

# Extensions to the `ltxdoc` class <sup>\*†</sup>

Arthur Ogawa <sup>‡</sup>

December 27, 2018

This file embodies the `ltxdocext` package, the implementation and its user documentation.

The distribution point for this work is [journals.aps.org/revtex](http://journals.aps.org/revtex), which contains prebuilt runtime files, documentation, and full source, ready to add to a TDS-compliant  $\text{\TeX}$  installation.

The `ltxdocext` package was commissioned by the American Physical Society and is distributed under the terms of the  $\text{\LaTeX}$  Project Public License, the same license under which all the portions of  $\text{\LaTeX}$  itself are distributed. Please see <http://ctan.tug.org/macros/latex/base/lppl.txt> for details.

To use this document class, you must have a working  $\text{\TeX}$  installation equipped with  $\text{\LaTeX} 2_{\epsilon}$  and possibly `pdftex` and Adobe Acrobat Reader or equivalent.

To install, retrieve the distribution, unpack it into a directory on the target computer, and move the files `ltxdocext.sty` and `acrofont.sty` into a location in your filesystem where they will be found by  $\text{\LaTeX}$ .

If you will be using the `acrofont` package, you must also install the virtual fonts `zpsynocmr`, `zptmno`, `zptmno`, and `zpcnocmry`. The corresponding `.tfm`, `.vf`, and `.vpl` files are part of this distribution.

To use, read the user documentation `ltxdocext.pdf`. The `.dtx` file, `ltxdocext.dtx`, constitutes in itself an instance of use of the `ltxdocext` package and the `acrofont` package.

## Contents

<b>1</b>	<b>Processing Instructions</b>	<b>3</b>
1.1	Build Instructions . . . . .	3
1.2	Change Log . . . . .	3
1.3	Bill of Materials . . . . .	3
1.3.1	Primary Source . . . . .	4
1.3.2	Generated by <code>latex ltxdocext.dtx</code> . . . . .	4
1.3.3	Generated by <code>tex ltxdocext.ins</code> . . . . .	4
1.3.4	Documentation . . . . .	4

---

<sup>\*</sup>This file has version number 1.0a, last revised 2018/12/26/17:10:03.

<sup>†</sup>Version 1.0a © 2019 American Physical Society

<sup>‡</sup><mailto:arthur.ogawa@sbcglobal.net>

1.3.5	Auxiliary . . . . .	4
<b>2</b>	<b>Code common to all modules</b>	<b>4</b>
<b>3</b>	<b>The driver module doc</b>	<b>5</b>
3.1	The Preamble . . . . .	5
3.1.1	Docstrip and info directives . . . . .	5
3.2	The “Read Me” File . . . . .	6
3.3	The Document Body . . . . .	8
<b>4</b>	<b>Using the ltxdoc and acrofont packages</b>	<b>8</b>
4.1	Extensions to the ltxdoc class . . . . .	9
4.1.1	Extensions to the verbatim environment and \verb command	9
4.1.2	The -matter Commands Work . . . . .	9
4.1.3	The \GetFileInfo command . . . . .	9
4.1.4	Self-Indexing Commands . . . . .	10
4.1.5	Unnumbered Tables . . . . .	10
4.1.6	Structuring Tables . . . . .	10
4.1.7	A Sectioning Command Below \subsection . . . . .	10
4.2	Alterations to the ltxdoc class . . . . .	11
<b>5</b>	<b>Extensions to the ltxdoc class</b>	<b>11</b>
5.1	Beginning of the package DOCSTRIP module . . . . .	11
5.2	Beginning of the kernel DOCSTRIP module . . . . .	11
5.3	Incorporate ltxguide.cls extensions . . . . .	11
5.4	Changes to the base class of the ltxdoc class . . . . .	14
5.5	Extensions to the base class of ltxdoc.cls . . . . .	15
5.6	In lieu of ltxdoc.cfg . . . . .	16
5.7	Extension to L <sup>A</sup> T <sub>E</sub> X’s filecontents Environment . . . . .	17
5.8	End of the kernel DOCSTRIP module . . . . .	18
5.9	Tail of the package DOCSTRIP module . . . . .	18
<b>6</b>	<b>Font Package for Acrobat Compatability</b>	<b>18</b>
6.1	Beginning of the fonts DOCSTRIP module . . . . .	18
6.2	Variants on psfonts packages . . . . .	19
6.3	Font definition files . . . . .	20
6.4	More math substitutions . . . . .	20
6.5	End of the fonts DOCSTRIP module . . . . .	20
<b>7</b>	<b>Programming Conventions</b>	<b>21</b>
7.1	Whitespace Conventions . . . . .	21
7.2	Conventions For Procedures . . . . .	22
7.3	Conventions For L <sup>A</sup> T <sub>E</sub> X . . . . .	23

# 1 Processing Instructions

The package files `ltxdocext.sty` and `acrofont.sty` are generated from this file, `ltxdocext.dtx`, using the DOCSTRIP facility of  $\text{\LaTeX}$  via `tex ltxdocext.dtx` (Note: do *not* use  $\text{\LaTeX}$  for this step). The typeset documentation that you are now reading is generated from the same file by typesetting it with  $\text{\LaTeX}$  or `pdftex` via `latex ltxdocext.dtx` or `pdflatex ltxdocext.dtx`.

## 1.1 Build Instructions

You may bootstrap this suite of files solely from `ltxdocext.dtx`. Prepare by installing  $\text{\LaTeX} 2_{\epsilon}$  (and either `tex` or `pdftex`) on your computer, then carry out the following steps:

1. Within an otherwise empty directory, typeset `ltxdocext.dtx` with  $\text{\TeX}$  or `pdftex`; thereby generating the package file `ltxdocext.sty`, and the package file `acrofont.sty`. Make sure that DOCSTRIP receives permission to overwrite existing versions of these packages.

2. Now typeset `ltxdocext.dtx` with  $\text{\LaTeX}$  or `pdflatex`; you will obtain the typeset documentation you are now reading, along with the file `README`.

