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

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 \texttt{\textbackslash DescribeEnv}, and is listed in the index by itself and also by category. Each item may be sorted further by an optional class. All index entries except code lines are hyperlinked.

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

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

Macros are provided for formatting the names of inline \LaTeX{} objects such as packages and booleans, as well as program and file names, file types, internet objects, the names of certain programs, a number of logos, and inline dashes and slashes. \texttt{dtxdescribe} works with the \texttt{ltxdoc} class, but also works with the standard classes as well, except that the \texttt{macro} and \texttt{environment} environments are not supported. Either \texttt{makeidx} or \texttt{splitidx} may be loaded by the user. \texttt{makeidx} will be used by default.

\texttt{dtxdescribe} works with \texttt{pdftex}, \texttt{xetex}, and \texttt{luatex}, and perhaps other engines as well.

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

Also, see change in setup of \texttt{hyperref} and \texttt{cleveref}: Section 2 on page 7.
## Contents

1 Introduction .................................................. 6

2 Using dt.xdescribe ........................................... 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 noindentmacro and noindentenvironment environments .. 13
  3.13 sourceverb, sourcedisplay, UIDisplay, docsidenav ....... 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 ................................... 16
    3.14.6 Acronyms, brand names, trademarks .................. 17
  3.15 Logos .................................................... 17
  3.16 Dashes and slashes ...................................... 17
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 .............................................. 35
6.5 Gobbling comment characters ............................. 36
6.6 Vertical spacing ............................................... 37
6.7 ltxdoc emulation .............................................. 37
6.8 doc emulation ................................................ 37
6.9 Support macros ............................................... 38
6.10 Key handling for object classes ......................... 42
6.11 Handling \marginpar inside a float ................... 43
6.12 Handling glossary ........................................... 43
6.13 Catcode handling to support underscores .......... 43
6.14 \DescribeMacro and \DescribeEnvironment ........... 47
6.15 New \Describe... macros .................................. 48
6.16 \DescribeDefault .......................................... 57
6.17 \ItemDescribeMacro, etc. ................................ 57
6.18 \margintag, \watchout .................................... 60
6.19 Nesting ....................................................... 61
6.20 The dtxexample environment .............................. 61
6.21 noidmacro and noindentenvironment ................... 64
6.22 sourcedisplay, UIDisplay, docsidenavbar ............ 64
6.23 Formatted objects .......................................... 66
   6.23.1 \LaTeX objects ....................................... 66
6.23.2 Programs and commands ........................................ 67
6.23.3 File types ....................................................... 68
6.23.4 Internet ......................................................... 69
6.23.5 Specific programs ............................................. 70
6.23.6 Acronyms, brand names, trademarks ....................... 70
6.24 Logos .............................................................. 71
6.25 Dashes and slashes ............................................... 72

7 Compiling dtxdescribe ............................................. 74

Change History and Index ........................................ 75

List of Examples

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

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

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

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

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

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

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

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

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

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

% hyperref now automatically loaded by \pkg{ltxdoc}.
\AtBeginDocument{
  \hypersetup{...}% If needed.
  \pdfstringdefDisableCommands{ ... }% If needed.
}
\AddToHook{begindocument/before}{% Before .aux file is loaded.
  \usepackage{cleveref}% If needed.
  \crefname{somename}{name}{names}% If needed.
}
...
\%</driver>

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

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

Various objects inside the .dtx or .tex file may be described with new macros such as \DescribeBoolean, \DescribeLength, \DescribeCounter, similar to the already-familiar \DescribeMacro and \DescribeEnv.
Optional "classes" may be assigned to the objects being described, including the new versions of `\DescribeMacro` and `\DescribeEnv`. These classes are printed in the margin tag and index entry for each item, and also generate additional index entries sorted by class. This is especially useful for key/value sets, where several sets may appear in the same document.

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.

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

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

3.1 Macros and environments

These are only provided by the \texttt{ltexdoc} and \texttt{doc} package to document a \texttt{.dtx} file, where comments are used by \texttt{docstrip} to disable these environments in the resulting \texttt{.sty} file. When using the regular document classes, the macro and environment environments would localize any definitions, and \DescribeMacro and \DescribeEnv should be used instead.

\begin{verbatim}
\DescribeMacro [⟨class⟩] {⟨name⟩}
\end{verbatim}

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

\begin{verbatim}
\DescribeEnv [⟨class⟩] {⟨environment name⟩}
\end{verbatim}

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

3.2 Arguments

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

\begin{verbatim}
\marg {⟨text⟩}
\end{verbatim}

Shows a mandatory argument for a macro or environment.

The results looks like \( ⟨\text{mandatory}⟩ \).

\begin{verbatim}
\oarg {⟨text⟩}
\end{verbatim}

Shows an optional argument for a macro or environment.

The results looks like \( ⟨\text{optional}⟩ \).

\begin{verbatim}
\parg {⟨text⟩}
\end{verbatim}

Used for “picture” arguments, such as coordinates.

The result looks like \( ⟨\text{coordinate}⟩ \).
May be used to describe actions taken when given certain macro arguments. These will be given an 'Arg' margin tag and will appear in the index. The class may be used to categorize arguments by their macro or environment name. See example 9 on page 24.

3.3 Booleans, lengths, counters, hooks, keys

See example 4 on page 21.

\DescribeBoolean \langle class \rangle \{ \langle name \rangle \}
Describes a boolean. Given a 'Bool' tag in the margin and index.

\DescribeLength \langle class \rangle \{ \langle name \rangle \}
Describes a length. Given a 'Len' tag in the margin and index.

\DescribeCounter \langle class \rangle \{ \langle name \rangle \}
Describes a counter. Given a 'Ctr' tag in the margin and index.

\DescribeHook \langle class \rangle \{ \langle name \rangle \}
Describes a hook. Given a 'Hook' tag in the margin and index. The class may be used to categorize hooks by package. Example:

\DescribeHook[LaTeX]{para/begin}

\DescribeKey \langle class \rangle \{ \langle name \rangle \}
Describes a key. Given a 'Key' tag in the margin and index. The class may be used to categorize keys by their kev/value group. See example 8 on page 23.

3.4 Packages, classes, options

\DescribePackage \langle class \rangle \{ \langle name \rangle \}
Describes a package. Given a 'Pkg' tag in the margin and index.

\DescribeClass \langle class \rangle \{ \langle name \rangle \}
Describes a \LaTeX class. Given a 'Cls' tag in the margin and index.

\DescribeOption \langle class \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 \{\langle class\rangle\} \{\langle name\rangle\}

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

\DescribeProgram \{\langle class\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 \{\langle class\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 \{\langle class\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 \ttfamily type. When a class is used, it is pre-pended to the margin tag, appended to the index entry, and a second index entry is created grouped by class. If a macro name is to be described, use \DescribeMacro instead. See example 10 on page 25.

