Describe additional object types in \texttt{dtx} or \texttt{tex} source files.

\textbf{Abstract}

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

Each described item is given a margin tag similar to \texttt{\DescribeEnv}, and is listed in the index by itself and also by type of object displayed in parentheses (length, filename, etc). Each item may be sorted further by an optional category displayed in brackets, such as \texttt{[category\_name]}.

The \texttt{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 \texttt{\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.

\texttt{dtxdescribe} works with the \texttt{ltxdoc} class, but also works with the standard classes as well, except that the macro and environment environments are not supported. Either \texttt{makeidx} or \texttt{splitidx} may be loaded by the user. \texttt{makeidx} will be used by default. \texttt{dtxdescribe} works with \texttt{pdf\LaTeX}, \texttt{\LaTeX{}}, and \texttt{ Lua\LaTeX}, and perhaps other engines as well.

\textit{See change in setup of hyperref and cleveref: Section 2 on page 7.}
# Contents

1 Introduction 6

2 Using dtxdescribe 7

3 The macros, and the dtxexample environment 9
   3.1 Macros and environments 9
   3.2 Arguments 9
   3.3 Booleans, lengths, counters, hooks, sockets, plugs, keys 10
   3.4 Packages, classes, options 10
   3.5 Files, programs, commands 11
   3.6 Other source objects 11
   3.7 In a description environment 12
   3.8 Defaults 12
   3.9 Nesting 13
   3.10 \margintag, \watchout 13
   3.11 dtxexample environment 13
   3.12 noindmacro and noindenvironment environments 14
   3.13 sourceverb, sourcedisplay, UIDisplay, docsidebar 14
   3.14 Formatted objects 15
      3.14.1 \LaTeX\ objects 15
      3.14.2 Programs and commands 16
      3.14.3 File types 16
      3.14.4 Internet 16
      3.14.5 Specific programs 17
      3.14.6 Acronyms, brand names, trademarks 17
   3.15 Logos 17
   3.16 Dashes and slashes 18
6.22.3 File types ........................................ 75
6.22.4 Internet ........................................... 76
6.22.5 Specific programs ................................. 77
6.22.6 Acronyms, brand names, trademarks .......... 77
6.23 Logos .................................................. 77
6.24 Dashes and slashes .................................. 79

7 Compiling dtxdescribe ................................. 80

A Change History ........................................ 81

B Index ...................................................... 82

List of Examples

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

1 A Figure .................................................. 28
1 Introduction

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

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

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

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

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

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

To use \texttt{dtxdescribe} with the \texttt{ltxdoc} class and .dtx files:

```latex
\documentclass{ltxdoc}
\usepackage{lmodern}
\usepackage{dtxdescribe}
\usepackage{packagename} % the name of your new package
\AtBeginDocument{
    \hypersetup{...}% If needed.
    \pdfstringdefDisableCommands{ ... }% If needed.
}
\AddToHook{begindocument/before}{% Before .aux file is loaded.
    \usepackage{cleveref}% If needed.
    \crefname{somename}{name}{names}% If needed.
}
\begin{document}
\end{document}
```

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

To use \texttt{dtxdescribe} with the regular classes, such as \texttt{article}, use:

```latex
\documentclass{article}
\usepackage{dtxdescribe}
\usepackage{hyperref}% If needed.
\hypersetup{...}% If needed.
\pdfstringdefDisableCommands{ ... }% If needed.
\usepackage{cleveref}% If needed.
\crefname{somename}{name}{names}% If needed.
\begin{document}
\end{document}
```
Optional "categories" may be assigned to the objects being described, including the new versions of \DescribeMacro and \DescribeEnv. These categories are printed in the margin tag and index entry for each item, and also generate additional index entries sorted by category. This is especially useful for key/value sets, where several sets may appear in the same document.

\textbf{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.

\texttt{\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.

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

The \texttt{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. \texttt{\listofdtxexamplefloats} prints the list of examples.
### 3 The macros, and the dtxexample environment

#### 3.1 Macros and environments

These are only provided by the \texttt{bdoc} class and \texttt{doc} package to document a .\texttt{dtx} file, where comments are used by \texttt{docstrip} to disable these environments in the resulting .\texttt{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.

\begin{verbatim}
\DescribeMacro \langle\textit{category}\rangle \{\langle\textit{name}\rangle\}
\end{verbatim}

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

\begin{verbatim}
\DescribeEnv \langle\textit{category}\rangle \{\langle\textit{environment name}\rangle\}
\end{verbatim}

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

#### 3.2 Arguments

The \Describe\ldots macros may be followed by \texttt{\textbackslash marg}, \texttt{\textbackslash oarg}, and \texttt{\textbackslash parg} to describe arguments passed to the macros.

\begin{verbatim}
\textbackslash marg \{\langle\textit{text}\rangle\}
\end{verbatim}

Shows a mandatory argument for a macro or environment.

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

\begin{verbatim}
\textbackslash oarg \{\langle\textit{text}\rangle\}
\end{verbatim}

Shows an optional argument for a macro or environment.

The results looks like [\langle\textit{optional}\rangle].

\begin{verbatim}
\textbackslash parg \{\langle\textit{text}\rangle\}
\end{verbatim}

Used for “picture” arguments, such as coordinates.

The result looks like \langle\textit{coordinate}\rangle.
\DescribeArgument \[\langle category \rangle \] \{ \langle argument \rangle \}

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 category may be used to categorize arguments by their macro or environment name. See example 10 on page 24.

### 3.3 Booleans, lengths, counters, hooks, sockets, plugs, keys

See example 4 on page 21.

\DescribeBoolean \[\langle category \rangle \] \{ \langle name \rangle \}

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

\DescribeLength \[\langle category \rangle \] \{ \langle name \rangle \}

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

\DescribeCounter \[\langle category \rangle \] \{ \langle name \rangle \}

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

\DescribeHook \[\langle category \rangle \] \{ \langle name \rangle \}

Describes a hook. Given a 'Hook' tag in the margin and index. The category may be used to categorize hooks by package. The hook name may or may not have a backslash. Example:

\DescribeHook{\hookname}
\DescribeHook{\LaTeX}{para/begin}

\DescribeSocket \[\langle category \rangle \] \{ \langle name \rangle \}

\DescribePlug \[\langle category \rangle \] \{ \langle name \rangle \}

Describes a socket or plug. Given a 'Socket' or 'Plug' tag in the margin and index. The category may be used to categorize sockets and plugs by package. Example:

\DescribeSocket{socketname}
\DescribePlug{plugname}

\DescribeKey \[\langle category \rangle \] \{ \langle name \rangle \}

Describes a key. Given a 'Key' tag in the margin and index. The category may be used to categorize keys by their key/value group. See example 9 on page 23.

### 3.4 Packages, classes, options

\DescribePackage \[\langle category \rangle \] \{ \langle name \rangle \}
Describes a package. Given a ‘Pkg’ tag in the margin and index.

\DescribeClass \{⟨category⟩\} \{⟨name⟩\}

Describes a \LaTeX{} class. Given a ‘Cls’ tag in the margin and index.

\DescribeOption \{⟨category⟩\} \{⟨name⟩\}

Describes a \LaTeX{} package or class option. Given an ‘Opt’ tag in the margin and index.

3.5 Files, programs, commands

\DescribeFile \{⟨category⟩\} \{⟨name⟩\}

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

\DescribeProgram \{⟨category⟩\} \{⟨name⟩\}

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

\DescribeCommand \{⟨category⟩\} \{⟨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 \{⟨category⟩\} \{⟨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 category is used, it is added to the margin tag, appended to the index entry, and a second index entry is created grouped by category. If a macro name is to be described, use \DescribeMacro instead. See example 11 on page 25.

\DescribeOther \{⟨category⟩\} \{⟨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 category is used, it is added to the margin tag, appended to the index entry, and a second index entry is created grouped by category. See example 12 on page 25.
3.7 In a description environment

To describe an object using a description environment, use the following. See example 13 on page 26.

\ItemDescribeMacro \([\textit{category}] \{\textit{name}\} \) A description.
\ItemDescribeEnv \([\textit{category}] \{\textit{name}\} \) A description.
\ItemDescribeArgument \([\textit{category}] \{\textit{argument}\} \) A description.
\ItemDescribeBoolean \([\textit{category}] \{\textit{name}\} \) A description.
\ItemDescribeLength \([\textit{category}] \{\textit{name}\} \) A description.
\ItemDescribeCounter \([\textit{category}] \{\textit{name}\} \) A description.
\ItemDescribeHook \([\textit{category}] \{\textit{name}\} \) A description.
\ItemDescribeSocket \([\textit{category}] \{\textit{name}\} \) A description.
\ItemDescribePlug \([\textit{category}] \{\textit{name}\} \) A description.
\ItemDescribeKey \([\textit{category}] \{\textit{package_name}\} \) With underscores.
\ItemDescribePackage \([\textit{category}] \{\textit{class_name}\} \) With underscores.
\ItemDescribeOption \([\textit{category}] \{\textit{name}\} \) A description.
\ItemDescribeFile \([\textit{category}] \{\textit{file_name}\} \) With underscores.
\ItemDescribeProgram \([\textit{category}] \{\textit{program_name}\} \) With underscores.
\ItemDescribeCommand \([\textit{category}] \{\textit{command_name}\} \) With underscores.
\ItemDescribeObject \([\textit{category}] \{\textit{name}\} \) A description.
\ItemDescribeOther \([\textit{category}] \{\textit{name}\} \) A description.

3.8 Defaults

\DescribeDefault \{\textit{value}\}  
\textbf{Default:} \textit{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 \{\textit{value}\}  
\textbf{Default:} \textit{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 Nesting

\shownesting * [(fraction of \linewidth)] {(container name)} {(contents)}

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

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

See example 14 on page 27 for an example.

3.10 \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 The color of the \margintag.
Default: blue!70!black

\watchout \{(text)\}

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

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

3.11 dtxexample environment

dtxexample (env) * [(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.

