Towards Better LaTeX Documentation With the `ltxguidex` Document Class

Rebecca Turner*

2019/04/15 0.2.0

Abstract

The `ltxguidex` document class extends `ltxguide` with a set of environments and commands that make writing beautiful LaTeX documentation easier and more natural.

`ltxguidex` is licensed under the LPPL version 1.3c, or any later version at your choosing.

This document is written with the `ltxguidex` document class.

NOTE This release of `ltxguidex` is an experimental public beta; it intends to demonstrate a hopeful new direction without committing to a stable public API.

Although `ltxguidex` is now suitable for use in your own documentation, do not be surprised if future versions break your docs.

NOTE Browse the sources, contribute, or complain at github.com/9999years/ltxguidex

Contents

*Brandeis University; rebeccaturner@brandeis.edu

1 The state of the docs

LaTeX documentation is easy enough to write that — in general — nobody has bothered to package the improvements made to the LaTeXdocumentation systems. If one examines the documentation for their favorite package, they'll likely find a few command definitions that make some aspect of documentation writing more ergonomic. In the case of complex packages like `listings` or — in an extreme case — `pgf`, it's commonplace to see packages define their own internal documentation packages containing hundreds-to-thousands of lines of documentation macros.

This class repackages useful macros from various packages' documentation, often changing their form (e.g. the macro's interface) but keeping its style. I've tried to balance versatility against specialization (i.e. determining which features are the *most* useful) as well as balancing short with descriptive names.

LaTeX documentation is enabled with two document classes and several packages. Document classes include:

`ltxdoc` Defines very little other than a few shorthands for documenting commands. Designed to be integrated with the DOCSTRIP system, but I've seen plenty of `.dtx` files documented with `ltxguide`. However, I haven't yet used DOCSTRIP, so my experience here is limited.

`ltxguide` Provides several ergonomic features absent in `ltxdoc`. However, `ltxguide` is almost entirely undocumented, a fact which is partially mitigated by the fact that it's only about 150 lines long. `ltxguidex` is, as the name implies, based on `ltxguide`.

And supporting packages include:

`hypdoc` One of many, many packages by Heiko Oberdiek. `hypdoc` undertakes the ambitious task of patching the `doc` package in order to generate better indexes. In my experience, `hypdoc` is not compatible with `ltxguide`; as such, it isn't loaded in `ltxguidex`.

`doctools` Provides many useful secondary commands such as `\ltxclass`, `\package`, and so on. Many are duplicated here.

`showexpl` Provides the `LTXexample` environment which typesets LaTeX code and displays it in a listing side-by-side. `showexpl` provides the functionality of `listings`' `\lstsample` command and more. `showexpl` does, however, rely on the fairly hefty `listings` package.

Compare to more "plain" LaTeX documentation, `ltxguidex` documentation can be expected to compile somewhat slower. This author is of the opinion that the improvements are so numerous that the slow-down is worth it.

2 The `ltxguidex` document class

Although `ltxguidex` provides many useful commands, much of its utility is in its aesthetics. Much LaTeX documentation is very ugly because producing beautiful documentation requires significantly more code than most package authors are interested in writing. This document is written with `ltxguidex` and one package loaded (the `bera` font family). Because `ltxguidex` is written with inherent beauty, it ends up being a bit heavier than its competitors; notably, it loads `xcolor`, `listings`, `graphicx`, and `calc` by default.

3 A note on typefaces

This document is set in Tiempos Text and Fira Sans (available on CTAN as `fira`).

For your own documents, I would recommend `bera` or `plex`, although neither has small caps, which I consider essential.

When deciding on a serif font for LaTeX documentation, I would recommend picking one with a tall x-height, as larger overall glyphs makes documents easier to read on small screens (nobody's going to be printing out your documentation). This will rule out most old-style serif typefaces, such as Garamond and Calson.

4 Commands provided by `ltxguide`

In `ltxguide`, pipe characters (`|`) mark verbatim text.

However, between two pipes, the angle brackets < and > typeset as pretty angle brackets with regular italics between them; therefore, `|<package>|` typesets as ⟨*package*⟩.

To write literal angle brackets, simply double the characters; `|<<|` will render as < and `|>>|` will render as >.