\DescribeOther \{\langle class\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 class is used, it is pre-pended to the margin tag, appended to the index entry, and a second index entry is created grouped by class. See example 11 on page 25.

3.7 In a description environment

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

\ItemDescribeMacro \{\langle class\rangle\} \{\langle name\rangle\} A description.
\ItemDescribeEnv \{\langle class\rangle\} \{\langle name\rangle\} A description.
\ItemDescribeArgument \{\langle class\rangle\} \{\langle argument\rangle\} A description.
\ItemDescribeBoolean \{\langle class\rangle\} \{\langle name\rangle\} A description.
\ItemDescribeLength \{\langle class\rangle\} \{\langle name\rangle\} A description.
\ItemDescribeCounter \{\langle class \rangle \} \{\langle name \rangle \} A description.
\ItemDescribeHook \{\langle class \rangle \} \{\langle name \rangle \} A description.
\ItemDescribeKey \{\langle class \rangle \} \{\langle name \rangle \} A description.
\ItemDescribePackage \{\langle class \rangle \} \{\langle package_name \rangle \} With underscores.
\ItemDescribeClass \{\langle class \rangle \} \{\langle class_name \rangle \} With underscores.
\ItemDescribeOption \{\langle class \rangle \} \{\langle name \rangle \} A description.
\ItemDescribeFile \{\langle class \rangle \} \{\langle file_name \rangle \} With underscores.
\ItemDescribeProgram \{\langle class \rangle \} \{\langle program_name \rangle \} With underscores.
\ItemDescribeCommand \{\langle class \rangle \} \{\langle command_name \rangle \} With underscores.
\ItemDescribeObject \{\langle class \rangle \} \{\langle name \rangle \} A description.
\ItemDescribeOther \{\langle class \rangle \} \{\langle name \rangle \} A description.

3.8 Defaults

\DescribeDefault \{\langle 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 fraction of \linewidth \rangle \} \{\langle container name \rangle \} \{\langle contents \rangle \}

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

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

See example 13 on page 27 for an example.
\section*{3.10 \texttt{\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.

\margintagexample

\margintag\texttt{color}
Default: blue!70!black

\watchout \[⟨text⟩\]

\△ \texttt{\watchout[example]}

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

\watchout\texttt{color}
Default: red!50!black

\section*{3.11 \texttt{dtxexample environment}}

\texttt{dtxexample(env.)} * \[⟨Notes/cross-references]\] \{⟨caption & label⟩\}

The \texttt{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 \texttt{dtxexample} environment are printed verbatim, then loaded and executed as \LaTeX\ code, showing the results just below the printed code. In the case of float commands, the floats are generated as expected somewhere nearby, and should be given their own labels. References to the float's labels may be placed in the optional argument to the \texttt{dtxexample} environment, and will be printed below the code.

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

See example 14 for a demonstration of how \texttt{dtxexample} works.

\texttt{\dtxexamplecodename}
Default: Code:

\texttt{\dtxexampleresultname}
Default: Result:

\section*{3.12 \texttt{noindmacro} and \texttt{noindenvironment} environments}

\△ \texttt{.dtx only}

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

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

Replace

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

with

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

and similarly for noindenvironment.

3.13 sourceverb, sourcedisplay, UIdisplay, docs-sidebar

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

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 15 on page 29.

sourcedisplay (env) Display source code with manual formatting. This is not a verbatim environment. \textcolor, \textbf, and \textit 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, \fquad, and \fqqquad may be used to force indenting. Because this is not a verbatim environment, it can be used inside a macro. See example 16 on page 29.

\fquad Single-level indent inside a sourcedisplay.

\fquad Double-level indent inside a sourcedisplay.

\fquad Triple-level indent inside a sourcedisplay.

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

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

\userentryname Default: Enter ⇒
docsidebar (env.) \[ ⟨title⟩ \] Creates a sidebar within the document. See example 18 on page 31.

### 3.14 Formatted objects

Macros to format references to various kinds of objects.

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

#### 3.14.1 \LaTeX objects

- \pkg{⟨packagename⟩} Prints as packagename. Also for a classname.
- \cs{⟨csname⟩} Prints as \csname.
- \env{⟨environment⟩} Prints as environment.
- \marg{⟨argument⟩} Prints \{⟨arg⟩\}. Mandatory argument.
- \oarg{⟨argument⟩} Prints \{⟨arg⟩\}. Optional argument.
- \parg{⟨argument⟩} Prints \{⟨arg⟩\}. Picture-mode argument.
- \ctr{⟨counter⟩} Prints as counter.
- \bool{⟨boolean⟩} Prints as boolean.
- \optn{⟨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

- \progcude Prints as inline program code: Escape underscores and other special characters such as \|, \%, $.
- \prog Prints as grep, make: A program name. Underscores allowed.
- \filenm Prints as file_name: Underscores allowed.
- \UI Prints as General user-interface text. What the user sees on the display. Also see the \UIdisplay environment.
- \cmds Prints as Commands to be entered: What the user enters. Escape underscores and other special characters such as \|, \%, $. Also see the \userentry macro.
3.14.3 File types
\ODT ODT OpenDocument Format word processing document
\SVG svg image format
\PNG PNG image format
\GIF GIF image format
\JPG JPG image format
\EPS EPS image format
\PDF PDF image format
\DVI DVI image format

3.14.4 Internet
\UTF UTF: Unicode
\URL URL: Uniform Resource Locator
\element \{\langle element name\rangle\} Prints as <element>, an HTML/CSS element
\attribute \{\langle attribute name\rangle\} Prints as attribute, an HTML/CSS attribute. pdfTEX and X\TeX\ only. Not for Lua\TeX.\attribute \{\langle attribute name\rangle\} Prints as attribute, an HTML/CSS attribute. pdfTEX, 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
\EPUB EPUB: E-book file format

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  \{\langle name\rangle\}  \textbf{BRANDNAME, COMPANY NAME}
\acro  \{\langle acronym\rangle\}  \textbf{ACRO: Acronym}
\supregistered  Superscript trademark symbol®

3.15  Logos

Several additional logos are provided.

Also see the \texttt{metalogo} and \texttt{metalogoX} packages.

