

The file `ltxdoc.dtx` for use with $\text{\LaTeX} 2_{\epsilon}$.*

It contains the code for `ltxdoc.cls`

David Carlisle

2021/12/07

This file is maintained by the \LaTeX Project team.
Bug reports can be opened (category `latex`) at
<https://latex-project.org/bugs.html>.

1 Documentation of the \LaTeX sources

This class file is designed for documenting the \LaTeX source files. You may however find it generally useful as a class for typesetting the documentation of files produced in ‘doc’ format.

Each documented file in the standard distribution comes with extension `dtx`. The appropriate class package or initex file will be extracted from the source by the docstrip system. Each `dtx` file may be directly processed with $\text{\LaTeX} 2_{\epsilon}$, for example

```
latex2e docclass.dtx
```

would produce the documentation of the Class and package interface.

Each file that is used in producing the $\text{\LaTeX} 2_{\epsilon}$ format (ie not including the standard class and packages) will be printed together in one document if you \LaTeX the file `sources2e.tex`. This has the advantage that one can produce a full index of macro usage across all the source files.

If you need to customise the typesetting of any of these files, there are two options:

- You can use DOCSTRIP with the module ‘driver’ to extract a small \LaTeX file that you may edit to use whatever class or package options you require, before inputting the source file.
- You can create a file `ltxdoc.cfg`. This configuration file will be read whenever the `ltxdoc` class is used, and so can be used to customise the typesetting of all the source files, without having to edit lots of small driver files.

The second option is usually more convenient. Various possibilities are discussed in the next section.

*This file has version number v2.1d, dated 2021/12/07.

2 Customisation

The simplest form of customisation is to pass more options to the `article` class which is loaded by `ltxdoc`. For instance if you wish all the documentation to be formatted for A4 paper, add the following line to `ltxdoc.cfg`:

```
\PassOptionsToClass{a4paper}{article}
```

All the source files are in two parts, separated by `\MaybeStop`. The first part (should) contain ‘user’ documentation. The second part is a full documented listing of the source code. The `doc` package provides the command `\OnlyDescription` which suppresses the code listings. This may also be used in the configuration file, but as the `doc` package is read later, you must delay the execution of `\OnlyDescription` until after the `doc` package has been read. The simplest way is to use `\AtBeginDocument`. Thus you could put the following in your `ltxdoc.cfg`.

```
\AtBeginDocument{\OnlyDescription}
```

If the full source listing `sources2e.tex` is processed, then an index and change history are produced by default, however indices are not normally produced for individual files.

As an example, consider `ltclass.dtx`, which contains the sources for the new class and package interface commands. With no `cfg` file, a 19 page document is produced. With the above configuration a slightly more readable document (4 pages) is produced.

Conversely, if you really want to read the source listings in detail, you will want to have an index. Again the index commands provided by the `doc` package may be used, but their execution must be delayed.

```
\AtBeginDocument{\CodelineIndex\EnableCrossrefs}
\AtEndDocument{\PrintIndex}
```

The `doc` package writes index files to be sorted using `MakeIndex` with the `gind` style, so one would then use a command such as

```
makeindex -s gind.ist ltclass.idx
```

and re-run \LaTeX .

Similarly to print a Change history, you would add

```
\AtBeginDocument{\RecordChanges}
\AtEndDocument{\PrintChanges}
```

to `ltxdoc.cfg`, and use `MakeIndex` with a command such as

```
makeindex -s gglo.ist -o ltclass.gls ltclass.glo
```

Finally if you do not want to list all the sections of `source2e.tex`, you can use `\includeonly` in the `cfg` file:

```
\includeonly{ltvers,ltboxes}
```

3 Options

```
1 \<class>
2 \DeclareOption{a5paper}{\@latexerr{Option not supported}%
3   {}}
```

Prevent loading of a config file.

```
4 \newif\ifltxdoc@load@cfg@ \ltxdoc@load@cfg@true
5 \DeclareOption{nocfg}{\ltxdoc@load@cfg@false}
6 \DeclareOption*{%
7   \PassOptionsToClass {\CurrentOption}{article}}
```

4 Option Processing

```
8 \ProcessOptions
```

5 Local configuration

Input a local configuration file, if it exists.

```
9 \ifltxdoc@load@cfg@
10 \InputIfFileExists{ltxdoc.cfg}
11     {\typeout{*****^J%
12               * Local config file ltxdoc.cfg used^J%
13               *****}}
14     {}
15 \else
16     \typeout{*****^J%
17             * Local config file ignored^J%
18             *****}
19 \fi
```