Note: you will have to run  $\text{\LaTeX}$ , then `makeindex -s gind.ist ltxdocext.idx`, then `makeindex -s gglo.ist -o ltxdocext.gls ltxdocext.glo`, then  $\text{\LaTeX}$  again in order to obtain a valid index and table of contents.

3. Install the following files into indicated locations within your TDS-compliant `texmf` tree (you may need root access):

- `$TEXMF/tex/latex/revtex/ltxdocext.sty` and `$TEXMF/tex/latex/revtex/acrofont.sty`
- `$TEXMF/source/latex/revtex/ltxdocext.dtx`
- `$TEXMF/doc/latex/revtex/ltxdocext.pdf`

where `$TEXMF/` stands for `texmf-local/`, or some other `texmf` tree in your installation.

4. Run `mktexlsr` on `$TEXMF/` (you may need root access).
5. Build and installation are now complete; now put a `\usepackage{ltxutil}` in your document preamble!

## 1.2 Change Log

## 1.3 Bill of Materials

Following is a list of the files in this distribution arranged according to provenance.

### 1.3.1 Primary Source

One single file generates all.

```
%ltxdocext.dtx
%
```

### 1.3.2 Generated by latex ltxdocext.dtx

Typesetting the source file under pdf<sub>l</sub>atex generates the readme and the documentation.

```
%README ltxdocext.pdf
%
```

### 1.3.3 Generated by tex ltxdocext.ins

Typesetting this file with T<sub>E</sub>X generates the package file.

```
%ltxdocext.sty    acrofont.sty
%
```

### 1.3.4 Documentation

The following are the online documentation:

```
%ltxdocext.pdf
%
```

### 1.3.5 Auxiliary

The following are auxiliary files generated in the course of running L<sup>A</sup>T<sub>E</sub>X:

```
%ltxdocext.aux ltxdocext.idx ltxdocext.ind ltxdocext.log ltxdocext.toc
%
```

## 2 Code common to all modules

We want to require only one place in this file where the version number is stated, and we also want to ensure that the version number is embedded into every generated file.

Now we declare that these files can only be used with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. An appropriate message is displayed if a different T<sub>E</sub>X format is used.

```
1 %<*driver|package|fonts>
2 \NeedsTeXFormat{LaTeX2e}[1995/12/01]%
3 %</driver|package|fonts>
```

As desired, the following modules all take common version information:

```

4 %<package>\ProvidesFile{ltxdocext.sty}%
5 %<font>\ProvidesFile{acrofont.sty}%
6 %<*driver>
7 \expandafter\ProvidesFile\expandafter{\jobname.dtx}%
8 %</driver>

```

The following line contains, for once and for all, the version and date information. By various means, this information is reproduced consistently in all generated files and in the typeset documentation.

```

9 %<version>
10 [2018/12/26/17:10:03 1.0a ltxdoc extensions package]% \fileversion

```

### 3 The driver module doc

This module, consisting of the present section, typesets the programmer’s documentation, generating the README-LTXDOCEXT as required.

Because the only uncommented-out lines of code at the beginning of this file constitute the doc module itself, we can simply typeset the .dtx file directly, and there is thus rarely any need to generate the “doc” DOCSTRIP module. Module delimiters are nonetheless required so that this code does not find its way into the other modules.

The `\end{document}` command concludes the typesetting run.

```

11 %<*driver>

```

#### 3.1 The Preamble

The programmers documentation is formatted with the `ltxdoc` document class, with local customizations, and with the usual code line indexing.

```

12 \documentclass[draft]{ltxdoc}
13 \RequirePackage{ltxdocext}%
14 \RequirePackage[colorlinks=true,linkcolor=blue]{hyperref}%
15 %\expandafter\ifx\csname package@font\endcsname\@undefined\else
16 % \expandafter\RequirePackage\expandafter{\csname package@font\endcsname}%
17 %\fi
18 \CodelineIndex\EnableCrossrefs % makeindex -s gind.ist ltxdocext
19 \RecordChanges % makeindex -s gglo.ist -o ltxdocext.gls ltxdocext.glo

```

##### 3.1.1 Docstrip and info directives

We use so many DOCSTRIP modules that we set the `StandardModuleDepth` counter to 1.

```

20 \setcounter{StandardModuleDepth}{1}

```

The following command retrieves the date and version information from this file.

```

21 \expandafter\GetFileInfo\expandafter{\jobname.dtx}%

```

### 3.2 The “Read Me” File

As promised above, here is the contents of the “Read Me” file. That file serves a double purpose, since it also constitutes the beginning of the programmer’s documentation. What better thing, after all, to have appear at the beginning of the typeset documentation?

A good discussion of how to write a ReadMe file can be found in Engst, Tonya, “Writing a ReadMe File? Read This” *MacTech* October 1998, p. 58.

Note the appearance of the `\StopEventually` command, which marks the dividing line between the user documentation and the programmer documentation.

The usual user will not be asked to do a full build, not to speak of the bootstrap. Instructions for carrying these processes begin the programmer’s manual.

```
22 \begin{filecontents*}{README-LTXDOCEXT}
23 \title{%
24 Extensions to the \classname{ltxdoc} class%
25 \thanks{%
26 This file has version number \fileversion,
27 last revised \filedate.%
28 }%
29 \thanks{%
30 Version \fileversion\ \copyright\ 2019 American Physical Society
31 }%
32 }%
33 \author{%
34 Arthur Ogawa%
35 \thanks{\texttt{mailto:arthur\_ogawa at sbcglobal.net}}}%
36 }%
37 %\iffalse
38 % For version number and date,
39 % search on "\fileversion" in the .dtx file,
40 % or see the end of the README file.
41 %\fi
42
43 \maketitle
44
45 This file embodies the \classname{ltxdocext} package,
46 the implementation and its user documentation.
47
48 The distribution point for this work is
49 \url{journals.aps.org/revtex},
50 which contains prebuilt runtime files, documentation, and full source,
51 ready to add to a TDS-compliant \TeX\ installation.
52
53 The \classname{ltxdocext} package was commissioned by the American Physical Society
54 and is distributed under the terms of the \LaTeX\ Project Public License,
55 the same license under which all the portions of \LaTeX\ itself are distributed.
56 Please see \url{http://ctan.tug.org/macros/latex/base/lppl.txt} for details.
57
58 To use this document class, you must have a working
```