\dviTeX \texttt{DVI TeX}
\dviLaTeX \texttt{DVI LaTeX}
\pdfTeX \texttt{PDF TeX}
\pdfLaTeX \texttt{PDF LaTeX}
\LuaTeX \texttt{LuaTeX}
\LuaLaTeX \texttt{LuaLaTeX}
\XeTeX \texttt{XeTeX}, with reversed E if \texttt{graphics} is loaded.
\XeLaTeX \texttt{XeLaTeX}, with reversed E if \texttt{graphics} is loaded.
\AmS \texttt{AMS}
\LyX \texttt{LyX}
\BibTeX \texttt{BibTeX}
\MakeIndex \texttt{MakeIndex}
\ConTeXt \texttt{ConTeXt}
\MiKTeX \texttt{MiKTeX}

3.16  Dashes and slashes

\thinspace \texttt{A breakable thin skip.}
\endash \texttt{An endash: –}
\textbackslash\text{emdash}  An emdash: —
\textbackslash\text{thinbrspace} A thin space which allows a line break.
\textbackslash\text{thinthinbrspace} A very thin space which allows a line break.
\textbackslash\text{Dash} An unbreakeable thin space, emdash, and breakable thin space: A—B
\textbackslash\text{dash} An unbreakeable thin space, endash, and breakable thin space: A–B
\textbackslash\text{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 \textbackslash\text{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 \textbackslash\text{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 \textbackslash\text{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

\ DeclareFloatingPhoto \photograph
\photocaptionsetup \photograph \cphotonameref \photograph

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

The re-defined \DescribeMacro, \DescribeEnv, and all the following macros create hyperlinked index entries, along with regular uses of \index.
Example 2: Environment

Code:
\DescribeEnv{myenvironment} \marg{argument} Short description.

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

add'l tags

index groups

⚠️ too much text

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.

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

Example 3: Second Environment

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

Result:
otherenvironment (env.) \{⟨opt args⟩\} ⟨⟨coordinates⟩\} A description.

The otherenvironment will be indexed by itself and also with myenvironment under the index entry "environments", and also under the class kindofenvironment.
Example 4: Booleans and Counters

Code:

\DescribeBoolean[examples]{sampleboolean} Some description.
\DescribeCounter[examples]{samplecounter} Some description.

Result:

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

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

Example 5: Lengths

Code:

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

Result:

\photowidth (Len) [photograph] Some description.

Lengths have a leading backslash, but are otherwise described the same as the rest of the objects.
Example 6: Packages, Classes, and Options

Code:

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

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

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

result:

| samplepackage (Pkg) [examples] | About a \LaTeX\ package. |
| sample_class (Cls) [examples]  | About a \LaTeX\ class.   |
| sampleoption (Opt) [examples]  | About an option for a package or class. |

Example 7: Files, Commands, and Programs

Code:

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

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

\DescribeFile{lone_file.txt} Some description.

\DescribeCommand(OS_command) An operating-system command.

\DescribeProgram(program_name) An operating-system program.

result:

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

Filenames, program names, and command names may have underscores, such as tested here. A class is used to group “bigfiles” together in the index.
Example 8: Keys

Code:

\DescribeKey[groupofkeys]{firstkey} About the first key of the \textit{groupofkeys} set.
\DescribeKey[groupofkeys]{secondkey} About the second key of \textit{groupofkeys}.
\DescribeKey[examples]{samplekey} About some key of \textit{otherkeys}.
\DescribeKey[examples]{sampletwokey} About another key of \textit{otherkeys}.
\DescribeKey{lonekey} A key without a class.

Result:

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

See the index key groups.
Example 9: Arguments

Code:

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

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

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

Result:

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

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

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

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

⚠️ **macro names** Note you may need to use \cs{mymacro} for the macro's name.
Example 10: Object

Code:

\DescribeObject[color]{somecolor}  
The color of something.
\DescribeObject[color]{othercolor}  
The other color.
\DescribeObject{randomobject} About some random object.

Result:

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

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

---

Example 11: Other

Code:

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

Describes an arbitrary non-programming object, using roman text.
Example 12: Description environments

Code:

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

Result:

\macroname [descexamples] \textbf{\macroname}: Describe the macro.

\booleanname (bool) [descexamples] \textbf{\booleanname}: Describe the boolean.

\lengthname (Len) [descexamples] \textbf{\lengthname}: Describe the length.

\keyname (Key) [descexamples] \textbf{\keyname}: Describe the key.

\package_name (Pkg) [descexamples] \textbf{\package_name}: Describe the package.

\classname (Cls) [descexamples] \textbf{\classname}: Describe the class.

\filename (file) [descexamples] \textbf{\filename}: Describe the file.

\program_name (Prog) [descexamples] \textbf{\program_name}: Describe the program.

\command_name (Cmd) [descexamples] \textbf{\command_name}: Describe the class.

Uses a description environment to describe objects.
Example 13: Nesting

*Code:*

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

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

*Result:*

```

<table>
<thead>
<tr>
<th>environmentA</th>
<th>\macroB</th>
<th>environmentC</th>
<th>The contents.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>sidebyside</th>
<th>minipage\hfill</th>
<th>minipage \hfill</th>
<th>Right contents.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left contents.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Note the use of the optional arguments to select less than full \linewidth, and the starred form for the second box to place it inline with the \hfill text.
Example 14: \texttt{dtxexample}

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

When the example was created:

1. The “float” of type \texttt{dtxexample} was created, with the caption “\texttt{dtxexample}” and the label \texttt{ex:dtxexample}, which points to example 14.
2. The code was displayed verbatim.
3. The code was written to the file \texttt{dtxexample\_cut.tex}.
4. The code was \texttt{\input} from \texttt{dtxexample\_cut.tex}.
5. Executing the code created the figure with caption “A Figure” and label \texttt{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 15: fsourceverb**

*Code:*

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

*Result:*

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

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

---

**Example 16: sourcedisplay**

*Code:*

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

*Result:*