`\pipe`
`\bs`

To write literal pipe characters, use the `\pipe` command. To write a literal backslash, use the `\bs` command.

|	`\pipe \\`
|	`\texttt{\pipe} \\`
/	`\textit{\pipe} \\`
|	`\textbf{\texttt{\pipe}} \\`
\	`\bs \\`
\	`\texttt{\bs} \\`
|	`\textit{\bs} \\`
****	`\textbf{\texttt{\bs}}`

`ltxguide` uses `shortvrb` to activate pipes as a synonym for short-verbatim material. There are some small conflicts with `ltxguidex`'s use of the `listings` package (in particular, pipes are silently gobbled from `lstlistings` environments, although they work normally within `verbatim`), which will hopefully be resolved with a coming change to `listings`; this simply depends on how quickly Jobst Hoffmann emails me back.

`ltxguide` also provides the `decl` environment that powers the `desc` environment.

`\m{`⟨*placeholder*⟩`}`
`\meta{`⟨*placeholder*⟩`}`

Prints ⟨*placeholder*⟩ in italics within angle-brackets.

`ltxguidex` provides `\meta` as a synonym for `\m`.

⟨*placeholder*⟩	`\m{placeholder}`

`\arg{`⟨*argument*⟩`} \oarg{`⟨*argument*⟩`}`

Shorthands for mandatory and optional arguments.

{⟨*foo*⟩}[⟨*bar*⟩]

```
\arg{foo}\oarg{bar}
```

`\NFSS`	NFSS
`\AmS`	$\mathcal{A}\mathcal{M}\mathcal{S}$
`\AmSLaTeX`	$\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX
`\babel`	babel
`\SLiTeX`	SLiTeX
`\ctanlogo`	CTAN

Various logos.

NOTE `ltxguide` actually defines the CTAN logo as `\ctan`, but this class uses `\ctan` to refer to a package, so the CTAN logo is typeset with `\ctanlogo`.

`\clsguide`	LaTeX 2ε for Class and Package Writers
`\usrguide`	LaTeX 2ε for Authors
`\fntguide`	LaTeX 2ε Font Selection
`\cfgguide`	Configuration options for LaTeX 2ε
`\cyrguide`	Cyrillic languages support in LaTeX
`\modguide`	Modifying LaTeX
`\sourcecode`	LaTeX: the program
`\LaTeXbook`	LaTeX: A Document Preparation System
`\LaTeXcomp`	The LaTeX Companion
`\LaTeXGcomp`	The LaTeX Graphics Companion
`\LaTeXWcomp`	The LaTeX Web Companion

The names of various documents, presumably intended only for the original `ltxguide` document.

`\eg`
`\ie`

Shortcuts for "e.g.," and "i.e.," followed by a non-breaking space.

i.e., the document class...
e.g., the package...

```
\ie the document class\dots\\
\eg the package\dots
```

`\NEWfeature{`⟨*version*⟩`}`
`\NEWdescription{`⟨*version*⟩`}`

Typeset their arguments in a `\marginpar`. This paragraph is prepended by `\NEWfeature{1.0.0} \NEWdescription{1.0.0}`.

New feature 1.0.0
New description 1.0.0

`\URL{`⟨*url*⟩`}`

Typesets its argument in `\texttt`. Obsolete given that `ltxguidex` loads `hyperref`.

5 New commands

`ltxguidex` provides several new commands for convenience.

`\begin{desc}...\end{desc}`

Describes a command or environment, setting it out into the margin and surrounding it with a frame. Originally written by Javier Bezos for the `enumitem` documenation.

EXAMPLE Unfortunately, a side-by-side listing doesn't seem to be possible here because pipes seem to be gobbled by the `listings` package (a side-effect of loading both `listings` and `shortvrb`, perhaps). However, here's how the `\email` command is described in this document:

```
\begin{desc}
|\email{<email>}|
\end{desc}
```

`\email{`⟨*email*⟩`}`

Typesets an email address with a `mailto:` link.

EXAMPLE Emails, along with other hyperlinks, are colored `magenta`, although `ltxguidex`'s default magenta is a bit closer to purple.

rebeccaturner@brandeis.edu `\email{rebeccaturner@brandeis.edu}`

`\https{`⟨*url*⟩`}` `\http{`⟨*url*⟩`}`

Typesets ⟨*url*⟩ with `https://` or `http://` prepended to the link address; this makes links display a bit prettier than `\url` might.

