

# The L<sup>A</sup>T<sub>E</sub>X `dtxdescribe` Package

v1.02 — 2019/07/16

© 2016–2018 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 L<sup>A</sup>T<sub>E</sub>X source .dtx format. The `dtxdescribe` package adds additional tools for describing booleans, lengths, counters, 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<sup>A</sup>T<sub>E</sub>X 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.

## Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Using dtxdescribe</b>	<b>7</b>
<b>3</b>	<b>The macros, and the dtxexample environment</b>	<b>8</b>
3.1	Macros and environments . . . . .	8
3.2	Arguments . . . . .	8
3.3	Booleans, lengths, counters, keys . . . . .	9
3.4	Packages, classes, options . . . . .	9
3.5	Files, programs, commands . . . . .	10
3.6	Other source objects . . . . .	10
3.7	In a description environment . . . . .	10
3.8	Defaults . . . . .	11
3.9	\marginpar, \watchout . . . . .	11
3.10	dtxexample environment . . . . .	12
3.11	noindmacro and noindenvironment environments . . . . .	12
3.12	sourceverb, sourcedisplay, UIdisplay, docsidebar . . . . .	13
3.13	Formatted objects . . . . .	14
3.13.1	L <sup>A</sup> T <sub>E</sub> X objects . . . . .	14
3.13.2	Programs and commands . . . . .	14
3.13.3	File types . . . . .	14
3.13.4	Internet . . . . .	15
3.13.5	Specific programs . . . . .	15
3.13.6	Acronyms, brand names, trademarks . . . . .	15
3.14	Logos . . . . .	16
3.15	Dashes and slashes . . . . .	16

<b>4 Examples</b>	<b>18</b>
<b>5 Usage notes</b>	<b>30</b>
<b>6 Code</b>	<b>31</b>
6.1 Required packages . . . . .	31
6.2 Gobbling comment characters . . . . .	33
6.3 Vertical spacing . . . . .	34
6.4 <code>ltxdoc</code> emulation . . . . .	34
6.5 <code>doc</code> emulation . . . . .	34
6.6 Support macros . . . . .	35
6.7 <code>\DescribeMacro</code> and <code>\DescribeEnvironment</code> . . . . .	39
6.8 New <code>\Describe...</code> macros . . . . .	41
6.9 <code>\DescribeDefault</code> . . . . .	45
6.10 <code>\ItemDescribeMacro</code> , etc. . . . .	46
6.11 <code>\margintag</code> , <code>\watchout</code> . . . . .	49
6.12 The <code>dtxexample</code> environment . . . . .	50
6.13 <code>noindmacro</code> and <code>noindenvironment</code> . . . . .	52
6.14 <code>sourcedisplay</code> , <code>UIdisplay</code> , <code>docsidebar</code> . . . . .	53
6.15 Formatted objects . . . . .	55
6.15.1 <code>LATEX</code> objects . . . . .	55
6.15.2 Programs and commands . . . . .	56
6.15.3 File types . . . . .	57
6.15.4 Internet . . . . .	58
6.15.5 Specific programs . . . . .	58
6.15.6 Acronyms, brand names, trademarks . . . . .	59
6.16 Logos . . . . .	59

6.17 Dashes and slashes . . . . .	60
<b>7 Compiling dtxdescribe</b>	<b>62</b>
<b>Change History and Index</b>	<b>63</b>

## List of Examples

1 Macros . . . . .	18
2 Environment . . . . .	19
3 Second Environment . . . . .	19
4 Booleans and Counters . . . . .	20
5 Lengths . . . . .	20
6 Packages, Classes, and Options . . . . .	21
7 Files, Commands, and Programs . . . . .	21
8 Keys . . . . .	22
9 Arguments . . . . .	23
10 Object . . . . .	24
11 Other . . . . .	24
12 Description environments . . . . .	25
13 dtxexample . . . . .	26
14 fsourceverb . . . . .	27
15 sourcedisplay . . . . .	27
16 UIDisplay . . . . .	28
17 docsidebar . . . . .	29

## List of Figures

1	A Figure	26
---	----------	----

## 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 L<sup>A</sup>T<sub>E</sub>X 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

`Env` macro 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.

`\DescribeMacro` `[\langle class \rangle] {\langle name \rangle}`

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 18 for examples.

`\DescribeEnv` `[\langle class \rangle] {\langle environment name \rangle}`

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 19.

#### 3.2 Arguments

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

`\marg` `{\langle text \rangle}`

Shows a mandatory argument for a macro or environment.

The results looks like `{\langle mandatory \rangle}`.

`\oarg` `{\langle text \rangle}`

Shows an optional argument for a macro or environment.

The results looks like `[\langle optional \rangle]`.

`\parg` `{\langle text \rangle}`

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 23.

### 3.3 Booleans, lengths, counters, keys

See example 4 on page 20.

\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 key/value group. See example 8 on page 22.

### 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<sup>A</sup>T<sub>E</sub>X class. Given a ‘Cls’ tag in the margin and index.

\DescribeOption [⟨class⟩] {⟨name⟩}

Describes a L<sup>A</sup>T<sub>E</sub>X 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 24.

\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 24.

### 3.7 In a description environment

To describe an object using a description environment, use the following. See example 12 on page 25.

\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>}
```

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** The color of the \watchout.  
Default: red!50!black

### 3.10 dtxexample environment

Env **dtxexample** \* [*<Notes/cross-references>*] {*<caption & label>*}

The dtxexample environment is useful for demonstrating a piece of L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 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

 **.dtx only** 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, the noindmacro and noindenvironment environments would localize any definitions, and \DescribeMacro and \DescribeEnv should be used instead.

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` `[(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` `[(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 27.

`xleftmargin=2em, frame=lines`

**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 27.

`\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 28. 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 ⇒

**Env** `docsidebar` `[(title)]` Creates a sidebar within the document. See example 17 on page 29.

### 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 L<sup>A</sup>T<sub>E</sub>X 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  Tikz: 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 X\TeX, with reversed E if *graphics* is loaded.  
\XeLaTeX X\LaTeX, with reversed E if *graphics* is loaded.  
\AmS \AmS  
\LyX \LyX  
\BibTeX \BibTeX  
\MakeIndex *MakelIndex*  
\ConTeXt \ConTeXt  
\MiKTeX \MiKTeX

### 3.15 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 \dash B	A-B (only breakable before the 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	(breakable before or after the dash)
A/B	A/B (not breakable)
A \Slash B	A/B (only breakable before the 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]` Create a photograph float.

`\DeclareFloatingEnvironment`

`[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 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:*

```
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]{sampletrokey} 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] sampletrokey 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	<b>\macroname:</b> Describe the macro.
Bool [descexamples] booleanname	<b>booleanname:</b> Describe the boolean.
Len [descexamples] \lengthname	<b>\lengthname:</b> Describe the length.
Key [descexamples] keyname	<b>keyname:</b> Describe the key.
Pkg [descexamples] package_name	<b>package_name:</b> Describe the package.
Cls [descexamples] class_name	<b>class_name:</b> Describe the class.
File [descexamples] file_name	<b>file_name:</b> Describe the file.
Prog [descexamples] program_name	<b>program_name:</b> Describe the program.
Cmd [descexamples] command_name	<b>command_name:</b> Describe the class.