```
\newcommand{dosomething}[1][whattodo]{
  doing #1
}
```
Example 17: UIdisplay

Code:

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

Result:

Select:

Preferences $\to$ Plugins $\to$ Files $\to$ HTML

For the field

Title heading:

Enter $\Rightarrow$ H1
Example 18: docsidebar

Code:

Main text.

More main text.

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

Additional main text.

Result:

Main text.

More main text.

\begin{docsidebar}[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.\footnote{Future versions may include \DeclareBoolean for use at the point where the boolean is defined, creating an index entry with a code line number, and \DescribeBoolean with a page number index entry for the related discussion in the User’s Manual portion of the document.} It may be useful to use \DeclareBoolean and friends both at the code location and also in the User’s Manual section.

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

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

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

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

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

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

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

6.1 Required packages

\makeidx\texttt{(Pkg)} \quad One of several index programs must be provided. One of several index programs must be provided.
\begin{verbatim}
\AtBeginDocument{
  \@ifpackageloaded{makeidx}{}
  \@ifpackageloaded{splitidx}{}
    \\RequirePackage{makeidx}
  \\makeindex
}
\end{verbatim}

\etoolbox\texttt{(Pkg)} \quad v2.6 or later for \texttt{BeforeBeginEnvironment}, \texttt{AfterEndEnvironment}
\begin{verbatim}
\RequirePackage{etoolbox}[2011/01/03]%
\end{verbatim}

\xparse\texttt{(Pkg)} \quad Used for the examples.
\begin{verbatim}
\RequirePackage{xparse}
\end{verbatim}

\calc\texttt{(Pkg)} \quad Used for \texttt{\shownesting}.
\begin{verbatim}
\RequirePackage{calc}
\end{verbatim}

\xcolor\texttt{(Pkg)} \quad Used for the examples.
\begin{verbatim}
\RequirePackage{xcolor}
  \definecolor{myurlcolor}{rgb}{0,0,.7}
  \definecolor{mylinkcolor}{rgb}{.7,0,0}
\end{verbatim}

\caption\texttt{(Pkg)} \quad Used for the examples.
\begin{verbatim}
\RequirePackage{caption}
\end{verbatim}

\newfloat\texttt{(Pkg)} \quad Used for the examples.
\begin{verbatim}
\RequirePackage{newfloat}
\end{verbatim}

\fancyvrb\texttt{(Pkg)} \quad Used for the examples.
\begin{verbatim}
\RequirePackage{fancyvrb}
\end{verbatim}

\xstring\texttt{(Pkg)} \quad Used for \texttt{\StrSubstitute} for \texttt{\DescribeFile}.
\begin{verbatim}
\RequirePackage{xstring}
\end{verbatim}

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

```latex
\newcommand*{\hyperpage}[1]{{#1}}
```

### pict2e (Pkg)

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

#### 6.2 Warning sign

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

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

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

#### 6.3 Special character handling

The literal backslash character:

```latex
\begingroup
\catcode`\|=0
\catcode`\|=12
\gdef\DTXD@backslash{{}}
\endgroup
```

#### 6.4 Patching hypdoc

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

