

Extensions to the `ltxdoc` class *

Arthur Ogawa (<mailto:ogawa@teleport.com>), 1.0d
Copyright (C) 1999 Arthur Ogawa

October 6, 2020

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

The distribution point for this work is <ftp://ftp.teleport.com/users/ogawa/macros/latex/contrib/supported/ltxdocext...>, which contains fully unpacked, prebuilt runtime files and documentation.

To use this document class, you must have a working \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 \LaTeX .

If you will be using the `acrofont` package, you must also install the virtual fonts `zpsynocmr`, `zptmnochr`, `zptmnochr`, and `zpzcnocmry`. 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

1	Processing Instructions	2
1.1	Build Instructions	3
1.2	Bill of Materials	3
1.2.1	Primary Source	3
1.2.2	Generated by <code>latex ltxutil.dtx</code>	3
1.2.3	Generated by <code>tex ltxutil.ins</code>	3
1.2.4	Documentation	4
1.2.5	Auxiliary	4
2	Code common to all modules	4

*This file has version number 1.0d, last revised 2020/09/30. For version number and date, search on "1.0d" in the `.dtx` file, or see the end of the `00readme.txt` file.

3	The driver module doc	4
3.1	The Preamble	5
3.1.1	Docstrip and info directives	5
3.2	The installer file	5
3.3	The “Read Me” File	6
3.4	The Document Body	8
4	Using this package	8
4.1	Invoking the package	9
5	Compatibility with L^AT_EX’s Required Packages	9
5.1	array	9
5.2	longtable	10
6	Implementation of package	10
6.1	Beginning of the ltxutil DOCSTRIP module	10
6.2	Banner	10
6.3	Errors and warnings	10
6.4	New Tools	11
6.5	Boolean Control	12
6.6	Begin Document Structure	15
6.7	Type Tools	18
6.8	Display Math	18
6.9	Footnotes	20
6.10	Floats	23
6.10.1	Usage notes	23
6.10.2	Robustifying fragile commands	24
6.10.3	Preparing for the hyperref package	24
6.10.4	Footnotes within floats, unfloating floats, float font	25
6.10.5	Writing floats out to a file	27
6.11	Counters	30
6.12	Customization of Sections	30
6.13	Patch the tabular and array Environments	34
6.14	Repair other broken parts of L ^A T _E X	55
6.15	Syntax	56
6.16	Auto-indented Contents	56
6.17	Lists	59
6.18	End of the ltxutil DOCSTRIP module	60
	Index	61

1 Processing Instructions

The package files `ltxdocext.sty` and `acrofont.sty` are generated from this file, `ltxdocext.dtx`, via the DOCSTRIP facility of L^AT_EX via `tex ltxdocext.ins`. The typeset documentation that you are now reading is generated from the

same file by typesetting it with L^AT_EX 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 L^AT_EX 2_ε (and either `tex` or `pdftex`) on your computer, then carry out the following steps:

1. Within an otherwise empty directory, typeset `ltxdocext.dtx` with L^AT_EX or `pdflatex`; you will obtain the typeset documentation you are now reading, along with the installer `ltxdocext.ins`, and the file `00readme.txt`.
2. Now typeset `ltxdocext.ins`, 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.
3. Install `ltxdocext.sty` and `acrofont.sty` by moving them to a location in your filesystem where they will be found by L^AT_EX.
4. Now complete the typesetting of the documentation by retypesetting `ltxdocext.dtx`. Note: you will have to run L^AT_EX twice, then `makeindex`, then L^AT_EX again in order to obtain a valid index and table of contents.

1.2 Bill of Materials

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

1.2.1 Primary Source

One single file generates all.

```
%ltxutil.dtx
%
```

1.2.2 Generated by `latex ltxutil.dtx`

Typesetting the source file under L^AT_EX generates the `readme` and the installer.

```
%00readme.txt  ltxutil.ins
%
```

1.2.3 Generated by `tex ltxutil.ins`

Typesetting the installer generates the package files.

```
%ltxutil.sty
%
```

1.2.4 Documentation

The following are the online documentation:

```
%ltxutil.pdf
%
```

1.2.5 Auxiliary

The following are auxiliary files generated in the course of running L^AT_EX:

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

2 Code common to all modules

The following may look a bit kloutchy, but 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^AT_EX 2_ε. An appropriate message is displayed if a different T_EX format is used.

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

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

```
4 %<ltxutil>\ProvidesFile{ltxutil.sty}%
5 %<*doc>
6 \expandafter\ProvidesFile\expandafter{\jobname.dtx}%
7 %</doc>
```

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.

```
8 %<*doc|ltxutil>
9 [2020/09/30 1.0d utilities package]% \fileversion
10 %</doc|ltxutil>
```

3 The driver module doc

This module, consisting of the present section, typesets the programmer's documentation, generating the `.ins` installer and `00readme.txt` 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 %<*doc>
```

3.1 The Preamble

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

```
12 \documentclass{ltxdoc}
13 \RequirePackage{ltxdocext}%
14 \let\url\undefined
15 \RequirePackage[colorlinks=true,linkcolor=blue]{hyperref}%
16 \pdfstringdefDisableCommands{%
17   \let\file\relax
18   \let\sc\relax
19 }
20 \makeatletter
21 \@ifundefined{package@font}{}%
22   {\expandafter\RequirePackage\expandafter{\csname package@font\endcsname}}
23 \makeatother
24 \CodelineIndex\EnableCrossrefs
```

3.1.1 Docstrip and info directives

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

```
25 \setcounter{StandardModuleDepth}{1}
```

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

```
26 \expandafter\GetFileInfo\expandafter{\jobname.dtx}%
```

3.2 The installer file

The installer `ltxutil.ins` appears here. If you have retrieved the standard distribution of this package, the installer file is already on your filesystem. If you are bootstrapping, the first typesetting of the `.dtx` file will cause the installer to be generated.

The following modules are used to direct DOCSTRIP in generating the external files:

Module	File	Description
doc	<code>ltxutil.drv</code>	driver for programmer's documentation
<code>ltxutil,ltxutil-krn</code>	<code>ltxutil.sty</code>	this package
<code>ltxutil-krn</code>		the portion of this package suitable for inclusion within another package

```
27 \begin{filecontents}{ltxutil.ins}
28 %% This file will generate documentation and runtime files
29 %% from ltxutil.dtx when run through LaTeX or TeX.
30 \input docstrip
31 \preamble
32
33 This is a generated file;
```

```

34 altering it directly is inadvisable;
35 instead, modify the original source file.
36 See the URL in the file 00readme.txt.
37
38 Copyright notice.
39
40 These files are distributed
41 WITHOUT ANY WARRANTY; without even the implied warranty of
42 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
43
44 \endpreamble
45 \keepsilent
46 \generate{%
47   \file{ltxutil.drv}{\from{ltxutil.dtx}{doc}}%
48   \file{ltxutil.sty}{%
49     \from{ltxutil.dtx}{ltxutil,ltxutil-krn}%
50   }%
51 }%
52 \ifToplevel{
53 \Msg{*****}
54 \Msg{*}
55 \Msg{* To finish the installation, please move}
56 \Msg{*   ltxutil.sty}
57 \Msg{* into a directory searched by TeX.}
58 \Msg{*}
59 \Msg{* To produce the documentation,
60     run ltxutil.dtx through LaTeX.}
61 \Msg{*}
62 \Msg{* Happy TeXing}
63 \Msg{*****}
64 }
65 \endbatchfile
66 \end{filecontents}

```

Note that, because all of the files generated by the installer are part of the standard distribution, it will be necessary to run the installer only when bootstrapping (or, of course, during development). Note, too, that it is rare to generate the doc module because it suffices to simply typeset the .dtx file itself.

3.3 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.

```
67 \begin{filecontents*}{00readme.txt}
68 \title{%
69 A \LaTeX\ Package of utility macros%
70 \thanks{%
71 This file has version number \fileversion,
72 last revised \filedate.%
73 % For version number and date,
74 % search on "\fileversion" in the .dtx file,
75 % or see the end of the 00readme.txt file.
76 }%
77 }%
78
79 \author{%
80 Arthur Ogawa (\texttt{mailto:ogawa@teleport.com}),
81 \fileversion\Copyright (C) 1999 Arthur Ogawa
82 }%
83 \maketitle
84
85 This file embodies the \classname{ltxutil} package,
86 the implementation and its user documentation.
87
88 The distribution point for this work is
89 \url{ftp://ftp.teleport.com/users/ogawa/macros/latex/contrib/supported/ltxutil...},
90 which contains fully unpacked, prebuilt runtime files and documentation.
91
92 The \classname{ltxutil} package was commissioned by the American Physical Society
93 and is distributed under the terms of the \LaTeX\ Project Public License,
94 the same license under which all the portions of \LaTeX\ itself is distributed.
95 Please see \url{http://ctan.tug.org/macros/latex/base/lppl.txt} for details.
96
97 To use this document class, you must have a working
98 \TeX\ installation equipped with \LaTeXe\
99 and possibly pdftex and Adobe Acrobat Reader or equivalent.
100
101 To install, retrieve the distribution,
102 unpack it into a directory on the target computer,
103 and move the file \file{ltxutil.sty}
104 into a location in your filesystem where it will be found by \LaTeX.
105
106 To use, read the user documentation \file{ltxutil.pdf}.
107
108 \tableofcontents
109
110 \section{Processing Instructions}
111
112 The package file \file{ltxutil.sty}
113 is generated from this file, \file{ltxutil.dtx},
```

```

114 using the {\sc docstrip} facility of \LaTeX
115 via |tex ltxutil.ins|.
116 The typeset documentation that you are now reading is generated from
117 the same file by typesetting it with \LaTeX\ or pdftex
118 via |latex ltxutil.dtx| or |pdflatex ltxutil.dtx|.
119
120 \subsection{Build Instructions}
121
122 You may bootstrap this suite of files solely from \file{ltxutil.dtx}.
123 Prepare by installing \LaTeXe\ (and either tex or pdftex) on your computer,
124 then carry out the following steps:
125 \begin{enumerate}
126 \item
127 Within an otherwise empty directory,
128 typeset \file{ltxutil.dtx} with \LaTeX\ or pdflatex;
129 you will obtain the typeset documentation you are now reading,
130 along with
131 the installer \file{ltxutil.ins},
132 and the file \file{00readme.txt}.
133
134 Note: you will have to run \LaTeX\ twice, then \file{makeindex}, then
135 \LaTeX\ again in order to obtain a valid index and table of contents.
136 \item
137 Now typeset \file{ltxutil.ins},
138 thereby generating the package file \file{ltxutil.sty}.
139 \item
140 Install \classname{ltxutil.sty}
141 by moving it to a location
142 in your filesystem where they will be found by \LaTeX.
143 \end{enumerate}
144 \end{filecontents*}

```

3.4 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.