---

Uses a description environment to describe objects.

Contents of the figure.

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{\dosomething}[1][whattodo]{
%   doing #1
% }
% \end{fsourceverb}
```

*Result:*

---

```
----- An fsourceverb example -----
\newcommand{\dosomething}[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 L<sup>A</sup>T<sub>E</sub>X 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.<sup>1</sup> 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 suppressed (there is no margin in a float), but the index entries are still created.

---

<sup>1</sup>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 `makeidx` One of several index programs must be provided. One of several index programs must be provided.

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

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

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

Pkg `xparse` Used for the examples.

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

Pkg `xifthen` Used for the examples.

```
10 \RequirePackage{xifthen}
```

Pkg `xcolor` Used for the examples.

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

Pkg `caption` Used for the examples.

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

Pkg `newfloat` Used for the examples.

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

Pkg `fancyvrb` Used for the examples.

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

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

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

Pkg **hyperref**  
**PDF bookmarks** If **hyperref** is loaded, disable some macros in 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\LuaTeX{\LuaTeX}%  
31             \def\XeTeX{\XeTeX}%  
32             \def\TeX{\TeX}%  
33             \def\LaTeX{\LaTeX}%  
34             \def\LaTeXe{\LaTeXe}%  
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\HTML{\HTML}%  
48             \def\ODT{\ODT}%  
49             \def\SVG{\SVG}%  
50             \def\PNG{\PNG}%  
51             \def\GIF{\GIF}%  
52             \def\JPG{\JPG}%  
53             \def\EPS{\EPS}%  
54             \def\PDF{\PDF}%  
55             \def\DVI{\DVI}%  
56             \def\UTF{\UTF}%  
57             \def\URL{\URL}%  
58             \def\element#1{\#1}%  
59             \def\attribute#1{\#1}%  
60             \def\HTML{\HTML}%  
61             \def\HTMLfive{\HTML5}%  
62             \def\CSS{\CSS}%  
63             \def\CSSthree{\CSS3}%
```

```

64      \def\EPUB{EPUB}%
65      \def\TOC{TOC}%
66      \def\LOF{LOF}%
67      \def\LOT{LOT}%
68    }
69  }

```

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

```

70  {
71    \newcommand*{\hyperpage}[1]{#1}
72  }
73 }

```

Pkg pict2e

```

74 \RequirePackage{pict2e}
75 \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.

```

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 }

```

## 6.2 Gobbling comment characters

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

```

84 @ifpackageloaded{doc}{
85   \newcommand*{\DTXD@gobble}{2}
86 }{
87   \newcommand*{\DTXD@gobble}{0}
88 }

```

### 6.3 Vertical spacing

```

89 \setlength{\marginparsep}{1em}
90 \setlength{\marginparpush}{.7ex}
91
92 \setlength{\parindent}{0em}
93 \setlength{\parskip}{2ex}
```

Len \IndexMin From **ltxdoc**.

```

94 \ifdef{\IndexMin}
95   {\setlength{\IndexMin}{40ex}}
96   {\newlength{\IndexMin}}
```

### 6.4 ltxdoc emulation

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

```

97 \@ifclassloaded{ltxdoc}{}{
98   \def\cmd#1{\cs{\expandafter\cmd@to@cs\string#1}}
99   \def\cmd@to@cs#1#2{\char\number‘#2\relax}
100  \DeclareRobustCommand\cs[1]{\texttt{\char‘\\#1}}
101  \providecommand\marg[1]{%
102    {\ttfamily\char‘\{}{\meta{#1}}{\ttfamily\char‘\}}}
103  \providecommand\oarg[1]{%
104    {\ttfamily[]}{\meta{#1}}{\ttfamily[]}}
105  \providecommand\parg[1]{%
106    {\ttfamily()}{\meta{#1}}{\ttfamily())}}
107  \providecommand\url{\texttt}
108 }
```

### 6.5 doc emulation

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

```

109 \AtBeginDocument{
110   \@ifpackageloaded{doc}{}{
111     \newenvironment*{macro}[1]{%
112       \PackageError{dtxdescribe}{%
113         {The ‘macro’ environment is only\MessageBreak
114          available when using the doc package\MessageBreak
115          with a .dtx source file}%
116         {This environment only makes sense for .dtx source.}%
117       }{}%
118     \newenvironment*{environment}[1]{%
119       \PackageError{dtxdescribe}{%
120         {The ‘environment’ environment is only\MessageBreak
```

```

121           available when using the doc package\MessageBreak
122           with a .dtx source file}
123           {This environment only makes sense for .dtx source.}
124       }{ }
125       \def\MacroFont{\fontencoding\encodingdefault
126           \fontfamily\ttdefault
127           \fontseries\mddefault
128           \fontshape\updefault
129           \small}%
130           \@ifundefined{actualchar}{\def\actualchar{@}}{}
131           \@ifundefined{quotechar}{\def\quotechar{"}}{}
132           \@ifundefined{levelchar}{\def\levelchar{!}}{}
133           \@ifundefined{encapchar}{\def\encapchar{|}}{}
134           \@ifundefined{verbatimchar}{\def\verbatimchar{+}}{}
135           \setlength\marginparpush{0pt} \setlength\marginparwidth{8pc}
136           \reversemarginpar
137           \DeclareRobustCommand\meta[1]{%
138               \ensuremath\langle
139               \ifmmode \expandafter \nfss@text \fi
140               \%
141               \meta@font@select
142               \edef\meta@hyphen@restore
143                   {\hyphenchar\the\font\the\hyphenchar\font}%
144                   \hyphenchar\font\m@ne
145                   \language\l@nohyphenation
146                   #1\%
147                   \meta@hyphen@restore
148                   }\ensuremath\rangle
149           }
150           \def\meta@font@select{\itshape}
151       }
152   }

```

## 6.6 Support macros

\PrintEnvName {*name*} Prints an environment name.

```

153 \providecommand*\PrintEnvName(){}
154 \renewcommand*\PrintEnvName[1]
155 {\strut\scriptsize\Env\quad\MacroFont#1\ }

```

\DTXD@printtype {*text*}

Used to print the object class in the margin:

```

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

```

---

\usage {⟨text⟩}

Allow hyperlinks in the “usage” index entries:

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

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

```
160 \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.

```
161 \newcommand*{\DTXD@margintag}[3]{%  
162 \@ifundefined{@capttype}{% not float?  
163 \leavevmode%  
164 \marginpar{  
165 \DTXD@printtype{  
166 #3% margintag  
167 \ifblank{#1}{}{ [#1]}% class  
168 }% Desc@Type  
169 \texttt{#2}% name  
170 }% marginpar  
171 }{}% not float?  
172 }
```

\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.