Prints the list of examples.

See example 15 for a demonstration of how dtxexample works.

The examples may be customized by redefining the following, perhaps for another language:

\dtxexamplecodename
\dtxexampleresultname

The text name of the code section.
The text name of the result section.

3.12 noindmacro and noindenvironment environments

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

\noindent
\begin{macro}{\macroname} \oarg{optional} \marg{mandatory}
\end{macro}

Replace

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

and similarly for noindenvironment.

3.13 sourceverb, sourcedisplay, UIDisplay, docsidebar

Display source code verbatim. Uses optional fancyverb 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.
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 16 on page 29.

Display source code with manual formatting. This is not a verbatim environment.

\quad Single-level indent inside a sourcedisplay.
\fquad Double-level indent inside a sourcedisplay.
\fqqquad Triple-level indent inside a sourcedisplay.

Displays a user interface, such as a dialog box entry or a menu selection. See example 18 on page 30. Also see the \UI macro.

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

Creates a sidebar within the document. See example 19 on page 31.

### 3.14 Formatted objects

Macros to format references to various kinds of objects.

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

#### 3.14.1 \LaTeX objects

\pkg \(<\text{packagename}>\)  Prints as packagename. Also for a classname.
\cs \(<\text{csname}>\)  Prints as csname.
\env \(<\text{environment}>\)  Prints as environment.
\marg \(<\text{argument}>\)  Prints \(<\text{arg}>\). Mandatory argument.
\oarg \(<\text{argument}>\)  Prints \([\text{arg}]\). Optional argument.
\parg \(<\text{counter}>\)  Prints \(<\text{arg}>\). Picture-mode argument.
\ctr \(<\text{counter}>\)  Prints as counter.
\bool \{ \langle boolean \rangle \} \quad \text{Prints as boolean.}

\optn \{ \langle option \rangle \} \quad \text{Prints as option, for example to a macro, package, class.}

\toc \text{TOC: Table of contents.}

\lof \text{LOF: List of figures.}

\lot \text{LOT: List of tables.}

3.14.2 Programs and commands

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

\prog \text{Prints as grep, make: A program name. Underscores allowed.}

\filenm \text{Prints as file_name: Underscores allowed.}

\ui \text{Prints as General user-interface text: What the user sees on the display. Also see the UIDisplay environment.}

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

3.14.3 File types

\odt \text{ODT OpenDocument Format word processing document}

\svg \text{SVG image format}

\png \text{PNG image format}

\gif \text{GIF image format}

\jpg \text{JPG image format}

\eps \text{EPS image format}

\pdf \text{PDF image format}

\dvi \text{DVI image format}

3.14.4 Internet

\utf \text{UTF: Unicode}

\url \text{URL: Uniform Resource Locator}

\element \{ \langle element name \rangle \} \quad \text{Prints as <element>, an HTML/CSS element}
\attribute \{⟨attribute name⟩\} Prints as attribute, an HTML/CSS attribute. \pdf\TeX\ and \X\TeX\ only. Not for \Lua\TeX.\n
\attrib \{⟨attribute name⟩\} Prints as attribute, an HTML/CSS attribute. \pdf\TeX, \X\TeX, or \Lua\TeX.\n
\HTML HTML: Hypertext Markup Language\n
\HTMLfive HTML5: Old-style figure if font supports\n
\CSS CSS: Cascading Style Sheet\n
\CSSthree CSS3: Old-style figure if font supports\n
\EPUB EPUB: E-book file format\n
### 3.14.5 Specific programs\n
\TikZ Ti\kZ: Package logo\n
\CTAN CTAN: Comprehensive \TeX\ Archive Network\n
\TDS TDS: \TeX\ Directory Structure\n
\MathML MathML: Mathematical Markup Language\n
\MathJax MathJAX: Math on the web.\n
### 3.14.6 Acronyms, brand names, trademarks\n
\brand \{⟨name⟩\} BRANDNAME, COMPANY NAME\n
\acro \{⟨acronym⟩\} ACRO: Acronym\n
\supregistered Superscript trademark symbol®\n
### 3.15 Logos\n
Several additional logos are provided.\n
Also see the metalogo and metalogox packages.
\LaTeX \quad \LaTeXe

\LaTeXe \quad \LaTeXe, with reversed E if graphics is loaded.

\LaTeXe \quad \LaTeXe, with reversed E if graphics is loaded.

\AMS \quad \AMS\n
\LyX \quad \LyX

\BibTeX \quad \BibTeX

\MakeIndex \quad \MakeIndex

\ConTeXt \quad \ConTeXt

\MiKTeX \quad \MiKTeX

\Dash \quad \Dash

\Dash \quad \Dash

\Slash \quad \Slash

\\thin\skip \quad \text{A breakable thin skip.}

\endash \quad \text{An endash: –}

\endash \quad \text{An emdash: —}

\\thin\space \quad \text{A thin space which allows a line break.}

\thin\thin\space \quad \text{A very thin space which allows a line break.}

\\Dash \quad \text{A unbreakable thin space, emdash, and breakable thin space: A—B}

\\dash \quad \text{An unbreakable thin space, endash, and breakable thin space: A–B}

\\Slash \quad \text{An unbreakable very thin space, a slash, and a breakable very thin space:}

\begin{tabular}{|c|c|c|}
\hline
\text{Command} & \text{Result} \\
\hline
A--B & A–B \quad \text{(not breakable)} \\
A \dash B & A – B \quad \text{(only breakable before the B)} \\
A -- B & A – B \quad \text{(breakable before or after the dash)} \\
\hline
A---B & A—B \quad \text{(not breakable)} \\
A \Dash B & A—B \quad \text{(only breakable before the B)} \\
A --- B & A — B \quad \text{(breakable before or after the dash)} \\
\hline
A/B & A/B \quad \text{(not breakable)} \\
A \Slash B & A/B \quad \text{(only breakable before the B)} \\
A / B & A / B \quad \text{(breakable before or after the slash)} \\
A~/~B & A / B \quad \text{(not breakable)} \\
\hline
\end{tabular}
4 Examples

Example 1: Macros

Code:

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

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

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

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

Result:

\mymacro
\DeclareFloatingPhoto[photograph]
\photocaptionsetup[photograph]
\cphotonameref[photograph]

The optional category is used to label and group tags and index entries. See this document’s index entries for examples of this “photograph” category and the \dtxexample category of macros.

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:

myenvironment (env) \{⟨argument⟩\} Short description.

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 receive 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:

otherenvironment (env) [⟨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 category kindofenvironment.
Example 4: Booleans and Counters

Code:
\DescribeBoolean[examples][sampleboolean] Some description.
\DescribeCounter[examples][samplecounter] Some description.

Result:
sampleboolean (bool) [examples] Some description.
samplecounter (Ctr) [examples] Some description.

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

Example 5: Lengths

Code:
\DescribeLength[photograph][\photowidth] Some description.

Result:
\photowidth (Len) [photograph] Some description.

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

Example 6: Hooks and Sockets

Code:
\DescribeHook{\hookname} A hook with a backslash.
\DescribeHook{para/begin} A hook without a backslash.
\DescribeSocket{socketname} A socket.
\DescribePlug{plugname} A socket plug.

Result:
\hookname (Hook) A hook with a backslash.
para/begin (Hook) A hook without a backslash.
socketname (Socket) A socket.
plugname (Plug) A socket plug.

Hooks may or may not have a leading backslash.
Example 7: Packages, Classes, and Options

Code:

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

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

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

Result:

samplepackage (\textit{Pkg}) [examples] About a \LaTeX\ package.

sample_class (\textit{Cls}) [examples] About a \LaTeX\ class.

sampleoption (\textit{Opt}) [examples] About an option for a package or class.

Example 8: 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:


OS\_command (\textit{Cmd}) An operating-system command.

program\_name (\textit{Prog}) An operating-system program.

Filenames, program names, and command names may have underscores, such as tested here. A category is used to group “bigfiles” together in the index.
Example 9: 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]{sampltwokey} About another key of \otherkeys.

\DescribeKey{lonekey} A key without a category.

Result:

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>firstkey (Key)</td>
<td>About the first key of the groupofkeys set.</td>
</tr>
<tr>
<td>secondkey (Key)</td>
<td>About the second key of groupofkeys.</td>
</tr>
<tr>
<td>samplekey (Key)</td>
<td>About some key of otherkeys.</td>
</tr>
<tr>
<td>sampltwokey (Key)</td>
<td>About another key of otherkeys.</td>
</tr>
<tr>
<td>lonekey (Key)</td>
<td>A key without a category.</td>
</tr>
</tbody>
</table>

See the index key groups.
Example 10: Arguments

Code:

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

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

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

Result:

- \texttt{[H] (Arg) [figure]} What happens when a figure is \texttt{H}ere.
- \texttt{[M] (Arg) [figure]} What happens when a figure is in the \texttt{M}argin.
- \texttt{bold (Arg) [mymacro]} What happens when \texttt{\texttt{mymacro}} is given the \texttt{bold} argument.

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

⚠️ \texttt{macro names} Note you may need to use \texttt{\texttt{mymacro}} for the macro's name.
Example 11: Object

Code:

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

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

\DescribeObject{randomobject} About some random object.

Result:

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

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

Example 12: Other

Code:

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

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

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

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

Result:

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

Describes an arbitrary non-programming object, using roman text.
Example 13: 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 command.
\end{description}

Result:

\macroname [descexamples] \macroname: Describe the macro.
booleanname (bool) \[desexamples] \textbf{booleanname}: Describe the boolean.
\lengthname (Len) \[desexamples] \textbf{\lengthname}: Describe the length.
keyname (Key) \[desexamples] \textbf{keyname}: Describe the key.
package_name (Pkg) \[desexamples] \textbf{package_name}: Describe the package.
class_name (Cls) \[desexamples] \textbf{class_name}: Describe the class.
file_name (file) \[desexamples] \textbf{file_name}: Describe the file.
program_name (Prog) \[desexamples] \textbf{program_name}: Describe the program.
command_name (Cmd) \[desexamples] \textbf{command_name}: Describe the command.

