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, 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 $\backslash \texttt{DescribeEnv}$, 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 $\backslash \texttt{input}$ 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 pdf\TeX, \Xe\TeX, and \Lua\TeX, 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, keys .......... 10
   3.4 Packages, classes, options .......................... 10
   3.5 Files, programs, commands .......................... 11
   3.6 Other source objects ................................ 11
   3.7 In a description environment ....................... 11
   3.8 Defaults ............................................. 12
   3.9 Nesting ............................................... 12
   3.10 \margintag, \watchout ............................... 13
   3.11 dtxexample environment ......................... 13
   3.12 noindmacro and noindenvironment environments . 14
   3.13 sourceverb, sourcedisplay, UIDisplay, docsbcbar . 14
   3.14 Formatted objects .................................... 15
      3.14.1 LaTeX objects .................................. 15
      3.14.2 Programs and commands ......................... 15
      3.14.3 File types ...................................... 16
      3.14.4 Internet ......................................... 16
      3.14.5 Specific programs ............................... 17
      3.14.6 Acronyms, brand names, trademarks .......... 17
   3.15 Logos ................................................. 17
   3.16 Dashes and slashes .................................. 18
4 Examples

5 Usage notes

6 Code

6.1 Required packages ........................................ 33
6.2 Warning sign .................................................. 35
6.3 Special character handling .............................. 35
6.4 Patching hypdoc, splitidx, doc ........................ 35
6.5 Gobbling comment characters ............................ 37
6.6 Vertical spacing .............................................. 38
6.7 Not ltxdoc: ltxdoc emulation ............................ 38
6.8 Not doc: doc emulation ..................................... 38
6.9 Support macros ............................................... 39
6.10 doc: Key handling for object categories .............. 43
6.11 doc: Handling \marginpar inside a float ............... 44
6.12 doc: Handling categories and detokenizing names .... 44
6.13 Not doc: \DescribeMacro and \DescribeEnvironment .... 53
6.14 New \Describe... macros ................................. 54
6.15 \DescribeDefault ......................................... 62
6.16 \ItemDescribeMacro, etc. ................................. 63
6.17 \margintag, \watchout .................................... 66
6.18 Nesting ...................................................... 67
6.19 The dtxexample environment ............................. 67
6.20 noindmacro and noindenviron ................................ 70
6.21 sourcedisplay, UIDisplay, docsSidebar .................. 70
6.22 Formatted objects ........................................... 72
   6.22.1 \LaTeX objects ........................................ 72
   6.22.2 Programs and commands .............................. 73
6.22.3 File types ................................................. 74
6.22.4 Internet .................................................. 75
6.22.5 Specific programs ....................................... 76
6.22.6 Acronyms, brand names, trademarks ............... 76
6.23 Logos ......................................................... 76
6.24 Dashes and slashes ........................................ 78

7 Compiling dtxdescribe ........................................ 79

Change History and Index ...................................... 80

List of Examples

1  Macros ....................................................... 19
2  Environment ............................................... 20
3  Second Environment ....................................... 20
4  Booleans and Counters .................................... 21
5  Lengths ...................................................... 21
6  Hooks ........................................................ 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
16 fsourceverb ............................................... 29
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 dtxdescribe with the ltxdoc class and .dtx files:

\%<\*driver>
\documentclass{ltxdoc}
\usepackage{lmodern}
\usepackage{dtxdescribe}
\usepackage{packagename} % the name of your new package
\AtBeginDocument{% hyperref now automatically loaded by \pkg{ltxdoc}.
  \hypersetup{...}% If needed.
  \pdfstringdefDisableCommands{ ... }% If needed.
  \AddToHook{begindocument/before}{% Before .aux file is loaded.
    \usepackage{cleveref}% If needed.
    \crefname{somename}{name}{names}% If needed.
  }
  ...
}\%</driver>

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

\documentclass{article}
\usepackage{dtxdescribe}
\usepackage{hyperref}% If needed.
\hypersetup{...}% If needed.
\pdfstringdefDisableCommands{ ... }% If needed.
\usepackage{cleveref}% If needed.
\crefname{somename}{name}{names}% If needed.
\begin{document}
  ...
  \end{document}

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

`\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{[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. `\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{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, the macro and environment environments would localize any definitions, and \texttt{\DescribeMacro} and \texttt{\DescribeEnv} should be used instead.

\texttt{\DescribeMacro} \texttt{\[\langle category\rangle\]} \texttt{\{\langle name\rangle\}}

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.

\texttt{\DescribeEnv} \texttt{\[\langle category\rangle\]} \texttt{\{\langle environment name\rangle\}}

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

\texttt{\marg} \texttt{\{\langle text\rangle\}}

Shows a mandatory argument for a macro or environment.

The results looks like \texttt{\{\texttt{mandatory}\}}.

\texttt{\oarg} \texttt{\{\langle text\rangle\}}

Shows an optional argument for a macro or environment.

The results looks like \texttt{\{\texttt{optional}\}}.

\texttt{\parg} \texttt{\{\langle text\rangle\}}

Used for “picture” arguments, such as coordinates.

The result looks like \texttt{\{\texttt{coordinate}\}}.
\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, keys

\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]

\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 \langle category \rangle \langle name \rangle

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

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

### 3.5 Files, programs, commands

\DescribeFile\quad [\langle category\rangle] \langle name\rangle

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

\DescribeProgram\quad [\langle category\rangle] \langle name\rangle

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

\DescribeCommand\quad [\langle category\rangle] \langle name\rangle

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\quad [\langle category\rangle] \langle name\rangle

Describes an arbitrary programming object, such as a color definition or caption setup. A margin tag and index entry are created with \texttt{family} 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\quad [\langle category\rangle] \langle name\rangle

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\quad [\langle category\rangle] \langle name\rangle\ A description.

\ItemDescribeEnv\quad [\langle category\rangle] \langle name\rangle\ A description.

\ItemDescribeArgument\quad [\langle category\rangle] \langle argument\rangle\ A description.
\ItemDescribeBoolean \[\langle\text{category}\rangle\] \{\langle\text{name}\rangle\} A description.
\ItemDescribeLength \[\langle\text{category}\rangle\] \{\langle\text{name}\rangle\} A description.
\ItemDescribeCounter \[\langle\text{category}\rangle\] \{\langle\text{name}\rangle\} A description.
\ItemDescribeHook \[\langle\text{category}\rangle\] \{\langle\text{name}\rangle\} A description.
\ItemDescribeKey \[\langle\text{category}\rangle\] \{\langle\text{package_name}\rangle\} With underscores.
\ItemDescribePackage \[\langle\text{category}\rangle\] \{\langle\text{package_name}\rangle\} With underscores.
\ItemDescribeClass \[\langle\text{category}\rangle\] \{\langle\text{class_name}\rangle\} With underscores.
\ItemDescribeOption \[\langle\text{category}\rangle\] \{\langle\text{name}\rangle\} A description.
\ItemDescribeFile \[\langle\text{category}\rangle\] \{\langle\text{file_name}\rangle\} With underscores.
\ItemDescribeProgram \[\langle\text{category}\rangle\] \{\langle\text{program_name}\rangle\} With underscores.
\ItemDescribeCommand \[\langle\text{category}\rangle\] \{\langle\text{command_name}\rangle\} With underscores.
\ItemDescribeObject \[\langle\text{category}\rangle\] \{\langle\text{name}\rangle\} A description.
\ItemDescribeOther \[\langle\text{category}\rangle\] \{\langle\text{name}\rangle\} A description.

### 3.8 Defaults

\DescribeDefault \{\langle\text{value}\rangle\}

Default: value

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

\DescribeDefaultcolor

Default: green!50!black

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

### 3.9 Nesting

\shownesting \*[\langle\text{fraction of \textwidth}\rangle] \{\langle\text{container name}\rangle\} \{\langle\text{contents}\rangle\}

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

The optional argument is the fraction of \textwidth 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 \linewithin for each box, and use \shownesting\mbox* 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.

\margintag\example

The color of the \margintag.

\margintag\color

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.

\watchout\example

The color of the \watchout.

\watchout\color

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 la-
beIs. 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.

listofdtxexamplefloats

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:

\dtxexample\codename

The text name of the code section.

\dtxexample\resultname

Default: Result:
### 3.12 noindmacro and noindentenvironment environments

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

**noindmacro** (env.) \begin{noindmacro}{\macroname} \oarg{optional} \marg{mandatory} ... \end{noindmacro}

**noindentenvironment** (env.) \begin{noindentenvironment}{\name} ... \end{noindentenvironment}

Replace

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

with

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

and similarly for `noindentenvironment`.

