
{%
    \quote\unskip\medskip
    \setlength{\parskip}{1.5ex}%
    \ifblank{#1}{}{\textit{#1}\newline}%
    \rule[.5\bigskipamount]{\linewidth}{.4pt}%
    \newline%
}
{%
    \leavevmode\par
    \rule[\bigskipamount]{\linewidth}{.4pt}
    \endquote\unskip
}
```

6.12 Formatted objects

Macros to format references to various kinds of objects.

6.12.1 LaTeX objects

\pkg or class

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

\env

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

\ctr

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

\bool

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

\optn

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

\TOC

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

\LOF

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

\LOT

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

6.12.2 Programs and commands

\cmds

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

\progcode

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

\prog

```
\newcommand*{\DTXD@prog}[1]{%
    \mbox{\textsf{\textsl{\detokenize{#1}}}}%
    \endgroup%
}

\providerobustcmd*{\prog}{%
    \begingroup%
    \catcode`\_=12%
    \DTXD@prog%
}
```

\filenm

```
\newcommand*{\DTXD@filenm}[1]{%
    \mbox{\texttt{\detokenize{#1}}}%
    \endgroup%
}

\providerobustcmd*{\filenm}{%
    \begingroup%
    \catcode`\_=12%
    \DTXD@filenm%
}
```

\UI General user-interface text.

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

6.12.3 File types

\ODT

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

\SVG

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

\PNG

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

```
      \GIF
517 \providerobustcmd*{\GIF}{\acro{GIF}}
      \JPG
518 \providerobustcmd*{\JPG}{\acro{JPG}}
      \EPS
519 \providerobustcmd*{\EPS}{\acro{EPS}}
      \PDF
520 \providerobustcmd*{\PDF}{\acro{PDF}}
      \DVI
521 \providerobustcmd*{\DVI}{\acro{DVI}}
```

6.12.4 Internet

```
      \UTF
522 \providerobustcmd*{\UTF}{\acro{UTF}}
      \URL
523 \providerobustcmd*{\URL}{\acro{URL}}
  \element
524 \providerobustcmd*{\element}[1]{\texttt{<#1>}}
\attribute
525 \providerobustcmd*{\attribute}[1]{\mbox{\texttt{#1}}}
     \HTML
526 \providerobustcmd*{\HTML}{\acro{HTML}}
```

\HTMLfive
```
527 \providerobustcmd*{\HTMLfive}{\HTML\textsc{5}}
```
\CSS
```
528 \providerobustcmd*{\CSS}{\acro{CSS}}
```
\CSSthree
```
529 \providerobustcmd*{\CSSthree}{\CSS\textsc{3}}
```
\EPUB
```
530 \providerobustcmd*{\EPUB}{\acro{EPUB}}
```

6.12.5 Specific programs

\tikz
```
531 \providerobustcmd*{\tikz}{Ti\textit{k}z}
```
\MathML
```
532 \providerobustcmd*{\MathML}{Math\acro{ML}}
```
\CTAN
```
533 \providerobustcmd*{\CTAN}{\acro{CTAN}}
```
\TDS
```
534 \providerobustcmd*{\TDS}{\acro{TDS}}
```

6.12.6 Acronyms, brand names, trademarks

\brand
```
535 \providerobustcmd*{\brand}[1]{\textsc{#1}}
```

\acro

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

\supregistered Superscript trademark symbol.

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

6.13 Logos

\LuaTeX LuaTEX

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

\LuaLaTeX LuaLATEX

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

\XeTeX XETEX, XELATEX
\XeLaTeX

```
\providerobustcmd*{\XeTeXrevE}
    {\hspace{-.1667em}\raisebox{-.5ex}{E}\hspace{-.125em}}

\AtBeginDocument{
\@ifpackageloaded{graphics}{
    \renewrobustcmd*{\XeTeXrevE}
        {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
}{}
}

\providerobustcmd*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
\providerobustcmd*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
```

\AmS $\mathcal{AMS}$

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

\LyX LyX

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

\BibTeX BibTeX

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

\MakeIndex *MakeIndex*

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

\ConTeXt ConTeXt

```
\providerobustcmd*{\ConTeXt}{\mbox{Con\TeX{}t}}
```

\MiKTeX MiKTeX

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

6.14 Dashes and slashes

\thinskip A breakable thin skip.