Uses a description environment to describe objects.
Example 14: Nesting

Code:

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

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

Result:

```
environmentA
\macroB
  environmentC
     The contents.
sidebyside
  minipage
    Left contents.
  \hfill
  \hfill
  minipage
    Right contents.
```

Note the use of the optional arguments to select less than full \linewidth, and the starred form for the second box to place it inline with the \hfill text.
Example 15: dtxexample

Code:

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

Result:

See fig. 1

Example 15, typeset above, was created with the following code:

\begin{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 dtxexamplefloat was created, with the caption “dtxexample” and the label ex:dtxexample, which points to example 15.
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 16: fsourceverb

**Code:**

```latex
% \begin{fsourceverb}[label=An fsourceverb example]
% \newcommand{fdosomething}[1][whattodo]{
%   \hspace{0pt}\textup{doing }#1
% }
% \end{fsourceverb}
```

**Result:**

```
An fsourcerverb example
\newcommand{fdosomething}[1][whattodo]{
  \hspace{0pt}\textup{doing }#1
}
```

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

Example 17: sourcedisplay

**Code:**

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

**Result:**

```
\newcommand{dosomething}[1][whattodo]{
  \hspace{0pt}\textup{doing }#1
}
```
Example 18: 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 19: 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.

\begin{quote}
\textit{A title}
\end{quote}

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.\footnote{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.} It may be useful to use \DeclareBoolean and friends both at the code location and also in the User’s Manual section.

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

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

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

\begin{verbatim}
\vspace*{-\baselineskip}
\end{verbatim}

\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.
6 Code

6.1 Required packages

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

\begin{verbatim}
\AtBeginDocument{
  \IfPackageLoadedTF{makeidx}{{
    \IfPackageLoadedTF{splitidx}{{
      \RequirePackage{makeidx}
      \makeindex
    }}
  }}
}\end{verbatim}

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

\begin{verbatim}
\RequirePackage{etoolbox}[2011/01/03]
\end{verbatim}

xparse (Pkg) Used for the examples.

\begin{verbatim}
\RequirePackage{xparse}
\end{verbatim}

calc (Pkg) Used for \shownesting.

\begin{verbatim}
\RequirePackage{calc}
\end{verbatim}

xcolor (Pkg) Used for the examples.

\begin{verbatim}
\RequirePackage{xcolor}
\definecolor{myurlcolor}{rgb}{0,0,.7}
\definecolor{mylinkcolor}{rgb}{.7,0,.7}
\end{verbatim}

caption (Pkg) Used for the examples.

\begin{verbatim}
\RequirePackage{caption}
\end{verbatim}

newfloat (Pkg) Used for the examples.

\begin{verbatim}
\RequirePackage{newfloat}
\end{verbatim}

fancyvrb (Pkg) Used for the examples.

\begin{verbatim}
\RequirePackage{fancyvrb}
\end{verbatim}

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

\begin{verbatim}
\RequirePackage{xstring}
\end{verbatim}

hyperref (Pkg) If hyperref is loaded, disable some macros in PDF bookmarks:

pdf bookmarks
\AtBeginDocument{
  \IfPackageLoadedTF{hyperref}{
    \pdfstringdefDisableCommands{
      \def\quad{}\%
      \def\\{}\%
      \def\pkg#1{#1}\%
      \def\ctr#1{#1}\%
      \def\bool#1{#1}\%
      \def\optn#1{#1}\%
      \def\env#1{#1}\%
      \def\cs#1{\textbackslash#1}\%
      \def\{}\%
      \def\LuaLaTeX{LuaLaTeX}\%
      \def\XeLaTeX{XeLaTeX}\%
      \def\TeX{TeX}\%
      \def\LaTeX{LaTeX}\%
      \def\LaTeXe{LaTeX2e}\%
      \def\LuaTeX{LuaTeX}\%
      \def\XeTeX{XeTeX}\%
      \def\AmS{AMS}\%
      \def\Dash{---}\%
      \def\dash{--}\%
      \def\Slash{/}\%
      \def\prog#1{\detokenize{#1}}\%
      \def\progcode#1{#1}\%
      \def\filenm#1{\detokenize{#1}}\%
      \def\brand#1{#1}\%
      \def\acro#1{#1}\%
      \def\ODT{ODT}\%
      \def\SVG{SVG}\%
      \def\PNG{PNG}\%
      \def\GIF{GIF}\%
      \def\JPG{JPG}\%
      \def\EPS{EPS}\%
      \def\PDF{PDF}\%
      \def\DVI{DVI}\%
      \def\UTF{UTF}\%
      \def\URL{URL}\%
      \def\element#1{#1}\%
      \def\attribute#1{#1}\%
      \def\attrib#1{#1}\%
      \def\HTML{HTML}\%
      \def\HTMLfive{HTML5}\%
      \def\CSS{CSS}\%
      \def\CSSthree{CSS3}\%
      \def\EPUB{EPUB}\%
      \def\TOC{TOC}\%
      \def\LOF{LOF}\%
      \def\LOT{LOT}\%
    }
  }}% yes hyperref}
If hyperref is not loaded, emulate \hyperpage here.

\begin{verbatim}
70  (% no hyperref
71 \newcommand*{\hyperpage}[1]{\textcolor{red}{#1}}
72  )
73 \end{verbatim}

\setlength{\unitlength}{1pt}

\input{pict2e}

\section{Warning sign}
\warningsign

\texttt{\warningsign} Prints an exclamation point inside a triangle. Displays as: \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.

\begin{verbatim}
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 )
\end{verbatim}

\section{Special character handling}

The literal backslash character:

\begin{verbatim}
84 \begingroup
85 \catcode\|=0
86 \catcode\\|=12
87 \gdef\D TX D[@backslash]{\}
88 \endgroup
\end{verbatim}

\section{Patching hypdoc, splitidx, doc}

If hyperref is disabled (by \l warp) then define the missing \hdclindex.

\begin{verbatim}
89 \IfPackageLoadedTF{doc}{
90 \IfPackageLoadedTF{hypdoc}{
91 \AddToHook{begindocument/before}[doc/hyperref]{%
92 \ifdoc@hyperref
93 \else
94 \def\hdclindex{!}
95 \fi
96 }%
97 }%
98 \end{verbatim}
Several changes for when \texttt{hypdoc} and \texttt{splitidx} are used together:

\begin{verbatim}
\AtBeginDocument{
  \IfPackageLoadedTF{doc}{
    \IfPackageLoadedTF{hypdoc}{
      \IfPackageLoadedTF{splitidx}{
        \renewcommand*{\@wrsindex}{%#2}{%#2}
        \ifx\relax#1\relax%
          \if@splitidx%
            \@wrsindex[\texttt{idx}]{#2}%
          \else%
            \def\@tempa{#2}%
            \if@verbindex\@onelevel@sanitize\@tempa\fi%
            \@wrindex{\@tempa}%
          \fi%
        \else%
          \def\@tempa{#2\encapchar hdpindex{}}%
          \csname index@#1@hook\endcsname%
          \expandafter\ifx\csname @@wrsindex\endcsname\relax%
            \@@@wrsindex{#1}{\@tempa}{\thepage}%
          \else%
            \def\@tempb{\@@wrsindex{#1}}%
            \expandafter\@tempb\@tempa||%\fi%
        \endgroup%
        \@esphack%
      }% \texttt{splitidx}
    }% \texttt{hypdoc}
  }% \texttt{doc}
}{% \texttt{hypdoc}
}{% \texttt{splitidx}
}% \texttt{splitidx}
\renewcommand{\@wrsindex}[2]{%#2}{%#2}
\ifx\relax#1\relax%
  \if@splitidx%
    \@wrsindex[\texttt{idx}]{#2}%
  \else%
    \def\@tempa{#2}%
    \if@verbindex\@onelevel@sanitize\@tempa\fi%
    \@wrindex{\@tempa}%
  \fi%
  \else%
    \def\@tempa{#2\encapchar hdpindex{}}%
    \csname index@#1@hook\endcsname%
    \expandafter\ifx\csname @@wrsindex\endcsname\relax%
      \@@@wrsindex{#1}{\@tempa}{\thepage}%
    \else%
      \def\@tempb{\@@wrsindex{#1}}%
      \expandafter\@tempb\@tempa||%\fi%
  \endgroup%
  \@esphack%
}{% \texttt{hypdoc}

\end{verbatim}

\texttt{hypdoc} guesses the \texttt{toc} level for the PDF bookmarks, but its algorithm seems to fail at \texttt{\StopEventually} for split indices.

“Paragraph ended before \texttt{\HD@@guesstoclevel} was complete.”

Its guess is fixed to level 1 until \texttt{\PrintChanges} or \texttt{\printindex}.

\begin{verbatim}
\def\HD@guesstoclevel#1{1}
\end{verbatim}

Patch \texttt{doc}'s \texttt{\PrintChanges} to reset the \texttt{toc} guess to top level.
hypdoc adds PDF bookmarks to letter groups in the index. hypdoc must take into account multiple indexes, otherwise, followup indices will have duplicate bookmarks.

Increment the index number at the start of each index, and append the index number to the PDF bookmark.

Also, change the TOC guess to top level from now on, presuming that the indices are at the end.