6 Loading article and doc

```
20 \LoadClass{article}
21 \RequirePackage{doc}
```

Make | be a ‘short verb’ character, but not in the document preamble, where an active character may interfere with packages that are loaded.

```
22 \AtBeginDocument{\MakeShortVerb{||}}
```

As ‘doc’ documents tend to have a lot of monospaced material, Set up some `tt` substitutions to occur silently.

```
23 \DeclareFontShape{OT1}{cmtt}{bx}{n}{<-> ssub * cmtt/m/n}{}
24 \DeclareFontFamily{OMS}{cmtt}{\skewchar\font 48} % '60
25 \DeclareFontShape{OMS}{cmtt}{m}{n}{<-> ssub * cmsy/m/n}{}
26 \DeclareFontShape{OMS}{cmtt}{bx}{n}{<-> ssub * cmsy/b/n}{}

```

This substitution is in the standard fd file, but not silent.

```
27 \DeclareFontShape{OT1}{cmss}{m}{it}{<->ssub*cmss/m/sl}{}
28 \CodelineNumbered
29 \DisableCrossrefs
```

Increase the text width slightly so that with the standard fonts 72 columns of code may appear in a `macrocode` environment.

```
30 \setlength{\textwidth}{355pt}
```

Increase the marginpar width slightly, for long command names. And increase the left margin by a similar amount

```
31 \addtolength\marginparwidth{30pt}
32 \addtolength\oddsidemargin{20pt}
33 \addtolength\evensidemargin{20pt}
34 \setcounter{StandardModuleDepth}{1}
```

7 Useful abbreviations

`\cmd{\foo}` Prints `\foo` verbatim. It may be used inside moving arguments. It can *not* be used to record commands that are defined as “`\outer`” nor is it possible to use it on conditionals such as `\iftrue` or defined by `\newif`. `\cs{foo}` already available with the `doc` package also prints `\foo`, for those who prefer that syntax. (This second form can be used to record all types of command so the above restrictions do not apply.

```
\cmd
\cs 35 \def\cmd#1{\cs{\expandafter\cmd@to\cs\string#1}}
      36 \def\cmd@to\cs#1#2{\char\number' #2\relax}
      37 %DeclareRobustCommand\cs[1]{\texttt{\char'\#1}}      % already part of doc.sty

\marg \marg{text} prints {\text}, ‘mandatory argument’.
      38 \providecommand\marg[1]{%
      39   {\ttfamily\char'\{ }\meta{#1}{\ttfamily\char'\}}

\oarg \oarg{text} prints [{text}], ‘optional argument’.
      40 \providecommand\oarg[1]{%
      41   {\ttfamily[]\meta{#1}{\ttfamily}}

\parg \parg{te,xt} prints (<te,xt>), ‘picture mode argument’.
      42 \providecommand\parg[1]{%
      43   {\ttfamily()\meta{#1}{\ttfamily}}}
```

8 Old Comments

The $\text{\LaTeX} 2_{\epsilon}$ sources contain a lot of code inherited from $\text{\LaTeX} 2.09$. The comments in this code were not designed to be typeset, and do not contain the necessary \LaTeX markup. The `oldcomments` environment typesets these comments, automatically sensing when any control sequence appears, and implicitly adding the `\verb`. This procedure does not produce particularly beautiful pages, but it allows us to fully document new sections, and have some form of typeset comments on all the old code.

Scan control names and put them in `tt`. Will actually (incorrectly) scan past `\\` but this does not matter as this is almost never followed by a letter in practice. (ie `\\foo`) would put `foo` in `\ttfamily`.

```
44 \def\oc@scan#1{%
45   \ifx\oc@bslash#1%
46                                     \egroup\let\next\oc@bslash\else
47   \ifcat a\noexpand#1%
```

```

48             #1\let\next\oc@scan\else
49 \ifx\oc@percent#1%
50             \def\next{\char'\%\egroup}%
51 \else
52             #1\let\next\egroup
53 \fi\fi\fi\next}

54 \def\oc@bslash{\bgroup\oc@ttf\char'\\\oc@scan}%

55 \def\oc@verb#1{%
56 \catcode'#1\active
57 \uccode'\~'#1%
58 \uppercase{\def~{\oc@ttf\char'#1}}}}

59 \begingroup
60 \obeyspaces%
61 \catcode'\/= \catcode'\\
62 /catcode'/\active
63 /catcode'<= \catcode'{%
64 /catcode'>= \catcode'}%
65 /catcode'/{\active%
66 /catcode'/}/active%
67 /gdef/oldc< \end{oldcomments}>%
68 /gdef/begmac< \begin{macrocode}>%
69 /gdef/obs</def <</oc@ttf/ >>>%
70 /endgroup%

71 \begingroup
72 \catcode'\/= \catcode'\\
73 \catcode'\\=13
74 /catcode'/|= \catcode'/%
75 /catcode'/%=13
76 /gdef/oldcomments{|
77 /makeatletter
78 /let/do\oc@verb/dospecials
79 /frenchspacing/@vobeyspaces/obs
80 /raggedright
81 /oc@verb/>|
82 /oc@verb/<|
83 /let\oc@bslash
84 /let%\oc@percent
85 /obeylines
86 /parindent/z@
87 /ttfamily/expandafter/let/expandafter/oc@ttf/the/font
88 /rmfamily
89 /textit{Historical /LaTeX/,2.09 comments (not necessarily accurate any more):}
90 /hfuzz/maxdimen
91 }
92 /endgroup

93 \begingroup
94 \sloppy%
95 \obeylines%
96 \gdef\oc@percent#1~M{%
97 \ifvmode%
98 \def\commentline{#1}%
99 \ifx\commentline\oldc%