59 \TeX\ installation equipped with \LaTeX\  
 60 and possibly pdftex and Adobe Acrobat Reader or equivalent.  
 61  
 62 To install, retrieve the distribution,  
 63 unpack it into a directory on the target computer,  
 64 and move the files \file{ltxdocext.sty} and \file{acrofont.sty}  
 65 into a location in your filesystem where they will be found by \LaTeX.  
 66  
 67 If you will be using the \classname{acrofont} package, you must  
 68 also install the virtual fonts  
 69 \file{zpsynocmr}, \file{zptmncmr},  
 70 \file{zptmncmr}, and \file{zpzcnocmr}.  
 71 The corresponding \file{.tfm}, \file{.vf}, and \file{.vpl}  
 72 files are part of this distribution.  
 73  
 74 To use, read the user documentation \file{ltxdocext.pdf}.  
 75 The \file{.dtx} file, \file{ltxdocext.dtx}, constitutes  
 76 in itself an instance of use of the \classname{ltxdocext}  
 77 package and the \classname{acrofont} package.  
 78  
 79 \tableofcontents  
 80  
 81 \section{Processing Instructions}  
 82  
 83 The package files \file{ltxdocext.sty} and \file{acrofont.sty}  
 84 are generated from this file, \file{ltxdocext.dtx},  
 85 using the {\sc docstrip} facility of \LaTeX  
 86 via |tex ltxdocext.dtx| (Note: do \emph{not} use \LaTeX\ for this step).  
 87 The typeset documentation that you are now reading is generated from  
 88 the same file by typesetting it with \LaTeX\ or pdftex  
 89 via |latex ltxdocext.dtx| or |pdflatex ltxdocext.dtx|.

90  
 91 \subsection{Build Instructions}  
 92  
 93 You may bootstrap this suite of files solely from \file{ltxdocext.dtx}.  
 94 Prepare by installing \LaTeX\ (and either tex or pdftex) on your computer,

95 then carry out the following steps:

- 96 \begin{enumerate}
- 97 \item
- 98 Within an otherwise empty directory,
- 99 typeset \file{ltxdocext.dtx} with \TeX\ or pdftex;
- 100 thereby generating the package file \file{ltxdocext.sty},
- 101 and the package file \file{acrofont.sty}.
- 102 Make sure that {\sc docstrip} receives permission
- 103 to overwrite existing versions of these packages.
- 104 \item
- 105 Now
- 106 typeset \file{ltxdocext.dtx} with \LaTeX\ or pdflatex;
- 107 you will obtain the typeset documentation you are now reading,
- 108 along with

```

109 the file \file{README}.
110
111 Note: you will have to run \LaTeX, then
112 \file{makeindex} \texttt{-s gind.ist ltxdocext.idx}, then
113 \file{makeindex} \texttt{-s gglo.ist -o ltxdocext.gls ltxdocext.glo}, then
114 \LaTeX\ again in order to obtain a valid index and table of contents.
115 \item
116 Install the following files into indicated locations within your
117 TDS-compliant \texttt{texmf} tree (you may need root access):
118 \begin{itemize}
119 \item
120 \file{$TEXMF/}\file{tex/}\file{latex/}\file{revtex/}\classname{ltxdocext.sty} and
121 \file{$TEXMF/}\file{tex/}\file{latex/}\file{revtex/}\classname{acrofont.sty}
122 \item
123 \file{$TEXMF/}\file{source/}\file{latex/}\file{revtex/}\classname{ltxdocext.dtx}
124 \item
125 \file{$TEXMF/}\file{doc/}\file{latex/}\file{revtex/}\classname{ltxdocext.pdf}
126 \end{itemize}
127 where \file{$TEXMF/} stands for \file{texmf-local/}, or some other \texttt{texmf} tree
128 in your installation.
129 \item
130 Run \texttt{mktexlsr} on \file{$TEXMF/} (you may need root access).
131 \item
132 Build and installation are now complete;
133 now put a \cmd\usepackage\texttt{\{ltxutil\}} in your document preamble!
134 \end{enumerate}
135
136 \subsection{Change Log}
137 \changes{1.0a}{2018/12/12}{(MD) Updated name of README file and use standard fonts when typeset}
138
139
140 \end{filecontents*}

```

### 3.3 The Document Body

Here is the document body, containing only a `\DocInput` directive—referring to this very file. This very cute self-reference is a common `ltxdoc` idiom.

```

141 \begin{document}%
142 \expandafter\DocInput\expandafter{\jobname.dtx}%
143 \PrintChanges
144 \end{document}
145 %</driver>

```

## 4 Using the `ltxdoc` and `acrofont` packages

These packages are an adjunct to the standard  $\text{\LaTeX}$  `ltxdoc` class and may be simply invoked as follows:



```
%\documentclass[draft]{ltxdoc}
%\RequirePackage{ltxdocext}%
%\RequirePackage{acrofont}%
%\CodelineIndex\EnableCrossrefs
%
```

Your document should simply cleave to the standards of the `ltxdoc` class, with extensions and alterations as noted.

## 4.1 Extensions to the `ltxdoc` class

### 4.1.1 Extensions to the verbatim environment and `\verb` command

The delimiters `<<` and `>>` within the scope of the verbatim environment or within the argument of the `\verb` command produce italics surrounded by angle brackets. This typographic convention usually indicates *metalanguage*, i.e., a placeholder.

To obtain the angle bracket character per se, double the character, viz., “the delimiter `\verb+<<<<+`”.

### 4.1.2 The `-matter` Commands Work

The sectioning commands `\frontmatter`, `\mainmatter`, and `\backmatter` of the standard L<sup>A</sup>T<sub>E</sub>X book class are operative in the `ltxdoc` class.

### 4.1.3 The `\GetFileInfo` command

You can use the `\GetFileInfo` command to extract the date, version, and file info of a file which has registered itself via the `\ProvidesFile` or `\ProvidesClass` command (employing the optional argument thereto).