```

145 \begin{document}%
146 \expandafter\DocInput\expandafter{\jobname.dtx}%
147 % ^^A\PrintChanges
148 \end{document}
149 %</doc>

```

4 Using this package

Once this package is installed on your filesystem, you can employ it in adding functionality to \LaTeX by invoking it in your document or document class.

4.1 Invoking the package

In your document, you can simply call it up in your preamble:

```
%\documentclass{book}%  
%\usepackage{ltxutil}%  
%\begin{document}  
%<your document here>  
%\end{document}  
%
```

However, the preferred way is to invoke this package from within your customized document class:

```
%\NeedsTeXFormat{LaTeX2e}[1995/12/01]%  
%\ProvidesClass{myclass}%  
%\RequirePackage{ltxutil}%  
%\LoadClass{book}%  
%<class customization commands>  
%\endinput  
%
```

Once loaded, the package gives you access to certain procedures, usually to be invoked by a \LaTeX command or environment, but not at the document level.

5 Compatibility with \LaTeX 's Required Packages

Certain packages, usually ones written by members of the \LaTeX Project itself, have been designated “required” and are distributed as part of standard \LaTeX . These packages have been placed in a privileged position vis á vis the \LaTeX kernel in that they override the definitions of certain kernel macros.

The `ltxutil` package will be incompatible with any package that redefines any of the kernel macros that `ltxutil` patches—if that package is loaded *after* `ltxutil`. This means that for greatest compatibility, `ltxutil` should be loaded *after*, say, `ftnright`, which overwrites \LaTeX 's kernel procedures `\@outputdblcol`, `\@startcolumn`, and `\@makecol`.

Hereinafter follows some notes on specific \LaTeX packages.

5.1 array

This package alters the way tabular environments are done, therefore it could run afoul of the \LaTeX “required” package `array` or any package that calls for it to be loaded. However, this package has provisions for remaining compatible with `array`. So long as the version of `array` that is used with this package has the appropriate meanings for the procedures it overwrites, all should be well.

5.2 longtable

David Carlisle's `longtable` package modifies both the \LaTeX kernel and the `array` package. This package must therefore alter `\LT@array`. For now, that job is handled by `ltxgrid`.

6 Implementation of package

Special acknowledgment: this package uses concepts pioneered and first realized by William Baxter (<mailto:web@superscript.com>) in his SuperScript line of commercial typesetting tools, and which are used here with his permission.

6.1 Beginning of the `ltxutil` DOCSTRIP module

```
150 %<*ltxutil>
151 \def\package@name{ltxutil}%
152 \expandafter\PackageInfo\expandafter{\package@name}{%
153 Utility macros for \protect\LaTeXe,
154 by A. Ogawa (ogawa@teleport.com)%
155 }%
156 %</ltxutil>
```

6.2 Banner

Credit where due.

```
157 %<*ltxutil-krn>
158 \typeout{%
159 ltxutil: portions licensed from W. E. Baxter (web@superscript.com)%
160 }%
```

6.3 Errors and warnings

```
\class@err A few shorthands for Class messages. Your document class should define
\class@warn \class@name.
\class@info 161 \def\class@err#1{\ClassError{\class@name}{#1}\@eha}%
162 \def\class@warn#1{\ClassWarningNoLine{\class@name}{#1}}%
163 \def\class@info#1{\ClassInfo{\class@name}{#1}}%
164 \def\obsolete@command#1{%
165 \class@warn@end{Command \string#1\space is obsolete.^^JPlease remove from your document}%
166 \global\let#1\@empty
167 #1%
168 }%
169 \def\replace@command#1#2{%
170 \class@warn@end{Command \string#1\space is obsolete;^^JUse \string#2\space instead}%
171 \global\let#1#2%
172 #1%
173 }%
174 \def\replace@environment#1#2{%
```

```

175 \class@warn@end{Environment #1 is obsolete;^^JUse #2 instead}%
176 \glet@environment{#1}{#2}%
177 \@nameuse{#1}%
178 }%
179 \def\incompatible@package#1{%
180 \@ifpackageloaded{#1}{%
181 \def\@tempa{I cannot continue. You must remove the \string\usepackage\ statement that caused
182 \ClassError{\class@name}{The #1 package cannot be used with \class@name}%
183 \@tempa\stop
184 }{%
185 \class@info{#1 was not loaded (OK!)}%
186 }%
187 }%
188 \def\class@warn@end#1{%
189 \gapdef\class@enddocumenthook{\class@warn{#1}}%
190 }%
191 \AtEndOfClass{%
192 \@ifundefined\class@name{\def\class@name{Generic Class}}{}}%
193 }%

```

6.4 New Tools

```

\t@
194 \def\t@{to}%

\dimen@iii
195 \dimendef\dimen@iii\thr@@

\halign@
196 \def\halign@{\halign\t@}%

\f@ur Analogous to \@ne, \tw@, and \thr@@.
197 \chardef\f@ur=4\relax
198 \chardef\cat@letter=11\relax
199 \chardef\other=12\relax

\let@environment The directive \let@environment takes care of a common programming idiom
\glet@environment whereby one environment is made a synonym for another.
200 \def\let@environment#1#2{%
201 \expandafter\let
202 \csname#1\expandafter\endcsname\csname#2\endcsname
203 \expandafter\let
204 \csname end#1\expandafter\endcsname\csname end#2\endcsname
205 }%
206 \def\glet@environment#1#2{%
207 \global\expandafter\let
208 \csname#1\expandafter\endcsname\csname#2\endcsname
209 \global\expandafter\let
210 \csname end#1\expandafter\endcsname\csname end#2\endcsname
211 }%

```

`\tracingplain` The command `\tracingplain` causes T_EX's tracing parameters to return to the values set by default. This command is sometimes useful when you have said `\tracingall` somewhere and want to restore. The `\traceoutput` command causes `\tracingoutput` diagnostics upon `\shipout`.

```
212 \newcommand\tracingplain{%
213 \tracingonline\z@\tracingcommands\z@\tracingstats\z@
214 \tracingpages\z@\tracingoutput\z@\tracinglostchars\@ne
215 \tracingmacros\z@\tracingparagraphs\z@\tracingrestores\z@
216 \showboxbreadth5\showboxdepth3\relax %\errorstopmode
217 }%
218 \newcommand\traceoutput{%
219 \appdef\@resetactivechars{\showoutput}%
220 }%
```

`\say` The commands `\say` and `\saythe` cause diagnostic messages in the T_EX log that give the value of a control sequence name or a register respectively.

```
221 \newcommand\say[1]{\typeout{<\noexpand#1=\meaning#1>}}%
222 \newcommand\saythe[1]{\typeout{<\noexpand#1=\the#1>}}%
```

`\fullinterlineskip` Resets the `\prevdepth` so that the full amount of `\baselineskip` glue will be inserted by the `\baselineskip` mechanism. Can be invoked just after a `\hrule` to undo its default suppression of base line skip.

```
223 \def\fullinterlineskip{\prevdepth\z@}%
```

```
\count@i
\count@ii 224 \countdef\count@i\@ne
225 \countdef\count@ii\tw@
```

6.5 Boolean Control

We introduce just enough of the Boolean calculus for T_EX. Alan Jeffrey was the pioneer here, with an article in TUGboat (Vol. 11, No. 2, page 237). This implementation owes a debt to William Baxter (web@superscript.com). See articles by Baxter and Ogawa in the proceedings of the 1994 TUG meeting, TUGboat Vol. 15, No. 3.

`\prepdef` Provide the capability of performing head- and tail patches. The procedure `\appdef` `\prepdef` prepends to the given macro the tokens specified in its second argument. `\gappdef` Likewise for `\appdef`, except that it appends. Note that the first 10 toks registers are utility registers, and we simply make a control sequence name, `\toks@ii`, for one of them.

```
226 \long\def\prepdef#1#2{%
227 \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
228 \toks@ii{#2}%
229 \edef#1{\the\toks@ii\the\toks@}%
230 }%
231 \long\def\appdef#1#2{%
```

```

232 \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
233 \toks@ii{#2}%
234 \edef#1{\the\toks@\the\toks@ii}%
235 }%
236 \long\def\gapdef#1#2{%
237 \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
238 \toks@ii{#2}%
239 \global\edef#1{\the\toks@\the\toks@ii}%
240 }%
241 \long\def\appdef@val#1#2{%
242 \appdef#1{#2}}%
243 }%
244 \long\def\appdef@e#1#2{%
245 \expandafter\appdef
246 \expandafter#1%
247 \expandafter{#2}%
248 }%
249 \long\def\appdef@eval#1#2{%
250 \expandafter\appdef@val
251 \expandafter#1%
252 \expandafter{#2}%
253 }%
254 \toksdef\toks@ii=\tw@

```

`\@ifxundefined` Certain utility procedures use `\@ifxundefined`, which is defined here in terms of `\@ifx`. Others use `\@ifnotrelax`, namely when the control sequence name is manufactured by the use of `\csname`.

`\@argswap` The procedures `\@argswap` and `\@argswap@val` are used to facilitate control of expansion.