### 3.13 sourceverb, sourcedisplay, UIdisplay, docsdisplay

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

**fsourceverb** (env.) \[\langle key/values \rangle \] 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.

**sourcedisplay** (env.) Display source code with manual formatting. This is not a verbatim environment. `\textcolor`, `\textbf`, and `\emph` may be used to highlight text. Macros must be escaped with `\cs`, characters such as `{` must be produced with `\{`, etc. `\` must be used to force a new line. `\fquad`, `\qqquad`, and `\qqquad` may be used to force indenting. Because this is **not** a verbatim environment, it **can** be used inside a macro. See example 17 on page 29.

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

\fquad Double-level indent inside a `sourcedisplay`.
\qqquad Triple-level indent inside a sourcedisplay.

\UIDisplay (env.) 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.

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

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

docsidebar (env) \[
\langle \text{title} \rangle \] 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 \{\textit{packagename}\} Prints as packagename. Also for a classname.

\cs \{\textit{csname}\} Prints as \csname.

\env \{\textit{environment}\} Prints as environment.

\marg \{\textit{argument}\} Prints \{\textit{arg}\}. Mandatory argument.

\oarg \{\textit{argument}\} Prints [\textit{arg}]. Optional argument.

\parg \{\textit{argument}\} Prints (\textit{arg}). Picture-mode argument.

\ctr \{\textit{counter}\} Prints as counter.

\bool \{\textit{boolean}\} Prints as boolean.

\optn \{\textit{option}\} Prints as option, for example to a macro, package, class.

\TOC TOC: Table of contents.

\LOF LOF: List of figures.

\LOT LOT: List of tables.

3.14.2 Programs and commands

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

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

\filenm \textbf{Prints as} file\_name: Underscores allowed.

\UI \textbf{Prints as} General user-interface text. What the user sees on the display. Also see the \texttt{UIdisplay} environment.

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

\textbf{3.14.3 File types}

\ODT odt OpenDocument Format word processing document

\SVG svg image format

\PNG PNG image format

\GIF GIF image format

\JPG JPG image format

\EPS EPS image format

\PDF PDF image format

\DVI DVI image format

\textbf{3.14.4 Internet}

\UTF UTF: Unicode

\URL URL: Uniform Resource Locator

\element \texttt{\{⟨element name⟩\}} \textbf{Prints as} \texttt{⟨element⟩}, an HTML/CSS element

\attribute \texttt{\{⟨attribute name⟩\}} \textbf{Prints as} attribute, an HTML/CSS attribute. pdf\TeX and X\TeX \textbf{only}. Not for Lua\TeX.

\attrib \texttt{\{⟨attribute name⟩\}} \textbf{Prints as} attribute, an HTML/CSS attribute. pdf\TeX, X\TeX, or Lua\TeX.

\HTML HTML: Hypertext Markup Language

\HTMLfive HTML5: Old-style figure if font supports

\CSS CSS: Cascading Style Sheet

\CSSthree CSS3: Old-style figure if font supports
3.14.5 Specific programs

\TikZ TikZ: Package logo
\CTAN CTAN: Comprehensive \TeX{} Archive Network
\TDS TDS: \TeX{} Directory Structure
\MathML MathML: Mathematical Markup Language
\MathJax MathJax: Math on the web.

3.14.6 Acronyms, brand names, trademarks

\brand \{(name)\} BRANDNAME, COMPANY NAME
\acro \{(acronym)\}acro: Acronym
\supregistered Superscript trademark symbol®

3.15 Logos

Several additional logos are provided.

Also see the metalogo and metalogox packages.

\dviTeX DVI \TeX{}
\dvilatex DVI \LaTeX{}
\pdftex PDF \TeX{}
\pdflatex PDF \LaTeX{}
\luatex \LaTeX{}
\luatex \LaTeX{}
\xetex X\TeX{}, with reversed E if graphics is loaded.
\xelatex X\LaTeX{}, with reversed E if graphics is loaded.
\amsmath \\mathcal{AMS}
\lyx LyX
\biblatex \\LaTeX{}

\epub EPUB: E-book file format
3.16 Dashes and slashes

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>A--B</td>
<td>A–B (not breakable)</td>
</tr>
<tr>
<td>A \dash B</td>
<td>A – B (only breakable before the B)</td>
</tr>
<tr>
<td>A -- B</td>
<td>A – B (breakable before or after the dash)</td>
</tr>
<tr>
<td>A---B</td>
<td>A—B (not breakable)</td>
</tr>
<tr>
<td>A \Dash B</td>
<td>A—B (only breakable before the B)</td>
</tr>
<tr>
<td>A --- B</td>
<td>A—B (breakable before or after the dash)</td>
</tr>
<tr>
<td>A/B</td>
<td>A/B (not breakable)</td>
</tr>
<tr>
<td>A \Slash B</td>
<td>A/B (only breakable before the B)</td>
</tr>
<tr>
<td>A / B</td>
<td>A / B (breakable before or after the slash)</td>
</tr>
<tr>
<td>A~/~B</td>
<td>A / B (not breakable)</td>
</tr>
</tbody>
</table>
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\[\langle\textit{optional}\rangle\] \{\langle\textit{mandatory}\rangle\} A typical macro definition.

Create a photograph float environment.

Caption settings for a photograph float.

cleveref name for the photograph float.

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:
\{\langle argument\rangle\} Short description.

\textit{myenvironment} (env.) \{\langle argument\rangle\} Short description.

\textit{add’l tags} The re-defined \DescribeEnv adds an ‘Env’ tag to the margin, and adds “(environment)” to its own index entry. Note that environments and all the other new objects defined by this package each receives two index entries, one by name, and one grouped with others of its kind.

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

Result:
\{\langle opt \textit{args}\rangle\} \{\langle coordinates\rangle\} A description.

\textit{otherenvironment} (env.) \{\langle opt \textit{args}\rangle\} \{\langle coordinates\rangle\} A description.

The \textit{otherenvironment} will be indexed by itself and also with \textit{myenvironment} under the index entry “environments”, and also under the category \textit{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

Code:
\DescribeHook{\hookname} A hook with a backslash.
\DescribeHook{para/begin} A hook without a backslash.

Result:
\hookname (Hook) A hook with a backslash.
para/begin (Hook) A hook without a backslash.

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

Code:

\b\texttt{\DescribePackage}[examples]{samplepackage} \hspace{1em} About a \LaTeX\ package.

\b\texttt{\DescribeClass}[examples]{sample\_class} \hspace{1em} About a \LaTeX\ class.

\b\texttt{\DescribeOption}[examples]{sampleoption} \hspace{1em} 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:

\b\texttt{\DescribeFile}[bigfiles]{really\_big\_file.txt} Some description.

\b\texttt{\DescribeFile}[bigfiles]{another\_big\_file.txt} Some description.

\b\texttt{\DescribeFile}{lone\_file.txt} Some description.

\b\texttt{\DescribeCommand}{OS\_command} An operating-system command.

\b\texttt{\DescribeProgram}{program\_name} An operating-system program.

Result:

| really\_big\_file.txt \textit{(file)} [bigfiles] | Some description. |
| another\_big\_file.txt \textit{(file)} [bigfiles] | Some description. |
| lone\_file.txt \textit{(file)} [bigfiles] | Some description. |
| 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]{sampletwokey} About another key of |otherkeys|.

\DescribeKey{lonekey} A key without a category.

Result:

<table>
<thead>
<tr>
<th>Key</th>
<th>About</th>
</tr>
</thead>
<tbody>
<tr>
<td>firstkey</td>
<td>About the first key of the groupofkeys set.</td>
</tr>
<tr>
<td>secondkey</td>
<td>About the second key of groupofkeys.</td>
</tr>
<tr>
<td>samplekey</td>
<td>About some key of otherkeys.</td>
</tr>
<tr>
<td>sampletwokey</td>
<td>About another key of otherkeys.</td>
</tr>
<tr>
<td>lonekey</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 [H]ere.

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

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

Result:

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

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

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

**Code:**

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

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

\DescribeObject{randomobject} About some random object.

**Result:**

<table>
<thead>
<tr>
<th>color</th>
<th>The</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>somecolor</td>
<td>color</td>
<td>The color of something.</td>
</tr>
<tr>
<td>othercolor</td>
<td>color</td>
<td>The other color.</td>
</tr>
<tr>
<td>randomobject</td>
<td></td>
<td>About some random object.</td>
</tr>
</tbody>
</table>

Describes an arbitrary programming object, using `\texttt{}` text.