```
\DeclareRobustCommand{\thinskip}{\hskip 0.16667em\relax}
```

\endash An endash: –

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

\emdash An emdash: —

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

\thinbrspace A thin space which allows a line break.

```
\newcommand{\thinbrspace}{\hspace{.16667em}\penalty\exhyphenpenalty\hspace{0pt}}
```

\thinthinbrspace A thin space which allows a line break.

```
\newcommand{\thinthinbrspace}{\hspace{.08333em}\penalty\exhyphenpenalty\hspace{0pt}}
```

\Dash An unbreakeable thin space, emdash, and breakable thin space.

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

\dash An unbreakeable thin space, endash, and breakable thin space.

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

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

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

Change History and Index

Change History

v0.10
General: 2016/12/08 Initial ver 1
v0.11
\DTXD@cmdmargintagindex: Index tag no longer plural. 34
\DTXD@index: Index tag no longer plural. 32
\watchout: Changed to \raggedleft. 44
General: 2018/03/30 1
v1.00
\DTXD@cmdmargintagindex: Sans tag font. 33
\DTXD@filemarginparindex: Fix: File class. 36
\DTXD@printtype: Sans tag font. . . . 30
\DescribeClass: Fix: Allow underscore. 38
\DescribeDefault: Added. 40
\DescribeDefaultcolor: Added. . . 40
\DescribeMacro: Sans tag font. . . . 34
\DescribePackage: Fix: Allow underscore. 37
\ItemDescribeArgument: Added. . . 41
\ItemDescribeBoolean: Added. . . . 41
\ItemDescribeClass: Added. 42
\ItemDescribeCommand: Added. . . . 43
\ItemDescribeCounter: Added. . . . 41
\ItemDescribeEnv: Added. 40
\ItemDescribeFile: Added. 42
\ItemDescribeKey: Added. 41
\ItemDescribeLength: Added. 41
\ItemDescribeMacro: Added. 40
\ItemDescribeObject: Added. 43
\ItemDescribeOption: Added. 42
\ItemDescribeOther: Added. 43
\ItemDescribePackage: Added. . . . 42
\ItemDescribeProgram: Added. . . . 43
\dtxexamplecodename: Added. 45
\dtxexampleresultname: Added. . . 45
\fqqquad: Added. 48
\fqquad: Added. 48
\fquad: Added. 47
\margintag: Uses \margintacolor. 44
\margintagcolor: Added. 44
\userentry: Added. 49
\userentryname: Added. 49
\watchoutcolor: Added. 44
General: 2019/01/11 1
Added formatted objects. 49
Added logos. 54
Cut file name changed to dtxexample_cut.tex 44
UIdisplay: Added. 48
docsidebar: Added. 49
noindenvironment: Added. 47
noindmacro: Added. 47
sourcedisplay: Added. 48
sourceverb: Added. 48
fsourceverb: Added. 48

Index

Numbers written in italic 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.

Symbols

[H] (argument) [figure] *21*
[M] (argument) [figure] *21*
[\mymacro]:
  bold (argument) *21*

A

\acro *14*, 536
Additional Item [otherclass] *22*
\AmS *15*, 552
another_big_file.txt (file) [bigfiles] . *19*
argument:
  [\mymacro]:
    bold *21*
  [figure]:
    [H] *21*
    [M] *21*
\attribute *14*, 525

B

Before myenvironment *22*
\BibTeX *15*, 557
[bigfiles]:
  another_big_file.txt (file) *19*
  really_big_file.txt (file) *19*
bold (argument) [\mymacro] *21*
\bool *13*, 486
boolean:
  [descexamples]:
    booleanname *23*
  [examples]:
    sampleboolean *18*
booleanname (boolean) [descexamples] *23*
\brand *14*, 535

C

caption (package) *29*
\captionsetup
  [dtxexample] *46*
  [photograph] *16*
class:
  [descexamples]:
    class_name *23*
  [examples]:
    sample_class *19*
class_name (class) [descexamples] *23*
\cmds *13*, 491
\cnameref
  [photograph] *16*
[color]:
  DTXD@examplerulecolor *44*
  othercolor *22*
  somecolor *22*
command:
  [descexamples]:
    command_name *23*
  OS_command *19*
command_name (command) [descexamples] *23*
\ConTeXt *15*, 559
counter:
  [examples]:
    samplecounter *18*
\crefname
  [dtxexample] *47*
\CSS *14*, 528
\CSSthree *14*, 529
\CTAN *14*, 533
\ctr *13*, 485

D

\Dash *15*, 566
\dash *15*, 567
\DeclareFloatingEnvironment
  [dtxexample] *46*
  [photograph] *16*
[descexamples]:
  booleanname (boolean) *23*
  class_name (class) *23*
  command_name (command) *23*
  file_name (file) *23*
  keyname (key) *23*
  \lengthname (length) *23*
  \macroname *23*
  package_name (package) *23*

program_name (program) 23
\DescribeArgument 7, 185
\DescribeBoolean 8, 187
\DescribeClass 8, 180
\DescribeCommand 9, 168
\DescribeCounter 8, 191
\DescribeDefault 10, 242
\DescribeDefaultcolor 10, 241
\DescribeEnv 7, 141
\DescribeFile 8, 156
\DescribeKey 8, 193
\DescribeLength 8, 189
\DescribeMacro 7, 110
\DescribeObject 9, 195
\DescribeOption 8, 183
\DescribeOther 9, 222
\DescribePackage 8, 174
\DescribeProgram 9, 162
docsidebar (environment) 12, 470
\DTX@filename 143
\DTXD@cmdmarginindex 83
\DTXD@DescribeClass 177
\DTXD@DescribeCommand 165
\DTXD@DescribeFile 153
\DTXD@DescribePackage 171
\DTXD@DescribeProgram 159
DTXD@examplerulecolor [color] 44
\DTXD@filemarginparindex 144
\DTXD@index 46
\DTXD@macroname 77
\DTXD@margintag 34
\DTXD@margintagindex 72
\DTXD@origwrindex 33
\DTXD@printtype 30
\DTXD@verbatimcmd 80
[dtxexample]:
\captionsetup 46
\crefname 47
\DeclareFloatingEnvironment . . . 46
dtxexample (environment) 11, 370
dtxexample_cut.tex (file) 44
\dtxexamplecodename 11, 368
\dtxexampleresultname 11, 369
\DVI 13, 521

E

\element 14, 524
\emdash 15, 563
\endash 15, 562
\env 13, 484
environment:
[kindofenvironment]:
otherenvironment 17
docsidebar 12
dtxexample 11
fsourceverb 12
myenvironment 17
noindenvironment 11
noindmacro 11
sourcedisplay 12
sourceverb 12
UIdisplay 12
environments:
docsidebar 470
dtxexample 370
fsourceverb 436
noindenvironment 424
noindmacro 418
sourcedisplay 439
sourceverb 433
UIdisplay 450
\EPS 13, 519
\EPUB 14, 530
etoolbox (package) 29
[examples]:
sample_class (class) 19
sampleboolean (boolean) 18
samplecounter (counter) 18
samplekey (key) 20
sampleoption (option) 19
samplepackage (package) 19
sampletwokey (key) 20

F

fancyvrb (package) 29
[figure]:
[H] (argument) 21
[M] (argument) 21
file:
[bigfiles]:
another_big_file.txt 19
really_big_file.txt 19
[descexamples]:
file_name 23
dtxexample_cut.tex 44
lone_file.txt 19

file_name (file) [descexamples] 23
\filenm 13, 503
firstkey (key) [groupofkeys] 20
\fqqquad 12, 432
\fqquad 12, 431
\fquad 12, 430
fsourceverb (environment) 12, 436