```

255 \long\def\@ifxundefined#1{\@ifx{\undefined#1}}%
256 \long\def\@ifnotrelax#1#2#3{\@ifx{\relax#1}{#3}{#2}}%
257 \long\def\@argswap#1#2{#2#1}%
258 \long\def\@argswap@val#1#2{#2{#1}}%
259 \def\@ifxundefined@cs#1{\expandafter\@ifx\expandafter{\csname#1\endcsname\relax}}%

```

`\rvtx@ifformat@geq` Some changes in the L^AT_EX kernel requires us to conditionally define some macros depending on the version of the kernel. `\rvtx@ifformat@geq` will check if the release date of the currently-running L^AT_EX 2_ε kernel is greater or equal to the argument (the argument should be in the format yyyy-mm-dd).

```

260 \ifx\IfFormatAtLeastTF\undefined
261 \def\rvtx@ifformat@geq{\ifl@t@r\fmtversion}%
262 \else
263 \let\rvtx@ifformat@geq\IfFormatAtLeastTF
264 \fi

```

`\@boolean` In order to define `\@ifx`, we first must create the “defining word” (term taken from our Forth vocabulary) `\@boole@def`, which employs `\@boolean` to do its job.

```

265 \def\@boolean#1#2{%
266 \long\def#1{%

```

```

267     #2% \if<something>
268     \expandafter\true@sw
269     \else
270     \expandafter\false@sw
271     \fi
272 }%
273 }%
274 \def\@boole@def#1#\boolean{#1}% Implicit #2

\@booleantrue The procedures \@booleantrue and \@booleanfalse are assignment operators
\@booleanfalse for Boolean flags.
275 \def\@booleantrue#1{\let#1\true@sw}%
276 \def\@booleanfalse#1{\let#1\false@sw}%

\@ifx We can now invoke the defining word to create the procedures \@ifx and friends.
\@ifx@empty Compatability Note: earlier versions of this package defined a procedure
\@ifempty \@ifempty. However, for compatability with AMS $\LaTeX$ , we must avoid the fol-
\@ifcat lowing three names: \@ifempty, \@xifempty, and \@ifnotempty.
\@ifdim 277 \@boole@def\@ifx#1{\ifx#1}%
\@ifeof 278 \@boole@def\@ifx@empty#1{\ifx\@empty#1}%
\@ifhbox 279 \@boole@def\@ifempty#1{\if!#1!}%
\@ifhmode 280 %\@boole@def\@if@sw#1{\csname if#1\endcsname}%
\@ifinner 281 \def\@if@sw#1#2{#1\expandafter\true@sw\else\expandafter\false@sw#2}%
\@ifmmode 282 \@boole@def\@ifdim#1{\ifdim#1}%
\@ifnum 283 \@boole@def\@ifeof#1{\ifeof#1}%
\@ifodd 284 \@boole@def\@ifhbox#1{\ifhbox#1}%
\@ifvbox 285 \@boole@def\@ifhmode{\ifhmode}%
\@ifvmode 286 \@boole@def\@ifinner{\ifinner}%
\@ifvoid 287 \@boole@def\@ifmmode{\ifmmode}%
288 \@boole@def\@ifnum#1{\ifnum#1}%
289 \@boole@def\@ifodd#1{\ifodd#1}%
290 \@boole@def\@ifvbox#1{\ifvbox#1}%
291 \@boole@def\@ifvmode{\ifvmode}%
292 \@boole@def\@ifvoid#1{\ifvoid#1}%

\true@sw Note that when a Boolean operator expands, it employs two macros that act as
\false@sw selectors, defined here.
293 \long\def\true@sw#1#2{#1}%
294 \long\def\false@sw#1#2{#2}%

\loopuntil Loop control using the Boolean idiom. Superior to \loop... \repeat because these
\loopwhile can be nested. The tail of the argument must have a Boolean predicate.
295 \long\def\loopuntil#1{#1}{\loopuntil{#1}}%
296 \long\def\loopwhile#1{#1}{\loopwhile{#1}}%

\@provide A defining word that refuses to clobber a prior meaning.
297 \def\@provide#1{%
298 \@ifx{\undefined#1}{\true@sw}{\@ifx{\relax#1}{\true@sw}{\false@sw}}%
299 {\def#1}{\def\j@nk}%
300 }%

```

6.6 Begin Document Structure

The standard L^AT_EX mechanism `\AtBeginDocument` is inadequate because the `\vsize` is bound much too early. We supply here a mechanism whereby decisions about the page layout can be deferred until `\AtBeginDocument` time.

The problem we are working around is that the `\AtBeginDocument` hook in `\document` appears long after the calculation of `\vsize` and `\hsize`, that is, L^AT_EX provides no mechanism for deferring the decision about the page grid until `\AtBeginDocument` time. We fix things by prepending a hook at the very beginning of `\document`.

`\document` We begin by installing hooks into `\document` that we will manage ourselves.

The 2020-10-01 L^AT_EX release got a new hook management system and several new hooks (several previously provided by `etoolbox`). The one we want here is `begindocument/before`, the first thing executed by `\document`, right after ending the group started by `\begin`.

Thus, if the L^AT_EX kernel date is 2020-10-01 we just add to that hook, otherwise resort to the old method, patching `\document`: end the group started by `\begin`, apply our hook, and conclude our shenanigans by absorbing the first token of the expansion of `\document`, which we assume to be `\endgroup` (true until the aforementioned release).

```
301 \rvtx@ifformat@geq{2020-10-01}%
302  {%
303   \AddToHook{begindocument/before}{%
304     \init@documenthook
305     \set@typesize@hook
306     \normalsize
307     \set@pica@hook
308   }%
309 }{%
310   \prepdef\document{%
311     \endgroup
312     \init@documenthook
313     \set@typesize@hook
314     \normalsize
315     \set@pica@hook
316     \true@sw{%
317   }%
318 }
```

`\class@documenthook` We install the first `\AtBeginDocument` hook, namely the procedure `\class@documenthook`.
`\class@enddocumenthook` Within the document class, we will use this hook exclusively, so as to avoid interference from other packages. Similarly with `\class@enddocumenthook`, installed via `\AtEndDocument`.

A document class using this package should do as this package does and just say, `\appdef\class@documenthook` and `\appdef\class@enddocumenthook` instead of `\AtBeginDocument` and `\AtEndDocument`.

```
319 \def\init@documenthook{%
```

```

320 \AtBeginDocument{%
321 \class@documenthook
322 }%
323 \AtEndDocument{%
324 \class@enddocumenthook
325 }%
326 \def\class@documenthook{}%
327 \def\class@enddocumenthook{}%

```

`\set@typesize@hook` The macros `\set@typesize@hook` and `\set@pica@hook` provide everything we
`\set@pica@hook` need. To use, simply `\appdef` your tokens to the appropriate hook.

```

328 \def\set@typesize@hook{}%
329 \def\set@pica@hook{}%

```

`\enddocument` The standard L^AT_EX `\end{document}` processing is a potential problem, particu-
`\check@aux` larly when the output routine has been changed by `ltxgrid`. We separate out the
`\do@check@aux` procedure that checks the auxiliary file at the end of the job so that later it can
be called from the safety of the output routine. We will do this to ensure that the
`\@mainaux` stream is not closed until the last page of the job is shipped out, and
that can only be done by coordinating with the output routine.

This approach, however, will only be done for older versions of the L^AT_EX kernel:

```

330 \rvtx@ifformat@geq{2020-10-01}{%
331 % <definitions for newer LaTeX later>
332 }{%
333 % <definitions for older LaTeX>
334 \def\enddocument{%
335 \@enddocumenthook
336 \@checkend{document}%
337 \clear@document
338 \check@aux
339 \deadcycles\z@
340 \@@end
341 }%
342 \def\clear@document{\clearpage}%
343 \def\check@aux{\do@check@aux}%
344 \def\do@check@aux{%
345 \@if@sw@if@filesw\fi{%
346 \immediate\closeout\@mainaux
347 \let\@setckpt\@gobbletwo
348 \let\@newl@bel\@testdef
349 \@tempwafalse
350 \makeatletter
351 \input\jobname.aux\relax
352 }}%
353 \@dofilelist
354 \@ifdim{\font@submax >\fontsubfuzz}\relax{%
355 \@font@warning{%
356     Size substitutions with differences\MessageBreak
357     up to \font@submax\space have occurred.\@gobbletwo

```



```

358 }%
359 }{}%
360 \@defaultsubs
361 \@refundefined
362 \@if@sw@if@filesw\fi{%
363 \@ifx{\@multiplelabels\relax}{%
364   \@if@sw@if@tempswa\fi{%
365     \@latex@warning@no@line{%
366       Label(s) may have changed.
367       Rerun to get cross-references right%
368     }%
369   }{}%
370 }{}%
371   \@multiplelabels
372 }%
373 }{}%
374 }%
375 }

```

`\rvtx@enddocument@patch` For newer L^AT_EX we'll try to be a bit more future-proof (no miracle though). The code for `\enddocument` (in pre-2020-10-01 L^AT_EX) is roughly:

```

% \def\enddocument{%
%   <hooks and bookkeeping>
%   \clearpage
%   <read main .aux and final checks>
%   \@@end
% }
%

```

and the patches above replace the `\clearpage` by its own `\clear@document`, and `<read main .aux and final checks>` by `\do@check@aux`, which it can later control the timing.

Now we will apply the same changes, but this time without redefining `\enddocument`: we will instead replace tokens on-the-fly, when `\enddocument` is expanded. This will grant us a slightly safer approach that won't depend so much on the internals of `\enddocument`.

This entire patch should work with the previous definition of `\enddocument` as well (except it cannot be used in the hook), but for now leave previous versions untouched.

The entire patching will reside in the `enddocument` hook:

```

376 \rvtx@ifformat@geq{2020-10-01}{%
377   \AddToHook{enddocument}{\rvtx@enddocument@patch{}}%
378 }{}

```

This macro will be executed after `\enddocument` has expanded, so all its tokens are now exposed. Here we will assume that `\enddocument` contains the tokens `\@checkend{document}` and `\endgroup`, and use them as delimiters:

```

379 \protected\long\def\rvtx@enddocument@patch#1#2\@checkend#3{%

```

```

380 \begingroup
381   \edef\x{\detokenize{#3}}%
382   \edef\y{\detokenize{document}}%
383 \expandafter\endgroup
384 \ifx\x\y
385   \expandafter\rvtx@enddocument@patch@end
386 \else
387   \expandafter\rvtx@enddocument@patch@more
388 \fi
389   {#1#2}{#3}}
390 \def\rvtx@enddocument@patch@more#1#2{%
391 \rvtx@enddocument@patch{#1\@checkend{#2}}

```

When the `\@checkend{document}` is reached, use `\clearpage` and `\enddocument` as delimiters for the `<read main .aux and final checks>` part, and save it in `\do@check@aux`:

```

392 \long\def\rvtx@enddocument@patch@end#1#2\clearpage#3\endgroup{%
393 \def\do@check@aux{#3\endgroup}}%

```

Then execute the code consumed in the previous step:

```

394 #1%
395 \@checkend{#2}%

```

Do `\clear@document` instead of `\clearpage` and `\check@aux` instead of the code grabbed.