Example 12: Other

**Code:**

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

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

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

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

**Result:**

<table>
<thead>
<tr>
<th>license agreement</th>
<th>The following is the fictional license agreement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before myenvironment</td>
<td>Actions to be done <code>\cs{BeforeBeginEnvironment}</code>.</td>
</tr>
<tr>
<td>Other Item [othercategory]</td>
<td>About the other item.</td>
</tr>
<tr>
<td>Additional Item [othercategory]</td>
<td>About the add'l item.</td>
</tr>
</tbody>
</table>

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}

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

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

Code:
```
\shownesting{\env{environmentA}}{
  \shownesting{\cs{macroB}}{
    \shownesting{\env{environmentC}}{
      \begin{itemize}
        \item The contents.
      \end{itemize}
    }
  }
}
\shownesting{\env{sidebyside}}{
  \shownesting[.35]{\minipage}{
    \begin{itemize}
      \item Left contents.
    \end{itemize}
  }
  \texttt{\hfill \cs{hfill} \hfill}
  \shownesting*[.35]{\minipage}{
    \begin{itemize}
      \item Right contents.
    \end{itemize}
  }
}
```

Result:
```
\begin{itemize}
  \item The contents.
\end{itemize}
\begin{minipage}{.35\textwidth}
  \begin{itemize}
    \item Left contents.
  \end{itemize}
\end{minipage}\texttt{\hfill \cs{hfill} \hfill}
\begin{minipage}{.35\textwidth}
  \begin{itemize}
    \item Right contents.
  \end{itemize}
\end{minipage}
```

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 \texttt{\hfill} text.
Example 15: \texttt{dtxexample}

\textit{Code:}

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

\textit{Result:}

\textit{See fig. 1}

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

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

When the example was created:

1. The “float” of type \texttt{dtxexamplefloat} was created, with the caption “\texttt{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 \texttt{\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 \texttt{dtxexample}.

7. The starred form of \texttt{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:

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

Result:

\begin{verbatim}
\newcommand{fdosomething}[1][whattodo]{
  doing #1
}
\end{verbatim}

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

Example 17: sourcedisplay

Code:

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

Result:

\begin{verbatim}
\newcommand{dosomething}[1][whattodo]{
  doing #1
}
\end{verbatim}
Example 18: Udisplay

Code:

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

Result:

Select:

Preferences → Plugins → Files → HTML

For the field

Title heading:

Enter ⇒ H1
Example 19: docsidebar

\begin{docsidebar}
\textbf{A title}

An aside, which may help explain something incidental to the main text.
\end{docsidebar}

Additional main text.
5 Usage notes

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

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

\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

\vspace{-\baselineskip}

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

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

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

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

6.1 Required packages

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

\makeindex

\AtBeginDocument{
  \IfPackageLoadedTF{makeidx}{}{
    \IfPackageLoadedTF{splitidx}{}{
      \RequirePackage{makeidx}
      \makeindex
    }
  }
}

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

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

xparse (Pkg) Used for the examples.

\RequirePackage{xparse}

\xparse

calc (Pkg) Used for \shownesting.

\RequirePackage{calc}

xcolor (Pkg) Used for the examples.

\definecolor{myurlcolor}{rgb}{0,0,.7}
\definecolor{mylinkcolor}{rgb}{.7,0,0}

caption (Pkg) Used for the examples.

\RequirePackage{caption}

newfloat (Pkg) Used for the examples.

\RequirePackage{newfloat}

fancyvrb (Pkg) Used for the examples.

\RequirePackage{fancyvrb}

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

\RequirePackage{xstring}

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

PDF bookmarks
\AtBeginDocument{
\IfPackageLoadedTF{hyperref}{
\pdfstringdefDisableCommands{%
  \def\quad{}%
  \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 \texttt{hyperref} is not loaded, emulate \texttt{\hyperpage} here.

\begin{verbatim}
\newcommand*{\hyperpage}{\hyperref[1][\#1]}
\end{verbatim}

\texttt{pict2e} (\texttt{Pkg})

\begin{verbatim}
\RequirePackage{pict2e}
\setlength{\unitlength}{1pt}
\end{verbatim}

\section{Warning sign}

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

\section{Special character handling}

The literal backslash character:

\begin{verbatim}
\begingroup
\catcode`\|=0
\catcode`\\=12
\edef\DTXD@backslash{{}\endgroup}
\end{verbatim}

\section{Patching \texttt{hypdoc}, \texttt{splitidx}, \texttt{doc}}

If \texttt{hyperref} is disabled (by \texttt{lwarp}) then define the missing \texttt{\hdclindex}.

\begin{verbatim}
\IfPackageLoadedTF{doc}{
\IfPackageLoadedTF{hypdoc}{
\AddToHook{begindocument/before}[doc/hyperref]{%
\ifdoc@hyperref
\else
\def\hdclindex#1#2{%}
\else
\def\hdclindex#1#2{%}
\end{verbatim}
Several changes for when hypdoc and splitidx are used together:

```latex
\AtBeginDocument{
  \IfPackageLoadedTF{doc}{
    \IfPackageLoadedTF{hypdoc}{
      \IfPackageLoadedTF{splitidx}{
        \renewcommand*{\@wrsindex}[2]{
          \ifx\relax#1\relax
            \if@splitidx
              \@wrsindex[idx]{#2}\fi
            \else
              \def@tempa{#2}\fi
            \else
              %
              % \def@tempa{#2}%
              \def@tempa{#2\encapchar hdpindex{}}% dtxdescribe
            \fi
          \else
            \csname index@#1@hook\endcsname
          \expandafter\ifx\csname @@wrsindex\endcsname\relax
            \@@@wrsindex{#1}{#2}{\thepage}\fi
            \def@tempb{\@@wrsindex{#1}}\expandafter\@tempb\@tempa||\fi
          \endgroup
          \@esphack
          \fi
        }
      }
    }
  }
}
```


```
\def\HD@guesstoclevel#1{1}
```

Patch doc's \PrintChanges to reset the 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}
}{}
{}% splitidx loaded
{}% hypdoc loaded
{}% doc loaded
% AtBeginDocument

6.5 Gobbling comment characters

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

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

\IndexMin \textit{(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.

\AtBeginDocument{}
\ifdefTF{\texttt{ltxdoc}}{}{% not \texttt{ltxdoc}
\newenvironment{macro}[1]{\PackageError{dtxdescribe}{The 'macro' environment is only available when using the \texttt{doc} package 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 \texttt{doc} package with a .dtx source file}{}{This environment only makes sense for .dtx source.}}{}

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

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

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

\PrintEnvName \{\langle name\rangle\} \quad Prints an environment name.

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

\usage \{\langle text\rangle\} \quad Allow hyperlinks in the “usage” index entries:
\DTXD@origwrindex Used to bypass hyperref index modifications.

\DTXD@margintag \{\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.

\DTXD@index \{\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.

\begin{verbatim}
258 \@bsphack
259 \begingroup
260 \DTXD@origwrindex{%

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

261 #2\actualchar{\protect\ttfamily#2} % name
262 (#4)% index tag
263 \ifblank{#1}{}{[#1]}%  
264 \encapchar #5}%

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

265 \begingroup
266 \DTXD@origwrindex{%  
267 #4:\levelchar% index tag
268 \ifblank{#1}{}{[#1]:\levelchar}%
269 #2\actualchar{\protect\ttfamily#2}% name
270 \encapchar #5}%

Possibly index by category and name:

271 \ifblank{#1}{}{% category given
272 \begingroup
273 \DTXD@origwrindex{%  
274 #1\actualchar[#1]:\levelchar% category
275 #2\actualchar{\protect\ttfamily#2} % name
276 (#4)% index tag
277 \encapchar #5}%
278 }% category given
279 \@esphack
280 \@esphack
281 \ignorespaces%
282
\end{verbatim}

\DTXD@margintagindex \{(category)\} \{(name)\} \{(margin tag)\} \{(index tag)\} \{(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 {⟨control sequence⟩}

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

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

\DTXD@verbatimcmd {⟨\name⟩}

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

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

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

\DTXD@cmdmargintagindex [5]{%}

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 \[\text{category}]:>name=csname (indextag)\]|usage

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

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

6.10 \texttt{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 doc: Handling \marginpar inside a float

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

\IfPackageLoadedTF{doc}{
  \def\@doc@describe#1#2{% dtxdescribe
    \ifdoc@noprint\else
      \@ifundefined{@captype}{% not float? dtxdescribe
        \marginpar{\raggedleft
          \strut
          \doc@providetarget
          \@nameuse{PrintDescribe#1}{#2}% dtxdescribe
        }%}
      }{}
    \fi
    \ifdoc@noindex\else
      \@nameuse{Special#1Index}{#2}%
    \fi
    \@esphack
  }
}

6.12 doc: Handling categories and detokenizing names

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

\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{}
    \fi
  }
}

\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 \ Add 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}[2]{%
\DTXD@maybecategory If there is a category, it is added verbatim.

\newcommand*{\DTXD@maybecategory}{% 
  \ifdefvoid{\DTXD@category}% 
  {}% 
  { 
    \DTXDbreak inserts a possible line break here, allowing a break if the following verbatim is too long. 
  }% 
} 

\DTXD@categorylevelname The simplified name, without backslash.

\newcommand*{\DTXD@categorylevelname}{\DTXD@maybecategorylevel}{% 
} 

\DTXD@maybecategorylevel \{⟨object\_type⟩\} 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.
Index:
categorylevelname=\verb!*+[category]:*=>name=\verb!*+[prefix]name+<break>{type}

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

\DTXD@gtempa@nobackslash A version without the leading backslash.