EXAMPLE The following two listings are equivalent:

ctan.org `\https{ctan.org}`

ctan.org `\href{https://ctan.org}{ctan.org}`

`\ctan{`⟨*package*⟩`}`
`\ctanlogo`

Typesets a package name with a link to `ctan.org/pkg/`⟨*package*⟩.

WARNING `ltxguidex`'s definition of `\ctan` differs from `ltxguide`'s, which simply typesets "CTAN" in small-caps. The CTAN logo is typeset with `\ctanlogo`.

the `listings` package... `the \ctan{listings} package\dots`

`\package{`⟨*package*⟩`}`
`\ltxclass{`⟨*document class*⟩`}`
`\option{`⟨*option name*⟩`}`
`\filename{`⟨*filename*⟩`}`
`\extension{`⟨*file extension*⟩`}`

Typesets a LaTeX package, option, file extension, etc. in `\texttt`.

NOTE Unlike those defined in the `doctools` package, these commands don't add entries to the index.

`.tex` files `\extension{tex} files`

`\begin{warning}...\end{warning}`
`\begin{note}...\end{note}`
`\begin{example}...\end{example}`
`\begin{bug}...\end{bug}`

These environments typeset "notices" with a hanging indent. Original definitions written by Javier Bezos for the `enumitem` documenation. `\noticestyle` is executed before the marker text ("warning," "note," etc.). New notice environments can be created with `\newnotice`.

BUG If the first content in a notice environment is vertical, the marker text is hidden. This can be avoided by starting the environment with `\leavevmode\\` or by adding some introductory material to the first line.

This is actually a bug in the `\list` command that the notice environments use.

EXAMPLE Although this example is short, note that subsequent lines will be indented. These environments only vary by text.

WARNING Lorem ipsum...

```
\begin{warning}
    Lorem ipsum\dots
\end{warning}
```

`\newnotice{`⟨*environment name*⟩`}{`⟨*marker text*⟩`}`

Creates a new notice environment in the style of `warning`, `note`, and so on. The marker text is automatically uppercased.

`\begin{LTXexample}[`⟨*options*⟩`]...\end{LTXexample}`

Typesets LaTeX code next to a listing of its source. Providing examples makes your user's lives easier, and should be done as much as possible. The `LTXexample` environment is provided by the `showexpl` package. Excerpted from `showexpl`'s documentation as of v0.3o 2016/12/11, valid options include:

`attachfile`=⟨`true`|`false`⟩ `false`

If set to true the sourcecode will be attached to the `.pdf` file—presumed that the document is processed by `pdflatex`.

codefile=⟨*filename*⟩ `\jobname.tmp`

Name of the (temporary) file that contains the code which will be formatted as source code. The default value is `\jobname.tmp`.

explpreset=⟨*key val list*⟩ `{language=[LaTeX]TeX,}`

A ⟨*key val list*⟩ which serves for presetting the properties of the formatting of the source code, for values see the documentation of the `listings` package. The default value is empty.[1]

graphic=⟨*filename*⟩

If present, includes and displays this file instead of the formatted code.

hsep=⟨*length*⟩

Defines the horizontal distance between the source code and the formatted text.

justification=⟨*code*⟩ `\raggedright`

Defines the justification of the formatted text: reasonable values are `\raggedleft`, `\raggedright`, `\centering`.

overhang=⟨*dimen*⟩ `0pt`

Defines the amount by which the formatted text and the source code can overlap the print space. The default value is 0 pt.

pos=⟨`t`|`b`|`l`|`r`|`o`|`i`⟩ `l`

Defines the relative position of the formatted text relating to the source code. Allowed values are `t`, `b`, `l`, `r`, `o`, and `i` for top, bottom, left, right, outer, and inner. The last values give sense only for two-sided printing, where there are outer and inner margins of a page.

preset=⟨*code*⟩

Any TEX code executed before the sample code but not visible in the listings area.

rangeaccept=⟨`true`|`false`⟩ `false`

If set to true, one can define ranges of lines that will be excerpted from the source code.

rframe=[`single`] *(empty)*

Defines the form of the frame around the formatted text. With a non-empty value (e. g. “single”) a simple frame will be drawn. In the future more kinds of frames will be supported. The default value is empty (no frame).

varwidth=⟨`true`|`false`⟩ `false`

If set to true, the formatted text is set with its “natural” width instead of a fixed width as given by the value of the option `width`.

vsep=⟨*dimen*⟩

Defines the vertical distance between the source code and the formatted text.

wide=⟨`true`|`false`⟩ `false`

If set to true, the source code and the formatted text overlap the print space and the margin area.

