

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

It contains the code for `ltxdoc.cls`

David Carlisle

2022/06/03

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 v2.1f, dated 2022/06/03.

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 your document relies on using the old `doc` version, you can request that the class loads `doc` version 2 by passing the option `doc2`.

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

Support rolling back doc to version 2:
6 \let\ltxdoc@doc@version\@empty % use current version by default
7 \DeclareOption{doc2}{\def\ltxdoc@doc@version{=v2}}

8 \DeclareOption*{%
9   \PassOptionsToClass {\CurrentOption}{article}}

```

4 Option Processing

```

10 \ProcessOptions

```

5 Local configuration

Input a local configuration file, if it exists.

```

11 \ifltxdoc@load@cfg@
12 \InputIfFileExists{ltxdoc.cfg}
13     {\typeout{*****^J%
14               * Local config file ltxdoc.cfg used^J%
15               *****}}
16     {}
17 \else
18     \typeout{*****^J%
19               * Local config file ignored^J%
20               *****}
21 \fi

```

6 Loading article and doc

```

22 \LoadClass{article}

```

By default, load the current doc version (`\ltxdoc@doc@version` is empty). If option `doc2` is given version 2 is loaded (`\ltxdoc@doc@version` contains `=v2`).

```

23 \RequirePackage{doc}[\ltxdoc@doc@version]

```

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

```

24 \AtBeginDocument{\MakeShortVerb{\|}}

```

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

```

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

```

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

```

29 \DeclareFontShape{OT1}{cmss}{m}{it}{<->ssub*cmss/m/sl}{}

```

```
30 \CodelineNumbered
31 \DisableCrossrefs
```

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

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

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

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

7 Useful abbreviations

`\cmd{\foo}` Prints `\foo` verbatim. It may be used inside moving arguments. It can *not* be use 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 37 \def\cmd#1{\cs{\expandafter\cmd@to@cs\string#1}}
    38 \def\cmd@to@cs#1#2{\char\number' #2\relax}
    39 %\DeclareRobustCommand\cs[1]{\texttt{\char'\#1}}      % already part of doc.sty

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

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

\parg \parg{te,xt} prints (<te,xt>), ‘picture mode argument’.
    44 \providecommand\parg[1]{%
    45   {\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`.

```

46 \def\oc@scan#1{%
47   \ifx\oc@bslash#1%
48     \egroup\let\next\oc@bslash\else
49   \ifcat a\noexpand#1%
50     #1\let\next\oc@scan\else
51   \ifx\oc@percent#1%
52     \def\next{\char'\%\egroup}%
53   \else
54     #1\let\next\egroup
55   \fi\fi\fi\next}

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

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

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

73 \begingroup
74   \catcode'\!=\catcode'\
75   \catcode'\|=13
76   /catcode'/|=catcode'/%
77   /catcode'/%=13
78   /gdef/oldcomments{|
79     /makeatletter
80     /let/do/oc@verb/dospecials
81     /frenchspacing/@vobeyspaces/obs
82     /raggedright
83     /oc@verb/>|
84     /oc@verb/<|
85     /let\oc@bslash
86     /let%\oc@percent
87     /obeylines
88     /parindent/z@
89     /ttfamily/expandafter/let/expandafter/oc@ttf/the/font
90     /rmfamily
91     /textit{Historical /LaTeX/,2.09 comments (not necessarily accurate any more):}
92     /hfuzz/maxdimen
93   }
94 /endgroup

95 \begingroup
96   \sloppy%
97   \obeylines%

```

```

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

```

9 DocInclude

```

115 \@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.

```

116 \def\partname{File}
117 \newcommand*{\DocInclude}[1]{%
118   \relax
119   \clearpage
120   \docincludeaux
121   \IfFileExists{#1.fdd}%
122   {\def\currentfile{#1.fdd}}%
123   {\def\currentfile{#1.dtx}}%
124   \ifnum\@auxout=\@partaux
125     \@latexerr{\string\include\space cannot be nested}\@eha
126   \else
127     \set@curr@file{#1}%
128     \edef\@curr@file{\@strip@tex@ext\@curr@file}%
129     \expandafter\@docinclude\expandafter{\@curr@file}
130   \fi}
131 \def\@docinclude#1 {\clearpage
132 \if@filesw \immediate\write\@mainaux{\string\@input{#1.aux}}\fi
133 \@tempswtrue\if@partsw \@tempswafalse\edef\@tempb{#1}\@for
134 \@tempa:=\@partlist\do{\ifx\@tempa\@tempb\@tempswtrue\fi}\fi
135 \if@tempswa \let\@auxout\@partaux \if@filesw
136 \immediate\openout\@partaux "#1.aux"
137 \immediate\write\@partaux{\relax}\fi
138 \@filehook@set@CurrentFile