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

\doc@createspecialindexes {⟨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.
\noexpand\DTXD@findgtempa{##1}\%
\@temptokena: Index (type).
\@temptokenb: Index group.
\@gtempa: Detokenized name of the particular object.
\DTXD@gtempa@nobackslash: Detokenized name without backslash.
\noexpand\@bsphack
\noexpand\DTXD@findcategorylevelname% 
\ifdoc@toplevel \noexpand\special@index{##1\noexpand\actualchar}%
\noexpand\DTXD@printobjectname% \noexpand\DTXD@gtempa@nobackslash}%%
\noexpand\@gtempa}%%
\ifx\@nil#2\@nil\else \the\@temptokena \fi
\noexpand\DTXD@maybecategory% \noexpand\encapchar main\%
\}%

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

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

If group is not empty, index:
\@nil#3\@nil\else
Maybe add the category index entry:

```
\noexpand\DTXD@maybe\categorylevel{#2}%
\if\doc@toplevel
\noexpand\doc@providetarget
% \noexpand\index{##1\actualchar{\string\ttfamily\space##1}%
\noexpand\index{\noexpand\DTXD@print\objectname\noexpand\DTXD@\tempa@\nobackslash}%
\noexpand\\DTXD@\tempa@\maybe\category% dtxdescribe
\noexpand\\encapchar main%}
\fi
\noexpand\@esphack}
```

**Define \Special\<name\>\Index {(/I: name)}**

- **\@temptokena**: Index (type).
- **\@temptokenb**: Index group.
- **\@gtempa**: Detokenized name of the particular object.
- **\DTXD@gtempa@\nobackslash**: Detokenized name without backslash.

```
\noexpand\\\DTXD@gtempa@\nobackslash% dtxdescribe
\noexpand\@esphack
\noexpand\DTXD@\find\categorylevelname% dtxdescribe
\if\doc@toplevel
% \noexpand\index{##1\actualchar{\string\ttfamily\space##1}%
\noexpand\index{\noexpand\DTXD@print\objectname\noexpand\DTXD@\tempa@\nobackslash}%
\noexpand\\DTXD@\tempa@\maybe\category% dtxdescribe
\noexpand\\doc@\handle\encap\usage%}
\fi
```

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

```
\if\doc@toplevel
\noexpand\doc@providetarget
```

Maybe add the category index entry:
```
\noexpand\DTXD@\maybe\categorylevel{#2}%
```
If group is not empty, index:

\begin{verbatim}
\ifx\@nil#3\@nil\else
\noexpand\index{%
  \the\@temptokenb
  \noexpand\levelchar%
  \string\ttfamily\space##1}
\fi
\end{verbatim}

Maybe add the category index entry:

\begin{verbatim}
\noexpand\DTXD@maybecategorylevel(#2)%
\fi
\noexpand\DTXD@maybecategoryname% dtxdescribe
\noexpand\@gtempa% dtxdescribe
\noexpand\doc@handleencap{usage}%
\end{verbatim}

\@doc@createspecialmacrolikeindexes
\def\doc@createspecialmacrolikeindexes#1#2#3{%
\@ tempteda{ (#2)}%\@ temptokenb{ #3:}%
\@nameedef{ Code#1 Index}##1##2{%\@bsphack
\noexpand\DTXD@findgtempa{##2}%
\@ temptokena : Index type.
\@ temptokenb : Index group.
\@ gtempa : Detokenized name of the particular object.
\@DTXD@gtempa@nobackslash : Detokenized name without backslash.
\noexpand\@bsphack
\noexpand\DTXD@findcategorylevelname% dtxdescribe
\noexpand\ifdoc@noindex\noexpand\else
\noexpand\@esphack}}

\@doc@createspecialmacrolikeindexes
\def\doc@createspecialmacrolikeindexes#1#2#3{%
\@ temptokena{ (#2)}%\@ temptokenb{ #3:}%
\@nameedef{ Code#1 Index}##1##2{%\@bsphack
\noexpand\DTXD@findgtempa{##2}%
\@ temptokena : Index type.
\@ temptokenb : Index group.
\@ gtempa : Detokenized name of the particular object.
\@DTXD@gtempa@nobackslash : Detokenized name without backslash.
\noexpand\@bsphack
\noexpand\DTXD@findcategorylevelname% dtxdescribe
\noexpand\ifdoc@noindex\noexpand\else
\noexpand\@esphack}}
If a top level object, index:
name=\verb!++\name+ (type) [category]|<main or usage>

\ifdoc@toplevel
\noexpand\special@index{% 
\noexpand\DTXD@printobjectname% dtxdescribe
(\noexpand\DTXD@gtempa@nobackslash)%
\noexpand\@gtempa%
\ifx\@nil#2\@nil\else \the\@temptokena \fi
\noexpand\DTXD@maybecategory% dtxdescribe
\noexpand\encapchar ##1%
}

Maybe add the category index entry:

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

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

\ifx\@nil#3\@nil\else
\noexpand\special@index{% 
\the\@temptokenb\noexpand\levelchar
dtxdescribe
\noexpand\DTXD@printobjectname% dtxdescribe
(\noexpand\DTXD@gtempa@nobackslash)%
\noexpand\@gtempa%
\noexpand\DTXD@maybecategory% dtxdescribe
\noexpand\encapchar ##1%
}

Maybe add the category index entry:

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

\@nameedef{SpecialMain#1Index}{main}{##1}

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

\@nameedef{SpecialMain#1Index}##1{% 
\expandafter\noexpand\csname Code#1Index\endcsname
\{main\}(##1})%

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

\@nameedef{Special#1Index}##1{% 
\noexpand\DTXD@findgtempa{##1}%
\@temptokena

\@temptokenb: Index type.
\@temptokena: 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:

doc loaded
\subsection{Not \texttt{doc: \DescribeMacro} and \texttt{\DescribeEnvironment}}

\DescribeMacro \[\langle \text{category}\rangle \] \{(\langle \text{name}\rangle)\}

Redefined to allow hyperlinked index entries and an optional category:

\begin{verbatim}
\IfPackageLoadedTF{doc}{}{% not doc
  \providecommand*{\DescribeMacro}{}
  \renewcommand*{\DescribeMacro}[2][2]{%\@bsphack
    \@ifundefined{@captype}{% not float?
      \leavevmode
      \marginpar{\{\hbadness=10000\%
        \hfuzz=5em\%
        \raggedleft\%
        \ifblank{#1}{}{{\scriptsize\textsf{[#1]}}}% category
        \cmd{#2}% name
      \}}% marginpar
    }{}% not float?
  \ifundefined{@captype}% not float?
    \leavevmode
    \marginpar{\{\hbadness=10000\%
      \hfuzz=5em\%
      \raggedleft\%
      \ifblank{#1}{}{{\scriptsize\textsf{[#1]}}}% category
      \cmd{#2}% name
    }% marginpar
  }{% category given
    \ifblank{#1}{}% no category
    % Again, and prepend the category:
    % Write category=[\text{category}]:>name=csname\verb+|usage+
    \begingroup%
    \DTXD@origwrindex{\#1\actualchar\DTXD@verbatimcmd{#2}%
      \DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2}%
    }% \DTXD@origwrindex
    \DTXD@origwrindex#1\actualchar\#1\actualchar\levelchar%
    \DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2}%
  }% category given
\end{verbatim}

Create the margin tag with the macro's name:

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>|\text{category}|\verb+|usage+

Only if a category was given:
The name may have underscores.

\DTXD@filename\{\detokenize{#2}\}\% Create a detokenized version of the filename...

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

\StrSubstitute{\DTXD@filename}\{\detokenize{\_}\}\{\detokenize{\_}\}\{\DTXD@filename\}\% The original filename is printed in the margin. Any underscore characters have already been disabled by the \catcode change.

\DTXD@margintag{#1}{#2}{#3}\% The detokenized and sanitized version is sent to the index file:
End the group with the disabled underscore, and clean up the extra space from the `\catcode` command:

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

\DescribeMacro
Redefine with new definitions.

\DescribeEnvironment
Redefine with new definitions.

\DescribeFile\{⟨name⟩\}

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

\DescribeFile\{⟨name⟩\}

The underscore character is temporarily disabled, then the name is passed directly to `\DTXD@DescribeFile`.
\DescribeProgram {⟨name⟩}

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

\DescribeCommand {⟨name⟩}

The underscore character is temporarily disabled, then the name is passed directly to \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}{%\overline{\begin{tabular}{l}
\DTXD@filemarginparindex{#1}{#2}{Pkg}{package}{usage}
\end{tabular}}%
\endgroup\vspace{-1pt}
}\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}{%\overline{\begin{tabular}{l}
\DTXD@filemarginparindex{#1}{#2}{Cls}{class}{usage}
\end{tabular}}%
\endgroup\vspace{-1pt}
The \texttt{\DescribeOption} macro can be used to describe options. It may be used to categorize options by their macro or environment name.