For instance, if your document contains the following:

```
%\RequirePackage{ltxdocext}%
%\GetFileInfo{ltxdocext.sty}%
%
```

then the following control sequence names will have a value corresponding to that package’s `\ProvidesFile` command: `\filedate`: the file’s date, `\fileversion`: the file’s version, and `\fileinfo`: the file’s info.

#### 4.1.4 Self-Indexing Commands

Certain commands automatically produce an index entry (or several related entries) according to the meaning.

meta-text	<code>\marg{&lt;text&gt;}</code>
command	<code>\cmd{&lt;csname&gt;}</code>
environment name	<code>\env{&lt;name&gt;}</code>
<code>\begin{foo}</code>	<code>\envb{&lt;foo&gt;}</code>
<code>\end{foo}</code>	<code>\enve{&lt;foo&gt;}</code>
argument	<code>\arg{&lt;name&gt;}</code>
optional	<code>\oarg{&lt;name&gt;}</code>
filename	<code>\file{&lt;name&gt;}</code>
url	<code>\url{&lt;name&gt;}</code>
document class	<code>\classname{&lt;name&gt;}</code>
document substyle	<code>\substyle{&lt;name&gt;}</code>
class option	<code>\classoption{&lt;name&gt;}</code>

#### 4.1.5 Unnumbered Tables

When your documentation requires the use of an unnumbered table, use the `unnumtable` environment:

```
%\begin{unnumtable}  
%\begin{tabular}{ll}  
%<table rows>  
%\end{tabular}  
%\end{unnumtable}  
%
```

#### 4.1.6 Structuring Tables

The commands `\toprule`, `\colrule`, and `\botrule` allow you to mark the beginning of the column heads the beginning of the table body, and the end of the table body, respectively. In context,

```
%\begin{tabular}{ll}  
%\toprule  
%<table head rows>  
%\colrule  
%<table rows>  
%\botrule  
%\end{tabular}  
%
```

#### 4.1.7 A Sectioning Command Below `\subsection`

The `\subsubsection` command is defined.

## 4.2 Alterations to the ltxdoc class

The following involve no new markup, but they do change the appearance of your formatted documentation:

1. Using the `acronym` package causes your document to be formatted using the standard Acrobat fonts to the greatest extent possible. This means that for most documents, Computer Modern is not used at all. Math that unavoidable must use CM still exists, however.
2. An index will be produced at the end of the document without your needing to explicitly mark it up, and it will have an entry in the TOC.
3. The `quote` environment has a slightly smaller left margin.
4. Array columns are set tight by default.
5. A host of `\DoNotIndex` directives are invoked. I intend this list to grow to encompass even more commands. Send me your suggestions.

## 5 Extensions to the ltxdoc class

The `package` `DOCSTRIP` module comprises the package `ltxdocext.sty`, which provides extensions to the standard L<sup>A</sup>T<sub>E</sub>X `ltxdoc` class.

### 5.1 Beginning of the package `DOCSTRIP` module

This portion of code is only present in the L<sup>A</sup>T<sub>E</sub>X package (`.sty` file), not in the kernel portion.

```
146 %<*package>
147 \def\class@name{ltxdocext}%
148 \expandafter\PackageInfo\expandafter{\class@name}{%
149   An extension to the \protect\LaTeXe\space ltxdoc class
150   by A. Ogawa (arthur\ogawa sbcglobal.net)%
151 }%
152 %</package>
```

### 5.2 Beginning of the kernel `DOCSTRIP` module

The bulk of the code is the kernel portion; a brief tail of `package` code then follows.

```
153 %<*kernel>
```

### 5.3 Incorporate `ltxguide.cls` extensions

Code extracted from `ltxguide.cls`, by Alan Jeffrey. “This code stolen from `ltxguide.cls`: Some hacks with `verbatim`... NB: this would be better done with the `verbatim` package, but this document has to run on any L<sup>A</sup>T<sub>E</sub>X installation.”

```
154 \RequirePackage{verbatim}%
```

```

155 \let\o@verbatim\verbatim
156 \def\verbatim{%
157   \ifhmode\unskip\par\fi
158   % \nopagebreak           % Overridden by list penalty
159   \ifx\@currsize\normalsize
160     \small
161   \fi
162   \o@verbatim
163 }%

```

Here we extend the font-setting command to include making <> active (i.e., adjusting the input encoding).

```

164 \renewcommand \verbatim@font {%
165   \normalfont \ttfamily
166   \catcode'\<=\active
167   \catcode'\>=\active
168 }%

```

Make |...| a synonym for \verb|...|.

```

169 \RequirePackage{shortvrb}
170 \AtBeginDocument{%
171   \MakeShortVerb{\|}%
172 }%

```

Make active bracket characters produce italics surrounded by angle brackets (used in `verbatim` and `\verb`). << produces a less-than, and >> produces a greater-than.

```

173 \begingroup
174   \catcode'\<=\active
175   \catcode'\>=\active
176   \gdef<{\@ifnextchar<\@lt\@meta}
177   \gdef>{\@ifnextchar>\@gt\@gtr@err}
178   \gdef\@meta#1>{\marg{#1}}
179   \gdef\@lt<{\char'\<}
180   \gdef\@gt>{\char'\>}
181 \endgroup
182 \def\@gtr@err{%
183   \ClassError{ltxguide}{%
184     Isolated \protect>%
185   }{%
186     In this document class, \protect<...\protect>
187     is used to indicate a parameter.\MessageBreak
188     I've just found a \protect> on its own.
189     Perhaps you meant to type \protect>\protect>?
190   }%
191 }
192 \def\verbatim@nolig@list{\do\'\do\,\do\''\do\~}

```

End of code stolen from `ltxguide.cls`. Thanks, Alan.