```

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

```

139 \let\@ltxdoc@PrintIndex\PrintIndex
140 \let\PrintIndex\relax
141 \let\@ltxdoc@PrintChanges\PrintChanges

```

```

142 \let\PrintChanges\relax
143 \let\@ltxdoc@theglossary\theglossary
144 \let\@ltxdoc@endtheglossary\endtheglossary
145 \part{\currentfile}%
146 {\let\ttfamily\relax
147 \xdef\filekey{\filekey, \thepart={\ttfamily\currentfile}}}%
148 \DocInput{\currentfile}%
149 \let\PrintIndex\@ltxdoc@PrintIndex
150 \let\PrintChanges\@ltxdoc@PrintChanges
151 \let\theglossary\@ltxdoc@theglossary
152 \let\endtheglossary\@ltxdoc@endtheglossary
153 \clearpage
154 \@writeckpt{#1}\if@filesw \immediate\closeout\@partaux \fi
155 \else\@nameuse{cp@#1}\fi\let\@auxout\@mainaux}
156 \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).

```

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

163 \let\filesep\@empty

```

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

```

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

```

`\docincludeaux`

```

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

```

```

187     Version \fileversion
188     \fi
189     \hfill\thepage}%
190 \let\@evenfoot\@oddfoot}%

```

\MaintainedByLaTeXTeam Generate boilerplate reference to bug database.

```

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

```

\url

```

221 \providecommand\url{\texttt}

222 \def\task#1#2{}

Some feaures from l3doc.cls
223 \providecommand\LuaTeX{Lua\TeX}
224 \providecommand\cls{\textsf}
225 \providecommand\pkg{\textsf}
226 \providecommand\enquote[1]{‘‘#1’’}
227 \DeclareRobustCommand\cs[1]{\texttt{\bslash\detokenize{#1}}}%
228 \AtBeginDocument{%
229 \renewcommand\PrintMacroName[1]{\MacroFont\detokenize{#1}}%
230 }

231 \end{class}

```


10 Configuration file

```
232 <*cfg>
233 %
234 % This is the ltxdoc configuration file we use to format the LaTeX
235 % kernel sources.
236 %
237 %
238 % Copyright 2006, 2007, 2011 Heiko Oberdiek
239 % Copyright 2014-2021 The LaTeX Project
240 %
241
242 \ProvidesFile{ltxdoc.cfg}%
243 [2021/12/07 v2.0c ltxdoc.cls configuration (LaTeX Project)]
244 \PassOptionsToClass{a4paper}{article}
245
246 % hyperref and hypdoc are now loaded late (or by the user) so we have to wait
247 % with any adjustments until that has happened
248
249 \AddToHook{package/hyperref/after}{%
250 %% \RequirePackage{hypdoc}%           % this is now triggered by doc
251 \RequirePackage{pdftexcmds}\relax
252 \ifnum\pdf@strcmp{\jobname}{inputenc}=0 %
253 \hypersetup{pdfencoding=auto}%
254 \pdfstringdefDisableCommands{%
255 \def\meta#1{% inputenc.dtx
256 \9060\010#1\9060\011%
257 }%
258 }%
259 \else
260 \fi
261 \pdfstringdefDisableCommands{%
262 \let\env\relax % longtable.dtx
263 \let\mytt\relax % tabularx.dtx
264 }%
265 }
266
267 % This should work well for documentation of packages outside the
268 % LaTeX kernel, but if not, you can prevent the loading with the
269 % option "nocfg", i.e.,
270 %
271 % \documentclass[nocfg]{ltxdoc}
272 %
273 % or by providing your own config file
274
275 \endinput
276 </cfg>
```