```latex
\ifpackageloaded{doc}{}
\ifpackageloaded{hypdoc}{}
\AddToHook{begindocument/before}[doc/hyperref]{%
\ifdoc@hyperref
\else
\def\hdclindex#1#2{%
```
splitidx is modified to add \hdpindex{} to work with hypdoc.

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

\ifpackageloaded{doc}{
  \newcommand*{\DTXD@gobble}{2}
}{
  \newcommand*{\DTXD@gobble}{0}
}
6.6 Vertical spacing

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

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

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

6.7 \texttt{ltxdoc} emulation

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

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

6.8 \texttt{doc} emulation

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

\AtBeginDocument{
\@ifpackageloaded{doc}{
\newenvironment*{macro}[1][]{
\PackageError{dtxdescribe}{The 'macro' environment is only available when using the \texttt{doc} package with a .\texttt{dtx} source file}
\PackageError{dtxdescribe}{This environment only makes sense for .\texttt{dtx} source.}
}{
\newenvironment*{environment}[1][]{
\PackageError{dtxdescribe}{The 'environment' environment is only available when using the \texttt{doc} package with a .\texttt{dtx} source file}
\PackageError{dtxdescribe}{This environment only makes sense for .\texttt{dtx} source.}
}{
}()}
6.9 Support macros

\PrintEnvName \{\textit{name}\}\} \quad Prints an environment name.
\providecommand*{\PrintEnvName}{}
\renewcommand*{\PrintEnvName}[1]{\strut\scriptsize{Env} \quad \MacroFont#1\ }

\DTXD@printtype \{\texttt{text}\}\}
\newcommand*{\DTXD@printtype}[1]{\texttt{\scriptsize\texttt{sffamily#1}}} \quad Used to print the object class in the margin:
\raggedleft\strut{\scriptsize\texttt{sffamily#1}}\quad\MacroFont
\usage \{\texttt{text}\}\}
\IfPackageLoadedTF{doc}{}{% not doc package

Allow hyperlinks in the “usage” index entries:
\IfPackageLoadedTF{doc}{}{% not doc package

\end{verbatim}
\DTXD@origwrindex  Used to bypass \hyperref\ index modifications.
\DTXD@margintag \{\langle\text{class}\rangle\} \{\langle\text{name}\rangle\} \{\langle\text{margin tag}\rangle\}

Creates the margin tag for the object being described.

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

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

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

The class is used to sub-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.

\@bsphack\% \begingroup\% \DTXD@origwrindex(\%

Index by name:

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

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

Index by tag and class:

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

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

Possibly index by class and name:

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

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

\DTXD@margintag(#1)(#2)(#3)

The index entries:

\DTXD@index(#1)(#2)(#3)(#4)(#5)

\DTXD@macroname \langle control sequence \rangle

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

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

\begingroup\lccode'|='\n\lowercase{\endgroup\def\removebs#1{\if#1|\else#1\fi}}
\newcommand*{\DTXD@macroname}{\expandafter\removebs\string#1}

\DTXD@verbatimcmd \langle\name\rangle

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

\newcommand*{\DTXD@verbatimcmd}{\string\verb\quotechar*\verbatimchar\string#1\verbatimchar}

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

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

\newcommand*{\DTXD@cmdmargintagindex}{\@bsphack}

Create a margin tag with the name of the macro:

@ifundefined{@captype}% not float?
\leavevmode
\marginpar{\hbadness=10000 \hfuzz=5em \DTXD@printtype{#3}{#1}{#2}{#4}{#5}{#6}}
Create an index entry sorted by the name without its leading backslash, followed by the macro name with the backslash, and the tag. Prepend with the class if given.

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

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

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

6.10 Key handling for object classes

If using doc, the optional key/value argument may also include an object class. This is supported by assigning any unknown key to be the class.
6.11 Handling `\marginpar` inside a float

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

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

6.12 Handling glossary

Fix for `\StopEventually` in some conditions:

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

6.13 Catcode handling to support underscores

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

334 \IfPackageLoadedTF{doc}{% 
335 336 \def{\@NewDocElement#1#2#3}{% 
337 \doc@macrolikefalse 
338 \doc@topleveltrue 
339 \def{\doc@idxtype(#3)}{
340 \def{\doc@idxgroup(#3)}{
341 \let{\doc@printtype}@empty 
342 \csname keys_set:nn\endcsname{doc}{#1}% 
343 \ifx{\doc@printtype}@empty 
344 \@emptokena []% 
345 \else
\DescribeMacro \[⟨class⟩\] {⟨\name⟩}

Redefined to allow hyperlinked index entries and an optional class:

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 class if given.
Only if a class was given:
\begin{macrocode}
\begingroup%
\DTXD@origwrindex{#1}\actualchar[#1]:\levelchar%
\DTXD@macroname{#2}\actualchar\DTXD@verbatimcmd{#2}
\encapchar usage}% class given
\@esphack%
\ignorespaces%
\endgroup%
\end{macrocode}

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

Redefined to allow hyperlinked index entries:
\begin{macrocode}
\IfPackageLoadedTF{doc}{}{% not doc
\providecommand*{\DescribeEnv}{\DTXD@margintagindex{#1}{#2}{Env}{environment}{usage}}% not doc
\end{macrocode}

6.15 New \texttt{\Describe... macros}

\DTX@filename \begin{macrocode} Stores the filename with a sanitized underscore.
\end{macrocode}

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

The name may have underscores.
\begin{macrocode}
\end{macrocode}

Create a detokenized version of the filename...
\begin{macrocode}
\renewcommand{\DTXD@filename}{\dtokenize{#2}}%
\end{macrocode}
... then replace any underscores with a detokenized \_, which will print as an underscore when read back from the index file:

572 \StrSubstitute{\DTXD@filename}{\detokenize{\_}}{\detokenize{\_}}{\DTXD@filename}\

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

573 \DTXD@margintag{#1}{#2}{#3}\

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

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

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

575 \endgroup\
576 \ignorespaces\
577 }

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

The name may have underscores.

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

\DescribeFile {{\name}}

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

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

The name may have underscores.

\newcommand*{\DTXDDescribeProgram}[2][]{\DTXDfilemarginparindex{#1}{#2}{Prog}{program}{usage}%
\DescribeProgram \{(name)\}

The underscore character is temporarily disabled, then the name is passed directly to \DTXDDescribeProgram.

\IfPackageLoadedTF{doc}{%
\NewDocElement[
macrolike=false,
toplevel=false,
idxtype=program,
idxgroup=Programs,
printtype=\textit{Prog}]
\DescribeProgram\{(program)\}%
}% not doc
\newcommand*{\DescribeProgram}{\begingroup\catcode\_=12 \DTXDDescribeProgram%}
\DTXDDescribeCommand \[(class)\] \{(name)\}

The name may have underscores.

\newcommand*{\DTXDDescribeCommand}[2][]{\DTXDfilemarginparindex{#1}{#2}{Cmd}{command}{usage}%
\DescribeCommand \{(name)\}