\verbatim
\DescribeOption [{\texttt{\langle category\rangle}}] {\langle name\rangle}
\endverbatim
\newdocelement[
  macrolike=false,
  toplevel=true,
  idxtype=boolean,
  idxgroup=Booleans,
  printtype=\textit{bool}]
{Boolean}{boolenv}

\newcommand*{\DescribeBoolean}{\DTXD@cindex{#1}{#2}{Bool}{boolean}{usage}}

\newcommand*{\DescribeLength}{\DTXD@cindex{#1}{#2}{Len}{length}{usage}}

\newcommand*{\DescribeCounter}{\DTXD@cindex{#1}{#2}{Ctr}{counter}{usage}}
\DescribeHook [⟨category⟩] {⟨name⟩}

\IfPackageLoadedTF{doc}{% doc
\NewDocElement[
  macrolike=true,
  toplevel=true,
  idxtype=hook,
  idxgroup=Hooks,
  printtype=\textit{Hook}
]{Hook}{hook}
\} {% not doc
\newcommand*{\DescribeHook}[2][]
{\DTXD@margintagindex{#1}{#2}{Hook}{hook}{usage}}
}% not doc

\DescribeKey [⟨category⟩] {⟨name⟩}

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

\IfPackageLoadedTF{doc}{% doc
\NewDocElement[
  macrolike=false,
  toplevel=true,
  idxtype=key,
  idxgroup=Keys,
  printtype=\textit{Key}
]{Key}{key}
\} {% not doc
\newcommand*{\DescribeKey}[2][]
{\DTXD@margintagindex{#1}{#2}{Key}{key}{usage}}
}% not doc

\DescribeObject [⟨category⟩] {⟨name⟩}

May be used to describe an arbitrary piece of code. Creates a margin tag and index entries with \texttt{family}. 

\IfPackageLoadedTF{doc}{% doc
\NewDocElement[
  macrolike=false,
  toplevel=true,
  idxtype=object,
  idxgroup=Objects,
  printtype=\texttt{family}
]{Object}{object}
\} {% not doc
\newcommand*{\DescribeObject}[2][]
{\DTXD@margintagindex{#1}{#2}{Object}{object}{usage}}
}% not doc
\DescribeOther \[⟨\textit{category}⟩\] \{⟨\textit{name}⟩\}

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

\IfPackageLoadedTF{doc}{% doc
\NewDocElement\[
\texttt{macrolike}=false,\n\texttt{toplevel}=true,\n\]...\}
\newcommand\DescribeDefaultcolor{green!50!black}
\DescribeDefault{⟨value⟩}

Creates a colored margin tag showing the boolean default value.
\itemDescribeMacro \[\langle category\rangle\] \{\langle name\rangle\}
\ItemDescribeEnv \[\langle category\rangle\] \{\langle name\rangle\}
\ItemDescribeArgument \[\langle category\rangle\] \{\langle argument\rangle\}
\ItemDescribeBoolean \[\langle category\rangle\] \{\langle name\rangle\}
\ItemDescribeLength \[\langle category\rangle\] \{\langle name\rangle\}
\ItemDescribeCounter [\langle category \rangle] [\langle name \rangle]

\newcommand{\ItemDescribeCounter}[2][% 
  \itemtexttt{#2}:]%
\setlength{\parskip}{1.5ex}
\DescribeCounter[#1]{#2}

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

\newcommand{\ItemDescribeHook}[2][% 
  \itemtexttt{#2}:]%
\setlength{\parskip}{1.5ex}
\DescribeHook[#1]{#2}

\ItemDescribeKey [\langle category \rangle] [\langle name \rangle]

\newcommand{\ItemDescribeKey}[2][% 
  \itemtexttt{#2}:]%
\setlength{\parskip}{1.5ex}
\DescribeKey[#1]{#2}

\ItemDescribePackage [\langle category \rangle] [\langle name \rangle]

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

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

\ItemDescribeClass [\langle category \rangle] [\langle name \rangle]

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

\newcommand{\ItemDescribeClass}{% 
  \begingroup\catcode\_=12 \DTXD@ItemDescribeClass%
}
\ItemDescribeOption \langle category \rangle \langle name \rangle

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

\ItemDescribeFile \langle category \rangle \langle name \rangle

\item \texttt{#2}:
\setlength{\parskip}{1.5ex}
\DescribeFile[#1]{#2}
\endgroup

\ItemDescribeProgram \langle category \rangle \langle name \rangle

\item \texttt{#2}:
\setlength{\parskip}{1.5ex}
\DescribeProgram[#1]{#2}
\endgroup

\ItemDescribeCommand \langle category \rangle \langle name \rangle

\item \texttt{#2}:
\setlength{\parskip}{1.5ex}
\DescribeCommand[#1]{#2}
\endgroup

\ItemDescribeObject \langle category \rangle \langle name \rangle

\item \texttt{#2}:
\ItemDescribeOther [\langle category\rangle] \{(\langle name\rangle)\}

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

6.17 \texttt{\textbackslash margintag, \textbackslash watchout}

\texttt{\textbackslash margintagcolor} The color of the \texttt{\textbackslash margintag}.

\newcommand*=\texttt{\textbackslash margintagcolor}\{blue!70!black\}

\texttt{\textbackslash margintag} \{\langle text\rangle\}

Prints a colored margin tag.

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

\texttt{\textbackslash watchoutcolor} The color of the \texttt{\textbackslash watchout}.

\newcommand*=\texttt{\textbackslash watchoutcolor}\{red!50!black\}

\texttt{\textbackslash watchout} \{\langle 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\text{container}\rangle\} \{\langle\text{contents}\rangle\}

6.19 The dtxexample environment

Also see example 15 on page 28.

dtxexample\_cut.tex (file) Used to store the \input example code.

DTX0@examplerulecolor [color] The color of the middle rule in the dtxexample.

\definecolor{DTX0@examplerulecolor}{rgb}{.9,.9,.9}

\dtxexamplecodename The text name of the code section.

\newcommand*{\dtxexamplecodename}{Code:}

\dtxexampleresultname The text name of the result section.

\newcommand*{\dtxexampleresultname}{Result:}

dtxexample (env.) \{\langle\text{caption \& \label}\rangle\}

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

\NewDocumentEnvironment{dtxexample}{s +O{} m}
{\% start dtxexample
Copy the environment's contents to the file dtxexample_cut.tex:
\VerbatimOut[gobble=\DTXD@gobble,tabsize=4]{dtxexample_cut.tex}%
\% start dtxexample
When the environment closes:
\% end dtxexample
\VerbatimOut% Finish the verbatim output:
\par\addvspace{\bigskipamount}
\IfBooleanTF{#1}{}% If unstarred, typeset the example in a minipage, else use a float:
{\% minipage
\minipage{\linewidth}%
\captionsetup{type=dtxexamplefloat}%
})%
{\% float
\begin{dtxexamplefloat}%
\hrule\medskip
\caption{#3}
Typeset the contents as verbatim:
\textcolor{DTXD@examplerulecolor}{\smallskip\hrule}
\smallskip
{\scriptsize\itshape\dtxexamplecodename}
\VerbatimInput{dtxexample_cut.tex}%
\unskip
\textcolor{DTXD@examplerulecolor}{\hrule}
\smallskip
{\scriptsize\itshape\dtxexampleresultname}
\ifstrempty{#2}{% Possible add the optional cross-references or notes:
{%}
If unstarred, close the float or \texttt{\minipage}.

\begin{dtxexample}
\IfBooleanTF{#1}{}
\{\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]

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

Name for \texttt{cleveref}. \texttt{\crefname} here is required for documents not using the \texttt{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.

\begin{verbatim}
noindmacro {env} {\langle name\rangle}
\end{verbatim}

\begin{verbatim}
\newenvironment{noindmacro}{\langle name\rangle}
\newenvironment{noindenvironment}{\langle name\rangle}
\end{verbatim}

6.21 sourcedisplay, UIDisplay, docsidenavbar

For use in a sourcedisplay:

\begin{verbatim}
\fquad Forces a quad indent.
\end{verbatim}

\begin{verbatim}
\fquad Forces a double-quad indent.
\end{verbatim}

\begin{verbatim}
\fqqquad Forces a triple-quad indent.
\end{verbatim}

sourceverb {env} To typeset a block of source code, verbatim.

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

fsourceverb {env} To typeset a framed block of source code, verbatim.
sourcedisplay (env.) To typeset a block of source code, allowing direct formatting.

\newenvironment{sourcedisplay}
\{
\leavevmode
\par
\mathpage\minipage{\linewidth-4em}
\ttfamily
\}
\{
\endminipage
\par
\}

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

\newenvironment{UIdisplay}
\{
\leavevmode
\par
\mathpage\minipage{\linewidth-4em}
\sffamily\bfseries
\}
\{
\endminipage
\par
\}

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

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

\userentry \{⟨text to enter⟩\}

Typesets text to be entered by the users.