```
173 \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.

```
174 \@bsphack%
175 \begingroup%
176 \DTXD@origwrindex{%
```

Index by name:

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

```
177 #2\actualchar{\protect\ttfamily#2} % name
178 (#4)% index tag
179 \ifblank{#1}{}{ [#1]}% class
180 \encapchar #5}%
```

Index by tag and class:

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

```
181 \begingroup%
182 \DTXD@origwrindex{%
183 #4:\levelchar% index tag
184 \ifblank{#1}{}{ [#1]:\levelchar}% class
185 #2\actualchar{\protect\ttfamily#2}% name
186 \encapchar #5}%
```

Possibly index by class and name:

```
187 \ifblank{#1}{}% class given
188 \begingroup%
189 \DTXD@origwrindex{%
190 #1\actualchar[#1]:\levelchar% class
191 #2\actualchar{\protect\ttfamily#2} % name
192 (#4)% index tag
193 \encapchar #5}%
194 }% class given
195 % \@esphack%
196 \@esphack%
197 \ignorespaces%
198 }
```

---

\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.

```
199 \newcommand*{\DTXD@margintagindex}[5]{%
200 % \@bsphack%
```

The margin tag and the name:

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

The index entries:

```
202 \DTXD@index{#1}{#2}{#3}{#4}{#5}%
203 }
```

\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](http://tex.stackexchange.com/questions/42318/removing-a-backslash-from-a-character-sequence)

```
204 \begingroup\lccode`'|='\\
205 \lowercase{\endgroup\def\removebs#1{\if#1|\else#1\fi}}
206 \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.

```
207 \newcommand*{\DTXD@verbatimcmd}[1]{%
208 \string\verb\quotechar*\verbatimchar\string#\1\verbatimchar%
209 }
```

\DTXD@cmdmargintagindex {*<class>*} {*<name>*} {*<margin tag>*} {*<index tag>*} {*<main/usage>*}

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

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

Create a margin tag with the name of the macro:

---

```

212 \@ifundefined{@capttype}{% not float?
213 \leavevmode%
214 \marginpar{%
215 \DTXD@printtype{%
216 #3% margin tag
217 \ifblank{#1}{}{ [#1]}% class
218 }% Desc@Type
219 \cmd{#2}% name
220 }% marginpar
221 }{}% 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

```

222 \begingroup%
223 \DTXD@origwrindex{%
224 \ifblank{#1}{}{#1\actualchar[#1]:\levelchar}% class
225 \DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2} % name
226 (#4)% index tag
227 \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

```

228 \begingroup%
229 \DTXD@origwrindex{%
230 #4:\levelchar% index tag
231 \ifblank{#1}{}{[#1]:\levelchar}% class
232 \DTXD@verbatimcmd{#2}% name
233 \encapchar #5}%
234 \esphack%
235 \ignorespaces%
236 }

```

## 6.7 \DescribeMacro and \DescribeEnvironment

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

Redefined to allow hyperlinked index entries and an optional class:

```

237 \providecommand*\DescribeMacro{}
238 \renewcommand*\DescribeMacro[2][]{%
239 \@bsphack%

```

Create the margin tag with the macro's name:

```

240 \@ifundefined{@capttype}{% not float?
241 \leavevmode%
242 \marginpar{%
243 \raggedleft%
244 \ifblank{\#1}{{\scriptsize\textrm{\#1}}}% class
245 \cmd{\#2}% name
246 }% marginpar
247 }{}% 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`

```

248 \begingroup%
249 \DTXD@origrindex{%
250 \DTXD@macroname{\#2}\actualchar\DTXD@verbatim\cmd{\#2}% name
251 \ifblank{\#1}{{\levelchar{\#1}}}% class
252 \encapchar usage}%

```

Only if a class was given:

```

253 \ifblank{\#1}%
254 {}% no class
255 {}% class given
256 % Again, and prepend the class:
257 %
258 % Write class=(class):>name=csname\verb+|usage+
259 %     \begin{macrocode}
260 \begingroup%
261 \DTXD@origrindex{%
262 \#1\actualchar{\#1}:\levelchar{%
263 \DTXD@macroname{\#2}\actualchar\DTXD@verbatim\cmd{\#2}%
264 \encapchar usage}%
265 }% class given
266 \esphack%
267 \ignorespaces%
268 }

```

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

Redefined to allow hyperlinked index entries:

```

269 \providecommand*\DescribeEnv{}
270 \renewcommand*\DescribeEnv[2][]
271 {\DTXD@margintagindex{\#1}{\#2}{\Env}{\environment}{\usage}}

```

## 6.8 New \Describe... macros

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

272 \newcommand\*{\DTXD@filename}{}%

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

The name may have underscores.

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

Create a detokenized version of the filename...

274 \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:

275 \StrSubstitute{\DTXD@filename}%  
276 {\detokenize{\\_}}{\detokenize{\\_}}[\DTXD@filename]%

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

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

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

278 \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:

279 \endgroup%  
280 \ignorespaces%  
281 }

\DTXD@DescribeFile [\langle class\rangle] {\langle name\rangle}

The name may have underscores.

282 \newcommand\*{\DTXD@DescribeFile}[2][]{}%  
283 \DTXD@filemarginparindex{\#1}{\#2}{File}{file}{usage}-%  
284 }

```
\DescribeFile {\langle name\rangle}
```

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

```
285 \newcommand*{\DescribeFile}{%
286 \begingroup\catcode`\_=12 \DTXD@DescribeFile%
287 }
```

```
\DTXD@DescribeProgram [\langle class\rangle] {\langle name\rangle}
```

The name may have underscores.

```
288 \newcommand*{\DTXD@DescribeProgram}[2][]{%
289 \DTXD@filemarginparindex{\#1}{\#2}{Prog}{program}{usage}%
290 }
```

```
\DescribeProgram {\langle name\rangle}
```

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

```
291 \newcommand*{\DescribeProgram}{%
292 \begingroup\catcode`\_=12 \DTXD@DescribeProgram%
293 }
```

```
\DTXD@DescribeCommand [\langle class\rangle] {\langle name\rangle}
```

The name may have underscores.

```
294 \newcommand*{\DTXD@DescribeCommand}[2][]{%
295 \DTXD@filemarginparindex{\#1}{\#2}{Cmd}{command}{usage}%
296 }
```

```
\DescribeCommand {\langle name\rangle}
```

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

```
297 \newcommand*{\DescribeCommand}{%
298 \begingroup\catcode`\_=12 \DTXD@DescribeCommand%
299 }
```

```
\DTXD@DescribePackage [\langle class\rangle] {\langle name\rangle} The name may have underscores.
```

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

```
301 \DTXD@filemarginparindex{#1}{#2}{Pkg}{package}{usage}%
302 }
```

```
\DescribePackage {⟨name⟩}
```

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

```
303 \newcommand*{\DescribePackage}{%
304 \begingroup\catcode`\_=12 \DTXD@DescribePackage%
305 }
```

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

The name may have underscores.

```
306 \newcommand*{\DTXD@DescribeClass}[2][]{%
307 \DTXD@filemarginparindex{#1}{#2}{Cls}{class}{usage}%
308 }
```

```
\DescribeClass {⟨name⟩}
```

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

```
309 \newcommand*{\DescribeClass}{%
310 \begingroup\catcode`\_=12 \DTXD@DescribeClass%
311 }
```

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

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

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

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

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

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

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

```
\DescribeLength [⟨class⟩] {⟨name⟩}  
318 \newcommand*{\DescribeLength}[2][]{  
319 {\DTXD@cmdmarginindex{#1}{#2}{Len}{length}{usage}}}
```

```
\DescribeCounter [⟨class⟩] {⟨name⟩}  
320 \newcommand*{\DescribeCounter}[2][]{  
321 {\DTXD@margintagindex{#1}{#2}{Ctr}{counter}{usage}}}
```

```
\DescribeKey [⟨class⟩] {⟨name⟩}
```

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

```
322 \newcommand*{\DescribeKey}[2][]{  
323 {\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`.