The underscore character is temporarily disabled, then the name is passed directly to \DTXDDescribeCommand.
\DescribeCommand {command}
\newcommand*{\DescribeCommand}{% not doc
\begingroup\catcode\_=12 \DTXD@DescribeCommand%}
% not doc

\DTXD@DescribePackage [\langle class\rangle] {\langle name\rangle} The name may have underscores.
\newcommand*{\DTXD@DescribePackage}[2][]{% not doc
\DTXD@filemarginparindex{#1}{#2}{Pkg}{package}{usage}%
}

\DescribePackage {\langle name\rangle}
The underscore character is temporarily disabled, then the name is passed directly to \DTXD@DescribePackage.
\IfPackageLoadedTF{doc}{% doc
\NewDocElement[
  macrolike=false,
  toplevel=false,
  idxtype=package,
  idxgroup=Packages,
  printtype=\textit{Pkg}
]{Package}{package}{% not doc
}
\DTXD@DescribeClass [\langle class\rangle] {\langle name\rangle} The name may have underscores.
\newcommand*{\DTXD@DescribeClass}[2][]{% not doc
\DTXD@filemarginparindex{#1}{#2}{Cls}{class}{usage}%
}

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

\[\text{\textbackslash IfPackageLoadedTF(doc)}\% \text{doc} \]
\[\NewDocElement[] \]
\[\macrolike=false, \]
\[\toplevel=false, \]
\[\idxtype=class, \]
\[\idxgroup=Classes, \]
\[\printtype=\textit{Cls} \]
\[\{Class\}\{class\} \]
\[\\% \text{not doc} \]
\[\\newcommand*\{\DescribeClass\}\{\DTXD@margintagindex{#1}{#2}{Opt}{option}{usage} \]
\[\\% \text{not doc} \]
\[\text{\textbackslash DescribeOption} \[\langle class \rangle \]\{\langle name \rangle\} \]
\[\text{\textbackslash IfPackageLoadedTF(doc)}\% \text{doc} \]
\[\NewDocElement[] \]
\[\macrolike=false, \]
\[\toplevel=false, \]
\[\idxtype=option, \]
\[\idxgroup=Options, \]
\[\printtype=\textit{Opt} \]
\[\{Option\}\{option\} \]
\[\\% \text{not doc} \]
\[\\newcommand*\{\DescribeOption\}\{1\}\{2\}\{\langle option \rangle\} \]
\[\\% \text{not doc} \]
\[\text{\textbackslash DescribeArgument} \[\langle class \rangle \]\{\langle name \rangle\} \]

The class may be used to categorize arguments by their macro or environment name.
\DescribeArgument \[ \text{class} \] \{ \langle \text{name} \rangle \}

\DescribeBoolean \[ \text{class} \] \{ \langle \text{name} \rangle \}

\DescribeLength \[ \text{class} \] \{ \langle \text{name} \rangle \}

\DescribeCounter \[ \text{class} \] \{ \langle \text{name} \rangle \}
The class may be used to categorize keys by their key/value group.
\DescribeObject \{⟨class⟩\} \{⟨name⟩\}

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

\IfPackageLoadedTF{doc}{% doc
\NewDocElement[ macrolike=false,
toplevel=false,
idxtype=object,
idxgroup=Objects,
printtype=,
](Object){object}
}% not doc
\Newcommand*{\DescribeObject}[2]{% not float?
\@ifundefined{\captype}{% not float?
\@bsphack
\leavevmode
\marginpar{\hbadness=10000\hfuzz=5em\raggedleft\ifblank{#1}{}{\raggedleft\scriptsize\[#1\]} \texttt{#2}}
\@esphack\ignorespaces
\DescribeOther [\langle \textit{class} \rangle] \{\langle \textit{name} \rangle\}

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

\describecolor \texttt{\textcolor{\descriptiondefaultcolor}{Default: #1}}

6.17 \ItemDescribeMacro, etc.

The following are for use inside a description.

\item\macro{\texttt{#2}}{\name{#2}}
\item\env{\texttt{#2}}{\name{#2}}
\item\texttt{#2}{\name{#2}}
\ItemDescribeBoolean \[\langle class \rangle \] \{ \langle name \rangle \}

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

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

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

\ItemDescribeKey \[\langle class \rangle \] \{ \langle name \rangle \}

\ItemDescribePackage \[\langle class \rangle \] \{ \langle name \rangle \}
\ItemDescribeClass \[\langle \text{class} \rangle \] \{\langle \text{name} \rangle \}

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

\ItemDescribeOption \[\langle \text{class} \rangle \] \{\langle \text{name} \rangle \}

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

\ItemDescribeFile \[\langle \text{class} \rangle \] \{\langle \text{name} \rangle \}

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

\ItemDescribeProgram \[\langle \text{class} \rangle \] \{\langle \text{name} \rangle \}

\newcommand{\DTXD@ItemDescribeProgram}[2][]{%
  \item[\texttt{#2}:]%
  \setlength{\parskip}{1.5ex}%
  \DescribeProgram[#1]{#2}%
  \endgroup%
}\ItemDescribeProgram
\ItemDescribeCommand \hfill \ItemDescribeObject \hfill \ItemDescribeOther \hfill \margintagcolor \hfill \watchoutcolor

\margintagcolor The color of the \margintag.
\margintag \hfill \margintagcolor\{\textcolor{\margintagcolor}{text}\}

Prints a colored margin tag.
\watchoutcolor The color of the \watchout.
\watchout \hfill \\textcolor{\watchoutcolor}{text}
\watchout \text{[⟨text⟩]}

Prints a warning sign and optional text.

\shownesting \text{[⟨fraction of \linewidth] \{⟨container⟩\} \{⟨contents⟩\}}

6.19 Nesting

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

6.20 The dtxexample environment

Also see example 14 on page 28.

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

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

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

\newcommand*{\dtxexampleresultname}{Result:}

\begin{dtxexample}(env.)*\[(\text{notes/cross-references})]\{\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}

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

\VerbatimOut[{gobble=\DTXD@gobble,tabsize=4}]{dtxexample_cut.tex}%

When the environment closes:

\Finish the verbatim output:

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

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

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

Typeset the contents as verbatim:
Possible add the optional cross-references or notes:

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

If unstarred, close the float or \minipage.

\IfBooleanTF{#1}\
{\endminipage}\
{\end{dtxexamplefloat}}

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

\AfterEndEnvironment{dtxexample}
%

Execute the code:

\par\input{dtxexample_cut.tex}%

Closing rule::

\medskip\hrule%

\newfloat{dtxexamplefloat} A new float type for the examples.

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

\captionsetup[dtxexamplefloat]{% Caption setup for the examples.
\crefname[dtxexamplefloat] Name for cleveref. \crefname here is required for documents not using the doc class:

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

6.21 noindmacro and noindenvironment

Similar to macro and environment, but not indexed.

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

\begin{Verbatim}
oindenvironment\{env\} \{\langle name\rangle\}
\end{Verbatim}

6.22 sourcedisplay, UIdisplay, docsidenav

For use in a sourcedisplay:

\begin{Verbatim}
fquad \text{Forces a quad indent.}
\end{Verbatim}

\begin{Verbatim}
fquad \text{Forces a double-quad indent.}
\end{Verbatim}
\fqqquad Forces a triple-quad indent.

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

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

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

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

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

\texttt{sourcedisplay (env.)} To typeset a block of source code, allowing direct formatting.

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

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

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

\texttt{\userentryname} Text to tell the user to enter the following item.

\newcommand*{\userentryname}{Enter~$\Rightarrow$}
\userentry\{}\langle\text{text to enter}\rangle\}\}
Typesets text to be entered by the users.