[1] `ltxguidex` redefines the default to perform syntax highlighting for LATEX, in addition to the general improvements made for all listings in the document.

width=⟨*dimen*⟩

Defines the width of the formatted text. The default value depends of the relative positions of the source code and the formatted text.

scaled=[⟨*scale factor*⟩]

Without a value the formatted text will be scaled to fit the given width of the result area. With a number as value the formatted text will be scaled by this number.

In addition to these options the kind of the result box (default: `\fbox`) can be changed. For example:

```
\renewcommand\ResultBox{\fcolorbox{green}{lightgray}}
\setlength\ResultBoxSep{5mm}%  default: \fboxsep
\setlength\ResultBoxRule{2mm}% default: \fboxrule
```

```
\begin{packages}...\end{packages}
\begin{classes}...\end{classes}
\begin{options}...\end{options}
```

Frequently, package authors need to describe a series of options, packages, or document classes. These environments wrap the `description` environment and provide an `\item` which wraps a command like `\package`. In the `packages` environment, `\item[listings]` translates to `\item[\package{listings}]`.

```
foo …

bar …
```

```
\begin{options}
    \item[foo] \dots
    \item[bar] \dots
\end{options}
```

```
\begin{advise}...\end{advise} ≡
\begin{faq}...\end{faq}
\Q   \A   \advisespace
```

Roughly copied from `listings`' internal `lstdoc` package, these environments represent a list of questions and answers.

→ Lorem ipsum dolor sit amet? Consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

→ Ut enim ad minim veniam, quis nostrud? Exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

```
\begin{faq}
\Q Lorem ipsum dolor sit amet?
\A Consectetur adipiscing elit,
   sed do eiusmod tempor
   incididunt ut labore et
   dolore magna aliqua.

\Q Ut enim ad minim veniam, quis
   nostrud?
\A Exercitation ullamco laboris
   nisi ut aliquip ex ea
   commodo consequat.
\end{faq}
```

Within these environments, `\Q` and `\A` indicate a question and an answer; they're defined to `\item` and `\advisespace`, respectively.

NOTE `faq` is an exact synonym for `advise`.

The list label for the `advise` environment is `\labeladvise`.

The font is set with `\advisestyle`.

`\alternative{`⟨*comma list*⟩`}`

Prints a comma-separated list delimited by vertical bars. Space around commas is not trimmed, and alternates are printed in `\textup{\texttt{...}}`.

This environment is from `lstdoc`.

`true|false`

```
\alternative{true,false}
```

```
\begin{keys}...\end{keys}
\key[⟨options⟩]{⟨key name⟩}[⟨key value⟩][⟨default value⟩]
\bool
```

Describes keys. Within a `keys` environment, `\bool` indicates a true/false value. This environment is a recreation of `lstdoc`'s `syntax` environment with a more elegant interface.

⟨*options*⟩ can include:

v=⟨*version*⟩

The version a key was introduced.

WARNING This key is currently ignored.

default=⟨*default value*⟩

An alias for the final argument; a default value if the key isn't given.

note=⟨*note*⟩

A note set in the left margin; might note a group of features or something else.

EXAMPLE Note the use of `\bool`:

addon

key=⟨*value*⟩ default

Lorem ipsum...

display=⟨`true|false`⟩ `true`

Lorem ipsum...

foo=⟨*foo*⟩

bar

Lorem ipsum...

```
\begin{keys}
  \key[note=addon]{key}
      [\m{value}][default]
  Lorem ipsum\dots

  \key{display}[\bool][true]
  Lorem ipsum\dots

  \key{foo}[\m{foo}]
  \key[v=1.3]{bar}
  Lorem ipsum\dots
\end{keys}
```

6 Changelog

0.1.1 Rebecca Turner (2019-04-15)

Added

- Renamed `\ltxguidex@noticestyle` to `\noticestyle` and committed it to the public API.
- The `\cs` and `\command` commands.