G

\GIF 13, 517
group of objects 17
[groupofkeys]:
  firstkey (key) 20
  secondkey (key) 20

H

\HTML 14, 526
\HTMLfive 14, 527

I

index
  by group 17
\ItemDescribeArgument 9, 260
\ItemDescribeBoolean 9, 265
\ItemDescribeClass 10, 295
\ItemDescribeCommand 10, 330
\ItemDescribeCounter 9, 275
\ItemDescribeEnv 9, 255
\ItemDescribeFile 10, 310
\ItemDescribeKey 9, 280
\ItemDescribeLength 9, 270
\ItemDescribeMacro 9, 250
\ItemDescribeObject 10, 340
\ItemDescribeOption 10, 305
\ItemDescribeOther 10, 345
\ItemDescribePackage 10, 285
\ItemDescribeProgram 10, 320

J

\JPG 13, 518

K

key:
  [descexamples]:
    keyname 23
  [examples]:
    samplekey 20
    sampletwokey 20
  [groupofkeys]:
    firstkey 20
    secondkey 20
  lonekey 20
keyname (key) [descexamples] 23
[kindofenvironment]:
  otherenvironment (environment) . 17

L

length:
  [descexamples]:
    \lengthname 23
  [photograph]:
    \photowidth 18
license agreement 22
\LOF 13, 489
lone_file.txt (file) 19
lonekey (key) 20
\LOT 13, 490
\LuaLaTeX 14, 539
\LuaTeX 14, 538
\LyX 15, 556

M

\macroname
  [descexamples] 23
\MakeIndex 15, 558
\marg 7
margin tag missing 28
\margintag 10, 351
\margintagcolor 10, 350
\MathML 14, 532
\MiKTeX 15, 560
myenvironment (environment) 17
\mymacro 16

N

newfloat (package) 29
noindenvironment (environment) . 11, 424
noindmacro (environment) 11, 418

O

\oarg 7
\ODT 13, 514
option:
  [examples]:
    sampleoption 19
\optn 13, 487
OS_command (command) 19

Other Item [otherclass] 22
[otherclass]:
Additional Item 22
Other Item 22
othercolor [color] 22
otherenvironment (environment) [kindofenvironment] 17

P

package:
[descexamples]:
package_name 23
[examples]:
samplepackage 19
caption 29
etoolbox 29
fancyvrb 29
newfloat 29
pict2e 29
xcolor 29
xifthen 29
xparse 29
xstring 29
package_name (package) [descexamples] 23
\parg 7
\PDF 13, 520
[photograph]:
\captionsetup 16
\cnameref 16
\DeclareFloatingEnvironment . . . 16
\photowidth (length) 18
pict2e (package) 29
\pkg 12, 483
\PNG 13, 516
\prog 13, 493
\progcode 13, 492
program:
[descexamples]:
program_name 23
program_name 19
program_name (program) 19
program_name (program) [descexamples] 23

R

randomobject 22
really_big_file.txt (file) [bigfiles] . . 19

S

sample_class (class) [examples] 19
sampleboolean (boolean) [examples] . . 18
samplecounter (counter) [examples] . . 18
samplekey (key) [examples] 20
sampleoption (option) [examples] 19
samplepackage (package) [examples] . . 19
sampletwokey (key) [examples] 20
secondkey (key) [groupofkeys] 20
\Slash 15, 568
somecolor [color] 22
sourcedisplay (environment) 12, 439
sourceverb (environment) 12, 433
\supregistered 14, 537
\SVG 13, 515

T

\TDS 14, 534
\thinbrspace 15, 564
\thinskip 15, 561
\thinthinbrspace 15, 565
\tikz 14, 531
\TOC 13, 488

U

\UI 13, 513
UIdisplay (environment) 12, 450
\URL 14, 523
\usage 32
\userentry 12, 462
\userentryname 12, 461
\UTF 14, 522

W

\warningsign 13
\watchout 10, 358
\watchoutcolor 10, 357

X

xcolor (package) 29
\XeLaTeX 15, 540
\XeTeX 15, 540
xifthen (package) 29
xparse (package) 29
xstring (package) 29