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

\begin{docsidebar}[env.]
To typeset a sidebar in the documentation.
\end{docsidebar}

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

\section{Formatted objects}

Macros to format references to various kinds of objects.

\subsection{\LaTeX objects}

\pkg{⟨name⟩} Also useable for class names.
\\providerobustcmd*{\pkg}{[1]\mbox{\textsf{⟨#1⟩}}}

\cs{⟨csname⟩} From ltxdoc.
\\providerobustcmd*{\cs}{[1]\texttt{⟨#1⟩}}

\env{⟨name⟩}
\\providerobustcmd*{\env}{[1]\mbox{\texttt{⟨#1⟩}}}

\marg{⟨argument⟩} From ltxdoc.
\providecommand\marg[1]{% 
\ttfamily\char'{}\meta{#1}\ttfamily\char'{}\}}
\oarg \langle\textit{argument}\rangle\quad\text{From ltxdoc.}

\providecommand\oarg[1]{% 
\ttfamily[}\meta{#1}\ttfamily\]}
\parg (\langle\textit{argument}\rangle)\quad\text{From ltxdoc.}

\providecommand\parg[1]{% 
\ttfamily(}\meta{#1}\ttfamily)}
\ctr \langle\textit{name}\rangle\quad\text{No processing is provided for special characters.}

\providecommand\ctr\{\mbox{\texttt{#1}}\}
\bool \langle\textit{name}\rangle\quad\text{No processing is provided for special characters.}

\providecommand\bool\{\mbox{\texttt{#1}}\}
\optn \langle\textit{name}\rangle\quad\text{No processing is provided for special characters.}

\providecommand\optn\{\mbox{\texttt{#1}}\}
\TOC\quad\text{No processing is provided for special characters.}

\providecommand\TOC\{\acro{TOC}\}
\LOF\quad\text{No processing is provided for special characters.}

\providecommand\LOF\{\acro{LOF}\}
\LOT\quad\text{No processing is provided for special characters.}

\providecommand\LOT\{\acro{LOT}\}

6.23.2 Programs and commands

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

\providecommand\cmds\{\mbox{\texttt{#1}}\}
\progcode \langle\textit{code to print}\rangle\quad\text{No processing is provided for special characters.}

\providecommand\progcode\{\mbox{\texttt{#1}}\}
\prog  \{\langle program name\rangle\}  Underscores are allowed.

\newcommand*{\DTXD@prog}{\mbox{\textsf{\textsl{\detokenize{#1}}}}}
\providerobustcmd{\prog}{\begingroup\catcode'\_=12\DTXD@prog\endgroup}

\filenm  \{\langle file name\rangle\}  Underscores are allowed.

\newcommand*{\DTXD@filenm}{\mbox{\texttt{\detokenize{#1}}}}
\providerobustcmd{\filenm}{\begingroup\catcode'\_=12\DTXD@filenm\endgroup}

\UI  General user-interface text.
\providerobustcmd{\UI}{\bf\textsf{#1}}

\section{File types}

\ODT
\newcommand*{\ODT}{\acro{ODT}}
\providerobustcmd{\ODT}{\acro{ODT}}

\SVG
\newcommand*{\SVG}{\acro{SVG}}
\providerobustcmd{\SVG}{\acro{SVG}}

\PNG
\newcommand*{\PNG}{\acro{PNG}}
\providerobustcmd{\PNG}{\acro{PNG}}

\GIF
\newcommand*{\GIF}{\acro{GIF}}
\providerobustcmd{\GIF}{\acro{GIF}}
Each of these is "provided", and any prior meaning will be unchanged. In particular, Lua\TeX{} uses \attribute, so its meaning is unchanged if using Lua\TeX{}.

\attribute {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}

\attribute {
\attrib {
\element {{\texttt{name}}}
}
6.23.5 Specific programs

\TikZ

6.23.6 Acronyms, brand names, trademarks

\brand {⟨name⟩}
\acro \{\acronym\}

\supregistered Superscript trademark symbol.

6.24 Logos

\dviTeX DVI\TeX

\dviLaTeX DVI\LaTeX

\pdfTeX PDF\TeX

\pdfLaTeX PDF\LaTeX

\LuaTeX Lua\TeX

\LuaLaTeX Lua\LaTeX

\XeTeX X\TeX, X\LaTeX

\XeLaTeX X\LaTeX
\textbf{6.25 Dashes and slashes}

\texttt{\thinspace} A breakable thin skip.

1244 \texttt{\DeclareRobustCommand{\thinspace}{\hspace{0.16667em}\relax}}

\texttt{\endash} An endash: –

1245 \texttt{\def\endash{\textendash}}

\texttt{\emdash} An emdash: —

1246 \texttt{\def\emdash{\textemdash}}

\texttt{\thinbrspace} A thin space which allows a line break.

1247 \texttt{\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 unbreakable thin space, emdash, and breakable thin space.

\newrobustcmd{\Dash}{\unskip\thinspace\emdash\thinthinbrspace}
\dash  An unbreakable thin space, endash, and breakable thin space.

\newrobustcmd{\dash}{\unskip\thinspace\endash\thinthinbrspace}
\Slash  An unbreakable 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>Date</th>
<th>General Changes</th>
<th>Document Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.10</td>
<td>2016/12/08</td>
<td>Initial version</td>
<td>Added formatted objects</td>
</tr>
<tr>
<td>v0.11</td>
<td>2018/03/30</td>
<td></td>
<td>Added logos</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cut file name changed to dtxexample_cut.tex</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mistag index tag font</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Package margin tag class in brackets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Other margin tag class in brackets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Length margin tag class in brackets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Macro margin tag class in brackets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Object margin tag class in brackets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Other margin tag class in brackets</td>
</tr>
<tr>
<td>v1.00</td>
<td>2019/01/11</td>
<td></td>
<td>Docbox display Reduced width</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>User entry display Reduced width</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Watchout color Added</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Noind macro Added</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sourceverb Added</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Source display Reduced width</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>User entry name Added</td>
</tr>
<tr>
<td>v1.01</td>
<td>2019/03/22</td>
<td></td>
<td>Sanitize PDF bookmarks</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Macro margin tag class in brackets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Object margin tag class in brackets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Other margin tag class in brackets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Filemarginparindex Fix if not doc package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fix if not hyperref package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fix if not ltxdoc class</td>
</tr>
<tr>
<td>v1.02</td>
<td>2019/07/16</td>
<td></td>
<td>User entry Fix if not ltxdoc class</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Watchout margin par</td>
</tr>
</tbody>
</table>
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.

\acro ............................. \textit{17, 1215} \captionsetup ............................. \textit{63}
\AmS .............................. \textit{17, 1235} Classes:
\captionsetup ............................. \textit{63}
Arguments:
[H] [figure] ............................ \textit{24} class_name [descexamples] ........................ \textit{26}
[M] [figure] ............................ \textit{24} sample_class [examples] ........................ \textit{22}
bold [mymacro] ........................ \textit{24} \cmds .............................. \textit{15, 1167}
\attrib .............................. \textit{16, 1201} \color:
\attribute ............................. \textit{16, 1201} DTXD@examplerulecolor ........................ \textit{61}
\BibTeX ............................ \textit{17, 1240} othercolor ........................ \textit{25}
[bigfiles]:
another_big_file.txt ........................ \textit{22} somecolor ........................ \textit{25}
really_big_file.txt ........................ \textit{22} DTXD@examplerulecolor ........................ \textit{61}
\bool .............................. \textit{15, 1162} \ConTeXt ............................ \textit{17, 1242}