```

396 \clear@document
397 \check@aux}
398 \def\check@aux{\do@check@aux}%
399 \def\clear@document{\clearpage}%

```

6.7 Type Tools

`\flushing` Undoes `\centering`. Should also undo `\raggedleft` and `\raggedright`.

```

400 \def\flushing{%
401   \let\\\@normalcr
402   \leftskip\z@skip
403   \rightskip\z@skip
404   \@rightskip\z@skip
405   \parfillskip\@flushglue
406 }%

```

6.8 Display Math

`\eqnarray@LaTeX` Team L^AT_EX has stated they will never repair Leslie's broken definition of `\eqnarray`. Let us be bold...

Note on `hyperref` package compatability: that package overrides `\eqnarray` by wrapping it up in a larger procedure, so its changes are compatible with this package's changes.

```

407 \def\eqnarray@LaTeX{%

```

```

408 \stepcounter{equation}%
409 \def\@currentlabel{\p@equation\theequation}%
410 \global\@eqnswtrue
411 \m@th
412 \global\@eqcnt\z@
413 \tabskip\@centering
414 \let\@@@eqncr
415 $$\everycr{\halign\t@{\displaywidth\bgroup
416 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
417 &\global\@eqcnt\@ne\hskip\tw@\arraycolsep\hfil${##}$\hfil
418 &\global\@eqcnt\tw@\hskip\tw@\arraycolsep
419 $\displaystyle{##}$\hfil\tabskip\@centering
420 &\global\@eqcnt\thr@@\hb@xt@\z@\bgroup\hss##\egroup
421 \tabskip\z@skip
422 \cr
423 }
424 \long\def\eqnarray@fleqn@fixed{%
425 \stepcounter{equation}\def\@currentlabel{\p@equation\theequation}%
426 \global\@eqnswtrue\m@th\global\@eqcnt\z@
427 \tabskip\mathindent
428 \let\@@@eqncr
429 \setlength\abovedisplayskip{\topsep}%
430 \ifvmode\addtolength\abovedisplayskip{\partopsep}\fi
431 \addtolength\abovedisplayskip{\parskip}%
432 \setlength\belowdisplayskip{\abovedisplayskip}%
433 \setlength\belowdisplayshortskip{\abovedisplayskip}%
434 \setlength\abovedisplayshortskip{\abovedisplayskip}%
435 $$%
436 \everycr{}%
437 \halign@\linewidth\bgroup
438 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
439 &\global\@eqcnt\@ne
440 \hskip\tw@\eqncolsep
441 \hfil${##}$\hfil
442 &\global\@eqcnt\tw@
443 \hskip\tw@\eqncolsep
444 $\displaystyle{##}$\hfil\tabskip\@centering
445 &\global\@eqcnt\thr@@\hb@xt@\z@\bgroup\hss##\egroup
446 \tabskip\z@skip
447 \cr
448 }%
449 \@ifx{\eqnarray\eqnarray@LaTeX}{-%
450 \class@info{Repairing broken LaTeX eqnarray}%
451 \let\eqnarray\eqnarray@fleqn@fixed
452 \newlength\eqncolsep
453 \setlength\eqncolsep\z@
454 \let\eqnarray@LaTeX\relax
455 \let\eqnarray@fleqn@fixed\relax
456 }{}%
457 \def\mathindent{\@centering}%

```

```
458 \def\set@eqnarray@skips{}
```

6.9 Footnotes

```
\footnote We repair an error in the LATEX kernel (see ltfloat.dtx) involving footnotes.
\footnotemark The symptom is that the \footnotemark command does not work properly
\@xfootnote within a minipage environment. The source of the problem is in the way the
\@xfootnotemark \footnotemark and \@xfootnotemark procedures are defined: they do not share
\@yfootnote the method used by the \footnote and other procedures that allows a context
switch to change the way footnotes behave within a minipage environment. This
is a LATEX bug of long standing; this fix dates to 1987.
```

While we are at it, we rewrite both the `\footnote` and `\footnotemark` procedures, achieving a slightly cleaner separation of syntax and semantics. Note that the `\@footnotemark` and `\@footnotetext` procedures are not altered here; they continue as the methods of formatting the footnote mark and footnote text, respectively.

A note about the context switch mentioned above: the `minipage` environment executes the following in order to alter the way footnotes behave:

```
%\def\@mpfn{mpfootnote}%
%\def\thempfn{\thempfootnote}%
%\let\@footnotetext\@mpfootnotetext
%\c@mpfootnote\z@
%
```

This code changes the counter used in autonumbered footnotes, the choice of footnote marker, and the procedure used on the footnote text. Changing the counter is needed because `minipage` footnotes are in their own sequence, and the footnote marker is customarily different within a `minipage`. The procedure that works on the footnote text must be different because the footnotes are placed at the bottom of the `minipage`, not the bottom of the text column.

Any procedure that establishes a `minipage`-like context (e.g., floats) can do the same.

```
459 \def\footnote{%
460   \@ifnextchar[\@xfootnote{\@yfootnote\@footnotetext}%
461 }%
462 \def\footnotemark{%
463   \@ifnextchar[\@xfootnotemark{\@yfootnote}%
464 }%
465 \def\@xfootnote[#1]{%
466   \@xfootnotemark[#1]%
467   \@footnotetext
468 }%
469 \def\@xfootnotemark@ltx[#1]{%
470   \begingroup
471     \csname c@\@mpfn\endcsname #1\relax
472     \unrestored@protected@xdef\@thefnmark{\thempfn}%
473   \endgroup
```

```

474 \H@@footnotemark
475 }%
476 \def\@yfootnote{%
477 \stepcounter\@mpfn
478 \protected@xdef\@thefnmark{\thempfn}%
479 \H@@footnotemark
480 }%

```

Note on `hyperref` compatability: In its “Automated L^AT_EX hypertext cross-references”, the `hyperref` package alters footnote processing, thereby imperiling these fixes and necessiating defensive measures.

The main thing `hyperref` does is to take over the `\@mpfootnotetext` and `\@footnotetext` procedures, feeding its own arguments to these macros. It also rewrites `\@footnotemark`, making it a hyperlink.

But at the same time, it attempts to turn off these changes during `\maketitle` processing, necessitating rewriting `\@xfootnotemark`. At this point it is on the slippery slope.

We make ourself `hyperref` friendly: we give `hyperref` what it needs, but avoid its change to `\@xfootnotemark`.

Any other package that rewrites L^AT_EX’s footnote macros will be incompatible with this package.

```

481 \appdef\class@documenthook{%
482 \@ifxundefined\H@@footnotemark{%
483 \let\H@@footnotemark\@footnotemark
484 }{}%
485 \let\@xfootnotemark\@xfootnotemark@ltx
486 }%

```

Two thoughts about `hyperref`: what for does it define `\realfootnote`? Also: a document class that desires high hypertext capabilities might well wish to reimplement `\maketitle` so that footnotes called out from there are hypertext links: the `hyperref` package’s “Automated L^AT_EX hypertext cross-references” does not do any of this:

But the special footnotes in `\maketitle` are much too hard to deal with properly. Let them revert to plain behaviour.

Note that the document class, in reimplementing `\maketitle`, must ensure that the `hyperref` package does not clobber its own definition!

<pre> \@footnotetext \@mpfootnotetext \@tpfootnotetext \make@footnotetext \set@footnotewidth </pre>	<p>The two procedures <code>\@footnotetext</code> and <code>\@mpfootnotetext</code> share code. We make that explicit here.</p> <p>Note that the procedure calling <code>\make@footnotetext</code> will open a group with <code>\bgroup</code> which is then closed by <code>\minipagefootnote@drop</code>.</p>
---	---

Difference from L^AT_EX: here we do not set `\floatingpenalty` to infinity. Doing this must date back to a time when L^AT_EX could not accomodate split insertions (footnotes). I cannot think of any other reason to do have done this. At any rate, with the `ltxgrid` package, split insertions are specifically properly taken care of, so we allow it.

We provide the hook `\set@footnotewidth` that sets the footnote on a particular measure. Some page grids are such as to set a footnote in a context where `\columnwidth` is not the right parameter to use for the set width of a footnote. In such a case, for the applicable scope, you should define `\set@footnotewidth` to perform this job correctly.

A procedure, `\set@footnotewidth@ii`, illustrates how to do this when in a two-column page grid. In general, remember that footnotes, like all insertions (including floats), are a step outside of the galley context, and all aspects of insertions need to be properly handled, including the set width.