\newcounter{DTXD@indexnumber}
\extendtheindex
{\addtocounter{DTXD@indexnumber}{1}\def\HD@guesstoclevel#1{0}}
{\def\HD@guesstoclevel#1{0}}
{\def\HD@@@bfseries\hfil#1\hfil{\ifx\#1\%\else\raisebox{\baselineskip}[0pt]{\kern-\HD@margin\relax%\pdfbookmark[\HD@toclevel@subindex]{$#1$}{HD.#1}}\pdfbookmark[\HD@toclevel@subindex]{$#1$}{dtxdescribe\arabic{DTXD@indexnumber}}\kern\HD@margin\relax%}}
{\def\HD@@@bfseries\hfil#1\hfil{\ifx\#1\%\else\raisebox{\baselineskip}[0pt]{\kern-\HD@margin\relax%\pdfbookmark[\HD@toclevel@subindex]{$#1$}{dtxdescribe\arabic{DTXD@indexnumber}}\kern\HD@margin\relax%}}}
{\def\HD@@@bfseries\hfil#1\hfil{\ifx\#1\%\else\raisebox{\baselineskip}[0pt]{\kern-\HD@margin\relax%\pdfbookmark[\HD@toclevel@subindex]{$#1$}{dtxdescribe\arabic{DTXD@indexnumber}}\kern\HD@margin\relax%}}}
{\def\HD@@@bfseries\hfil#1\hfil{\ifx\#1\%\else\raisebox{\baselineskip}[0pt]{\kern-\HD@margin\relax%\pdfbookmark[\HD@toclevel@subindex]{$#1$}{dtxdescribe\arabic{DTXD@indexnumber}}\kern\HD@margin\relax%}}}

6.5 Gobbling comment characters

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

\setlength{\marginparsep}{1em}
\setlength{\marginparpush}{.7ex}
\setlength{\parindent}{0em}
\setlength{\parskip}{2ex}

\IndexMin (Len) From \texttt{ltxdoc}.

\ifdef{\IndexMin}
\setlength{\IndexMin}{40ex}
\newlength{\IndexMin}

6.7 Not \texttt{ltxdoc}: \texttt{ltxdoc} emulation

If the \texttt{ltxdoc} class is not used, some of its macros are replicated here.

\@ifclassloaded{\texttt{ltxdoc}}{}
\def\cmd#1{\cs{\expandafter\cmd@to@cs\string#1}}
\def\cmd@to@cs#1#2{\char\number'#2\relax}
\DeclareRobustCommand\cs[1]{\texttt{\char'\#1}}
\providecommand\marg[1]{\texttt{\char'\{}\meta{#1}\texttt{\char'\}}}
\providecommand\oarg[1]{\texttt{[}\meta{#1}\texttt{]}}
\providecommand\parg[1]{\texttt{(}\meta{#1}\texttt{)}}
\providecommand\url{	exttt}

6.8 Not \texttt{doc}: \texttt{doc} emulation

If the \texttt{doc} class is not used, some of its macros are replicated here.

\AtBeginDocument{
\@ifpackageloadedTF{doc}{}{% not doc
\newenvironment*{macro}[1]{%
\PackageError{dtxdescribe}{The 'macro' environment is only available when using the doc package\MessageBreak
with a .dtx source file}{This environment only makes sense for .dtx source.}
}%
\newenvironment*{environment}[1]{%
\PackageError{dtxdescribe}{The 'environment' environment is only available when using the doc package\MessageBreak
with a .dtx source file}{This environment only makes sense for .dtx source.}
}%
\}
6.9 Support macros

\PrintEnvName \{\textit{name}\}\quad Prints an environment name.

\providecommand*{\PrintEnvName}{\par}
\renewcommand*{\PrintEnvName}{\strut\scriptsize{Env}\quad\MacroFont\textit{name}}

\DTXD@printtype \{\textit{text}\}\quad Used to print the object category in the margin:

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

\usage \{\textit{text}\}\quad Allow hyperlinks in the “usage” index entries:

\IfPackageLoadedTF{doc}{% not doc package

\DTXorigindex Used to bypass hyperref index modifications.
\let\DTXorigwrindex@wrindex

\DTXmargintag \textit{⟨category⟩} \{⟨name⟩\} \{⟨margin tag⟩\}

Creates the margin tag for the object being described.
The category is used to sub-categorize keys into their key/value groups.

\DTXindex \textit{⟨category⟩} \{⟨name⟩\} \{⟨margin tag⟩\} \{⟨index tag⟩\} \{⟨main/usage⟩\}

Creates the index entries for the object being described, where name has no backslash or underscore.
The category is used to sub-categorize keys into their key/value groups. main prints code lines in the index, and usage prints page numbers.

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.

\DTXD@origwrindex(%

Index by name:
Write the name, the formatted name, the index tag, and the category:

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

Index by tag and category:
Write the tag and category as a group, under which is the name and the formatted name.

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

Possibly index by category and name:

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

\DTXD@margintagindex {{\category}} {{\name}} {{\margintag}} {{\indextag}} {{\main\usage}}

Creates the margin tag and the index entries. The category is used to sub-categorize keys into their key/value groups.
The margin tag and the name:
\DTXD@margintag{#1}{#2}{#3}%

The index entries:
\DTXD@index{#1}{#2}{#3}{#4}{#5}%

\DTXD@macroname \langle control sequence \rangle

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'\|='\textbackslash
\lowercase{\endgroup\def\removebs#1{\if#1|\else#1\fi}}
\newcommand*{\DTXD@macroname}[1]{\expandafter\removebs\string#1}

\DTXD@verbatimcmd \langle \name \rangle

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 \langle category \rangle \langle name \rangle \langle margin tag \rangle \langle index tag \rangle \langle main/usage \rangle

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

\DTXD@cmdmargintagindex \langle category \rangle \langle name \rangle \langle margin tag \rangle \langle index tag \rangle \langle main/usage \rangle

Create a margin tag with the name of the macro:
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 category if given.

Write \[\text{[category]}>:\text{name}=\text{csname} \text{[indextag]}\] usage

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

Write \[\text{indextag}>:\text{[category]}>:\text{csname}\] usage

6.10 doc: Key handling for object categories

If using \texttt{doc}, the optional key/value argument may also include an object category. This is supported by assigning any unknown key to be the category.

The category may be given using the key \texttt{c}, or as the default action when an unknown key is given.
6.11 \textit{doc}: Handling \texttt{marginpar} inside a float

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

\begin{verbatim}
\IfPackageLoadedTF{doc}{
    \def\@doc@describe#1#2{%
        \ifdoc@noprint\else%
            \ifundef{@captype}{% not float?
                \marginpar{\raggedleft
                    \strut\doc@providetarget% \@nameuse{PrintDescribe#1}{#2}\
                    \ifdefvoid{\DTXD@category}{}{\space{\footnotesize[\mbox{\DTXD@category}]}}%
                \}
            }%
            \fi%
        \ifdoc@noindex\else%
            \@nameuse{Special#1Index}{#2}%
        \fi%
    }{
    }\fi%
\endgroup%
}\ignorespaces%
\end{verbatim}

6.12 \textit{doc}: Handling categories and detokenizing names

File names and such may include underscores or other characters, so patch \texttt{doc} to neutralized while processing.

\begin{verbatim}
\IfPackageLoadedTF{doc}{%
    \def\@NewDocElement#1#2#3{%
        \doc@macrolikefalse%
        \doc@topleveltrue%
        \def\doc@idxtype{#3}%
        \def\doc@idxgroup{#3s}%
        \let\doc@printtype\@empty%
        \csname keys_set:nn\endcsname{doc}{#1}%
        \ifx\doc@printtype\@empty%
            \@temptokena{}%
        \else%
            \texttt{\doc@printtype}@empty%
        \fi%
        \@esphack%
    }%
\endgroup%
\end{verbatim}
\DTXDbreak Inserts a possible line break point. Used in the index to allow line breaks before verbatim category names.

\newcommand*{\DTXDbreak}{\space\penalty200}

\DTXD@printobjectname \{\langle name w/o backslash\rangle\}\{\langle name\rangle\} Adds the object name to the index in verbatim. These are passed as arguments instead of directly used here because they must have their value when the index is written instead of when \DTXD@printobjectname is used when the index is read back.

\newcommand*{\DTXD@printobjectname}{\langle name w/o backslash\rangle\{\langle name\rangle\}}
If there is a category, it is added verbatim.

\DTXD@maybecategory If there is a category, it is added verbatim.

\DTXD@categorybreak inserts a possible line break here, allowing a break if the following verbatim is too long.

\DTXD@categorylevelname The simplified name, without backslash.

\DTXD@categorylevelname \langle object_type \rangle If a category is assigned, index by category.

\arg 1: Type of object, shown between parens \( (), \) such as macro, boolean, etc.

\DTXD@category: The name of the category, printed in brackets \( [ ] \), such as \macroname.

\DTXD@categorylevelname: The simplified name of the category, such as \macroname.

@\@gtempa: The name of this particular object.
dtxdescribe

Index:
categorylevelname=\verb!*+[category]-->name=\verb!*+[prefix]name+<break>(type)

\DTXD@findcategorylevelname Given \DTXD@category, create \DTXD@categorylevelname, a detokenized name without backslash.

\DTXD@findgtempa Detokenize \@gtempa and also find another version without any leading backslash.

\doccreatespecialindexes \{1: name\} \{2: index type\} \{3: index group\}
\def\doc@createspecialindexes#1#2#3{% 
  \@temptokena{\space (#2)}% 
  \@temptokenb{#3:}% 
}

Define \SpecialMain<name>Index \{I: name\}
\@nameedef{SpecialMain#1Index}##1{%

Find \@gtempa, the name of the object.
\@bsphack
\noexpand\DTXD@findcategorylevelname%
\ifdoc@toplevel%
\noexpand\special@index{##1\noexpand\actualchar}%
\fi%

Index:
name=\verb!*+name+ (type) [category]|main
\noexpand\special@index(%
\noexpand\@bsphack%
\noexpand\DTXD@printobjectname%
\noexpand\DTXD@mobackslash%
\noexpand\@gtempa%
\if\@nil#2\@nil\else\the\@temptokena\fi%
\noexpand\DTXD@maybecategory% 
\noexpand\@bsphack%
\noexpand\encapchar main%
)%

Maybe add the category index entry:
\noexpand\DTXD@maybecategorylevel(#2)%
\fi%
\fi%

If group is not empty, index:
group:>name=\verb!*+name+ [category]|main
\if\@nil#3\@nil\else%
Maybe add the category index entry:

\DTXD@maybecategorylevel{#2}%
\fi%
\noexpand\@esphack}

\Define\Special<name>Index \{(l: name)\}
\@nameedef{Special#1Index}{%  
\noexpand\DTXD@findgtempa{##1}%
\@temptokena: Index (type).
\@temptokenb: Index group.
\@gtempa: Detokenized name of the particular object.
\DTXD@gtempa@nobackslash: Detokenized name without backslash.