Booleans:
booleanname [descexamples] ........................ \textit{26} \counters ............................ \textit{21}
sampleboolean [examples] ........................ \textit{21} \cphotonameref ........................ \textit{19}
\brand .............................. \textit{17, 1214} \crefname ............................. \textit{64}
\captionsetup ............................. \textit{63}
\acro ............................. \textit{17, 1215} \captionsetup ............................. \textit{63}
### Desceexamples:

<table>
<thead>
<tr>
<th>Key</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>booleanname</td>
<td>26</td>
</tr>
<tr>
<td>class_name</td>
<td>26</td>
</tr>
<tr>
<td>command_name</td>
<td>26</td>
</tr>
<tr>
<td>file_name</td>
<td>26</td>
</tr>
<tr>
<td>keyname</td>
<td>26</td>
</tr>
<tr>
<td>package_name</td>
<td>26</td>
</tr>
<tr>
<td>program_name</td>
<td>26</td>
</tr>
<tr>
<td>DescribeArgument</td>
<td>9</td>
</tr>
<tr>
<td>DescribeBoolean</td>
<td>10</td>
</tr>
<tr>
<td>DescribeClass</td>
<td>10</td>
</tr>
<tr>
<td>DescribeCommand</td>
<td>11</td>
</tr>
<tr>
<td>DescribeCounter</td>
<td>10</td>
</tr>
<tr>
<td>DescribeDefault</td>
<td>12</td>
</tr>
<tr>
<td>DescribeDefaultcolor</td>
<td>12</td>
</tr>
<tr>
<td>DescribeEnv</td>
<td>9</td>
</tr>
<tr>
<td>DescribeFile</td>
<td>11</td>
</tr>
<tr>
<td>DescribeHook</td>
<td>10</td>
</tr>
<tr>
<td>DescribeKey</td>
<td>10</td>
</tr>
<tr>
<td>DescribeLength</td>
<td>10</td>
</tr>
<tr>
<td>DescribeMacro</td>
<td>9</td>
</tr>
<tr>
<td>DescribeObject</td>
<td>11</td>
</tr>
<tr>
<td>DescribeOption</td>
<td>10</td>
</tr>
<tr>
<td>DescribeOther</td>
<td>11</td>
</tr>
<tr>
<td>DescribePackage</td>
<td>10</td>
</tr>
<tr>
<td>DescribeProgram</td>
<td>11</td>
</tr>
<tr>
<td>docsidebar (env.)</td>
<td>15</td>
</tr>
<tr>
<td>DTX@login</td>
<td>569</td>
</tr>
<tr>
<td>DTX@cmdmarginatagindex</td>
<td>266</td>
</tr>
<tr>
<td>DTX@descClass</td>
<td>659</td>
</tr>
<tr>
<td>DTX@describeFile</td>
<td>618</td>
</tr>
<tr>
<td>DTX@describeCommand</td>
<td>578</td>
</tr>
<tr>
<td>DTX@describePackage</td>
<td>633</td>
</tr>
<tr>
<td>DTX@describeProgram</td>
<td>598</td>
</tr>
<tr>
<td>DTX@filemarginparindex</td>
<td>570</td>
</tr>
<tr>
<td>DTX@gobble</td>
<td>132</td>
</tr>
<tr>
<td>DTX@index</td>
<td>229</td>
</tr>
<tr>
<td>DTX@macroname</td>
<td>260</td>
</tr>
<tr>
<td>DTX@marginatag</td>
<td>213</td>
</tr>
<tr>
<td>DTX@marginatagindex</td>
<td>255</td>
</tr>
<tr>
<td>DTX@origwindindex</td>
<td>212</td>
</tr>
<tr>
<td>DTX@printtype</td>
<td>204</td>
</tr>
<tr>
<td>DTX@verbatimcmd</td>
<td>263</td>
</tr>
<tr>
<td>dtxexample (env.)</td>
<td>13</td>
</tr>
<tr>
<td>dtxexample codename</td>
<td>13</td>
</tr>
</tbody>
</table>

### Environments:

<table>
<thead>
<tr>
<th>Key</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>docsidebar</td>
<td>15</td>
</tr>
<tr>
<td>dtxexample</td>
<td>9</td>
</tr>
<tr>
<td>environment</td>
<td>9</td>
</tr>
<tr>
<td>fsourcerverb</td>
<td>14</td>
</tr>
<tr>
<td>macro</td>
<td>9</td>
</tr>
<tr>
<td>myenvironment</td>
<td>20</td>
</tr>
<tr>
<td>noindentenvironment</td>
<td>14</td>
</tr>
<tr>
<td>noindentmacro</td>
<td>14</td>
</tr>
<tr>
<td>otherenvironment</td>
<td>20</td>
</tr>
<tr>
<td>sourcedisplay</td>
<td>14</td>
</tr>
<tr>
<td>sourcerverb</td>
<td>14</td>
</tr>
<tr>
<td>UIdisplay</td>
<td>14</td>
</tr>
<tr>
<td>\EPS</td>
<td>16</td>
</tr>
<tr>
<td>\EPUB</td>
<td>16</td>
</tr>
</tbody>
</table>

### Files:

<table>
<thead>
<tr>
<th>Key</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>another_big_file.txt</td>
<td>22</td>
</tr>
<tr>
<td>dtxexample_cut.tex</td>
<td>61</td>
</tr>
<tr>
<td>file_name [desceexamples]</td>
<td>26</td>
</tr>
<tr>
<td>lone_file.txt</td>
<td>22</td>
</tr>
<tr>
<td>really_big_file.txt</td>
<td>22</td>
</tr>
<tr>
<td>\fqquad</td>
<td>14</td>
</tr>
<tr>
<td>\fquadv</td>
<td>14</td>
</tr>
<tr>
<td>\fquadv</td>
<td>14</td>
</tr>
<tr>
<td>\fqquad</td>
<td>14</td>
</tr>
<tr>
<td>\fqquad</td>
<td>14</td>
</tr>
<tr>
<td>fsourcerverb (env.)</td>
<td>14</td>
</tr>
</tbody>
</table>

### Group of objects

<table>
<thead>
<tr>
<th>Key</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>\GIF</td>
<td>16</td>
</tr>
</tbody>
</table>
[groupofkeys]:
  firstkey .......................... 23
  secondkey .......................... 23

H
HTML .......................... 16, 1204
HTMLfive .......................... 16, 1205

I
index
  by group .......................... 20
\ItemDescribeArgument  ............ 11, 999
\ItemDescribeBoolean  ............... 11, 904
\ItemDescribeClass  .................. 12, 939
\ItemDescribeCommand  ............... 12, 974
\ItemDescribeCounter  ............... 12, 914
\ItemDescribeEnv  .................... 11, 894
\ItemDescribeFile  ................... 12, 954
\ItemDescribeHook  ................. 12, 919
\ItemDescribeKey  .................... 12, 924
\ItemDescribeLength  ............... 11, 909
\ItemDescribeMacro  ................. 11, 889
\ItemDescribeObject  ............... 12, 984
\ItemDescribeOption  ............... 12, 949
\ItemDescribeOther  ................. 12, 989
\ItemDescribePackage  .............. 12, 929
\ItemDescribeProgram  .............. 12, 964

J
\JPG .......................... 16, 1194

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

L
Lengths:
  \IndexMin .......................... 37
  \lengthname ........................ 26
  \ photowidth ........................ 21
\LOF .......................... 15, 1165
\LOT .......................... 15, 1166
\LuaLaTeX .......................... 17, 1222
\LuaTeX .......................... 17, 1221
\LyX .......................... 17, 1239

M
\macro (env.) ........................ 9
\macroname .......................... 26
\MakeIndex .......................... 17, 1241
\marg .................................. 9, 15, 1155
\margin tag missing .................. 32
\margin intag ........................ 13, 995
\margin intagcolor .................. 13, 994
\MathJax .......................... 17, 1213
\MathML .......................... 16, 1212
\MiKTeX .......................... 17, 1243
\myenvironment (env.) ............ 20
[\mymacro]:
  bold .......................... 24
\mymacro .......................... 19

N
\noindentenvironment (env.) ....... 14, 1093
\noindmacro (env.) .................. 14, 1087

O
\oarg .......................... 9, 15, 1157

Objects:
  \DTXD@examplerulecolor[\color]  .. 61
  othercolor [\color] ................ 25
  randomobject  ..................... 25
  somecolor [\color] ............... 25
\ODT .......................... 16, 1190

Options:
  \sampleoption [\examples] ........ 22
\optn .......................... 15, 1163

Other:
  Additional Item [otherclass] .... 25
  Before \myenvironment ............ 25
  license agreement ................ 25
  Other Item [otherclass] ........ 25
[otherclass]:
  Additional Item .................. 25
  Other Item  ....................... 25
  \otherenvironment (env.) ....... 20

P
Packages:
  calc .......................... 33
  caption .......................... 33
  etoolbox ........................ 33
  fancyverb ........................ 33
  hyperref ........................ 33
  makeidx .......................... 33
  newfloat .......................... 33
  package_name [descexamples] ... 26
  pict2e .......................... 35
  samplepackage [\examples] ...... 22
  splitidx ........................ 33
  xcolor .......................... 33
  xparse .......................... 33
  xstring ........................ 33
\parg .......................... 9, 15, 1159