```

487 \long\def\@footnotetext{%
488 \insert\footins\bgroup
489 \make@footnotetext
490 }%
491 \long\def\@mpfootnotetext{%
492 \minipagefootnote@pick
493 \make@footnotetext
494 }%
495 \def\make@footnotetext#1{%
496 \reset@font\footnotesize
497 \interlinepenalty\interfootnotelinepenalty
498 \splittopskip\footnotesep
499 \splitmaxdepth\dp\strutbox
500 % \floatingpenalty\@MM
501 \set@footnotewidth
502 \@parboxrestore
503 \protected@edef\@currentlabel{%
504 \csname p@footnote\endcsname\@thefnmark
505 }%
506 \color@begingroup
507 \@makefntext{%
508 \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox
509 }%
510 \color@endgroup
511 \minipagefootnote@drop
512 }%
513 \def\set@footnotewidth{%
514 \hsize\columnwidth
515 \linewidth\hsize
516 }%
517 \def\set@footnotewidth@ii{%
518 \hsize\textwidth
519 \advance\hsize\columnsep
520 \divide\hsize\tw@
521 \advance\hsize-\columnsep
522 \linewidth\hsize
523 }%

```

6.10 Floats

6.10.1 Usage notes

We extend the \LaTeX kernel for three purposes:

1. When the `\footnote` command is used within the scope of a float, we do as `minipage` does.
2. We provide a mechanism to write floats out to an external stream for temporary storage (deferred floats).
3. We provide mechanism for placing a float **here** invariably, that is, floats are unfloat. This mechanism is used to read the external stream mentioned above.

To use these mechanisms, the document class should define a float, say, `figure` as per usual, and in addition:

1. Optionally define an alternative, say `figure@write` as follows:

```
%\newenvironment{figure@write}{%  
% \write@float{figure}%  
%}%  
% \endwrite@float  
%}%  
%
```

That is, the alternative environment executes `\write@float` instead of `\@float`. Note that this step is not needed if the float environment is defined in the simple way of `classes.dtx`. However, an environment like `longtable` will require it.

2. Install into `\AtBeginDocument` a call to `\do@if@floats`, with the float name and an appropriate file extension as its arguments.

```
%\AtBeginDocument{\do@if@floats{figure}{.fgx}}%  
%
```

3. Optionally define a text entity `\figuresname` that will be the text of the head that is set over the deferred floats. If not defined, there will be no head.
4. Optionally define a user-level command to allow the document to determine where the figures are printed out (default is to print at end of document).
E.g.,

```
%\newcommand\printfigures{\print@float{figure}}%  
%
```

5. Install into `\appdef\class@enddocumenthook` a call to `\printfigures`, or, if the latter is not defined, as follows:

```
%\appdef\class@enddocumenthook{\print@float{figure}}%
%
```

Note that installing this command into `\AtBeginDocument` is best done earlier than calls that assume the last page of the document is at hand.

6.10.2 Robustifying fragile commands

Certain of L^AT_EX's commands cannot be written out to a file or appear within a `\mark` command argument because they do calculations during expansion. We provide for a little help, but without changing the meanings of these commands.

```
\addtocontents
524 \def\addtocontents#1#2{%
525   \protected@write\@auxout{%
526     \let \label \@gobble \let \index \@gobble \let \glossary \@gobble
527     \def\({\string\}%
528     \def\)\string\}%
529     \def\\{\string\\}%
530   }\string \@writefile {#1}{#2}}%
531 }%
```

6.10.3 Preparing for the hyperref package

`\addcontentsline` The `hyperref` package assumes that the `\contentsline` command will be given four arguments. Therefore it cannot successfully process a `ltxutil.dtx.toc` file that had been written by standard L^AT_EX. We fix things up by always writing that fourth argument and by supplying a `\contentsline` command that can read them.

We also give the `\newlabel` command's second argument five tokens.

This means that a document class that uses this package will itself have trouble taking over a `ltxutil.dtx.toc` file that was written by standard L^AT_EX. Sigh.

```
532 \def\addcontentsline#1#2#3{%
533   \addtocontents{#1}{%
534     \protect\contentsline{#2}{#3}{\thepage}{}%
535   }%
536 }%
537 \def\contentsline#1#2#3#4{%
538   \csname l@#1\endcsname{#2}{#3}%
539 }%
540 \def\label#1{%
541   \@bsphack
542   \protected@write\@auxout{%
543     \string\newlabel{#1}{\@currentlabel}{\thepage}{}{}}%
544   }%
545   \@esphack
546 }%
```


6.10.4 Footnotes within floats, unfloating floats, float font

`\caption` DPC: Er a bit of a hack, but seems best way of supporting normal L^AT_EX syntax at this point: If a caption is used below a table, then put out the footnotes before the caption.

```
547 \appdef\class@documenthook{%
548 \prepdef\caption{\minipagefootnote@here}%
549 }%
```

Note on `hyperref` compatability: this change to the `\caption` command is compatible with the “Automated L^AT_EX hypertext cross-references” patches of that package.

All the same, I think Sebastian’s changes to `\caption` and `\@caption` could bear with some improvement. The following implementation requires knowing only the pattern part of the `\@caption` macro:

```
%\def\caption{%
% \H@refstepcounter\@capttype
% \hyper@makecurrent{\@capttype}%
% \@dblarg{\H@caption\@capttype}%
%}%
%\def\H@caption#1[#2]#3{%
% \@caption{#1}[#2]{%
% \ifHy@nesting
% \hyper@@anchor{\@currentHref}{#3}%
% \else
% \hyper@@anchor{\@currentHref}{\relax}#3%
% \fi
% }%
%}%
%
%
```

`\minipagefootnote@init` Procedure to deal with footnotes accumulated within a minipage environment.

`\minipagefootnote@here` These procedures encapsulate all uses of the `\@mpfootins` box.

`\minipagefootnote@foot` Note: `\minipagefootnote@here` must *not* be executed within the MVL!

```
\minipagefootnote@pick 550 \def\minipagefootnote@init{%
\minipagefootnote@drop 551 \setbox\@mpfootins\box\voidb@x
552 }%
553 \def\minipagefootnote@pick{%
554 \global\setbox\@mpfootins\vbox\bgroup
555 \unvbox\@mpfootins
556 }%
557 \def\minipagefootnote@drop{%
558 \egroup
559 }%
560 \def\minipagefootnote@here{%
561 \par
562 \@ifvoid\@mpfootins{}{%
563 \vskip\skip\@mpfootins
```

```

564     \fullinterlineskip
565     \@ifinner{%
566         \vtop{\unvcopy\@mpfootins}%
567         {\setbox\z@\lastbox}%
568     }{%
569     \unvbox\@mpfootins
570     }%
571 }%
572 \def\minipagefootnote@foot{%
573 \@ifvoid\@mpfootins{}{%
574 \insert\footins\bgroup\unvbox\@mpfootins\egroup
575 }%
576 }%
577 \def\endminipage{%
578     \par
579     \unskip
580     \minipagefootnote@here
581     \@minipagefalse %% added 24 May 89
582 \color@endgroup
583 \egroup
584 \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}%
585 }%

```

`\floats@sw` The Boolean `\floats@sw` signifies that floats are to be floated; if false, that floats are to be deferred to the end of the document. Note that the state of this Boolean is to be changed by the document class in response to user-selected options. Here we display model code that assigns a default value at `\AtBeginDocument` time.

```

%\AtBeginDocument{%
% \@ifxundefined\floats@sw{\@booleantrue\floats@sw}{}%
%}%
%

```

`\@xfloat` The float start-code is redefined to set up footnotes in the style of minipage. Also, `\@mpmakefn` the `\floats@sw` Boolean informs us that floats are to be all placed here. Note that, to protect against the Boolean being undefined at this late hour, we default it globally to true.

FIXME: why does hyperref override `\@xfootnotenext`?

```

586 \let\@xfloat@LaTeX\@xfloat
587 \def\@xfloat#1[#2]{%
588 \@xfloat@prep
589 \@nameuse{fp@proc@#2}%
590 \@ifxundefined\floats@sw{\global\@booleantrue\floats@sw}{}%
591 \floats@sw{\@xfloat@LaTeX{#1}[#2]}{\@xfloat@anchored{#1}[]}%
592 }%
593 \def\@xfloat@prep{%
594 \let\footnote\footnote@latex
595 \def\@mpfn{mpfootnote}%
596 \def\thempfn{\thempfootnote}%

```

```

597 % \def\thefootnote{\thempfootnote}%
598 \c@mpfootnote\z@
599 \let\@footnotetext\@mpfootnotetext
600 \let\H@footnotetext\@mpfootnotetext
601 \let\@makefntext\@mpmakefntext
602 % \samepage
603 }%
604 \appdef\class@documenthook{%
605 \let\footnote@latex\footnote
606 }%
607 %\def\fp@proc@h{\@booleanfalse\floats@sw}%
608 %\def\fp@proc@H{\@booleanfalse\floats@sw}%
609 \def\@xfloat@anchored#1[#2]{%
610 \def\@capttype{#1}%
611 \begin@float@pagebreak
612 %\vbox\bgroup
613 \let\end@float\end@float@anchored
614 \let\end@dblfloat\end@float@anchored
615 % do as \@xfloat does:
616 \hsize\columnwidth
617 \@parboxrestore
618 \@floatboxreset
619 \minipagefootnote@init
620 % \pagegrid@col\@ne % Klootch to avoid processing as a float
621 }%
622 \def\end@float@anchored{%
623 \minipagefootnote@here
624 \par\vskip\z@skip %% \par\vskip\z@ added 15 Dec 87
625 %\egroup
626 \par
627 \end@float@pagebreak
628 }%
629 \def\begin@float@pagebreak{\par\addvspace\intextsep}%
630 \def\end@float@pagebreak{\par\addvspace\intextsep}%
631 \def\@mpmakefntext#1{%
632 \parindent=1em
633 \noindent
634 \hb@xt@1em{\hss\@makefnmark}%
635 #1%
636 }%

```

6.10.5 Writing floats out to a file

`\do@if@floats` The procedure `\do@if@floats` should be executed at `\AtBeginDocument` time, and arranges to write out the floats of the given class to a temporary file, to be read back later (deferred floats), given that `\floats@sw` is false. Note that, to protect against the Boolean being undefined at this late hour, we default it globally to true.