Add functionality from `doc.dtx`: (code stolen from `doc.dtx`):

```

193 \def\GetFileInfo#1{%

```

```

194 \def\filename{#1}%
195 \def\@tempb##1 ##2 ##3\relax##4\relax{%
196   \def\filedate{##1}%
197   \def\fileversion{##2}%
198   \def\fileinfo{##3}%
199   \edef\@tempa{\csname ver@#1\endcsname}%
200   \expandafter\@tempb\@tempa\relax? ? \relax\relax}

(end of code stolen from doc.dtx. Thanks FMi.)
Various forms of self-indexing commands:
201 \DeclareRobustCommand{\marg}[1]{%
202   \meta{#1}%
203   \index{#1=\string\meta{#1} placeholder}\index{placeholder>#1=\string\meta{#1}}%
204 }%
205 \DeclareRobustCommand\meta[1]{%
206   \mbox{\LANGLE\itshape#1/\RANGLE}%
207 }%
208 \def\LANGLE{$\langle$}%
209 \def\RANGLE{$\rangle$}%
210 \DeclareRobustCommand{\arg}[1]{%
211   {\ttfamily\string{}\meta{#1}{\ttfamily\string}}%
212   \index{#1=\string\ttt{#1}, argument}\index{argument>#1=\string\ttt{#1}}%
213 }%
214 \let\oarg\undefined
215 \DeclareRobustCommand{\oarg}[1]{%
216   {\ttfamily[%]
217   }\meta{#1}{\ttfamily%[
218   ]}%
219   \index{#1=\string\ttt{#1}, optional argument}%
220   \index{argument, optional>#1=\string\ttt{#1}}%
221 }%
222 \DeclareRobustCommand\cmd{\begingroup\makeatletter\@cmd}%
223 \long\def\@cmd#1{%
224   \endgroup
225   \cs{\expandafter\cmd@to@cs\string#1}%
226   \expandafter\cmd@to@index\string#1\@nil
227 }%
228 \def\cmd@to@cs#1#2{\char\number' #2\relax}%
229 \def\cmd@to@index#1#2\@nil{%
230   \index{#2=\string\cmd#1#2}\index{command>#2=\string\cmd#1#2}%
231 }%
232 \DeclareRobustCommand\cs[1]{\ttfamily\char'\@#1}%
233 \def\scmd#1{%
234   \cs{\expandafter\cmd@to@cs\string#1}%
235   \expandafter\scmd@to@index\string#1\@nil
236 }%
237 \def\scmd@to@index#1#2\@nil#3{%
238   \index{\string$#3=\string\cmd#1#2---#3}%
239   \index{command>\string$#3=\string\cmd#1#2---#3}%
240 }%

```

```

241 \DeclareRobustCommand\env{\name@idx{environment}}}%
242 \DeclareRobustCommand\envb[1]{%
243   {\ttfamily\string\begin\string{}\env{#1}{\ttfamily\string}}}%
244 }%
245 \DeclareRobustCommand\enve[1]{\{\ttfamily\string\end\string{}\env{#1}{\ttfamily\string}}}%
246 \DeclareRobustCommand{\file}{\begingroup\@sanitize\@file}%
247 \long\def\@file#1{\endgroup
248   {\ttfamily#1}%
249   \index{#1=\string\ttt{#1}}\index{file>#1=\string\ttt{#1}}}%
250 }%
251 \DeclareRobustCommand\substyle{\name@idx{document substyle}}%
252 \DeclareRobustCommand\classoption{\name@idx{document class option}}%
253 \DeclareRobustCommand\classname{\name@idx{document class}}%
254 \def\name@idx#1#2{%
255   {\ttfamily#2}%
256   \index{#2\space#1=\string\ttt{#2}\space#1}\index{#1>#2=\string\ttt{#2}}}%
257 }%
258 \DeclareRobustCommand\url@ltxdocext{\begingroup\catcode'\active\catcode'\active\catcode'\active
259 \AtBeginDocument{%
260   \ifx\url\undefined\let\url\url@ltxdocext\fi
261 }%
262 \def\@url#1{%
263   \url@break{\ttfamily#1}%
264   \url@char\edef\@tempa{#1=\string\url{#1}}%
265   \expandafter\index\expandafter{\@tempa}%
266   \expandafter\index\expandafter{\expandafter u\expandafter r\expandafter l\expandafter >\@tempa}
267   \endgroup
268 }%
269 {\catcode'\active\aftergroup\def\aftergroup:}{\active@colon}%
270 \def\colon@break{\colon@char\allowbreak}%
271 \def\colon@char{:}%
272 {\catcode'\active\aftergroup\def\aftergroup/}{\active@slash}%
273 \def\slash@break{\slash@char\allowbreak}%
274 \def\slash@char{/}%
275 {\catcode'\active\aftergroup\def\aftergroup.}{\active@dot}%
276 \def\dot@break{\dot@char\allowbreak}%
277 \def\dot@char{.}%
278 \def\url@break{\let\active@slash\slash@break\let\active@dot\dot@break\let\active@colon\colon@br
279 \def\url@char{\let\active@slash\slash@char\let\active@dot\dot@char\let\active@colon\colon@char}

```

Modify the `index` environment so that it produces a TOC entry

```

286             \columnseprule \z@
287             \columnsep 35\p@
288 \def\see##1##2{\textit{See} ##1}%
289 \def\seealso##1##2{\textit{See also} ##1}%
290 \long\def\cmd##1{\cs{\expandafter\cmd@to@cs\string##1}}%
291 \def\@url##1{\url@break\ttt{##1}\endgroup}%
292 \def\ttt{\begingroup\@sanitize\ttfamily\@ttt}%
293 \def\@ttt##1{##1\endgroup}%
294 \mathchardef\save@secnumdepth\c@secnumdepth
295 \c@secnumdepth\m@ne
296             \twocolumn[\section{\indexname}]%
297 %             \mkboth{\MakeUppercase\indexname}%
298 %             {\MakeUppercase\indexname}%
299 \c@secnumdepth\save@secnumdepth
300             \thispagestyle{plain}\parindent\z@
301             \parskip\z@ \@plus .3\p@\relax
302             \let\item\@idxitem}
303             {\if@restonecol\onecolumn\else\clearpage\fi}
304 \renewenvironment{quote}
305             {\list{}{%
306             \leftmargin1em\relax
307             \rightmargin\leftmargin
308             }%
309             \item\relax}
310             {\endlist}

