

The file `ltxdoc.dtx` for use with  $\text{\LaTeX} 2_{\epsilon}$ .<sup>\*</sup>  
It contains the code for `ltxdoc.cls`

David Carlisle

2022/06/22

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

## 1 Documentation of the $\text{\LaTeX}$ sources

This class file is designed for documenting the  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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.

---

<sup>\*</sup>This file has version number v2.1i, dated 2022/06/22.

## 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 %\DeclareRobustCommand\cs[1]...           % defined later
    38 %\def\cmd#1{\cs{\expandafter\cmd@to@cs\string#1}} % can't use with new \cs
    39 \def\cmd#1{\texttt{\char'\'\expandafter\cmd@to@cs\string#1}}
    40 \def\cmd@to@cs#1#2{\char\number'#2\relax}

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

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

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

## 8 Old Comments

The  $\LaTeX 2_{\epsilon}$  sources contain a lot of code inherited from  $\LaTeX 2.09$ . 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.

```

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

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

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

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

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

```

```

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

```

## 9 DocInclude

```
116 \@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.

```

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

```

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

```

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

157 \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).

```

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

164 \let\filesep\@empty

```

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

```

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

```

`\docincludeaux`

```

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

```

```

185 \GetFileInfo{\currentfile}%
186 File \thepart: {\ttfamily\filename} %
187 Date: \filedate\ %
188 Version \fileversion
189 \fi
190 \hfill\thepage}%
191 \let\@evenfoot\@oddfoot}%

```

`\MaintainedByLaTeXTeam` Generate boilerplate reference to bug database.

```

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

```

Some features from `l3doc.cls` Eventually, `\cs` should get the definition from `l3doc` but for now we revert to the simple one from `doc`.

```

223 %\DeclareRobustCommand\cs[1]{\texttt{\backslash #1}}% -- def in doc.sty
224 \AtBeginDocument{%
225 % \renewcommand\PrintMacroName[1]{\MacroFont\string #1\ }% -- def in doc.sty

```

We provide those delated in case somebody has loaded `csquotes` or makes some definitions in the preamble.

```

226 \providecommand\LuaTeX{Lua\TeX}
227 \providecommand\cls{\textsf}
228 \providecommand\pkg{\textsf}

```

```

229 \providecommand\enquote[1]{‘‘#1’’}
230 \providecommand\url{\texttt}
231 }
232 </class>

```

## 10 Configuration file

```

233 <*cfg>
234 %
235 % This is the ltxdoc configuration file we use to format the LaTeX
236 % kernel sources.
237 %
238 %
239 % Copyright 2006, 2007, 2011 Heiko Oberdiek
240 % Copyright 2014-2021 The LaTeX Project
241 %
242
243 \ProvidesFile{ltxdoc.cfg}%
244 [2022/06/14 v2.0d ltxdoc.cls configuration (LaTeX Project)]
245 \PassOptionsToClass{a4paper}{article}
246
247 % hyperref and hypdoc are now loaded late (or by the user) so we have to wait
248 % with any adjustments until that has happened
249
250 \AddToHook{package/hyperref/after}{%
251 %% \RequirePackage{hypdoc}% % this is now triggered by doc
252 \RequirePackage{pdftexcmds}\relax
253 \ifnum\pdf@stricmp{\jobname}{inputenc}=0 %
254 \hypersetup{pdfencoding=auto}%
255 \pdfstringdefDisableCommands{%
256 \def\meta#1{% inputenc.dtx
257 \9060\010#1\9060\011%
258 }%
259 }%
260 \else
261 \fi
262 \pdfstringdefDisableCommands{%
263 \let\env\relax % longtable.dtx
264 \let\mytt\relax % tabularx.dtx
265 }%
266 }
267
268 % This should work well for documentation of packages outside the
269 % LaTeX kernel, but if not, you can prevent the loading with the
270 % option "nocfg", i.e.,
271 %
272 % \documentclass[nocfg]{ltxdoc}
273 %
274 % or by providing your own config file
275
276 \endinput
277 </cfg>

```