```
324 \newcommand*{\DescribeObject}[2][]{%  
325 \@ifundefined{@capttype}{% not float?  
326 @bsphack%  
327 \leavevmode%  
328 \marginpar{  
329 \raggedleft%  
330 \ifblank{#1}{}{\raggedleft\scriptsize{#1}} }  
331 \texttt{#2}%  
332 }%  
333 }{}% not float?  
334 \ifblank{#1}{  
335 {\begingroup%  
336 \DTXD@origwrindex{  
337 #2\actualchar{\protect\ttfamily#2}}%  
338 \encapchar usage%  
339 }%  
340 }%  
341 {  
342 \begingroup%  
343 \DTXD@origwrindex{  
344 #2\actualchar{\protect\ttfamily#2} [#1]}%  
345 \encapchar usage%  
346 }%  
347 \begingroup%  
348 \DTXD@origwrindex{%
```

---

```

349 #1\actualchar[#1]:\levelchar#2\actualchar{\protect\ttfamily#2}%
350 \encapchar usage%
351 }%
352 }%
353 \@esphack%
354 \ignorespaces%
355 }

```

\DescribeOther [⟨class⟩] {⟨name⟩}

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

```

356 \newcommand*\DescribeOther[2][]{%
357 \@ifundefined{@capttype}{% not float?
358 \@bsphack%
359 \leavevmode%
360 \marginpar{%
361     \raggedleft%
362     \ifblank{#1}{}{\raggedleft\scriptsize{#1}} }%
363     #2%
364 }%
365 }% not float?
366 \ifblank{#1}%
367 {%
368 \begingroup%
369 \DTXD@origwrindex{#2\encapchar usage}%
370 }%
371 {%
372 \begingroup%
373 \DTXD@origwrindex{#2 [#1]\encapchar usage}%
374 \begingroup%
375 \DTXD@origwrindex{#1\actualchar[#1]:\levelchar#2\encapchar usage}%
376 }%
377 \@esphack%
378 \ignorespaces%
379 }

```

## 6.9 \DescribeDefault

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

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

\DescribeDefault {⟨value⟩}

---

Creates a colored margin tag showing the booleandefault value.

```

381 \newcommand{\DescribeDefault}[1]{%
382     \marginpar{%
383         \footnotesize%
384         \textcolor{\DescribeDefaultcolor}{%
385             Default: \texttt{\#1}%
386         }%
387     }%
388 }
```

## 6.10 \ItemDescribeMacro, etc.

The following are for use inside a description.

```
\ItemDescribeMacro  [<class>] {<name>}

389 \newcommand{\ItemDescribeMacro}[2][]{%
390 \item[\cmd{\#2}: ]%
391 \setlength{\parskip}{1.5ex}%
392 \DescribeMacro[\#1]{\#2}%
393 }

\ItemDescribeEnv  [<class>] {<name>}

394 \newcommand{\ItemDescribeEnv}[2][]{%
395 \item[\env{\#2}: ]%
396 \setlength{\parskip}{1.5ex}%
397 \DescribeEnv[\#1]{\#2}%
398 }

\ItemDescribeArgument  [<class>] {<argument>}

399 \newcommand{\ItemDescribeArgument}[2][]{%
400 \item[\texttt{\#2}: ]%
401 \setlength{\parskip}{1.5ex}%
402 \DescribeArgument[\#1]{\#2}%
403 }

\ItemDescribeBoolean  [<class>] {<name>}

404 \newcommand{\ItemDescribeBoolean}[2][]{%
405 \item[\texttt{\#2}: ]%
406 \setlength{\parskip}{1.5ex}%

```

```
407 \DescribeBoolean[#1]{#2}%
408 }

\ItemDescribeLength  [<class>] {<name>}
409 \newcommand{\ItemDescribeLength}[2][]{%
410 \item[\cmd{#2}:]%
411 \setlength{\parskip}{1.5ex}%
412 \DescribeLength[#1]{#2}%
413 }

\ItemDescribeCounter [<class>] {<name>}
414 \newcommand{\ItemDescribeCounter}[2][]{%
415 \item[\texttt{#2}:]%
416 \setlength{\parskip}{1.5ex}%
417 \DescribeCounter[#1]{#2}%
418 }

\ItemDescribeKey   [<class>] {<name>}
419 \newcommand{\ItemDescribeKey}[2][]{%
420 \item[\texttt{#2}:]%
421 \setlength{\parskip}{1.5ex}%
422 \DescribeKey[#1]{#2}%
423 }

\ItemDescribePackage [<class>] {<name>}
424 \newcommand{\DTXD@ItemDescribePackage}[2][]{%
425 \item[\texttt{#2}:]%
426 \setlength{\parskip}{1.5ex}%
427 \DescribePackage[#1]{#2}%
428 \endgroup
429 }
430
431 \newcommand{\ItemDescribePackage}{%
432 \begingroup\catcode`\_=12 \DTXD@ItemDescribePackage%
433 }

\ItemDescribeClass  [<class>] {<name>}
434 \newcommand{\DTXD@ItemDescribeClass}[2][]{%
435 \item[\texttt{#2}:]%
436 \setlength{\parskip}{1.5ex}%
437 \DescribeClass[#1]{#2}%
438 }
```

```
438 \endgroup
439 }
440
441 \newcommand{\ItemDescribeClass}{%
442 \begingroup\catcode`\_=12 \DTXD@ItemDescribeClass%
443 }

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

444 \newcommand{\ItemDescribeOption}[2][]{%
445 \item[\texttt{#2}:]%
446 \setlength{\parskip}{1.5ex}%
447 \DescribeOption[#1]{#2}%
448 }

\ItemDescribeFile [⟨class⟩] {⟨name⟩}

449 \newcommand{\DTXD@ItemDescribeFile}[2][]{%
450 \item[\texttt{#2}:]%
451 \setlength{\parskip}{1.5ex}%
452 \DescribeFile[#1]{#2}%
453 \endgroup
454 }
455
456 \newcommand{\ItemDescribeFile}{%
457 \begingroup\catcode`\_=12 \DTXD@ItemDescribeFile%
458 }

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

459 \newcommand{\DTXD@ItemDescribeProgram}[2][]{%
460 \item[\texttt{#2}:]%
461 \setlength{\parskip}{1.5ex}%
462 \DescribeProgram[#1]{#2}%
463 \endgroup
464 }
465
466 \newcommand{\ItemDescribeProgram}{%
467 \begingroup\catcode`\_=12 \DTXD@ItemDescribeProgram%
468 }