If is a top level object, index:
name=\verb!*+name+ (type) [category]usage

\ifdoc@toplevel%
\noexpand\doc@providetarget%
\noexpand\index{##1\actualchar{\string\ttfamily\space##1}}%
\noexpand\index{%
\noexpand\DTXD@printobjectname%
\noexpand\DTXD@gtempa@nobackslash%
\noexpand\gtempa%
\noexpand\DTXD@maybecategory%
\noexpand\doc@handleencap{usage}%
}%

Maybe add the category index entry:

\DTXD@maybecategorylevel{#2}%
If group is not empty, index:

\verb|\verb|!+name+ [category]|usage

Maybe add the category index entry:

Define \Code<name>Index {⟨1: main or usage⟩} {⟨2: name⟩}
If a top level object, index:
name=\verb!*++\name+ (type) [category]|<main or usage>

```latex
\ifdoc@toplevel
  \noexpand\special@index{
    \noexpand\DTXD@printobjectname{\noexpand\DTXD@gtempa
      \noexpand\@gtempa}{\ifx\@nil#2\@nil\else \the\@temptokena \fi}
    \noexpand\DTXD@maybecategory{\noexpand\encapchar ##1}
  }

  Maybe add the category index entry:

  \noexpand\DTXD@maybecategorylevel{#2}

  \fi
\fi
```

If group is not empty, index:
group>:name=\verb!*++\name+ [category]|<main or usage>

```latex
\ifx\@nil#3\@nil\else
  \noexpand\special@index{
    \the\@temptokenb\noexpand\levelchar
      \noexpand\DTXD@printobjectname{\noexpand\DTXD@gtempa
        \noexpand\@gtempa}{\noexpand\DTXD@maybecategory{\noexpand\encapchar ##1}}
  }

  Maybe add the category index entry:

  \noexpand\DTXD@maybecategorylevel{#2}

  \fi
\fi
```

Define \SpecialMain<name>Index {{I: name}}

```latex
@nameedef{SpecialMain#1Index}##1{\expandafter\noexpand\csname Code#1Index\endcsname{main}{##1}}
```

Define \Special<name>Index {{I: name}}

```latex
@nameedef{Special#1Index}##1{%\noexpand\DTXD@findgtempa{##1}%=main\@nameedef{\@temptokena}{\@temptokenb}{Index type.\@temptokenb}{Index group.}}
```
\@gtempa: Detokenized name of the particular object.
\DTXD@gtempa@nobackslash: Detokenized name without backslash.

If a top level object, index:
name=\verb!*+[name+] (type) [category]|usage

Maybe add the category index entry:

If group is not empty, index:
group:>name=\verb!*+[name+] [category]|usage

Maybe add the category index entry:

6.13 Not doc: `\DescribeMacro` and `\DescribeEnvironment`

\DescribeMacro [⟨category⟩] {⟨\name⟩}

Redefined to allow hyperlinked index entries and an optional category:

\IfPackageLoadedTF{doc}{}{% not doc
\providecommand*{\DescribeMacro}{}
\renewcommand*{\DescribeMacro}[2][{}]{%
\@bsphack%
Create the margin tag with the macro's name:
\@ifundefined{@captype}{% not float?
\leavevmode%
\marginpar{%
\hbadness=10000%
\hfuzz=5em%
\raggedleft%
\ifblank{#1}{}{{\scriptsize\textsf{[#1]}} }% category
\cmd{#2}% name
\}%marginpar
\)% not float?
\begingroup%
Write the index sorted by the name without the backslash, followed by the actual
name with the backslash. Append the category if given.
Write name=csname>|category| usage
\begingroup%
\DTXD@origwrindex{%
\DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2}% name
\ifblank{#1}{}{\levelchar{[#1]}% category
\encapchar usage%
\}%
\begingroup%
Only if a category was given:
\ifblank{#1}%
\{% no category
\{% category given
\% Again, and prepend the category:
\%
\% Write category=[category]:>name=csname\verb+|usage+
\begin{macrocode}
\% \begingroup%
\begingroup%
\DTXD@origwrindex{%
#\actualchar\DTXD@verbatimcmd{#2}%
\begingroup%
\begingroup%
\DTXD@origwrindex{%
#\actualchar\DTXD@verbatimcmd{#2}%
642 \encapchar usage}%
643 )% category given
644 \@esphack%
645 \ignorespaces%
646 )
647
648 )% not doc

\DescribeEnv \[ ⟨category⟩ \] \{ ⟨environment name⟩ \}

Redefined to allow hyperlinked index entries:

649 \IfPackageLoadedTF{doc}{}{% not doc
650 \providecommand*{\DescribeEnv}{}
651 \renewcommand*{\DescribeEnv}[2]{\DTXD@margintagindex{#1}{#2}{Env}{environment}{usage}}
652 \renewcommand*{\DescribeEnv}{% not doc

6.14 New \Describe... macros

\DTXD@filemarginparindex \{ ⟨category⟩ \} \{ ⟨name⟩ \} \{ ⟨margin tag⟩ \} \{ ⟨index tag⟩ \} \{ ⟨main/usage⟩ \}

The name may have underscores.

656 \IfPackageLoadedTF{doc}{}{% not doc
657 \newcommand*{\DTXD@filename}{}
658 \newcommand*{\DTXD@filemarginparindex}[5]{% not doc

Create a detokenized version of the filename...

660 \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:

661 \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.

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

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

663 \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:

```latex
\endgroup
\ignorespaces
```

\DescribeMacro

Redefine with new definitions.

\DescribeEnvironment

Redefine with new definitions.

\IfPackageLoadedTF{doc}{{ doc
\RenewDocElement[macrolike = true ,
  idxtype = ,
  idxgroup = ,
  printtype = ]{Macro}{macro}
\RenewDocElement[macrolike = false ,
  idxtype = env. ,
  idxgroup = environments ,
  printtype = \textit{env.} ]{Env}{environment}
\}}{% not doc
\IfPackageLoadedTF{doc}{{ doc
\NewDocElement[ macrolike=false,
  toplevel=true,
  idxtype=file,
  idxgroup=Files,
  printtype=\textit{file} ]{File}{file}
\}}{% not doc
\newcommand*{\DTXD@DescribeFile}[2][]{% 
\DTXD@filemarginparindex{#1}{#2}{File}{file}{usage} 
\}
\newcommand*{\DescribeFile}{% 
\begingroup\catcode`\_=12 \DTXD@DescribeFile 
}
\DescribeProgram {⟨name⟩}

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

\IfPackageLoadedTF{doc}{% doc
\NewDocElement[
  macrolike=false,
  toplevel=true,
  idxtype=program,
  idxgroup=Programs,
  printtype=\textit{Prog}
]{Program}{program}{% not doc
\newcommand*{\DTXD@DescribeProgram}[2][2]{%
  \DTXD@filemarginparindex[#1]{#2}{Prog}{program}{usage}%
}\newcommand*{\DescribeProgram}{%
\begingroup\catcode'\_=12 \DTXD@DescribeProgram%
}{% not doc
\newcommand*{\DTXD@DescribeProgram}[2][2]{%
  \DTXD@filemarginparindex[#1]{#2}{Prog}{program}{usage}%
}\newcommand*{\DescribeProgram}{%
\begingroup\catcode'\_=12 \DTXD@DescribeProgram%
}{% not doc
\newcommand*{\DTXD@DescribeCommand}[2][2]{%
  \DTXD@filemarginparindex[#1]{#2}{Cmd}{command}{usage}%
}\newcommand*{\DescribeCommand}{%
\begingroup\catcode'\_=12 \DTXD@DescribeCommand%
}{% not doc
\newcommand*{\DTXD@DescribeCommand}[2][2]{%
  \DTXD@filemarginparindex[#1]{#2}{Cmd}{command}{usage}%
}\newcommand*{\DescribeCommand}{%
\begingroup\catcode'\_=12 \DTXD@DescribeCommand%
}
\DescribePackage {⟨name⟩}

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

\IfPackageLoadedTF{doc}{% doc
\NewDocElement[
    macrolike=false,
    toplevel=true,
    idxtype=package,
    idxgroup=Packages,
    printtype=\textit{Pkg}
]{Package}{package}
}{% not doc
\newcommand*{\DTXD@DescribePackage}[2][]{%
    \DTXD@filemarginparindex(#1)(#2)(Pkg)(package)(usage)%
}\newcommand*{\DescribePackage}{%
    \begingroup\catcode\_=12 \DTXD@DescribePackage%
}\endgroup
}

\DescribeClass {⟨name⟩}

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

\IfPackageLoadedTF{doc}{% doc
\NewDocElement[
    macrolike=false,
    toplevel=true,
    idxtype=class,
    idxgroup=Classes,
    printtype=\textit{Cls}
]{Class}{class}
}{% not doc
\newcommand*{\DTXD@DescribeClass}[2][]{%
    \DTXD@filemarginparindex(#1)(#2)(Cls)(class)(usage)%
}\newcommand*{\DescribeClass}{%
    \begingroup\catcode\_=12 \DTXD@DescribeClass%
}\endgroup
}
The category may be used to categorize arguments by their macro or environment name.
\newdocelement[
  macrolike=false,
  toplevel=true,
  idxtype=boolean,
  idxgroup=Booleans,
  printtype=\textit{bool}
]{Boolean}{boolenv}
\}% not doc
\newcommand*{\DescribeBoolean}[2][{}]
  {\DTXD@margintagindex{#1}{#2}{Bool}{boolean}{usage}}
\}% not doc
\describe{\langle category \rangle}{\langle name \rangle}
% not doc
\ifpackageloadedTF{doc}{% doc
  \newdocelement[
    macrolike=true,
    toplevel=true,
    idxtype=length,
    idxgroup=Lengths,
    printtype=\textit{Len}
  ]{Length}{length}
% not doc
\newcommand*{\DescribeLength}[2][{}]
  {\DTXD@cmdmargintagindex{#1}{#2}{Len}{length}{usage}}
% not doc
\}% not doc
\describe{\langle category \rangle}{\langle name \rangle}
% not doc
\ifpackageloadedTF{doc}{% doc
  \newdocelement[
    macrolike=false,
    toplevel=true,
    idxtype=counter,
    idxgroup=Counters,
    printtype=\textit{Ctr}
  ]{Counter}{counter}
% not doc
\newcommand*{\DescribeCounter}[2][{}]
  {\DTXD@margintagindex{#1}{#2}{Ctr}{counter}{usage}}
% not doc
\}% not doc
\DescribeHook [{\langle category\rangle}] \{\langle name\rangle\}