```

## 5.5 Extensions to the base class of ltxdoc.cls

Matter commands from book.cls

```

311 \newif\if@mainmatter
312 \newif\if@openright
313 \@openrighttrue
314 \DeclareRobustCommand\frontmatter{%
315   \cleartorecto
316   \@mainmatterfalse
317   \pagenumbering{roman}%
318 }%
319 \DeclareRobustCommand\mainmatter{%
320   \cleartorecto
321   \@mainmattertrue
322   \pagenumbering{arabic}%
323 }%
324 \DeclareRobustCommand\backmatter{%
325   \if@openright
326     \cleartorecto
327   \else
328     \clearpage
329   \fi
330   \@mainmatterfalse

```

```

331 }%
332 \ifx\undefined\cleartorecto
333 \def\cleartorecto{\cleardoublepage}%
334 \fi

```

Unnumbered tables

`unnumtable` An unnumbered table does not float.

```

335 \def\@to{to}%
336 \newenvironment{unnumtable}{%
337 \par
338 \addpenalty\predisplaypenalty
339 \addvspace\abovedisplayskip
340 \hbox\@to\hsize\bgroup\hfil\ignorespaces
341 \let\@Hline\@empty
342 }{%
343 \unskip\hfil\egroup
344 \penalty\postdisplaypenalty
345 \vskip\belowdisplayskip
346 \aftergroup\ignorespaces
347 \@endpetrue
348 }%

```

Emulate `\toprule` and friends

```

349 \providecommand\toprule{\hline\hline}%
350 \providecommand\colrule{\hline\hline}%
351 \providecommand\botrule{\hline\hline}%

```

Define sectioning command below `\subsubsection`.

```

352 \DeclareRobustCommand\subsubsection{%
353 \@startsection{subsubsection}{4}%
354 {\z@}{-15\p@}{\p@}{\p@}{\p@}%
355 {5\p@}{\normalfont\normalsize\itshape}%
356 }%

```

## 5.6 In lieu of `ltxdoc.cfg`

We don't want everything to appear in the index

```

357 \DoNotIndex{\',\.,\@M,\@input,\@Alph,\@alph,\@addtoreset,\@arabic}
358 \DoNotIndex{\@badmath,\@centercr,\@cite}
359 \DoNotIndex{\@dotsep,\@empty,\@float,\@gobble,\@gobbletwo,\@ignoretrue}
360 \DoNotIndex{\@input,\@ixpt,\@m,\@minus,\@mkboth}
361 \DoNotIndex{\@one,\@nil,\@nomath,\@plus,\@roman,\@set@topoint}
362 \DoNotIndex{\@tempboxa,\@tempcnta,\@tempdima,\@tempdimb}
363 \DoNotIndex{\@tempswafalse,\@tempswatruer,\@vipt,\@viipt,\@vipt}
364 \DoNotIndex{\@vpt,\@warning,\@xipt,\@xipt,\@xivpt,\@xpt,\@xvipt}
365 \DoNotIndex{\@xxpt,\@xxvpt,\@, \ ,\addpenalty,\addtolength,\addvspace}
366 \DoNotIndex{\advance,\ast,\begin,\begingroup,\bfseries,\bgroup,\box}
367 \DoNotIndex{\bullet}
368 \DoNotIndex{\cdot,\cite,\CodelineIndex,\cr,\day,\DeclareOption}

```



```

369 \DoNotIndex{\def,\DisableCrossrefs,\divide,\DocInput,\documentclass}
370 \DoNotIndex{\DoNotIndex,\egroup,\ifdim,\else,\fi,\em,\endtrivlist}
371 \DoNotIndex{\EnableCrossrefs,\end,\end@dblfloat,\end@float,\endgroup}
372 \DoNotIndex{\endlist,\everycr,\everypar,\ExecuteOptions,\expandafter}
373 \DoNotIndex{\fbox}
374 \DoNotIndex{\filedate,\filename,\fileversion,\fontsize,\framebox,\gdef}
375 \DoNotIndex{\global,\halign,\hangindent,\hbox,\hfil,\hfill,\hrule}
376 \DoNotIndex{\hsize,\hskip,\hspace,\hss,\if@tempswa,\ifcase,\or,\fi,\fi}
377 \DoNotIndex{\ifhmode,\ifvmode,\ifnum,\iftrue,\ifx,\fi,\fi,\fi,\fi,\fi}
378 \DoNotIndex{\input}
379 \DoNotIndex{\jobname,\kern,\leavevmode,\let,\leftmark}
380 \DoNotIndex{\list,\llap,\long,\m@ne,\m@th,\mark,\markboth,\markright}
381 \DoNotIndex{\month,\newcommand,\newcounter,\newenvironment}
382 \DoNotIndex{\NeedsTeXFormat,\newdimen}
383 \DoNotIndex{\newlength,\newpage,\nobreak,\noindent,\null,\number}
384 \DoNotIndex{\numberline,\OldMakeindex,\OnlyDescription,\p@}
385 \DoNotIndex{\pagestyle,\par,\paragraph,\paragraphmark,\parfillskip}
386 \DoNotIndex{\penalty,\PrintChanges,\PrintIndex,\ProcessOptions}
387 \DoNotIndex{\protect,\ProvidesClass,\raggedbottom,\raggedright}
388 \DoNotIndex{\refstepcounter,\relax,\renewcommand}
389 \DoNotIndex{\rightmargin,\rightmark,\rightskip,\rlap,\rmfamily}
390 \DoNotIndex{\secdef,\selectfont,\setbox,\setcounter,\setlength}
391 \DoNotIndex{\settowidth,\sfcode,\skip,\sloppy,\slshape,\space}
392 \DoNotIndex{\symbol,\the,\trivlist,\typeout,\tw@,\undefined,\uppercase}
393 \DoNotIndex{\usecounter,\usefont,\usepackage,\vfil,\vfill,\viipt}
394 \DoNotIndex{\vipt,\vpt,\vskip,\vspace}
395 \DoNotIndex{\wd,\xiipt,\year,\z@}
396 \DoNotIndex{\next}

```

Direct ltxdoc to produce an index.

```

397 \AtEndDocument{\PrintIndex\PrintChanges}%