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

469 \newcommand{\DTXD@ItemDescribeCommand}[2][]{%
470 \item[\texttt{#2}:]%
471 \setlength{\parskip}{1.5ex}%
472 \DescribeCommand[#1]{#2}%
473 }
```

```

473 \endgroup
474 }
475
476 \newcommand{\ItemDescribeCommand}{%
477 \begingroup\catcode`\_=12 \DTXD@ItemDescribeCommand%
478 }

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

479 \newcommand{\ItemDescribeObject}[2][]{%
480 \item[\texttt{#2}:]%
481 \setlength{\parskip}{1.5ex}%
482 \DescribeObject[#1]{#2}%
483 }

\ItemDescribeOther [⟨class⟩] {⟨name⟩}

484 \newcommand{\ItemDescribeOther}[2][]{%
485 \item[\texttt{#2}:]%
486 \setlength{\parskip}{1.5ex}%
487 \DescribeOther[#1]{#2}%
488 }

```

## 6.11 \margintag, \watchout

\margintagcolor The color of the \margintag.

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

\margintag {⟨text⟩}

Prints a colored margin tag.

```

490 \newcommand{\margintag}[1]{%
491 \@ifundefined{@capttype}{% not float?
492 \marginpar{\raggedleft\textcolor{\margintagcolor}{#1}}%
493 \ignorespaces%
494 }{}% not float?
495 }
```

\watchoutcolor The color of the \watchout.

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

---

\watchout [*<text>*]

Prints a warning sign and optional text.

```
497 \newcommand{\watchout}[1][]{%
498 \@ifundefined{@capttype}{% not float?
499   \marginpar{%
500     \raggedleft%
501     \textcolor{\watchoutcolor}{\warningsign\normalsize\quad#1}%
502   }%
503   \ignorespaces%
504 }{}% not float?
505 }
```

## 6.12 The dtxexample environment

Also see example 13 on page page 26.

File dtxexample\_cut.tex Used to store then \input example code.

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

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

\dtxexamplecodename The text name of the code section.

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

\dtxexampleresultname The text name of the result section.

```
508 \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.

```
509 \NewDocumentEnvironment{dtxexample}{s +0{} m}
510 {% start dtxexample
```

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

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

When the environment closes:

```
513 {%
514 \endVerbatimOut
515 \par
516 \addvspace{\bigskipamount}
```

Finish the verbatim output:

```
514 \endVerbatimOut
515 \par
516 \addvspace{\bigskipamount}
```

If unstarred, typeset the example in a minipage:

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

Emulated a float of type “example”:

```
518 \captionsetup{type=dtxexample}%
519 \hrule\medskip
520 \caption{#3}
```

Typeset the contents as verbatim:

```
521 \textcolor{DTXD@examplerulecolor}{\smallskip\hrule}
522 \smallskip
523 {\scriptsize\itshape\dtxexamplecodename}
524 \VerbatimInput[tabsize=4]{dtxexample_cut.tex}
525 \unskip
526 \textcolor{DTXD@examplerulecolor}{\hrule}
527 \smallskip
528 {\scriptsize\itshape\dtxexampleresultname}
529
```

Possible add the optional cross-references or notes:

```
530 \ifstrempty{#2}
531 {}
532 {{\itshape\small #2}}
```

If unstarred, close the `\minipage`.

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

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

```
535 \AfterEndEnvironment{dtxexample}
536 {%
```

Execute the code:

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

Closing rule::

```
538 \medskip\hrule%
539 }
```

[dtxexample] A new float type for the examples.

\DeclareFloatingEnvironment

```
540 \DeclareFloatingEnvironment[
541 fileext=lox,
542 listname={List of Examples},
543 name=Example,
544 placement=hbp
545 ]{dtxexample}
```

[dtxexample] \captionsetup Caption setup for the examples.

```
546 \captionsetup*[dtxexample]{
547 format=hang,
548 font=bf,
549 justification=raggedright,
550 singlelinecheck=false,
551 skip=0pt,
552 position=top,
553 }
```

[dtxexample] \crefname Name for cleveref.

```
554 \AtBeginDocument{
555 \@ifpackageloaded{cleveref}{\crefname{dtxexample}{example}{examples}}{}%
556 }
```

## 6.13 noindmacro and noindenvironment

Similar to macro and environment, but not indexed.

```

Env  noindmacro  {⟨name⟩}

557 \newenvironment{noindmacro}[1]
558 {
559     \setlength{\parskip}{\marginparpush}
560     \leavevmode\par\DTXD@margintag{}{\cmd{#1}}{}
561 }
562 {\unskip}

```

```

Env  noindenvironment  {⟨name⟩}

563 \newenvironment{noindenvironment}[1]
564 {
565     \setlength{\parskip}{\marginparpush}
566     \leavevmode\par\DTXD@margintag{}{\#1}{Env}
567 }
568 {\unskip}

```

## 6.14 sourcedisplay, UIdisplay, docsidebar

For use in a sourcedisplay:

\fquad Forces a quad indent.

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

\fqquad Forces a double-quad indent.

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

\fqqquad Forces a triple-quad indent.

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

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

```

572 \DefineVerbatimEnvironment{sourceverb}{Verbatim}
573     {gobble=\DTXD@gobble,tabsize=4,xleftmargin=2em}
574 \BeforeBeginEnvironment{sourceverb}{\vspace*{-5\parskip}}

```

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

---

```

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

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

```

578 \newenvironment{sourcedisplay}
579 {
580     \leavevmode
581     \par
582     \fqqquad\minipage{\linewidth-4em}
583     \ttfamily
584 }
585 {%
586     \endminipage
587     \par
588 }
```

Env `UIdisplay` To typeset a user interface display.

```

589 \newenvironment{UIdisplay}
590 {
591     \leavevmode
592     \par
593     \fqqquad\minipage{\linewidth-4em}
594     \sffamily\bfseries
595 }
596 {
597     \endminipage
598     \par
599 }
```

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

```
600 \newcommand*{\userentryname}{Enter~$\rightarrow$}
```

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

Typesets text to be entered by the users.

```

601 \newcommand{\userentry}[1]{%
602 \par
603 \fqqquad%
604 \begin{minipage}{\linewidth-2em}
605     {\footnotesize \userentryname}\quad\cmds{#1}
606 \end{minipage}
607 \par
```

```
608 }
```

Env docsidebar To typeset a sidebar in the documentation.