\IfPackageLoadedTF{doc}{% doc
\NewDocElement[
macrolike=true,
toplevel=true,
dxtype=hook,
dxgroup=Hooks,
printtype=\textit{Hook}]
\{Hook\}(hook)
\}% not doc
\newcommand*{\DescribeHook}[2][]{\DTXD@margintagindex{#1}{#2}{Hook}{hook}{usage}}%
\DescribeSocket [{\langle category\rangle}] \{\langle name\rangle\}

\IfPackageLoadedTF{doc}{% doc
\NewDocElement[
macrolike=false,
toplevel=true,
dxtype=socket,
dxgroup=Sockets,
printtype=\textit{Socket}]
\{Socket\}(socket)
\}% not doc
\newcommand*{\DescribeSocket}[2][]{\DTXD@margintagindex{#1}{#2}{Socket}{socket}{usage}}%
\DescribePlug [{\langle category\rangle}] \{\langle name\rangle\}

\IfPackageLoadedTF{doc}{% doc
\NewDocElement[
macrolike=false,
toplevel=true,
dxtype=plug,
dxgroup=Plugs,
printtype=\textit{Plug}]
\{Plug\}(plug)
\}% not doc
\newcommand*{\DescribePlug}[2][]{\DTXD@margintagindex{#1}{#2}{Plug}{plug}{usage}}%
The category may be used to categorize keys by their key/value group.

May be used to describe an arbitrary piece of code. Creates a margin tag and index entries with `\ttfamily`.
\DescribeOther \[\langle\text{category}\rangle\] \{\langle\text{name}\rangle\}

May be used to describe an arbitrary non-programming object. Creates a margin tag and index entries with roman type.
6.15 \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 boolean default value.
\newcommand{\DescribeDefault}[1]{%
  \margintag{%
    \footnotesize%
    \textcolor{\DescribeDefaultcolor}{% Default: \texttt{#1}%
  }
}%
}

6.16 \ItemDescribeMacro, etc.

The following are for use inside a description.
\ItemDescribeMacro [⟨category⟩] {⟨name⟩}
\newcommand{\ItemDescribeMacro}[2][]{%
  \item{\cmd{#2}:}%
  \setlength{\parsep}{1.5ex}%
  \DescribeMacro[#1]{#2}%
}
\ItemDescribeEnv \[\langle category\rangle\] \{\langle name\rangle\}

\begin{verbatim}
1036 \newcommand{\ItemDescribeEnv}[2][% 
1037 \item[\texttt{#2}]:% 
1038 \setlength{\parskip}(1.5ex)% 
1039 \DescribeEnv[#1][#2]% 
1040 ]
\end{verbatim}

\ItemDescribeArgument \[\langle category\rangle\] \{\langle argument\rangle\}

\begin{verbatim}
1041 \newcommand{\ItemDescribeArgument}[2][% 
1042 \item[\texttt{#2}]:% 
1043 \setlength{\parskip}(1.5ex)% 
1044 \DescribeArgument[#1][#2]% 
1045 ]
\end{verbatim}

\ItemDescribeBoolean \[\langle category\rangle\] \{\langle name\rangle\}

\begin{verbatim}
1046 \newcommand{\ItemDescribeBoolean}[2][% 
1047 \item[\texttt{#2}]:% 
1048 \setlength{\parskip}(1.5ex)% 
1049 \DescribeBoolean[#1][#2]% 
1050 ]
\end{verbatim}

\ItemDescribeLength \[\langle category\rangle\] \{\langle name\rangle\}

\begin{verbatim}
1051 \newcommand{\ItemDescribeLength}[2][% 
1052 \item[\texttt{#2}]:% 
1053 \setlength{\parskip}(1.5ex)% 
1054 \DescribeLength[#1][#2]% 
1055 ]
\end{verbatim}

\ItemDescribeCounter \[\langle category\rangle\] \{\langle name\rangle\}

\begin{verbatim}
1056 \newcommand{\ItemDescribeCounter}[2][% 
1057 \item[\texttt{#2}]:% 
1058 \setlength{\parskip}(1.5ex)% 
1059 \DescribeCounter[#1][#2]% 
1060 ]
\end{verbatim}

\ItemDescribeHook \[\langle category\rangle\] \{\langle name\rangle\}

\begin{verbatim}
1061 \newcommand{\ItemDescribeHook}[2][% 
1062 \item[\texttt{#2}]:% 
1063 \setlength{\parskip}(1.5ex)% 
1064 \DescribeHook[#1][#2]% 
1065 ]
\end{verbatim}

\ItemDescribeSocket \[\langle category\rangle\] \{\langle name\rangle\}
\newcommand{\ItemDescribeSocket}[2][\{}%  
\item[#2]:\%  
\setlength{\parskip}{1.5ex}\%  
\DescribeSocket[#1]{#2}\%  
\}\%  

\ItemDescribePlug[\langle category\rangle]{\langle name\rangle}\%  
\newcommand{\ItemDescribePlug}[2][\{}%  
\item[#2]:\%  
\setlength{\parskip}{1.5ex}\%  
\DescribePlug[#1]{#2}\%  
\}\%  

\ItemDescribeKey[\langle category\rangle]{\langle name\rangle}\%  
\newcommand{\Item DescribeKey}[2][\{}%  
\item[#2]:\%  
\setlength{\parskip}{1.5ex}\%  
\DescribeKey[#1]{#2}\%  
\}\%  

\ItemDescribePackage[\langle category\rangle]{\langle name\rangle}\%  
\newcommand{\DTXD@ItemDescribePackage}[2][\{}%  
\item[#2]:\%  
\setlength{\parskip}{1.5ex}\%  
\DescribePackage[#1]{#2}\%  
\endgroup\%  
\}\%  

\ItemDescribeClass[\langle category\rangle]{\langle name\rangle}\%  
\newcommand{\DTXD@ItemDescribeClass}[2][\{}%  
\item[#2]:\%  
\setlength{\parskip}{1.5ex}\%  
\DescribeClass[#1]{#2}\%  
\endgroup\%  
\}\%  

\ItemDescribeOption[\langle category\rangle]{\langle name\rangle}\%  
\newcommand{\ItemDescribeOption}[2][\{}%  
\item[#2]:\%  
\setlength{\parskip}{1.5ex}\%  
\DescribeOption[#1]{#2}\%  
\}
\newcommand{\ItemDescribeOption}[2]{%\item{\texttt{#2}:}%\setlength{\parskip}{1.5ex}\DescribeOption[#1]{#2}%}

\ItemDescribeFile \[\langle category \rangle \] {\langle name \rangle}

\newcommand{\DTXD@ItemDescribeFile}[2]{%\item{\texttt{#2}:}%\setlength{\parskip}{1.5ex}\DescribeFile[#1]{#2}\endgroup%}
\newcommand{\ItemDescribeFile}{%\begingroup\catcode'\_=12 \DTXD@ItemDescribeFile%}

\ItemDescribeProgram \[\langle category \rangle \] {\langle name \rangle}

\newcommand{\DTXD@ItemDescribeProgram}[2]{%\item{\texttt{#2}:}%\setlength{\parskip}{1.5ex}\DescribeProgram[#1]{#2}\endgroup%}
\newcommand{\ItemDescribeProgram}{%\begingroup\catcode'\_=12 \DTXD@ItemDescribeProgram%}

\ItemDescribeCommand \[\langle category \rangle \] {\langle name \rangle}

\newcommand{\DTXD@ItemDescribeCommand}[2]{%\item{\texttt{#2}:}%\setlength{\parskip}{1.5ex}\DescribeCommand[#1]{#2}\endgroup%}
\newcommand{\ItemDescribeCommand}{%\begingroup\catcode'\_=12 \DTXD@ItemDescribeCommand%}

\ItemDescribeObject \[\langle category \rangle \] {\langle name \rangle}

\newcommand{\ItemDescribeObject}[2]{%\item{\texttt{#2}:}%\setlength{\parskip}{1.5ex}\DescribeObject[#1]{#2}%.}
\ItemDescribeOther \[\langle\text{category}\rangle\] \{\langle\text{name}\rangle\}

6.17 \margintag, \watchout

\margintagcolor The color of the \margintag.

\newcommand*{\margintagcolor}{blue!70!black}
\margintag \[\langle\text{text}\rangle\]

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 \[\langle\text{text}\rangle\]

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?
}

6.18 Nesting

Shows a box enclosing a label for the container, and the container’s contents. May be nested.
\shownesting \{\langle fraction of \linewidth \rangle \langle container \rangle \langle contents \rangle \}

\NewDocumentCommand{\shownesting}{s O{1} m m}{
\IfBooleanF{#1}{\par\smallskip}
\fbox{
\begin{minipage}{#2\linewidth-2em}
\hbadness=10000\relax%
#3\par\smallskip
\hspace{1em}
\begin{minipage}{\linewidth-1.5em}
#4
\end{minipage}
\end{minipage}}
}

6.19 The \texttt{dtxexample} environment

Also see example 15 on page 28.

dtxexample_cut.tex (file) Used to store the \texttt{input} example code.