```

```

100 \textit{End of historical \LaTeX\,2.09 comments.}
101 \end{oldcomments}%
102 \else%
103 \ifx\commentline\begmac%
104 \begin{macrocode}%
105 \else%
106 \leavevmode%
107 #1^~M%
108 \fi\fi%
109 \else%
110 {\oc@ttf\char'\%}#1^~M%
111 \fi}%
112 \endgroup%

```

9 DocInclude

```

113 \@addtoreset{CodelineNo}{part}

\DocInclude More or less exactly the same as \include, but uses \DocInput on a dtx file, not
\input on a tex file.

114 \def\partname{File}

115 \newcommand*{\DocInclude}[1]{%
116   \relax
117   \clearpage
118   \docincludeaux
119   \IfFileExists{#1.fdd}%
120     {\def\currentfile{#1.fdd}}%
121     {\def\currentfile{#1.dtx}}%
122   \ifnum\@auxout=\@partaux
123     \@latexerr{\string\include\space cannot be nested}\@eha
124   \else
125     \set@curr@file{#1}%
126     \edef\@curr@file{\@strip@tex@ext\@curr@file}%
127     \expandafter\@docinclude\expandafter{\@curr@file}
128   \fi}
129 \def\@docinclude#1 {\clearpage

130 \if@filesw \immediate\write\@mainaux{\string\input{#1.aux}}\fi
131 \@tempwattrue\if@partsw \@tempwafalse\edef\@tempb{#1}\@for
132 \@tempa:=\@partlist\do{\ifx\@tempa\@tempb\@tempwattrue\fi}\fi
133 \if@tempswa \let\@auxout\@partaux \if@filesw
134 \immediate\openout\@partaux "#1.aux"
135 \immediate\write\@partaux{\relax}\fi
136 \@filehook\set@CurrentFile

```

We need to save (and later restore) various index-related commands which might be changed by the included file.

```

137 \let\@ltxdoc@PrintIndex\PrintIndex
138 \let\PrintIndex\relax
139 \let\@ltxdoc@PrintChanges\PrintChanges
140 \let\PrintChanges\relax
141 \let\@ltxdoc@theglossary\theglossary
142 \let\@ltxdoc@endtheglossary\endtheglossary
143 \part{\currentfile}%

```

```

144 {\let\ttfamily\relax
145 \xdef\filekey{\filekey, \thepart={\ttfamily\currentfile}}}%
146 \DocInput{\currentfile}%
147 \let\PrintIndex\@ltxdoc@PrintIndex
148 \let\PrintChanges\@ltxdoc@PrintChanges
149 \let\theglossary\@ltxdoc@theglossary
150 \let\endtheglossary\@ltxdoc@endtheglossary
151 \clearpage
152 \@writeckpt{#1}\if@filesw \immediate\closeout\@partaux \fi
153 \else\@nameuse{cp@#1}\fi\let\@auxout\@mainaux}

154 \gdef\codeline@wrindex#1{\if@filesw

```

Set `\protect` to a suitable value in the index entries (we can't use `\set@display@protect` as that would result in different number of spaces after a command depending on the number of expansion happening prior to writing the index).

```

155     \begingroup
156     \let\protect\noexpand
157     \immediate\write\@indexfile
158         {\string\indexentry{#1}%
159         {\filesep\number\c@CodelineNo}}}%
160     \endgroup\fi}

161 \let\filesep\@empty