```
609 \newenvironment{docsidebar}[1][]{%
610 {%
611   \quote\unskip\medskip
612   \setlength{\parskip}{1.5ex}%
613   \ifblank{#1}{}{\textit{#1}\newline}%
614   \rule[.5\bigskipamount]{\ linewidth}{.4pt}%
615   \newline%
616 }%
617 {%
618   \leavevmode\par
619   \rule[\bigskipamount]{\ linewidth}{.4pt}%
620   \endquote\unskip
621 }
```

## 6.15 Formatted objects

Macros to format references to various kinds of objects.

### 6.15.1 L<sup>A</sup>T<sub>E</sub>X objects

\pkg or class

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

\env

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

\ctr

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

\bool

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

\optn

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

```
\TOC  
627 \providerobustcmd*\{\TOC\}{\acro{TOC}}  
  
\LOF  
628 \providerobustcmd*\{\LOF\}{\acro{LOF}}  
  
\LOT  
629 \providerobustcmd*\{\LOT\}{\acro{LOT}}
```

### 6.15.2 Programs and commands

```
\cmds  
630 \providerobustcmd*\{\cmds\}[1]{\mbox{\textbf{\texttt{#1}}}}  
  
\progcode  
631 \providerobustcmd*\{\progcode\}[1]{\mbox{\texttt{#1}}}  
  
\prog  
632 \newcommand*\{\DTXD@prog\}[1]{%  
633   \mbox{\textsf{\textsl{\detokenize{#1}}}}}%  
634   \endgroup%  
635 }  
636  
637 \providerobustcmd*\{\prog\}{%  
638   \begingroup%  
639   \catcode`\_=12%  
640   \DTXD@prog%  
641 }  
  
\filenm  
642 \newcommand*\{\DTXD@filenm\}[1]{%  
643   \mbox{\texttt{\detokenize{#1}}}}%  
644   \endgroup%  
645 }  
646  
647 \providerobustcmd*\{\filenm\}{%  
648   \begingroup%
```

```
649     \catcode`\_=12%
650     \DTXD@filenm%
651 }
```

\UI General user-interface text.

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

### 6.15.3 File types

\ODT

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

\SVG

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

\PNG

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

\GIF

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

\JPG

```
657 \providerobustcmd*\{\JPG\}{\acro{JPG}}
```

\EPS

```
658 \providerobustcmd*\{\EPS\}{\acro{EPS}}
```

\PDF

```
659 \providerobustcmd*\{\PDF\}{\acro{PDF}}
```

\DVI

```
660 \providerobustcmd*\{\DVI\}{\acro{DVI}}
```

#### 6.15.4 Internet

\UTF

661 \providerobustcmd\*\{\UTF\}{\acro{UTF}}

\URL

662 \providerobustcmd\*\{\URL\}{\acro{URL}}

\element

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

\attribute

664 \providerobustcmd\*\{\attribute\}[1]{\mbox{\texttt{<\#1>}}}

\HTML

665 \providerobustcmd\*\{\HTML\}{\acro{HTML}}

\HTMLfive

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

\CSS

667 \providerobustcmd\*\{\CSS\}{\acro{CSS}}

\CSSthree

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

\EPUB

669 \providerobustcmd\*\{\EPUB\}{\acro{EPUB}}

#### 6.15.5 Specific programs

\tikz

670 \providerobustcmd\*\{\tikz\}{\tikz\textit{k}z}

```
\MathML  
671 \providerobustcmd*\{\MathML\}{Math\acro{ML}}  
  
\CTAN  
672 \providerobustcmd*\{\CTAN\}{\acro{CTAN}}  
  
\TDS  
673 \providerobustcmd*\{\TDS\}{\acro{TDS}}
```

### 6.15.6 Acronyms, brand names, trademarks

```
\brand  
674 \providerobustcmd*\{\brand\}[1]{\textsc{\#1}}  
  
\acro  
675 \providerobustcmd*\{\acro\}[1]{\textsc{\lowercase{\#1}}}  
  
\supregistered Superscript trademark symbol.  
676 \providerobustcmd*\{\supregistered\}{\textsuperscript{\textregistered}}
```

## 6.16 Logos

```
\LuaTeX \LuaTeX  
677 \providerobustcmd*\{\LuaTeX\}{\mbox{\LuaTeX}}  
  
\LuaLaTeX \LuaLaTeX  
678 \providerobustcmd*\{\LuaLaTeX\}{\mbox{\LuaLaTeX}}  
  
\XeTeX XETEX, XELATEX  
\XeLaTeX  
679 \providerobustcmd*\{\XeTeXrevE\}  
680 { \hspace{-.1667em} }\raisebox{-.5ex}{E}\hspace{-.125em}
```

```

681
682 \AtBeginDocument{
683 \@ifpackageloaded{graphics}{
684     \renewrobustcmd*{\XeTeXrevE}
685         {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
686 }()
687 }
688
689 \providerobustcmd*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
690 \providerobustcmd*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}

```

\AmS *A*<sub>M</sub>S

```

691 \providerobustcmd*{\AmS}{%
692     \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex\hbox{$\mathcal M$}$}%
693     \hbox{$\mathcal M\kern-.2em\mathcal S$}%
694 }

```

\LyX *LyX*

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

\BibTeX *BIB*<sub>T</sub><sub>E</sub>X