\texttt{DTXD@examplerulecolor} [color] The color of the middle rule in the \texttt{dtxexample}.

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

\texttt{dtxexamplecodename} The text name of the code section.

\newcommand*{\dtxexamplecodename}{Code:}

\texttt{dtxexampleresultname} The text name of the result section.

\newcommand*{\dtxexampleresultname}{Result:}

dtxexample (env.) * \{\langle notes/cross-references \rangle \langle caption & label \rangle \}

Reads the code listing as a verbatim input using the \texttt{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}
Copy the environment’s contents to the file dtxexample_cut.tex:

\begin{dtxexample}
% start dtxexample

When the environment closes:

% end dtxexample

Finish the verbatim output:

\end{VerbatimOut}
\par
\addvspace{\bigskipamount}

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

\IfBooleanTF{#1}
\{
% minipage
\minipage{\linewidth}
\captionsetup{type=dtxexamplefloat}
\}
\{
% float
\begin{dtxexamplefloat}
\hrule\medskip
\caption{#3}
\textcolor{DTXD@examplerulecolor}{\smallskip\hrule}
\smallskip\scriptsize\itshape\dtxexamplecodename
\VerbatimInput{tabsize=4}{dtxexample_cut.tex}
\unskip
\textcolor{DTXD@examplerulecolor}{\hrule}
\smallskip\scriptsize\itshape\dtxexampleresultname
\}
\end{dtxexamplefloat}
\IfBooleanTF{#1}
\{
\endminipage
\}
\end{dtxexamplefloat}
\} % end dtxexample

Typeset the contents as verbatim:

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

Possible add the optional cross-references or notes:

\ifstrempty{#2}
\{}
\{}\itshape\small #2\}
\IfBooleanTF{#1}
\{\endminipage\}
\{\end{dtxexamplefloat}\}
\} % 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

A new float type for the examples.

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

Caption setup for the examples.

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

Name for \cref. \crefname here is required for documents not using the \doc class:

\AddToHook{begindocument/after}{%
  Before .aux file is loaded.
  \AtBeginDocument{
    \ifdef{\crefname}{%
      \crefname{dtxexamplefloat}{example}{examples}%
    %}
    %}
  }{}}%

6.20 noindmacro and noindenvironment

Similar to macro and environment, but not indexed.

\noindmacro\{env\} \{\langle name\rangle\}
For use in a sourcedisplay:

\fquad Forces a quad indent.

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

\fquadquad Forces a double-quad indent.

\fqqquad \newcommand*{\fqqquad}{\hspace*{2em}}

\fqqqqquad Forces a triple-quad indent.

\fqqqqquad \newcommand*{\fqqqqquad}{\hspace*{3em}}

sourceverb \textit{(env.\quad)} To typeset a block of source code, verbatim.

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

fsourceverb \textit{(env.\quad)} To typeset a framed block of source code, verbatim.

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

sourcedisplay \textit{(env.\quad)} To typeset a block of source code, allowing direct formatting.
\newenvironment{sourcedisplay}
{
\leavevmode
\par
\fqquad\minipage{\linewidth-4em}
\texttt
}{
\endminipage
\par}

\UIdisplay (env.) To typeset a user interface display.

\newenvironment{UIdisplay}
{
\leavevmode
\par
\fqquad\minipage{\linewidth-4em}
\textbf
}{
\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][1]{
&\par
&\fqquad
&\footnotesize \userentryname \quad \texttt{#1}
&\endminipage
&\par}

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

\newenvironment{docsidebar}[1][1][]
{\quote\unskip\medskip
&\setlength{\parskip}{1.5ex}
&\ifblank{#1}{}{\textit{#1}\newline}
&\rule[.5\bigskipamount]{\linewidth}{.4pt}
&\end{minipage}
&\par}

\docsidebar{⟨text⟩}
6.22 Formatted objects

Macros to format references to various kinds of objects.

6.22.1 \TeX\ objects

\pkg{⟨name⟩} Also useable for class names.
\cs{⟨csname⟩} From \texttt{ltxdoc}.
\env{⟨name⟩} From \texttt{ltxdoc}.
\marg{⟨argument⟩} From \texttt{ltxdoc}.
\oarg{⟨argument⟩} From \texttt{ltxdoc}.
\parg{⟨argument⟩} From \texttt{ltxdoc}.
\ctr{⟨name⟩}
\bool \{\langle name\rangle\}

\optn \{\langle name\rangle\}

\TOC

\LOF

\LOT

6.22.2 Programs and commands

\cmds \{\langle commands to print\rangle\} No processing is provided for special characters.

\progcode \{\langle code to print\rangle\} No processing is provided for special characters.

\prog \{\langle program name\rangle\} Underscores are allowed.

\filenm \{\langle file name\rangle\} Underscores are allowed.
General user-interface text.

6.22.3 File types

\ODT

\SVG

\PNG

\GIF

\JPG

\EPS

\PDF

\DVI
6.22.4 Internet

\UTF

\URL

\element \texttt{<name>}

\attribute \texttt{#1}

Each of these is "provided", and any prior meaning will be unchanged. In particular, Lua\TeX uses \attribute, so its meaning is unchanged if using Lua\TeX.

\HTML

\HTMLfive

\CSS

\CSSthree

\EPUB
6.22.5 Specific programs

\TikZ

\CTAN

\TDS

\MathML

\MathJax

6.22.6 Acronyms, brand names, trademarks

\brand \langle \mathit{name} \rangle

\acro \langle \mathit{acronym} \rangle

\supregistered Superscript trademark symbol.

6.23 Logos

\dvi\TeX \textsc{DVI\TeX}

\dvi\LaTeX \textsc{DVI\LaTeX}
6.24 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 thin space which allows a line break.

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

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

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

To compile the dtxdescribe package:

Enter ⇒ \texttt{pdflatex dtxdescribe.ins}

To compile the dtxdescribe documentation

Enter ⇒ \texttt{pdflatex dtxdescribe.dtx}

(Several times)

Enter ⇒ \texttt{makeindex -s gglo.ist -o dtxdescribe.gls dtxdescribe.glo}

Enter ⇒ \texttt{makeindex -s gind.ist dtxdescribe}

Enter ⇒ \texttt{pdflatex dtxdescribe.dtx}

(Several times)
A Change History

v0.10
General: 2016/12/08 Initial ver ... 1

v0.11
General: 2018/03/30 ... 1
\DTXD@cmdmargintagindex: Index
tag no longer plural. ... 43
\DTXD@index: Index tag no longer plural. ... 41
\watchout: Changed to \raggedleft. ... 67

v1.00
General: 2019/01/11 ... 1
Added formatted objects. ... 73
Added logos. ... 77
Cut file name changed to \dtxexample_cut.tex ... 68
\DescribeClass: Fix: Allow underscore. ... 57
\DescribeDefault: Added. ... 63
\DescribeDefaultcolor: Added. ... 63
\DescribeMacro: Sans tag font. ... 53
\DescribePackage: Fix: Allow underscore. ... 57
docsidebar: Added. ... 72
\DTXD@cmdmargintagindex: Sans tag font. ... 42
\DTXD@filemarginparindex: Fix: File category. ... 54
\DTXD@printtype: Sans tag font. ... 39
\dtxexamplecodename: Added. ... 68
\dtxexampleresultname: Added. ... 68
\fqquad: Added. ... 71
\fqquad: Added. ... 71
\fqquad: Added. ... 71
\fsourceverb: Added. ... 71
\ItemDescribeArgument: Added. ... 64
\ItemDescribeBoolean: Added. ... 64
\ItemDescribeClass: Added. ... 65
\ItemDescribeCommand: Added. ... 66
\ItemDescribeCounter: Added. ... 64
\ItemDescribeEnv: Added. ... 64
\ItemDescribeFile: Added. ... 66
\ItemDescribeKey: Added. ... 65
\ItemDescribeLength: Added. ... 64
\ItemDescribeMacro: Added. ... 63
\ItemDescribeObject: Added. ... 66
\ItemDescribeOption: Added. ... 65
\ItemDescribeOther: Added. ... 67
\ItemDescribePackage: Added. ... 65
\ItemDescribeProgram: Added. ... 66
\margintag: Uses \margintacolor. ... 67
\margintagcolor: Added. ... 67
noindenvironment: Added. ... 71

noindmacro: Added. ... 70
sourcedisplay: Added. ... 71
sourceverb: Added. ... 71
\UIdisplay: Added. ... 72
\userentry: Added. ... 72
\userentryname: Added. ... 72
\watchoutcolor: Added. ... 67

v1.01
General: 2019/03/22 ... 1
Sanitize PDF bookmarks. ... 33
\DescribeMacro: Put margin tag category in brackets. ... 53
\DescribeObject: Put margin tag category in brackets. ... 61
\DescribeOther: Put margin tag category in brackets. ... 62
\DTXD@cmdmargintagindex: Put margin tag category in brackets. ... 42
\DTXD@filenameindex: Put margin tag category in brackets. ... 43
\DTXD@filemarginparindex: Fix: File category. ... 43
\sourcedisplay: Reduced width. ... 71
\UIdisplay: Reduced width. ... 72

v1.02
General: 2019/07/16 ... 1
Fix if not doc package. ... 38
Fix if not hyperref package. ... 33
Fix if not ltxdoc class. ... 33, 38
\DescribeEnv: Fix if not ltxdoc class. ... 54
\DescribeMacro: Fix if not ltxdoc class. ... 53
\DescribeObject: \raggedleft margin par. ... 61
\usage: Fix if not ltxdoc class. ... 39
\DTXD@gobble: Fix if not doc package. ... 37
\usage: Fix if not ltxdoc class. ... 39