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

docs sidebar (env.) To typeset a sidebar in the documentation.
6.22 Formatted objects

Macros to format references to various kinds of objects.

6.22.1 \LaTeX objects

\pkg{⟨name⟩}  Also usable for class names.

\cs{⟨csname⟩}  From \texttt{ltxdoc}.

\env{⟨name⟩}

\marg{⟨argument⟩}  From \texttt{ltxdoc}.

\oarg{⟨argument⟩}  From \texttt{ltxdoc}.

\parg{⟨argument⟩}  From \texttt{ltxdoc}.
\counter{⟨name⟩}
\bool{⟨name⟩}
\optional{⟨name⟩}
\tableofcontents
\listoffigures
\listoftables

6.22.2 Programs and commands

\commands{⟨commands to print⟩} No processing is provided for special characters.
\progcodelisting{⟨code to print⟩} No processing is provided for special characters.
\program{⟨program name⟩} Underscores are allowed.
\filenm \{file name\}  Underscores are allowed.

1290 \newcommand*{\DTXD@filenm}[1]{
1291 \mbox{\texttt{\detokenize{#1}}}\\
1292 \endgroup
1293 }
1294
1295 \providernobustcmd{\filenm}[%}
1296 \begingroup%
1297 \catcode\_=12%
1298 \DTXD@filenm%
1299 %}

\UI General user-interface text.

1300 \providernobustcmd{\UI}[1]{\textbf{\textsf{#1}}}

6.22.3 File types

\ODT

1301 \providernobustcmd{\ODT}{\acro{ODT}}

\SVG

1302 \providernobustcmd{\SVG}{\acro{SVG}}

\PNG

1303 \providernobustcmd{\PNG}{\acro{PNG}}

\GIF

1304 \providernobustcmd{\GIF}{\acro{GIF}}

\JPG

1305 \providernobustcmd{\JPG}{\acro{JPG}}

\EPS

1306 \providernobustcmd{\EPS}{\acro{EPS}}

\PDF

1307 \providernobustcmd{\PDF}{\acro{PDF}}
6.22.4 Internet

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

\texttt{TikZ}

\texttt{CTAN}

\texttt{TDS}

\texttt{MathML}

\texttt{MathJax}

6.22.6 Acronyms, brand names, trademarks

\texttt{brand }\{\langle name\rangle\}

\texttt{acro }\{\langle acronym\rangle\}

\supregistered Superscript trademark symbol.

6.23 Logos

\texttt{dviTeX DVI TeX}

\texttt{dviLaTeX DVI LaTeX}
\ConTeXt Con\TeX t

(providerobustcmd*{\ConTeXt}{\mbox{\TeX}})

\MiKTeX MiK\TeX

(providerobustcmd*{\MiKTeX}{\mbox{\TeX}})

6.24 Dashes and slashes

\thinskip A breakable thin skip.

(DeclareRobustCommand{\thinskip}{\hskip 0.16667em\relax})

\endash An endash: –

(\def\endash{–})

\emdash An emdash: —

(\def\emdash{—})

\thinbrspace A thin space which allows a line break.

(newcommand{\thinbrspace}{%\hspace{.16667em}\penalty\exhyphenpenalty\hspace{0pt}%})

\thinthinbrspace A thin space which allows a line break.

(newcommand{\thinthinbrspace}{%\hspace{.08333em}\penalty\exhyphenpenalty\hspace{0pt}%})

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

(newrobustcmd{\Dash}{\unskip\thinspace\emdash\thinbrspace})

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

(newrobustcmd{\dash}{\unskip\thinspace\endash\thinbrspace})

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

(newrobustcmd{\Slash}{\unskip\hspace{.08333em}/\thinthinbrspace})
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)
# Change History and Index

## Change History

<table>
<thead>
<tr>
<th>Version</th>
<th>General</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>v0.10</strong></td>
<td>2016/12/08</td>
<td>Initial version</td>
</tr>
<tr>
<td><strong>v0.11</strong></td>
<td>2018/03/30</td>
<td>\DTXD@cmdmargintagindex: Index tag no longer plural.</td>
</tr>
<tr>
<td><strong>v1.00</strong></td>
<td>2019/01/11</td>
<td>Added formatted objects. Added logos. Cut file name changed to dtxexample_cut.tex.</td>
</tr>
<tr>
<td><strong>v1.01</strong></td>
<td>2019/03/22</td>
<td>Sanitize PDF bookmarks.</td>
</tr>
<tr>
<td><strong>v1.02</strong></td>
<td>2019/07/16</td>
<td>Fix if not doc package. Fix if not hyperref package. Fix if not ltxdoc class.</td>
</tr>
</tbody>
</table>

## Index

- ItemDescribeLength: Added. 63
- ItemDescribeMacro: Added. 63
- ItemDescribeObject: Added. 65
- ItemDescribeOption: Added. 65
- ItemDescribeOther: Added. 66
- ItemDescribePackage: Added. 64
- ItemDescribeProgram: Added. 65
- \margintag: Uses \margintagcolor. 66
- \margintagcolor: Added. 66
- noindenvironment: Added. 70
- noindmacro: Added. 70
- sourcedisplay: Added. 71
- sourceverb: Added. 70
- \userentry: Added. 71
- \userentryname: Added. 71
- \watchoutcolor: Added. 66
- \watchoutcolor: Uses \watchoutcolor. 66
- \watchout: Changed to \raggedleft. 66
- \DescribeClass: Fix: Allow underscore. 57
- \DescribeDefault: Added. 62
- \DescribeDefaultcolor: Added. 62
- \DescribeMacro: Put margin tag category in brackets. 53
- \DescribePackage: Fix: Allow underscore. 57
doc-sidebar: Added. 71
\DTXD@cmdmargintagindex: Sans tag font. 42
\DTXD@filenameparindex: Fix: File category. 54
\DTXD@printtype: Sans tag font. 39
\dtxexamplecodename: Added. 67
\dtxexampleresultname: Added. 67
\fqquad: Added. 70
\fquad: Added. 70
\fsourcverb: Added. 70
\ItemDescribeArgument: Added. 63
\ItemDescribeBoolean: Added. 63
\ItemDescribeClass: Added. 64
\ItemDescribeCommand: Added. 65
\ItemDescribeCounter: Added. 64
\ItemDescribeEnv: Added. 63
\ItemDescribeFile: Added. 65
\ItemDescribeKey: Added. 64
\ItemDescribeLength: Added. 63
\ItemDescribeMacro: Added. 63
\ItemDescribeObject: Added. 65
\ItemDescribeOption: Added. 65
\ItemDescribeOther: Added. 66
\ItemDescribePackage: Added. 64
\ItemDescribeProgram: Added. 65
\margintag: Uses \margintagcolor. 66
\margintagcolor: Added. 66
noindenvironment: Added. 70
noindmacro: Added. 70
sourcedisplay: Added. 71
sourceverb: Added. 70
\userentry: Added. 71
\userentryname: Added. 71
\watchoutcolor: Added. 66
\watchoutcolor: Uses \watchoutcolor. 66
\watchout: Changed to \raggedleft. 66

Index

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

<table>
<thead>
<tr>
<th>Symbols</th>
<th>[M] [figure]</th>
<th>[H] (argument) [figure]</th>
<th>[M] (argument) [figure]</th>
<th>[oarg]</th>
<th>\attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>@NewDocElement</td>
<td>364</td>
<td>bold [\mymacro]</td>
<td>24</td>
<td>attrib</td>
<td>16, 1312</td>
</tr>
<tr>
<td>[H] [argument] [figure]</td>
<td>24</td>
<td>attributable</td>
<td>16, 1312</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[M] [argument] [figure]</td>
<td>24</td>
<td>attrib</td>
<td>16, 1312</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\acro</td>
<td>17, 1326</td>
<td>Before_myenvironment (other)</td>
<td>25</td>
<td>\BibTeX</td>
<td>17, 1351</td>
</tr>
<tr>
<td>Additional_Item (other) [othercategory]</td>
<td>25</td>
<td>another_big_file.txt (file)</td>
<td>22</td>
<td>[bigfiles]:</td>
<td>\another_big_file.txt (file)</td>
</tr>
<tr>
<td>\AmS</td>
<td>17, 1346</td>
<td>another_big_file.txt (file)</td>
<td>22</td>
<td>\bold</td>
<td>[\mymacro]</td>
</tr>
<tr>
<td>another_big_file.txt (file) [bigfiles]</td>
<td>22</td>
<td>really_big_file.txt (file)</td>
<td>22</td>
<td>\bool</td>
<td>15, 1273</td>
</tr>
<tr>
<td>Arguments:</td>
<td>24</td>
<td>\bold (argument)</td>
<td>[\mymacro]</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>[H] [figure]</td>
<td>24</td>
<td>\bold</td>
<td>[\mymacro]</td>
<td>24</td>
<td></td>
</tr>
</tbody>
</table>