```
696 \providerobustcmd*{\BibTeX}{\mbox{B\textrm{textrm}{ib}\TeX}}
```

\MakeIndex *MakelIndex*

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

\ConTeXt *ConTeXt*

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

\MiKTeX *MiK*<sub>T</sub><sub>E</sub>X

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

## 6.17 Dashes and slashes

\thinspace A breakable thin skip.

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

\endash An endash: –

701 \def\endash{–}

\emdash An emdash: —

702 \def\emdash{—}

\thinbrspace A thin space which allows a line break.

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

\thinthinbrspace A thin space which allows a line break.

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

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

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

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

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

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

707 \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 . . . . .
v0.11	
\DTXD@cmdmargintagindex:	Index tag no longer plural. . . . . 39
\DTXD@index:	Index tag no longer plural. . . . . 37
\watchout:	Changed to \raggedleft. 50
General:	2018/03/30 . . . . . 1
v1.00	
\DTXD@cmdmargintagindex:	Sans tag font. . . . . 38
\DTXD@filemarginparindex:	Fix: File class. . . . . 41
\DTXD@printtype:	Sans tag font. . . . . 35
\DescribeClass:	Fix: Allow underscore. . . . . 43
\DescribeDefault:	Added. . . . . 46
\DescribeDefaultcolor:	Added. . . . . 45
\DescribeMacro:	Sans tag font. . . . . 39
\DescribePackage:	Fix: Allow underscore. . . . . 43
\ItemDescribeArgument:	Added. . . . . 46
\ItemDescribeBoolean:	Added. . . . . 46
\ItemDescribeClass:	Added. . . . . 47
\ItemDescribeCommand:	Added. . . . . 48
\ItemDescribeCounter:	Added. . . . . 47
\ItemDescribeEnv:	Added. . . . . 46
\ItemDescribeFile:	Added. . . . . 48
\ItemDescribeKey:	Added. . . . . 47
\ItemDescribeLength:	Added. . . . . 47
\ItemDescribeMacro:	Added. . . . . 46
\ItemDescribeObject:	Added. . . . . 49
\ItemDescribeOption:	Added. . . . . 48
\ItemDescribeOther:	Added. . . . . 49
\ItemDescribePackage:	Added. . . . . 47
\ItemDescribeProgram:	Added. . . . . 48
\dtxexamplecodename:	Added. . . . . 50
\dtxexampleresultname:	Added. . . . . 50
\fqqquad:	Added. . . . . 53
\fqquad:	Added. . . . . 53
\fquad:	Added. . . . . 53
\margintag:	Uses \margintacolor. 49
\margintacolor:	Added. . . . . 49
\userentry:	Added. . . . . 54
\userentryname:	Added. . . . . 54
\watchoutcolor:	Added. . . . . 49
General:	2019/01/11 . . . . . 1
	Added formatted objects. . . . . 55
	Added logos. . . . . 59
	Cut file name changed to dtxexample_cut.tex . . . . . 50
UIDisplay:	Added. . . . . 54
docsidebar:	Added. . . . . 55
noindenvironment:	Added. . . . . 53
noindmacro:	Added. . . . . 53
sourcedisplay:	Added. . . . . 54
sourceverb:	Added. . . . . 53
fsourceverb:	Added. . . . . 53
v1.01	
\DTXD@cmdmargintagindex:	Put margin tag class in brackets. . . . . 38
\DTXD@margintag:	Put margin tag class in brackets. . . . . 36
\DescribeMacro:	Put margin tag class in brackets. . . . . 39
\DescribeObject:	Put margin tag class in brackets. . . . . 44
\DescribeOther:	Put margin tag class in brackets. . . . . 45
General:	2019/03/22 . . . . . 1
	Sanitize PDF bookmarks. . . . . 32
UIDisplay:	Reduced width. . . . . 54
sourcedisplay:	Reduced width. . . . . 54
v1.02	
\DescribeEnv:	Fix if not ltxdoc class. 40
\DescribeMacro:	Fix if not ltxdoc class. . . . . 39

\DescribeObject: \raggedleft	
margin par.	44
Added \ignorespaces.	44
\DescribeOther: \raggedleft	
margin par.	45
Added \ignorespaces.	45
\PrintEnvName: Fix if not ltxdoc	
class.	35
\usage:	Fix if not ltxdoc class.
	36
General:	2019/07/16
	1
Fix if not doc package.	34
Fix if not hyperref package.	32
Fix if not ltxdoc class.	31, 34
DTXD@gobble:	Fix if not doc package.
	33
dtxexample:	Fix if not doc package.
	50

## 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		C
[H] (argument) [figure]	23	booleanname (boolean) [descexamples] <span style="color:red">25</span>
[M] (argument) [figure]	23	\brand ..... <span style="color:red">15, 674</span>
[\\mymacro]:		
bold (argument)	23	caption (package) ..... <span style="color:red">31</span>
		\captionsetup
		[dtxexample] ..... <span style="color:red">52</span>
		[photograph] ..... <span style="color:red">18</span>
A		class:
\\acro	<span style="color:red">16, 675</span>	[descexamples]:
Additional Item [otherclass]	<span style="color:red">24</span>	class_name ..... <span style="color:red">25</span>
\\AmS	<span style="color:red">16, 691</span>	[examples]:
another_big_file.txt (file) [bigfiles]	<span style="color:red">21</span>	sample_class ..... <span style="color:red">21</span>
argument:		class_name (class) [descexamples] ... <span style="color:red">25</span>
[\\mymacro]:		\\cmds ..... <span style="color:red">14, 630</span>
bold	23	\\cnameref
[figure]:		[photograph] ..... <span style="color:red">18</span>
[H]	23	
[M]	23	
\\attribute	<span style="color:red">15, 664</span>	[color]:
		DTXD@examplerulecolor ..... <span style="color:red">50</span>
B		othercolor ..... <span style="color:red">24</span>
Before myenvironment	<span style="color:red">24</span>	somecolor ..... <span style="color:red">24</span>
\\BibTeX	<span style="color:red">16, 696</span>	command:
[bigfiles]:		[descexamples]:
another_big_file.txt (file)	<span style="color:red">21</span>	command_name ..... <span style="color:red">25</span>
really_big_file.txt (file)	<span style="color:red">21</span>	OS_command ..... <span style="color:red">21</span>
bold (argument) [\\mymacro]	23	command_name (command) [descexamples] .....
\\bool	<span style="color:red">14, 625</span>	25
boolean:		\\ConTeXt ..... <span style="color:red">16, 698</span>
[descexamples]:		counter:
booleanname	25	[examples]:
[examples]:		samplecounter ..... <span style="color:red">20</span>
sampleboolean	20	

\crefname		\DTXD@gobble .....	84
[dtxexample]	52	\DTXD@index .....	173
\CSS	15, 667	\DTXD@macroname .....	204
\CSSthree	15, 668	\DTXD@margintag .....	161
\CTAN	15, 672	\DTXD@margintagindex .....	199
\ctr	14, 624	\DTXD@origwrindex .....	160
		\DTXD@printtype .....	156
		\DTXD@verbatimcmd .....	207
<b>D</b>			
\Dash	16, 705	[dtxexample]:	
\dash	16, 706	\captionsetup .....	52
\DeclareFloatingEnvironment		\crefname .....	52
[dtxexample]	52	\DeclareFloatingEnvironment ...	52
[photograph]	18	dtxexample (environment) .....	12, 509
[descexamples]:		dtxexample_cut.tex (file) .....	50
booleanname (boolean)	25	\dtxexamplecodename .....	12, 507
class_name (class)	25	\dtxexampleresultname .....	12, 508