```

## 5.7 Extension to L<sup>A</sup>T<sub>E</sub>X's filecontents Environment

We want to coax the version number into filecontents-generated files. Note that we expect `\fileversion` and `\filedate` to hold the needed information. For this to be the case, your document should execute the `\GetFileInfo` command (as documented in section 4.1.3) before any instances of `filecontents`.

```

398 \makeatletter
399 \def\endfilecontents{%
400   \immediate\write\reserved@c{%
401     \string\iffalse\space ltxdoc kloothch^J%
402     \ifx\undefined\fileversion\else
403     \ifx\undefined\filedate\else
404       This file has version number \fileversion, last revised \filedate.%
405     \fi\fi
406     \string\fi
407   }%
408   \immediate\closeout\reserved@c

```

```

409 \def\T##1##2##3{%
410   \ifx##1\@undefined\else
411     \@latex@warning@no@line{##2 has been converted to Blank ##3e}%
412   \fi
413 }%
414 \T\L{Form Feed}{Lin}%
415 \T\I{Tab}{Spac}%
416 \immediate\write\@unused{}%
417 }%
418 \expandafter\let\csname endfilecontents*\endcsname\endfilecontents
419 \makeatother

    Alter formatting in arrays; set them tight.
420 \setlength\arraycolsep{0pt}%

```

## 5.8 End of the kernel DOCSTRIP module

```
421 %</kernel>
```

## 5.9 Tail of the package DOCSTRIP module

Here is the remainder of the package code.

```
422 %<*package>
```

Currently, there is little.

```
423 %</package>
```

# 6 Font Package for Acrobat Compatability

The package `acrofont` substitutes Acrobat-standard fonts for Computer Modern where possible, even in math mode. Documents typeset with this package in effect will require as little downloaded font data as possible, but will not be exemplars of fine math typesetting.

## 6.1 Beginning of the fonts DOCSTRIP module

The document class module comprises this and the next four sections.

`\class@base` We define in exactly one spot the base class. Typically that class will be one of `book`, `article`, or `report`. The base class effectively defines the use and the markup scheme of the class of documents to be handled by this class.

This class is a variant of the standard L<sup>A</sup>T<sub>E</sub>X book class: `ftp://ctan.tug.org/tex-archive/macros/latex/unpacked`.

```

424 %<*fonts>
425 \def\class@name{ltxdocext}%
426 \expandafter\ClassInfo\expandafter{\class@name}{%
427   Written for \protect\LaTeXe\space
428   by A. Ogawa (arthur_ogawa at sbcglobal.net)%
429 }%

```

## 6.2 Variants on psfonts packages

The following uses `times.sty` from `/packages/psnfss/psfonts.dtx`

```
430 \RequirePackage{times}%
```

The following uses `mathptm.sty` from `/packages/psnfss/psfonts.dtx`

```
431 \RequirePackage{mathptm}%
```

The following is a customization of `ot1ptmcm.fd`. The virtual font referred to here `zptmnocmr` is a variant of Sebastian Rahtz's `zptmcmr`, but with even more glyphs moved from CM to Acrobat-standard fonts.

```
432 \DeclareFontFamily{OT1}{ptmcm}{}
433 \DeclareFontShape{OT1}{ptmcm}{m}{n}{
434     <-> zptmnocmr
435 }{}
436 \DeclareFontShape{OT1}{ptmcm}{l}{n}{<->ssub * ptmnocm/m/n}{}

```

The following is a customization of `omlptmcm.fd`. The virtual font referred to here `zptmnocmr` is a variant of Sebastian Rahtz's `zptmcmr`, but with even more glyphs moved from CM to Acrobat-standard fonts.

```
437 \DeclareFontFamily{OML}{ptmcm}{\skewchar \font =127}
438 \DeclareFontShape{OML}{ptmcm}{m}{it}{
439     <-> zptmnocmr
440 }{}
441 \DeclareFontShape{OML}{ptmcm}{l}{it}{<->ssub * ptmcm/m/it}{}
442 \DeclareFontShape{OML}{ptmcm}{m}{sl}{<->ssub * ptmcm/m/it}{}
443 \DeclareFontShape{OML}{ptmcm}{l}{sl}{<->ssub * ptmcm/m/sl}{}

```

The following is a customization of `omspzccm.fd`. The virtual font referred to here `zpzcncmry` is a variant of Sebastian Rahtz's `zpzcmmr`, but with even more glyphs moved from CM to Acrobat-standard fonts.

```
444 \DeclareFontFamily{OMS}{pzccm}{}
445 \DeclareFontShape{OMS}{pzccm}{m}{n}{
446     <-> zpzcncmry
447 }{}% cmsy10 Symbol Zapf Chancery Medium-Italic Times-Roman
448 \DeclareFontShape{OMS}{pzccm}{l}{n}{<->ssub * pzccm/m/n}{}

```

The following is a customization of `omxpsycm.fd`. The virtual font referred to here `zpsynocmr` is a variant of Sebastian Rahtz's `zpsycmr`, but with even more glyphs moved from CM to Acrobat-standard fonts.

```
449 \DeclareFontFamily{OMX}{psycm}{}
450 \DeclareFontShape{OMX}{psycm}{m}{n}{
451     <-> zpsynocmr
452 }{}
453 \DeclareFontShape{OMX}{psycm}{l}{n}{<->ssub * psycm/m/n}{}
454 %
455 \DeclareFontEncoding{8r}{}{}% from file: 8renc.def
456 \DeclareFontFamily{8r}{cmr}{\hyphenchar\font45}% from file: 8rcmr.fd
457 \DeclareFontShape{8r}{cmr}{m}{n}{
458     <-> cmr10
459 }{}

```

### 6.3 Font definition files

The following forces L<sup>A</sup>T<sub>E</sub>X to do now what it would do anyway: load the ‘font definition’ information for the fonts that we use. In this way, we prepare for faster processing through the `\dump` of a preformatted macro package that will not need to read in any packages or font definitions from disk.

```
460 \input{8rphv.fd}%
461 \input{8rptm.fd}%
462 \input{ot1phv.fd}%
463 \input{ot1ptm.fd}%
464 \input{t1ptm.fd}%
```