v1.03
General: 2022/02/01 1
No longer requires xifthen. 33
\attrib: For LuaTEX. 75
\cs: Provided. 72
\DescribeHook: Added. 60
\DescribeMacro: Reduce hbox warnings. 53
\DescribeObject: Reduce hbox warnings. 60
\DescribeOther: Reduce hbox warnings. 61
\DTX@cmdmargintagindex: Reduce hbox warnings. 42
\DTXD@gobble: Fix if not \doc package. 37
\DTXD@findgtempa: Detokenized names. 47
\DTXD@maybecategory: \verb*. 46
\DTXD@maybecategorylevel: \verb*. 46
\ItemDescribeHook: Added. 64
\marg: Provided. 72
\MathJax: Added. 76
\oarg: Provided. 72
\parg: Provided. 72
\pdfLaTeX: Added. 77
\pdfTeX: Added. 77
\shownesting: Added. 67

v1.04
General: 2022/02/01 1
Corrected copyright date. 1

v1.05
General: 2022/09/08 1
\TikZ: Renamed from \tikz, cap Z. 76

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
\doc@createspecialmacrolikeindexes: Added category names for macros. 50
\DTX@findtempa: Detokenized names. 47
\DTXD@maybecategory: \verb*. 46
Improved line breaks. 46
\DTXD@maybecategorylevel: \verb*. 46
Fix occasional crash. 47
\DTXDbreak: Improved line breaks. 45
booleanname (boolean) [descexamples] 26
Booleans:
  booleanname [descexamples] ... 26
  sampleboolean [examples] ... 21
\brand ........................................ 17,1325

C
calc (package) .................................. 33
caption (package) ............................... 33
\captionsetup [dtxexamplefloat] .......... 69
class_name (class) [descexamples] ... 26
Classes:
  class_name [descexamples] ... 26
  sample_class [examples] ... 22
\cmds ........................................ 16,1278
[color]:
  DTXD@examplerulecolor (object) .. 67
  othercolor (object) ................. 25
  somecolor (object) ................. 25
  command_name (command) [descexamples] ........................................ 26
Commands:
  command_name [descexamples] ... 26
  OS_command ................................. 22
\ConText ....................................... 18,1353
Counters:
  samplecounter [examples] ... 21
\crefname [photograph] .......... 19
\cphotonameref [photograph] .......... 19
\cs ......................................... 15,1264
\CSS ......................................... 16,1317
\CSSthree .................................. 16,1318
\CTAN ........................................ 17,1321
\ctr ......................................... 15,1272

D
\Dash ........................................ 18,1364
\dash ........................................ 18,1365
\DeclareFloatingEnvironment [dtxexamplefloat] ........................................ 69
\DeclareFloatingPhoto [photograph] ........................................ 19
[descexamples]:
  booleanname (boolean) ................. 26
  class_name (class) ....................... 26
  command_name (command) ............... 26
  file_name (file) .......................... 26
  keyname (key) ............................. 26
  lengthname (length) ..................... 26
  macroname .................................. 26
  package_name (package) ............... 26
  program_name (program) ............... 26
\DescribeArgument ............................ 10,805
\DescribeBoolean ............................. 10,821
\DescribeClass .............................. 10,768
\DescribeCommand ............................ 11,725
\DescribeCounter ............................ 10,853
\DescribeDefault .................................. 12,991
\DescribeDefaultheight .................................. 12,990
\DescribeEnv .................................. 9,649
\DescribeEnvironment ............................ 668
\DescribeFile .................................. 11,683
\DescribeHook .................................. 10,869
\DescribeKey .................................. 10,885
\DescribeLength .................................. 10,837
\DescribeMacro .................................. 9,608,668
\DescribeObject .................................. 11,901
\DescribeOption .................................. 10,789
\DescribeOther .................................. 11,950
\DescribePackage .................................. 10,746
\DescribeProgram .................................. 11,704
\doc@createspecialindexes .................. 471
\doc@createspecialmacrolikeindexes ........ 541
docsiderbar (env) ........................... 1250
docsiderbar (env) ........................... 15
\DTXD@categorylevelname .......................... 435
\DTXD@cmdmarginindex .......................... 294
\DTXD@examplerulecolor (object) [color] ........................................ 67
\DTXD@findcategorylevelname .................. 456
\DTXD@findtemplatename ....................... 456
\DTXD@gobble .................................. 464
\DTXD@gobble .................................. 160
\DTXD@gtempa@nobackslash .................... 462
\DTXD@index .................................. 257
\DTXD@macroname .................................. 288
\DTXD@margintag .................................. 241
\DTXD@margintagindex .......................... 243
\DTXD@maybecategory ........................... 422
\DTXD@maybecategorylevelname ................. 436
\DTXD@origwindex .................................. 240
\DTXD@printindex .................................. 413
\DTXD@printtype .................................. 232
\DTXD@verbatimcmd .................................. 291
\DTXD@verbatimcmd .................................. 412
dtxexample (env) .................................. 1139
dtxexample (env) .................................. 13
dtxexample_cut (file) ......................... 67
dtxexamplecodename ............................ 13,1137
[dtxexamplefloat]:
  \captionsetup .................................. 69
  \crefname .................................. 69
  \DeclareFloatingEnvironment .................. 69
  \dtxexampleresultname ....................... 13,1138
\DVI ........................................ 16,1308
\dvilatex .................................... 17,1329
\dviteX ...................................... 17,1328

82
E
\element ....... 16, 1311
\emdash ............ 18, 1357
\endash ............ 18, 1356
\env ............ 15, 1265
environment (env.) ............ 9

environments:
docsidebar ............ 1250
docsidebar ............ 15
dtxexample ............ 1139
dtxexample ............ 13
environment ............ 9
dsourceverb ............ 1216
dsourceverb ............ 14
macro ............ 9
myenvironment ............ 20
noindentenvironment ............ 1204
noindentenvironment ............ 14
noindentmacro ............ 1198
noindentmacro ............ 14
otherenvironment ............ 20
sourcedisplay ............ 1219
sourcedisplay ............ 14
dsourceverb ............ 1213
dsourceverb ............ 14
display ............ 1230
display ............ 15
\EPS ............ 16, 1306
\EPUB ............ 17, 1319
toolbox (package) ............ 33
[examples]:
sample_class (class) ............ 22
sampleboolean (boolean) ............ 21
samplecounter (counter) ............ 21
samplekey (key) ............ 23
sampleoption (option) ............ 22
samplepackage (package) ............ 22
dsampletwokey (key) ............ 23

F
fancyverb (package) ............ 33
[figure]:
[H] (argument) ............ 24
[M] (argument) ............ 24
file_name (file) [descexamples] ............ 26
\f ilem ............ 16, 1290
Files:
another_big_file.txt [bigfiles] ............ 22
dtxexample_cut.txt ............ 67
file_name [descexamples] ............ 26
lone_file.txt ............ 22
really_big_file.txt [bigfiles] ............ 22
firstkey (key) [groupofkeys] ............ 23
\f qquad ............ 15, 1212

G
\g f ..... 16, 1304

H
\h ookname (hook) ............ 21
Hooks:
\hookname ............ 21
para/begin ............ 21
\HTML ............ 16, 1315
\HTMLfive ............ 16, 1316
hyperref (package) ............ 33
I
index
by group ............ 20
\IndexMin (length) ............ 38
\ItemDescribeArgument ............ 11, 1009
\ItemDescribeBoolean ............ 12, 1014
\ItemDescribeClass ............ 12, 1049
\ItemDescribeCommand ............ 12, 1084
\ItemDescribeCounter ............ 12, 1024
\ItemDescribeEnv ............ 11, 1004
\ItemDescribeFile ............ 12, 1064
\ItemDescribeHook ............ 12, 1029
\ItemDescribeKey ............ 12, 1034
\ItemDescribeLength ............ 12, 1019
\ItemDescribeMacro ............ 11, 999
\ItemDescribeObject ............ 12, 1094
\ItemDescribeOption ............ 12, 1059
\ItemDescribeOther ............ 12, 1099
\ItemDescribePackage ............ 12, 1039
\ItemDescribeProgram ............ 12, 1074

J
\JPG ............ 16, 1305

K
keyname (key) [descexamples] ............ 26

Keys:
firstkey [groupofkeys] ............ 23
keyname [descexamples] ............ 26
lonekey ............ 23
samplekey [examples] ............ 23
dsampletwokey [examples] ............ 23
secondkey [groupofkeys] ............ 23