command_name (command)	25	\DVIm .....	15, 660
file_name (file)	25		
keyname (key)	25	<b>E</b>	
\lengthname (length)	25	\element .....	15, 663
\macro	25	\emdash .....	16, 702
package_name (package)	25	\endash .....	16, 701
program_name (program)	25	\env .....	14, 623
\DescribeArgument	9, 314	environment (environment) .....	8
\DescribeBoolean	9, 316	environment:	
\DescribeClass	9, 309	[kindofenvironment]:	
\DescribeCommand	10, 297	otherenvironment .....	19
\DescribeCounter	9, 320	docsidebar .....	13
\DescribeDefault	11, 381	dtxexample .....	12
\DescribeDefaultcolor	11, 380	environment .....	8
\DescribeEnv	8, 269	fsourceverb .....	13
\DescribeFile	10, 285	macro .....	8
\DescribeKey	9, 322	myenvironment .....	19
\DescribeLength	9, 318	noindenvironment .....	12
\DescribeMacro	8, 237	noindmacro .....	12
\DescribeObject	10, 324	sourcedisplay .....	13
\DescribeOption	9, 312	sourceverb .....	13
\DescribeOther	10, 356	UIdisplay .....	13
\DescribePackage	9, 303	environments:	
\DescribeProgram	10, 291	docsidebar .....	609
docsidebar (environment)	13, 609	dtxexample .....	509
\DTX@filename	272	fsourceverb .....	575
\DTXD@cmdmargintagindex	210	noindenvironment .....	563
\DTXD@DescribeClass	306	noindmacro .....	557
\DTXD@DescribeCommand	294	sourcedisplay .....	578
\DTXD@DescribeFile	282	sourceverb .....	572
\DTXD@DescribePackage	300	UIdisplay .....	589
\DTXD@DescribeProgram	288	\EPS .....	15, 658
DTXD@examplerulecolor [color]	50	\EPUB .....	15, 669
\DTXD@filemarginparindex	273	etoolbox (package) .....	31

[examples]:	\ItemDescribeCounter .....	<i>11, 414</i>	
sample_class (class) .....	<i>21</i> \ItemDescribeEnv .....	<i>10, 394</i>	
sampleboolean (boolean) .....	<i>20</i> \ItemDescribeFile .....	<i>11, 449</i>	
samplecounter (counter) .....	<i>20</i> \ItemDescribeKey .....	<i>11, 419</i>	
samplekey (key) .....	<i>22</i> \ItemDescribeLength .....	<i>11, 409</i>	
sampleoption (option) .....	<i>21</i> \ItemDescribeMacro .....	<i>10, 389</i>	
samplepackage (package) .....	<i>21</i> \ItemDescribeObject .....	<i>11, 479</i>	
sampletwokey (key) .....	<i>22</i> \ItemDescribeOption .....	<i>11, 444</i>	
	\ItemDescribeOther .....	<i>11, 484</i>	
<b>F</b>	\ItemDescribePackage .....	<i>11, 424</i>	
fancyvrb (package) .....	<i>31</i> \ItemDescribeProgram .....	<i>11, 459</i>	
[figure]:			
[H] (argument) .....	<i>23</i>		
[M] (argument) .....	<i>23</i> \JPG .....	<i>15, 657</i>	
file:			
[bigfiles]:			
another_big_file.txt .....	<i>21</i>	key:	
really_big_file.txt .....	<i>21</i>	[descexamples]:	
[descexamples]:		keyname .....	<i>25</i>
file_name .....	<i>25</i>	[examples]:	
dtxexample_cut.tex .....	<i>50</i>	samplekey .....	<i>22</i>
lone_file.txt .....	<i>21</i>	sampletwokey .....	<i>22</i>
file_name (file) [descexamples] .....	<i>25</i>	[groupofkeys]:	
\filenm .....	<i>14, 642</i>	firstkey .....	<i>22</i>
firstkey (key) [groupofkeys] .....	<i>22</i>	secondkey .....	<i>22</i>
\fqqquad .....	<i>13, 571</i>	lonekey .....	<i>22</i>
\fqquad .....	<i>13, 570</i>	keyname (key) [descexamples] .....	<i>25</i>
\fquad .....	<i>13, 569</i>	[kindofenvironment]:	
fsourceverb (environment) .....	<i>13, 575</i>	otherenvironment (environment) ..	<i>19</i>
<b>G</b>			
\GIF .....	<i>15, 656</i>		
group of objects .....	<i>19</i>	length:	
[groupofkeys]:		[descexamples]:	
firstkey (key) .....	<i>22</i>	\lengthname .....	<i>25</i>
secondkey (key) .....	<i>22</i>	[photograph]:	
<b>H</b>			
\HTML .....	<i>15, 665</i>	\photowidth .....	<i>20</i>
\HTMLfive .....	<i>15, 666</i>	\IndexMin .....	<i>34</i>
hyperref (package) .....	<i>32</i>	license agreement .....	<i>24</i>
<b>I</b>		\LOF .....	<i>14, 628</i>
index		lone_file.txt (file) .....	<i>21</i>
by group .....	<i>19</i>	lonekey (key) .....	<i>22</i>
\IndexMin (length) .....	<i>34</i>	\LOT .....	<i>14, 629</i>
\ItemDescribeArgument .....	<i>10, 399</i>	\LuaLaTeX .....	<i>16, 678</i>
\ItemDescribeBoolean .....	<i>11, 404</i>	\LuaTeX .....	<i>16, 677</i>
\ItemDescribeClass .....	<i>11, 434</i>	\LyX .....	<i>16, 695</i>
\ItemDescribeCommand .....	<i>11, 469</i>	[descexamples] .....	<i>25</i>
<b>M</b>			
		macro (environment) .....	<i>8</i>
		\macroname .....	

makeidx (package) . . . . .	31	package_name (package) [descexamples] . . . . .	25
\MakeIndex . . . . .	16, 697	\parg . . . . .	8
\marg . . . . .	8	\PDF . . . . .	15, 659
margin tag missing . . . . .	30	[photograph]:	
\marginpar . . . . .	11, 490	\captionsetup . . . . .	18
\marginparcolor . . . . .	11, 489	\cnameref . . . . .	18
\MathML . . . . .	15, 671	\DeclareFloatingEnvironment . . . . .	18
\MiKTeX . . . . .	16, 699	\photowidth (length) . . . . .	20
myenvironment (environment) . . . . .	19	pict2e (package) . . . . .	33
\mymacro . . . . .	18	\pkg . . . . .	14, 622
		\PNG . . . . .	15, 655
		\PrintEnvName . . . . .	153
		\prog . . . . .	14, 632
		\progcode . . . . .	14, 631
		program:	
		[descexamples]:	
		program_name . . . . .	25
		program_name . . . . .	21
		program_name (program) . . . . .	21
		program_name (program) [descexamples] . . . . .	25
			R
		randomobject . . . . .	24
		really_big_file.txt (file) [bigfiles] . . . . .	21
			S
		sample_class (class) [examples] . . . . .	21
		sampleboolean (boolean) [examples] . . . . .	20
		samplecounter (counter) [examples] . . . . .	20
		samplekey (key) [examples] . . . . .	22
		sampleoption (option) [examples] . . . . .	21
		samplepackage (package) [examples] . . . . .	21
		sampletwokey (key) [examples] . . . . .	22
		secondkey (key) [groupofkeys] . . . . .	22
		\\$lash . . . . .	16, 707
		somecolor [color] . . . . .	24
		sourcedisplay (environment) . . . . .	13, 578
		sourceverb (environment) . . . . .	13, 572
		splitidx (package) . . . . .	31
		\supregistered . . . . .	16, 676
		\\$VG . . . . .	14, 654
			T
		\TDS . . . . .	15, 673
		\thinbrspace . . . . .	16, 703
		\thinspace . . . . .	16, 700
		\thinthinbrspace . . . . .	16, 704
		\tikz . . . . .	15, 670
		\TOC . . . . .	14, 627

<b>U</b>	<b>X</b>
\UI . . . . . <i>14</i> , <u>652</u>	\watchout . . . . . <i>11</i> , <u>497</u>
UIdisplay (environment) . . . . . <i>13</i> , <u>589</u>	\watchoutcolor . . . . . <i>12</i> , <u>496</u>
\URL . . . . . <i>15</i> , <u>662</u>	
\usage . . . . . <i>158</i>	
\userentry . . . . . <i>13</i> , <u>601</u>	xcolor (package) . . . . . <i>31</i>
\userentryname . . . . . <i>13</i> , <u>600</u>	\XeLaTeX . . . . . <i>16</i> , <u>679</u>
\UTF . . . . . <i>15</i> , <u>661</u>	\XeTeX . . . . . <i>16</i> , <u>679</u>
	xifthen (package) . . . . . <i>31</i>
<b>W</b>	xparse (package) . . . . . <i>31</i>
\warningsign . . . . . <i>76</i>	xstring (package) . . . . . <i>31</i>