```

`\aalph` Special form of `\alph` as currently `source2e.tex` includes more than 26 files .

```

162 \def\aalph#1{\@aalph{\csname c@#1\endcsname}}
163 \def\@aalph#1{%
164     \ifcase#1\or a\or b\or c\or d\or e\or f\or g\or h\or i\or
165         j\or k\or l\or m\or n\or o\or p\or q\or r\or s\or
166         t\or u\or v\or w\or x\or y\or z\or A\or B\or C\or
167         D\or E\or F\or G\or H\or I\or J\or K\or L\or M\or
168         N\or O\or P\or Q\or R\or S\or T\or U\or V\or W\or
169         X\or Y\or Z\else\@ctrerr\fi}

```

`\docincludeaux`

```

170 \def\docincludeaux{%
171     \def\thepart{\aalph{part}}\def\filesep{\thepart-}%
172     \let\filekey\@gobble
173     \g@addto@macro\index@prologue{%
174         \gdef\@oddfoot{\parbox[t]{\textwidth}{\strut\footnotesize
175             \raggedright{\bfseries File Key:} \filekey}}%
176         \let\@evenfoot\@oddfoot}%
177     \global\let\docincludeaux\relax
178     \gdef\@oddfoot{%
179         \expandafter\ifx\csname ver@\currentfile\endcsname\relax
180             File \thepart: {\ttfamily\currentfile} %
181         \else
182             \GetFileInfo{\currentfile}%
183             File \thepart: {\ttfamily\filename} %
184             Date: \filedate\ %
185             Version \fileversion
186             \fi
187             \hfill\thepage}%
188     \let\@evenfoot\@oddfoot}%

```

```

\MaintainedByLaTeXTeam Generate boilerplate reference to bug database.
189 \def\MaintainedBy#1{\gdef\@maintainedby{#1}}
190 \let\@maintainedby\@empty
191 \def\MaintainedByLaTeXTeam#1{%
192 {\gdef\@maintainedby{%
193 This file is maintained by the \LaTeX{} Project team.\\%
194 Bug reports can be opened (category \texttt{#1}) at\\%
195 \url{https://latex-project.org/bugs.html}.}}}%
196 \def\@maketitle{%
197   \newpage
198   \null
199   \vskip 2em%
200   \begin{center}%
201     \let \footnote \thanks
202     {\LARGE \@title \par}%
203     \vskip 1.5em%
204     {\large
205       \lineskip .5em%
206       \begin{tabular}[t]{c}%
207         \@author
208         \end{tabular}\par}%
209     \vskip 1em%
210     {\large \@date}%
211     \ifx\@maintainedby\@empty
212     \else
213     \vskip 1em%
214     \fbox{\fbox{\begin{tabular}{@{}l@{}}\@maintainedby\end{tabular}}}%
215     \fi
216   \end{center}%
217   \par
218   \vskip 1.5em}

\url
219 \providecommand\url{\texttt}

220 \def\task#1#2{}
221 \end{class}

```

10 Configuration file

```

222 <*cfg>
223 %
224 % This is the ltxdoc configuration file we use to format the LaTeX
225 % kernel sources.
226 %
227 %
228 % Copyright 2006, 2007, 2011 Heiko Oberdiek
229 % Copyright 2014-2021 The LaTeX Project
230 %
231
232 \ProvidesFile{ltxdoc.cfg}%
233 [2021/12/07 v2.0c ltxdoc.cls configuration (LaTeX Project)]

```



```

234 \PassOptionsToClass{a4paper}{article}
235
236 % hyperref and hypdoc are now loaded late (or by the user) so we have to wait
237 % with any adjustments until that has happened
238
239 \AddToHook{package/hyperref/after}{%
240 %% \RequirePackage{hypdoc}%           % this is now triggered by doc
241 \RequirePackage{pdftexcmds}\relax
242 \ifnum\pdf@strcmp{\jobname}{inputenc}=0 %
243 \hypersetup{pdfencoding=auto}%
244 \pdfstringdefDisableCommands{%
245 \def\meta#1{% inputenc.dtx
246 \9060\010#1\9060\011%
247 }%
248 }%
249 \else
250 \fi
251 \pdfstringdefDisableCommands{%
252 \let\env\relax % longtable.dtx
253 \let\mytt\relax % tabularx.dtx
254 }%
255 }
256
257 % This should work well for documentation of packages outside the
258 % LaTeX kernel, but if not, you can prevent the loading with the
259 % option "nocfg", i.e.,
260 %
261 % \documentclass[nocfg]{ltxdoc}
262 %
263 % or by providing your own config file
264
265 \endinput
266 </cfg>

```