v1.03
General: 2022/02/01 ... 1
No longer requires xifthen. ... 33
\attrib: For LuaTeX. ... 76
\cs: Provided. ... 73
\DescribeHook: Added. ... 60
\DescribeMacro: Reduce hbox warnings. ... 53
\DescribeObject: Reduce hbox warnings. ... 61
DescribeOther: Reduce hbox warnings. ........................................... 62
\DTXD@cmdmargintagindex: Reduce hbox warnings. ........................................... 42
\DTXD@margintag: Reduce hbox warnings. ........................................... 40
\dviLaTeX: Added. ........................................... 77
\dviTeX: Added. ........................................... 77
\ItemDescribeHook: Added. ........................................... 64
\marg: Provided. ........................................... 73
\MathJax: Added. ........................................... 77
\oarg: Provided. ........................................... 73
\parg: Provided. ........................................... 73
\pdfLaTeX: Added. ........................................... 78
\pdfTeX: Added. ........................................... 78
\shownesting: Added. ........................................... 68
v1.04
General: 2022/02/01 ........................................... 1
Corrected copyright date. ........................................... 1
v1.05
General: 2022/09/08 ........................................... 1
\TikZ: Renamed from \tikz, cap Z. ........................................... 77
v1.06
General: 2022/12/07 ........................................... 1
Fixed for updated doc. 35, 36, 43, 44
@NewDocElement: Fixed for updated doc. ........................................... 44
v1.07
General: 2023/01/03 ........................................... 1
Improved PDF bookmarks. ........................................... 36, 37
\doccreatespecialmacrolikeindexes: Added category names for macros. ........................................... 50
\DTXD@findgtempa: Detokenized names. ........................................... 47
\DTXD@maybecategory: \verb*. ........................................... 46
Improved line breaks. ........................................... 46
\DTXD@maybecategorylevel: \verb*. ........................................... 46
Fix occasional crash. ........................................... 47
\DTXD@break: Improved line breaks. ........................................... 45

v1.08
General: 2024/01/08 ........................................... 1
Updated contact info. ........................................... 1
\DescribePlug: Added. ........................................... 60
\DescribeSocket: Added. ........................................... 60
\ItemDescribePlug: Added. ........................................... 65
\ItemDescribeSocket: Added. ........................................... 64

v1.09
General: 2024/01/20 ........................................... 1
Improved TOC change log and index entries. ........................................... 1
Removed extra spaces. ........................................... 44
\doccreatespecialindexes: Removed extra spaces. ........................................... 49
\doccreatespecialmacrolikeindexes: Removed extra spaces. ........................................... 51

B Index

Symbols
\@NewDocElement ........................................... 364
[H] (argument) [figure] ........................................... 24
[M] (argument) [figure] ........................................... 24

A
\acro ........................................... 17, 1368
Additional_Item (other) [othercategory] ........................................... 25
\AmS ........................................... 18, 1388
another_big_file.txt (file) [bigfiles] ........................................... 22
Arguments:
[H] [figure] ........................................... 24
[M] [figure] ........................................... 24
bold [\mymacro] ........................................... 24
attrib ........................................... 17, 1354
attribute ........................................... 17, 1354

B Before_myenvironment (other) ........................................... 25
\BibTeX ........................................... 18, 1393
[bigfiles]:
another_big_file.txt (file) ........................................... 22
really_big_file.txt (file) ........................................... 22
bold (argument) [\mymacro] ........................................... 24
\bool ........................................... 16, 1315
booleanname (boolean) [descexamples] ........................................... 26
Booleans:
booleanname [descexamples] ........................................... 26
sampleboolean [examples] ........................................... 21
\brand ........................................... 17, 1367

C calc (package) ........................................... 33
caption (package) ........................................... 33
\captionsetup [dtxexamplefloat] ........................................... 70
class_name (class) [descexamples] ........................................... 26
environment 9
fsourceverb 1258
fsourceverb 15
macro 9
myenvironment 20
noindentenvironment 1246
noindentenvironment 14
noindentmacro 1240
noindentmacro 14
otherenvironment 20
sourcedisplay 1261
sourcedisplay 15
sourceverb 1255
sourceverb 14
UIdisplay 1272
UIdisplay 15
\EPS 16, 1348
\EPUB 17, 1361
etoolbox (package) 33
[examples]:
sample_class (class) 22
sampleboolean (boolean) 21
samplecounter (counter) 21
samplekey (key) 23
sampleoption (option) 22
samplepackage (package) 22
sampletwokey (key) 23

F
fancyverb (package) 33
[figure]:
[H] (argument) 24
[M] (argument) 24
file_name (file) [descexamples] 26
\filename 16, 1332
Files:
another_big_file.txt [bigfiles] 22
dtxexample_cut.txt 68
file_name [descexamples] 26
lone_file.txt 22
really_big_file.txt [bigfiles] 22
firstkey (key) [groupofkeys] 23
\fquad 15, 1254
\fquad 15, 1253
\fquad 15, 1252
fsourceverb (env.) 1258
fsourceverb (env.) 15

G
\GIF 16, 1346
group of objects 20
[groupofKeys]:
firstkey (key) 23
secondkey (key) 23

H
\hookname (hook) 21
Hooks:
\hookname 21
para/begin 21
\HTML 17, 1357
\HTMLfive 17, 1358
hyperref (package) 33

I
index
by group 20
\IndexMin (length) 38
\ItemDescribeArgument 12, 1041
\ItemDescribeBoolean 12, 1046
\ItemDescribeClass 12, 1091
\ItemDescribeCommand 12, 1126
\ItemDescribeCounter 12, 1056
\ItemDescribeEnv 12, 1036
\ItemDescribeFile 12, 1106
\ItemDescribeHook 12, 1061
\ItemDescribeKey 12, 1076
\ItemDescribeLength 12, 1051
\ItemDescribeMacro 12, 1031
\ItemDescribeObject 12, 1136
\ItemDescribeOption 12, 1101
\ItemDescribeOther 12, 1141
\ItemDescribePackage 12, 1081
\ItemDescribePlug 12, 1071
\ItemDescribeProgram 12, 1116
\ItemDescribeSocket 12, 1066

J
\JP 16, 1347

K
keyname (key) [descexamples] 26
Keys:
firstkey [groupofkeys] 23
keyname [descexamples] 26
lonekey 23
samplekey [examples] 23
sampletwokey [examples] 23
secondkey [groupofkeys] 23

L
\lengethname (length) [descexamples] 26
Lengths:
\IndexMin 38
\lengethname [descexamples] 26
\photowidth [photograph] 21
license_agreement (other) 25
listofdtxexamplefloats 14
\LOF 16, 1318
lone_file.txt (file) 22
<table>
<thead>
<tr>
<th>Package Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\LOT</code></td>
<td>16, 1319</td>
</tr>
<tr>
<td><code>\LuaLaTeX</code></td>
<td>18, 1375</td>
</tr>
<tr>
<td><code>\LuaTeX</code></td>
<td>18, 1374</td>
</tr>
<tr>
<td><code>\LyX</code></td>
<td>18, 1392</td>
</tr>
</tbody>
</table>

**M**

- `macro (env)` .................. 9
- `macroname [descexamples]` .... 26
- `makeidx (package)` ........... 33
- `\MakeIndex` ................... 18, 1394
- `\marg` ........................ 9, 15, 1308
- `margin tag missing` .......... 32
- `\marginpar` ................... 13, 1147
- `\marginparcolor` ............. 13, 1146
- `\MathJax` ...................... 17, 1366
- `\MathML` ....................... 17, 1365
- `\MiKTeX` ...................... 18, 1396
- `myenvironment (env)` .......... 20
- `[\mymacro]` ................... 24
- `\mymacro` ...................... 19

**N**

- `newfloat (package)` .......... 33
- `noindenvironment (env)` ..... 1246
- `noindenvironment (env)` ..... 14
- `noindmacro (env)` ............ 1240
- `noindmacro (env)` ............ 14

**O**

- `\oarg` ........................ 9, 15, 1310

**Objects:**

- `\DTXexamplerulecolor [color]` 68
- `othercolor [color]` .......... 25
- `randomobject` ................. 25
- `somecolor [color]` .......... 25
- `\DOT` ........................ 16, 1343

**Options:**

- `sampleoption [examples]` ..... 22
- `\optn` ........................ 16, 1316
- `OS_command (command)` ....... 22
- `Other_Item (other) [othercategory]` 25

**Other:**

- `Additional_Item [othercategory]` 25
- `Before_myenvironment` ....... 25
- `license_agreement` .......... 25
- `Other_Item [othercategory]` 25

**P**

- `package_name (package) [descexamples]` 26

**Packages:**

- `calc` .......................... 33
- `caption` ........................ 33
- `etoolbox` ........................ 33
- `fancyvrb` ........................ 33
- `hyperref` ....................... 33
- `makeidx` ........................ 33
- `newfloat` ........................ 33
- `package_name [descexamples]` ... 26
- `pic\texttwo` ................... 35
- `samplepackage [examples]` ..... 22
- `splitidx` ........................ 33
- `xcolor` .......................... 33
- `xparse` .......................... 33
- `xstring` .......................... 33

**Para/Begin (hook)** ............. 21

- `\par` .......................... 9, 15, 1312
- `\pdf` .......................... 16, 1349
- `\pdfLaTeX` ...................... 17, 1373
- `\pdfTeX` ........................ 17, 1372
- `\photocaptionsetup [photograph]` ... 19

**[photograph]:**

- `\photonomeref` ............... 19
- `\DeclareFloatingPhoto` ....... 19
- `\photocaptionsetup` .......... 19
- `\photowidth (length)` ........ 21
- `\photowidth (length) [photograph]` ... 21
- `pic\texttwo (package)` ....... 35
- `\pkg` .......................... 15, 1305
- `\plugin (plug)` ............... 21

- `\plugname` ..................... 21

**Plugs:**

- `\plugname (plug)` ............. 21

**R**

- `randomobject (object)` ..... 25
- `really_big_file.txt (file) [bigfiles]` 22
- `sample_class (class) [examples]` 22
- `sampleboolean (boolean) [examples]` 21
- `samplecounter (counter) [examples]` 21
- `samplekey (key) [examples]` .... 23
- `sampleoption (option) [examples]` 22

85