### 6.4 More math substitutions

The following definitions arrange to get certain glyphs from the text font instead of out of math pi fonts. In particular, the copyright and registered symbols are single glyphs instead of composites involving the big circle from the `cmsy` font.

```
465 \def\eightRChar#1{\def\encodingdefault{8r}\fontencoding\encodingdefault\selectfont\char"#1}}%
466 \def\LANGLE{<}>\eightRChar{8B}}%
467 \def\RANGLE{>}>\eightRChar{9B}}%
468 %\def\ASTER{\eightRChar{2A}}%
469 %\def\DAGGER{\eightRChar{86}}%
470 %\def\DDAGGER{\eightRChar{87}}%
471 \def\BULLET{\eightRChar{95}}%
472 %\def\SECTION{\eightRChar{A7}}%
473 %\def\PARAGRAPH{\eightRChar{B6}}%
474 \def\VERTBAR{\eightRChar{7C}}%
475 \def\COPYRIGHT{\eightRChar{A9}}%
476 \def\REGISTERED{\eightRChar{AE}}%
477 \def\textbar{\VERTBAR}%
478 \def\textbullet{\BULLET}%
479 \def\textcopyright{\COPYRIGHT}%
480 \def\textregistered{\REGISTERED}%
481 \def\@makefnmark{\@thefnmark}%
482 \def\@fnsymbol#1{\ifcase#1\or *\or \dagger\or \ddagger\or
483 \textsection\or \textparagraph\or \|\or **\or \dagger\dagger
484 \or \ddagger\ddagger \else\@ctrerr\fi}}
```

I have removed `\ensuremath` from the following definition, and all commands like `\mathsection` have been converted to e.g., `\textsection`.

```
481 \def\@makefnmark{\@thefnmark}%
482 \def\@fnsymbol#1{\ifcase#1\or *\or \dagger\or \ddagger\or
483 \textsection\or \textparagraph\or \|\or **\or \dagger\dagger
484 \or \ddagger\ddagger \else\@ctrerr\fi}}
```

### 6.5 End of the fonts DOCSTRIP module

Here ends the module.

```
485 %</fonts>
```

## 7 Programming Conventions

In writing the above code, I cleave to certain conventions, noted here. My goal in explaining them is to encourage others maintaining this body of code to consider following them as well. The benefits are twofold: Some of the coding conventions aim to avoid programming pitfalls; following them reduces maintenance costs. Others make the code easier to read; following these eases the process of understanding how the code works.

And, for my part, I prefer to read code of this type.

### 7.1 Whitespace Conventions

Exactly where code lines break and indent, and where additional whitespace is inserted is explained here.

- Each new macro definition or register assignment begins a new line. Therefore, `\def`, `\newcommand`, and their ilk will start in column 1.
- Code is indented one space for each level of nesting within braces `{}`.
- Likewise, if possible, for `\if...` and matching `\fi`.
- However, the closing brace or `\fi` is outdented by one so that it falls at the same level of indentation as its matching brace or `\if`, and it appears alone on its line.
- Use of the `tab` character is deprecated (tabs are not standardized across all applications and operating systems).
- Lines of code are limited to 72 characters. I follow this convention mostly to ease the transmission of files via email (a deprecated practice) and to accomodate people with small monitors. But `ltxdoc` output looks better with the shorter lines, too.
- Extraneous whitespace in the replacement part of a macro definition is avoided by using the comment character `%`. In most cases, if falling at the end of a line of code, a brace will be immediately followed by a comment character, as will the macro parameter `#1` and any one-letter control sequence, like `\`.
- Extraneous whitespace in the package file is also avoided. When `TEX` reads in the `.sty` file, it will process `\defs`, and other commands, but will not process blank spaces. This practice is simply a discipline. You don't need to do this. But sometimes `TEX` has to read in a file while it is in horizontal mode, at which point this practice is essential.

These conventions taken together are illustrated by the following:

```

%\def\prepdef#1#2{%
% \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
% \toks@ii{#2}%
% \edef#1{\the\toks@ii\the\toks@}%
%}%
%

```

In the above, the definition of `\prepdef` would not fit on a single line, and required breaking. The first and last lines have matching braces, and are at the same level of indentation, with the last line containing a single brace.

Each of the three intervening lines has balanced braces and is indented by one space. Each line that would otherwise end in a single brace character is terminated by a comment character.

Some coders rely on the fact that a space character seen by  $\TeX$ 's scanner while in vertical mode is a no-op. Be that as it may, I eliminate them unless actually intentional.

## 7.2 Conventions For Procedures

Here are some of my preferences when writing procedures:

- I dislike defining a macro within another macro, especially when the pattern part is non-nil. You know, you are not saving much space in `mem` when you do this, right? You do save space in the hash table and the string pool, though. On the other hand, we are not dealing with small  $\TeX$  engines anymore; Team  $\LaTeX$  has made sure of this.
- If two or more macros have very similar replacement parts, consider layering.
- Macros may perform parsing, may maintain tokens or registers, or may set type (produce marks). I try to avoid combining the three functions in a single macro.
- When a procedure both does assignments and sets type, I try to have a clean separation between the two activities. Try to avoid:

```

% \vskip10pt
% \parindent=0pt
% \leavevmode
%

```

- The Boolean calculus (cf. `\@ifx`) is very useful and produces code that executes nicely. Using it also helps avoid your having to debug problems where `\if...` and `\fi` are not properly balanced (a nightmare, in case you have not already experienced it).

### 7.3 Conventions For L<sup>A</sup>T<sub>E</sub>X

Team L<sup>A</sup>T<sub>E</sub>X make certain recommendations in `clsguide.tex`. Ones that I particularly pay attention to are:

- For the sake of “color safety”, use `\sbox` rather than `\setbox`, `\mbox` rather than `\hbox`, and `\parbox` or `minipage` rather than `\vbox`.
- Use `\newcommand` and `\newenvironment` to declare user-level commands and environments. Avoid the idiom `\def\foo`, `\def\endfoo` to define an environment. Ideally, all user-level markup could be extracted from the document class by grepping on `\newcommand` and `\newenvironment`.
- Prefer to use `\setlength` to assign registers.

I cannot help but notice that much of the code of L<sup>A</sup>T<sub>E</sub>X itself fails to comply with many of the coding recommendations of Team L<sup>A</sup>T<sub>E</sub>X.