```

637 \def\do@if@floats#1#2{%

```

```
638 \@ifxundefined\floats@sw{\global\@booleantrue\floats@sw}{}%
639 \floats@sw}{-%
```

Open the stream to save out the document's floats of this class.

```
640 \expandafter\newwrite
641         \csname#1write\endcsname
642 \expandafter\def
643         \csname#1@stream\endcsname{\jobname#2}%
644 \expandafter\immediate
645 \expandafter\openout
646         \csname#1write\endcsname
647         \csname#1@stream\endcsname\relax
```

Swap environments. If the class writer has defined, e.g., `figure@write`, then we use this as the procedure to execute for writing the float out to the external stream. Otherwise, the replacement of `\@float` by `\write@float` should do the right thing for float environments defined in the simple way of `classes.dtx`.

```
648 \@ifxundefined\@float@LaTeX{%
649 \let\@float@LaTeX\@float
650 \let\@dblfloat@LaTeX\@dblfloat
651 \let\@float\write@float
652 \let\@dblfloat\write@floats
653 }{-%
654 \let@environment{#1@float}{#1}%
655 \let@environment{#1@floats}{#1*}%
656 \@ifxundefined@cs{#1@write}{}{-%
657 \let@environment{#1}{#1@write}%
658 }%
659 }%
660 }%
```

`\print@float` The procedure `\print@float` prints out the deferred floats.

```
661 \def\triggerpar{\leavevmode\@par}%
662 \def\onepage{\def\begin@float@pagebreak{\newpage}\def\end@float@pagebreak{\newpage}}%
663 \def\print@float#1#2{%
664 \@ifxundefined@cs{#1write}{}{-%
665 \begingroup
666 \booleanfalse\floats@sw
667 #2%
668 \raggedbottom
669 \def\array@default{v}% floats must
670 \let\@float\@float@LaTeX
671 \let\@dblfloat\@dblfloat@LaTeX
672 \let\trigger@float@par\triggerpar
673 \let@environment{#1}{#1@float}%
674 \let@environment{#1*}{#1@floats}%
675 \expandafter\prepdef\csname#1\endcsname{\trigger@float@par}%
676 \expandafter\prepdef\csname#1*\endcsname{\trigger@float@par}%
677 \namedef{fps@#1}{h!}%
678 \expandafter\immediate
```

```

679 \expandafter\closeout
680         \csname#1write\endcsname
681 \everypar{%
682   \global\let\trigger@float@par\relax
683   \global\everypar{}\setbox\z@\lastbox
684   \@ifundefined@cs{#1sname}{-}{%
685     \begin@float@pagebreak
686     \expandafter\section
687     \expandafter*%
688     \expandafter{%
689       \csname#1sname\endcsname
690       }%
691   }%
692 }%
693 \input{\csname#1@stream\endcsname}%
694 \endgroup
695 \global\expandafter\let\csname#1write\endcsname\relax
696 }%
697 }%

```

`\write@float` Handles the case where the name of the float is the same as that of the stream.
`\write@floats` Note that `longtable` does *not* fit this case. Note also: `\write@float` is *not* a
`\write@@float` user-level environment therefore it is properly not defined with `\newenvironment`.

```

698 \def\write@float#1{\write@@float{#1}{#1}}%
699 \def\endwrite@float{\@Esphack}%
700 \def\write@floats#1{\write@@float{#1*}{#1}}%
701 \def\endwrite@floats{\@Esphack}%

```

`\write@@float`

```

702 \def\write@@float#1#2{%
703   \ifhmode
704     \@bsphack
705   \fi
706   \chardef\@tempc\csname#2write\endcsname
707   \toks@\{\begin{#1}}%
708   \def\@tempb{#1}%
709   \expandafter\let\csname end#1\endcsname\endwrite@float
710   \catcode'\^M\active
711   \@makeother\{\@makeother\}\@makeother\%
712   \write@floatline
713 }%

```

`\write@floatline` The procedure `\write@floatline` only parses, and passes its result to `\@write@floatline`,
`\@write@floatline` which writes the line to output, then tests the line for the `\end{float}` tokens
`\float@end@tag` with aid of the `\float@end@tag` procedure.

```

714 \begingroup
715 \catcode'\[\the\catcode'\{\catcode'\}\the\catcode'\}\@makeother\{\@makeother\}%
716 \gdef\float@end@tag#1\end{#2}#3\nul[%
717   \def\@tempa[#2]%

```

```

718 \@ifx[\@tempa\@tempb][\end[#2]][\write@floatline]%
719 ]%
720 \obeylines%
721 \gdef\write@floatline#1^M[%
722 \begingroup%
723 \newlinechar‘^M%
724 \toks@\expandafter[\the\toks@#1]\immediate\write\@tempc[\the\toks@]%
725 \endgroup%
726 \toks@[]%
727 \float@end@tag#1\end{}}\@nul%
728 ]%
729 \endgroup

```

6.11 Counters

The following definitions override those of the L^AT_EX kernel, providing for a greater range of inputs.

```

730 \def\@alph#1{\ifcase#1\or a\or b\or c\or d\else\@ialph{#1}\fi}
731 \def\@ialph#1{\ifcase#1\or \or \or \or \or e\or f\or g\or h\or i\or j\or
732 k\or l\or m\or n\or o\or p\or q\or r\or s\or t\or u\or v\or w\or x\or
733 y\or z\or aa\or bb\or cc\or dd\or ee\or ff\or gg\or hh\or ii\or jj\or
734 kk\or ll\or mm\or nn\or oo\or pp\or qq\or rr\or ss\or tt\or uu\or
735 vv\or ww\or xx\or yy\or zz\else\@ctrerr\fi}

```

6.12 Customization of Sections

Patch the standard L^AT_EX sectioning procedure to:

- Allow a sectioning command to trigger the title page, or more generally to recognize that it is the first object in the document, so we headpatch `\@startsection`.
- Allow a tail command in #6 to uppercase the title, so we retain DPC’s braces.
- Allow each type of sectioning command to format its number differently, so we generalize `\@secCNTformat`.
- Allow each type of sectioning command to format its argument differently, so we generalize `\@hangfrom`.
- Allow the starred form of the command to mark (the running head) and make an entry in the TOC, so we put `\@ssect` on the same footing as `\@sect`.

Note that the tokens passed to the TOC now are *not* the optional argument of the command, but the required. This means that the user can no longer use the former to put variant content in to the TOC as the Manual says.

Instead, the optional argument is used to put an alternative title into the running headers, a better choice.

`\@startsection` Patch a head hook into the basic sectioning command. Treat `\@sect` and `\@ssect` on an equal footing; now their pattern parts are identical.

```

736 \def\@startsection#1#2#3#4#5#6{%
737   \@startsection@hook
738   \ifnoskipsec \leavevmode \fi
739   \par
740   \@tempskipa #4\relax
741   \@afterindenttrue
742   \ifdim \@tempskipa <\z@
743     \@tempskipa -\@tempskipa \@afterindentfalse
744   \fi
745   \if@nobreak
746     \everypar{}%
747   \else
748     \addpenalty\@secpenalty\addvspace\@tempskipa
749   \fi
750   \@ifstar
751     {\@dblarg{\@ssect@ltx{#1}{#2}{#3}{#4}{#5}{#6}}}%
752     {\@dblarg{\@sect@ltx {#1}{#2}{#3}{#4}{#5}{#6}}}%
753 }%
754 \def\@startsection@hook{}%

```

`\@sect` When defining `\@svsec`, do not expand `\@secntformat`. Put brace characters back where they were before David Carlisle got at them (i.e., as if `\@hangfrom` had two arguments). Protect the mark mechanism from an undefined meaning. Pass #8 to the TOC instead of #7. Remove `\relax` from the replacement part of `\@svsec`.

The procedure `\@hangfrom` and `\@runin@to` can be used to process the argument of the head. The head can define, e.g., `\@hangfrom@section`, to do its own processing.

In using `\H@refstepcounter` in place of `\refstepcounter` we rely on either loading before any package that patches the latter, or the convention that the former is the original L^AT_EX procedure.

```

755 \class@info
756   {Repairing broken LaTeX \string\@sect}%
757 \def\@sect@ltx#1#2#3#4#5#6[#7]#8{%
758   \@ifnum{#2}>\c@secnumdepth}{%
759     \def\H@svsec{\phantomsection}%
760     \let\@svsec\@empty
761   }{%
762     \H@refstepcounter{#1}%
763     \def\H@svsec{%
764       \phantomsection
765     }%
766     \protected@edef\@svsec{#{1}}%
767     \@ifundefined{@#1cntformat}{%
768       \prepdef\@svsec\@secntformat
769     }{%

```

```

770     \expandafter\prepdef
771     \expandafter\@svsec
772         \csname @#1cntformat\endcsname
773     }%
774 }%
775 \@tempskipa #5\relax
776 \@ifdim{\@tempskipa>\z@}{%
777     \begingroup
778         \interlinepenalty \@M
779         #6{%
780             \ifundefined{@hangfrom@#1}{\@hang@from}{\csname @hangfrom@#1\endcsname}%
781             {\hskip#3\relax\H@svsec}{\@svsec}{#8}%
782         }%
783         \@@par
784     \endgroup
785     \ifundefined{#1mark}{\@gobble}{\csname #1mark\endcsname}{#7}%
786     \addcontentsline{toc}{#1}{%
787         \@ifnum{#2}>\c@secnumdepth}{%
788             \protect\numberline{}}%
789         }{%
790             \protect\numberline{\csname the#1\endcsname}}%
791         }%
792         #8}%
793 }{%
794     \def\@svsechd{%
795         #6{%
796             \ifundefined{@runin@to@#1}{\@runin@to}{\csname @runin@to@#1\endcsname}%
797             {\hskip#3\relax\H@svsec}{\@svsec}{#8}%
798         }%
799         \ifundefined{#1mark}{\@gobble}{\csname #1mark\endcsname}{#7}%
800         \addcontentsline{toc}{#1}{%
801             \@ifnum{#2}>\c@secnumdepth}{%
802                 \protect\numberline{}}%
803             }{%
804                 \protect\numberline{\csname the#1\endcsname}}%
805             }%
806             #8}%
807     }%
808 }%
809 \@xsect{#5}%
810 }%
811 \def\@hang@from#1#2#3{\@hangfrom{#1#2}#3}%
812 \def\@runin@to #1#2#3{#1#2#3}%

```

\@ssect Put brace characters back where they were before David Carlisle got at them (as if `\@hangfrom` has two arguments). Possibly set a mark. Make a TOC entry.

Note that, for compatibility with the `hyperref` package, we need to provide the interface required by that package (actually required by `pdfmark.def` and `nameref.sty`), namely the definition of `\@currentlabelname` (but now removed), the insertion of the procedure `\Sectionformat` (but why is this needed?), and the

call to `\phantomsection` (which must precede the call to `\addcontentsline`). We also have to sidestep the patch to `\@ssect` in that same file, therefore we use a different control sequence name in the call from `\@startsection`.