83
\textbf{L}
\begin{itemize}
\item \texttt{\textbackslash lengthname (length)} [descexamples] \textbullet \text{26}
\item \textbf{Lengths}:
\begin{itemize}
\item \texttt{\textbackslash IndexMin} \textbullet \text{38}
\item \texttt{\textbackslash lengthname [descexamples]} \textbullet \text{26}
\item \texttt{\textbackslash photowidth [photograph]} \textbullet \text{21}
\end{itemize}
\item \texttt{license\_agreement (other)} \textbullet \text{25}
\item \texttt{listoftdexamplefloats} \textbullet \text{13}
\item \texttt{\textbackslash LOF} \textbullet \text{15, 1276}
\item \texttt{\textbackslash lone\_file.txt (file)} \textbullet \text{22}
\item \texttt{\textbackslash lonekey (key)} \textbullet \text{23}
\item \texttt{\textbackslash LOT} \textbullet \text{15, 1277}
\item \texttt{\textbackslash LuaLaTeX} \textbullet \text{17, 1333}
\item \texttt{\textbackslash LuaTeX} \textbullet \text{17, 1332}
\item \texttt{\textbackslash LyX} \textbullet \text{17, 1350}
\end{itemize}
\textbf{M}
\begin{itemize}
\item \texttt{macro (env.)} \textbullet \text{9}
\item \texttt{macroname [descexamples]} \textbullet \text{26}
\item \texttt{makeidx (package)} \textbullet \text{33}
\item \texttt{\textbackslash MakeIndex} \textbullet \text{18, 1352}
\item \texttt{\textbackslash marg} \textbullet \text{9, 15, 1296}
\item \texttt{margin\_tag\_missing} \textbullet \text{32}
\item \texttt{\textbackslash margintag} \textbullet \text{13, 1105}
\item \texttt{\textbackslash margintagcolor} \textbullet \text{13, 1104}
\item \texttt{\textbackslash MathJax} \textbullet \text{17, 1324}
\item \texttt{\textbackslash MathML} \textbullet \text{17, 1323}
\item \texttt{\textbackslash MikTeX} \textbullet \text{18, 1354}
\item \texttt{\textbackslash myenvironment (env.)} \textbullet \text{20}
\item \texttt{[\textbackslash mymacro]}:
\begin{itemize}
\item \texttt{\textbackslash bold (argument)} \textbullet \text{24}
\item \texttt{\textbackslash mymacro} \textbullet \text{19}
\end{itemize}
\item \texttt{newfloat (package)} \textbullet \text{33}
\item \texttt{noindenvironment (env.)} \textbullet \text{1204}
\item \texttt{noindenvironment (env.)} \textbullet \text{14}
\item \texttt{noindmacro (env.)} \textbullet \text{1198}
\item \texttt{noindmacro (env.)} \textbullet \text{14}
\end{itemize}
\textbf{N}
\begin{itemize}
\item \texttt{\textbackslash arg} \textbullet \text{9, 15, 1268}
\item \textbf{Objects}:
\begin{itemize}
\item \texttt{DTX\@examplerulecolor [color]} \textbullet \text{67}
\item \texttt{othercolor [color]} \textbullet \text{25}
\item \texttt{randomobject} \textbullet \text{25}
\item \texttt{somecolor [color]} \textbullet \text{25}
\end{itemize}
\item \texttt{\textbackslash DDT} \textbullet \text{16, 1301}
\item \textbf{Options}:
\begin{itemize}
\item \texttt{sampleoption [examples]} \textbullet \text{22}
\item \texttt{\textbackslash optn} \textbullet \text{15, 1274}
\item \texttt{OS\_command [command]} \textbullet \text{22}
\item \texttt{Other\_Item [other] [othercategory]} \textbullet \text{25}
\end{itemize}
\item \texttt{Other}:
\begin{itemize}
\item \texttt{Additional\_Item [othercategory]} \textbullet \text{25}
\end{itemize}
\end{itemize}
\textbf{O}
\begin{itemize}
\item \texttt{\textbackslash oarg} \textbullet \text{9, 15, 1268}
\item \textbf{Objects}:
\begin{itemize}
\item \texttt{DTX\@examplerulecolor [color]} \textbullet \text{67}
\item \texttt{othercolor [color]} \textbullet \text{25}
\item \texttt{randomobject} \textbullet \text{25}
\item \texttt{somecolor [color]} \textbullet \text{25}
\end{itemize}
\item \texttt{\textbackslash DDT} \textbullet \text{16, 1301}
\item \textbf{Options}:
\begin{itemize}
\item \texttt{sampleoption [examples]} \textbullet \text{22}
\item \texttt{\textbackslash optn} \textbullet \text{15, 1274}
\item \texttt{OS\_command [command]} \textbullet \text{22}
\item \texttt{Other\_Item [other] [othercategory]} \textbullet \text{25}
\end{itemize}
\item \texttt{Other}:
\begin{itemize}
\item \texttt{Additional\_Item [othercategory]} \textbullet \text{25}
\end{itemize}
\end{itemize}
\textbf{P}
\begin{itemize}
\item \texttt{\textbackslash package\_name (package)} [descexamples] \textbullet \text{26}
\item \textbf{Packages}:
\begin{itemize}
\item \texttt{calc} \textbullet \text{33}
\item \texttt{caption} \textbullet \text{33}
\item \texttt{etoolbox} \textbullet \text{33}
\item \texttt{fancyvrb} \textbullet \text{33}
\item \texttt{hyperref} \textbullet \text{33}
\item \texttt{makeidx} \textbullet \text{33}
\item \texttt{newfloat} \textbullet \text{33}
\item \texttt{package\_name [descexamples]} \textbullet \text{26}
\item \texttt{pict\_2e} \textbullet \text{35}
\item \texttt{samplepackage [examples]} \textbullet \text{22}
\item \texttt{splitidx} \textbullet \text{33}
\item \texttt{xcolor} \textbullet \text{33}
\item \texttt{xparse} \textbullet \text{33}
\item \texttt{xstring} \textbullet \text{33}
\end{itemize}
\item \texttt{\textbackslash parain\_begin (hook)} \textbullet \text{21}
\item \texttt{\textbackslash parg} \textbullet \text{9, 15, 1270}
\item \texttt{\textbackslash PDF} \textbullet \text{16, 1307}
\item \texttt{\textbackslash pdfLaTeX} \textbullet \text{17, 1331}
\item \texttt{\textbackslash pdfTeX} \textbullet \text{17, 1330}
\item \texttt{\textbackslash photocaptionsetup [photograph]} \textbullet \text{19}
\item \texttt{\textbackslash photonameref} \textbullet \text{19}
\item \texttt{\textbackslash DeclarefloatingPhoto} \textbullet \text{19}
\item \texttt{\textbackslash photocaptionsetup} \textbullet \text{19}
\item \texttt{\textbackslash photowidth (length) [photograph]} \textbullet \text{21}
\item \texttt{\textbackslash pict\_2e (package)} \textbullet \text{35}
\item \texttt{\textbackslash pkg} \textbullet \text{15, 1263}
\item \texttt{\textbackslash PNG} \textbullet \text{16, 1303}
\item \texttt{\textbackslash PrintEnvName} \textbullet \text{229}
\item \texttt{\textbackslash prog} \textbullet \text{16, 1280}
\item \texttt{\textbackslash procode} \textbullet \text{15, 1279}
\item \texttt{\textbackslash program\_name (program)} \textbullet \text{22}
\item \texttt{\textbackslash program\_name (program) [descexamples]} \textbullet \text{26}
\end{itemize}
\textbf{R}
\begin{itemize}
\item \texttt{\textbackslash randomobject (object)} \textbullet \text{25}
\item \texttt{\textbackslash \texttt{	extbackslash really\_big\_file.txt (file) [bigfiles]} \textbullet \text{22}}
\end{itemize}
S

sample_class (class) [examples] .... 22
sampleboolean (boolean) [examples] ... 21
samplecounter (counter) [examples] ... 21
samplekey (key) [examples] .......... 23
sampleoption (option) [examples] .... 22
samplepackage (package) [examples] ... 22
secondkey (key) [groupofkeys] ..... 23
\shownesting .................. 12, 1121
\slash ........................ 18, 1366
somecolor (object) [color] ......... 25
sourcedisplay (env) .............. 1219
sourcedisplay (env) .......... 14
sourceverb (env) .............. 1213
sourceverb (env) .......... 14
splitidx (package) .......... 33
\supregistered ................. 17, 1327
\SVG ........................ 16, 1302

T

\TDS ........................ 17, 1322
\thinspace ..................... 18, 1358
\thinskip ...................... 18, 1355
\thinnbrspace .................. 18, 1361
\TikZ .......................... 17, 1320
\tableofcontents ............... 15, 1275
\U

\UI ............................. 16, 1300
\UIdisplay (env) .............. 1230
\UIdisplay (env) .............. 15
\url ................................ 16, 1310
\usage ........................... 234
\userentry ..................... 15, 1242
\userentryname ................ 15, 1241
\UTF .......................... 16, 1309
\W

\warningsign .................... 26
\watchout ....................... 13, 1112
\watchoutcolor .................. 13, 1111
\X

xcolor (package) ............... 33
\XeLaTeX ....................... 17, 1334
\XeTeX ........................ 17, 1334
xparse (package) ............... 33
xstring (package) .............. 33