```

813 \def\@ssect@ltx#1#2#3#4#5#6[#7]#8{%
814 % \def\@currentlabelname{#8}%
815 \def\H@svsec{\phantomsection}%
816 \@tempskipa #5\relax
817 \@ifdim{\@tempskipa>\z@}{%
818   \begingroup
819     \interlinepenalty \@M
820     #6{%
821       \@ifundefined{@hangfroms@#1}{\@hang@froms}{\csname @hangfroms@#1\endcsname}%
822 %     {\hskip#3\relax\H@svsec}{\Sectionformat{#8}{#1}}%
823     {\hskip#3\relax\H@svsec}{#8}%
824     }%
825     \@@par
826   \endgroup
827   \@ifundefined{#1smark}{\@gobble}{\csname #1smark\endcsname}{#7}%
828   \addcontentsline{toc}{#1}{\protect\numberline{#8}}%
829 }{%
830 \def\@svsechd{%
831   #6{%
832     \@ifundefined{@runin@tos@#1}{\@runin@tos}{\csname @runin@tos@#1\endcsname}%
833 %   {\hskip#3\relax\H@svsec}{\Sectionformat{#8}{#1}}%
834   {\hskip#3\relax\H@svsec}{#8}%
835   }%
836   \@ifundefined{#1smark}{\@gobble}{\csname #1smark\endcsname}{#7}%
837   \addcontentsline{toc}{#1}{\protect\numberline{#8}}%
838 }%
839 }%
840 \@xsect{#5}%
841 }%
842 \def\@hang@froms#1#2{#1#2}%
843 \def\@runin@tos #1#2{#1#2}%

```

`\init@documenthook` Document classes that incorporate this package will be `hyperref-savvy`. (To accomplish this, we ensure that `\hyperanchor` and `\hyper@last` are both defined.) Being `hyperref-savvy` levels some requirements on us, but the benefits are many.

One is that the TOC will not get amnesia and require a full set of three typesetting runs before its formatting is stable. Instead, only two runs are required: the first updates the auxiliary file, the second the TOC. However, the formatting of the document does not change.

Another aspect of being `hyperref-savvy` is that the syntax of commands in the `ltxutil.dtx.aux` file will now change if `hyperref` is turned on or off.

Note that `\hyper@anchorstart` and `\hyper@anchorend` constitute the programming interface for a hypertext anchor (the target of a hypertext link); `\hyper@linkstart` and `\hyper@linkend` are the interface for a hypertext link.

```

844 \appdef\init@documenthook{%

```

```

845 \providecommand\phantomsection{}%
846 %\@ifx{\Sectionformat\undefined}{\let\Sectionformat\@firstoftwo}{}%
847 \providecommand\hyper@anchor[1]{}%
848 \providecommand\hyper@last{}%
849 \providecommand\Hy@raisedlink[1]{#1}%
850 \providecommand\hyper@anchorstart[1]{}%
851 \providecommand\hyper@anchorend{}%
852 \providecommand\hyper@linkstart[2]{}%
853 \providecommand\hyper@linkend{}%
854 }%
855 \let\H@refstepcounter\refstepcounter

```

`\sec@upcase` Upper case for sections (optional upper case items). These are created so that some headings can be toggled between mixed case and upper case readily. Headings that might be changed can be wrapped in the style file in `\sec@upcase{<text>}` constructs; the expansion of `\sec@upcase` is controlled here. It is `\relax` by default (mixed case heads), and can easily be changed to `\uppercase` if desired. If mixed-case headings are wanted by the editor, authors *must* supply mixed case text, although this is what authors should be doing anyway. (Mixed can be converted to upper, but the reverse transformation cannot be automated.)

The following setting gives the L^AT_EX default.

```
856 \def\sec@upcase#1{\relax{#1}}%
```

6.13 Patch the tabular and array Environments

`\endtabular` We headpatch the begin processing and tailpatch the end processing of the `\endarray` `\endarray` tabular and array environments. A document class can define these hooks as needed.

We proceed with care to make further patches to support tabulars that break over pages. Our patches will not necessarily be effective for other packages that replace the L^AT_EX array and tabular environments. I know of none that do so.

```

857 \appdef\class@documenthook{%
858 \@ifpackageloaded{array}{\switch@array}{\switch@tabular}%
859 \prepdef\endtabular{\endtabular@hook}%
860 \@provide\endtabular@hook{}%
861 \prepdef\endarray{\endarray@hook}%
862 \@provide\endarray@hook{}%
863 \providecommand\array@hook{}%

```

Install, effectively, a head patch to `\tabular`. In order to avoid interference from, e.g., the `array` package, we must perform this patch only *after* packages load.

```

864 \prepdef\@tabular{\tabular@hook}%
865 \@provide\tabular@hook{}%
866 }%

```

`\switch@tabular` The two procedures `\switch@tabular` and `\switch@array` apply needed patches to the various tabular procedures, the former applying to the L^AT_EX kernel, the `\switch@array`

latter to the required array package (and to the number of other required packages that load it).

```

867 \def\switch@tabular{%
868 \let\@array@sw\@array@sw@array
869 \@ifx{\@array\@array@LaTeX}{%
870 \@ifx{\multicolumn\multicolumn@LaTeX}{%
871 \@ifx{\@tabular\@tabular@LaTeX}{%
872 \@ifx{\@tabarray\@tabarray@LaTeX}{%
873 \@ifx{\array\array@LaTeX}{%
874 \@ifx{\endarray\endarray@LaTeX}{%
875 \@ifx{\endtabular\endtabular@LaTeX}{%
876 \@ifx{\@mkpream\@mkpream@LaTeX}{%
877 \@ifx{\@addamp\@addamp@LaTeX}{%
878 \@ifx{\@arrayacol\@arrayacol@LaTeX}{%
879 \@ifx{\@tabacol\@tabacol@LaTeX}{%
880 \@ifx{\@arrayclassz\@arrayclassz@LaTeX}{%
881 \@ifx{\@tabclassiv\@tabclassiv@LaTeX}{%
882 \@ifx{\@arrayclassiv\@arrayclassiv@LaTeX}{%
883 \@ifx{\@tabclassz\@tabclassz@LaTeX}{%
884 \@ifx{\@classv\@classv@LaTeX}{%
885 \@ifx{\hline\hline@LaTeX}{%
886 \@ifx{\@tabularcr\@tabularcr@LaTeX}{%
887 \@ifx{\@xtabularcr\@xtabularcr@LaTeX}{%
888 \@ifx{\@xargarraycr\@xargarraycr@LaTeX}{%
889 \@ifx{\@yargarraycr\@yargarraycr@LaTeX}{%
890 \true@sw
891 }{%
892 \false@sw
893 }%
894 }{%
895 \false@sw
896 }%
897 }{%
898 \false@sw
899 }%
900 }{%
901 \false@sw
902 }%
903 }{%
904 \false@sw
905 }%
906 }{%
907 \false@sw
908 }%
909 }{%
910 \false@sw
911 }%
912 }{%
913 \false@sw

```

```

914         }%
915     }{%
916         \false@sw
917     }%
918 }{%
919     \false@sw
920 }%
921 }{%
922     \false@sw
923 }%
924 }{%
925     \false@sw
926 }%
927 }{%
928     \false@sw
929 }%
930 }{%
931     \false@sw
932 }%
933 }{%
934     \false@sw
935 }%
936 }{%
937     \false@sw
938 }%
939 }{%
940     \false@sw
941 }%
942 }{%
943     \false@sw
944 }%
945 }{%
946     \false@sw
947 }%
948 }{%
949     \false@sw
950 }%
951 }{%
952     \false@sw
953 }%
954 {%
955 \class@info{Patching LaTeX tabular.}%
956 }{%
957 \class@info{Unrecognized LaTeX tabular. Please update this document class! (Proceeding with f
958 }%
959 \let\@array\@array@ltx
960 \let\multicolumn\multicolumn@ltx
961 \let\@tabular\@tabular@ltx
962 \let\@tabarray\@tabarray@ltx
963 \let\array\array@ltx

```

```

964 \let\endarray\endarray@ltx
965 \let\endtabular\endtabular@ltx
966 \let\@mkpream\@mkpream@ltx
967 \let\@addamp\@addamp@ltx
968 \let\@arrayacol\@arrayacol@ltx
969 \let\@tabacol\@tabacol@ltx
970 \let\@arrayclassz\@arrayclassz@ltx
971 \let\@tabclassiv\@tabclassiv@ltx
972 \let\@arrayclassiv\@arrayclassiv@ltx
973 \let\@tabclassz\@tabclassz@ltx
974 \let\@classv\@classv@ltx
975 \let\hline\hline@ltx
976 \let\@tabularcr\@tabularcr@ltx
977 \let\@xtabularcr\@xtabularcr@ltx
978 \let\@xarraycr\@xarraycr@ltx
979 \let\@yarraycr\@yarraycr@ltx
980 }%
981 \def\switch@array{%
982 \let\@array@sw\@array@sw@LaTeX
983 \@ifx{\@array\@array@array}{%
984 \@ifx{\@tabular\@tabular@array}{%
985 \@ifx{\@tabarray\@tabarray@array}{%
986 \@ifx{\array\array@array}{%
987 \@ifx{\endarray\endarray@array}{%
988 \@ifx{\endtabular\endtabular@array}{%
989 \@ifx{\@mkpream\@mkpream@array}{%
990 \@ifx{\@classx\@classx@array}{%
991 \@ifx{\insert@column\insert@column@array}{%
992 \@ifx{\@arraycr\@arraycr@array}{%
993 \@ifx{\@xarraycr\@xarraycr@array}{%
994 \@ifx{\@xarraycr\@xarraycr@array}{%
995 \@ifx{\@yarraycr\@yarraycr@array}{%
996 \true@sw
997 }{%
998 \false@sw
999 }%
1000 }{%
1001 \false@sw
1002 }%
1003 }{%
1004 \false@sw
1005 }%
1006 }{%
1007 \false@sw
1008 }%
1009 }{%
1010 \false@sw
1011 }%
1012 }{%
1013 \false@sw

```

```

1014     }%
1015   }{%
1016     \false@sw
1017   }%
1018 }{%
1019   \false@sw
1020 }%
1021 }{%
1022   \false@sw
1023 }%
1024 }{%
1025   \false@sw
1026 }%
1027 }{%
1028   \false@sw
1029 }%
1030 }{%
1031   \false@sw
1032 }%
1033 }{%
1034   \false@sw
1035 }{%
1036 \class@info{Patching array package.}%
1037 }{%
1038 \class@info{Unrecognized array package. Please update this document class! (Proceeding with f
1039 }%
1040 \let\@array    \@array@array@new
1041 \let\@@array   \@array % Così fan tutti
1042 \let\@tabular \@tabular@array@new
1043 \let\@tabarray \@tabarray@array@new
1044 \let\@array    \@array@array@new
1045 \let\endarray  \endarray@array@new
1046 \let\endtabular\endtabular@array@new
1047 \let\@mkpream  \@mkpream@array@new
1048 \let\@classx   \@classx@array@new
1049 \let\@arrayacol\@arrayacol@ltx
1050 \let\@tabacol  \@tabacol@ltx
1051 \let\insert@column\insert@column@array@new
1052 \expandafter\let\csname endtabular*\endcsname\endtabular % Così fan tutti
1053 \let\@arraycr  \@arraycr@new
1054 \let\@xarraycr \@xarraycr@new
1055 \let\@xargarraycr\@xargarraycr@new
1056 \let\@yargarraycr\@yargarraycr@new
1057 }%

```

`\@array@sw` The Boolean `\@array@sw` must be different depending on whether the array package is loaded.

```

1058 \def\@array@sw@LaTeX{\@ifx{\@tabularcr}}%
1059 \def\@array@sw@array{\@ifx{\dollarbegin\begingroup}}%

```

`\@tabular` We provide the old versions of `\@tabular` along with the respective new versions. The change here is to avoid committing to LR mode. That will be done later (as late as possible, naturally).

```

1060 \def\@tabular@LaTeX{%
1061 \leavevmode
1062 \hbox\bgroup$%
1063 \let\@acol\@tabacol
1064 \let\@classz\@tabclassz
1065 \let\@classiv\@tabclassiv
1066 \let\@tabularcr
1067 \@tabarray
1068 }%
1069 \def\@tabular@ltx{%
1070 \let\@acoll\@tabacoll
1071 \let\@acolr\@tabacolr
1072 \let\@acol\@tabacol
1073 \let\@classz\@tabclassz
1074 \let\@classiv\@tabclassiv
1075 \let\@tabularcr
1076 \@tabarray
1077 }%
1078 \def\@tabular@array{%
1079 \leavevmode
1080 \hbox\bgroup$%
1081 \col@sep\tabcolsep
1082 \let\d@llarbegin\beginngroup
1083 \let\d@llarend\endngroup
1084 \@tabarray
1085 }%
1086 \def\@tabular@array@new{%
1087 \let\@acoll\@tabacoll
1088 \let\@acolr\@tabacolr
1089 \let\@acol\@tabacol
1090 \let\col@sep\@undefined
1091 \let\d@llarbegin\beginngroup
1092 \let\d@llarend\endngroup
1093 \@tabarray
1094 }%

```

`\@tabarray` Here we provide old and new versions of the `\@tabarray` procedure. The change here is to parametrize the default vertical alignment, which is 'c' in standard L^AT_EX. Under some circumstances, we want to change this to, say, 'v'.

FIXME: must decouple `array` and `tabular`.

```

1095 \def\@tabarray@LaTeX{%
1096 \m@th\@ifnextchar[\@array{\@array[c]}%
1097 }%
1098 \def\@tabarray@ltx{%
1099 \m@th\@ifnextchar[\@array{\expandafter\@array\expandafter[\@array@default]}%
1100 }%

```

```

1101 \def\@tabarray@array{%
1102   \@ifnextchar[{\@array}{\@array[c]}%
1103 }%
1104 \def\@tabarray@array@new{%
1105   \@ifnextchar[{\@array}{\expandafter\@array\expandafter[\array@default]}%
1106 }%

```

`\@tabularcr` We provide for the `\@` command within `tabular` to provide control over page breaking, just the same as that of `eqnarray`.

`\@tabularcr` The count register `\intertabularlinepenalty` is similar to `\interdisplaylinepenalty`: it is the penalty associated with each row of a `tabular`. When it is set to `\@M`, the `tabular` will cleave together.

`\@xargarraycr` The count register `\@tbpen` is similar to `\@eqpen`: it memorizes the penalty to use after the current `tabular` row. If the `\@` command is in its star form, then `\@xarraycr` `\@eqpen` is set to `\@M`.

We append code to `\samepage` so that a `tabular` within its scope will cleave together.

We keep the standard definition of `\@tabularcr` in `\@tabularcr@LaTeX` for reference, and provide a new definition that works like `\@eqnocr`: it sets `\@tbpen` to `\@M` if the star was given.

We also provide new versions of `\@xtabularcr`, `\@xargarraycr`, and `\@yargarraycr`, all of which invoke `\@tbpen`.

The `\switch@tabular` procedure switches in the new definitions.

```

1107 \newcount\intertabularlinepenalty
1108 \intertabularlinepenalty=100
1109 \newcount\@tbpen
1110 \apdef\samepage{\intertabularlinepenalty\@M}%
1111 \def\@tabularcr@LaTeX{\ifnum 0='}\fi \ifstar \@xtabularcr \@xtabularcr}%
1112 \def\@tabularcr@ltx{\ifnum 0='}\fi \ifstar {\global \@tbpen \@M \@xtabularcr }{\global \@tbpen \@xtabularcr}%
1113 \def\@xtabularcr@LaTeX{\ifnextchar [\@argtabularcr {\ifnum 0='{\fi }\cr }}%
1114 \def\@xtabularcr@ltx{\ifnextchar [\@argtabularcr {\ifnum 0='{\fi }\cr \noalign {\penalty \@tbpen}}%
1115 \def\@xargarraycr@LaTeX#1{\@tempdima #1\advance \@tempdima \dp \@arstrutbox \vrule \@height \z@ \width \@width}%
1116 \def\@xargarraycr@ltx#1{\@tempdima #1\advance \@tempdima \dp \@arstrutbox \vrule \@height \z@ \width \@width}%
1117 \def\@yargarraycr@LaTeX#1{\cr \noalign {\vskip #1}}%
1118 \def\@yargarraycr@ltx#1{\cr \noalign {\penalty \@tbpen \vskip #1}}%

```

If the `array` package has been loaded, we must alter the meanings of `\@arraycr`, `\@xarraycr`, `\@xargarraycr`, and `\@yargarraycr`. In this case, it is `\switch@array` that switches in the new definitions.

```

1119 \def\@arraycr@array{%
1120   \relax
1121   \iffalse{\fi\ifnum 0='}\fi
1122   \@ifstar \@xarraycr \@xarraycr
1123 }%
1124 \def\@arraycr@new{%
1125   \relax
1126   \iffalse{\fi\ifnum 0='}\fi
1127   \@ifstar {\global \@tbpen \@M \@xarraycr }{\global \@tbpen \intertabularlinepenalty \@xarraycr}

```



```

1128 }%
1129 \def\xarraycr@array{%
1130 \@ifnextchar [%]
1131 \@argarraycr {\ifnum 0='{}\fi\cr}%
1132 }%
1133 \def\xarraycr@new{%
1134 \@ifnextchar [%]
1135 \@argarraycr {\ifnum 0='{}\fi\cr \noalign {\penalty \@tbpn }}%
1136 }%
1137 \def\xargarraycr@array#1{%
1138 \unskip
1139 \@tempdima #1\advance\@tempdima \dp\@arstrutbox
1140 \vrule \@depth\@tempdima \@width\z@
1141 \cr
1142 }%
1143 \def\xargarraycr@new#1{%
1144 \unskip
1145 \@tempdima #1\advance\@tempdima \dp\@arstrutbox
1146 \vrule \@depth\@tempdima \@width\z@
1147 \cr
1148 \noalign {\penalty \@tbpn }}%
1149 }%
1150 \def\yargarraycr@array#1{%
1151 \cr
1152 \noalign{\vskip #1}%
1153 }%
1154 \def\yargarraycr@new#1{%
1155 \cr
1156 \noalign{\penalty \@tbpn \vskip #1}%
1157 }%

```

`\array` We provide old and new versions of the `\array` procedure for both L^AT_EX and the `array` package. The change here is to accomodate the new procedures that will be called for the array boundaries, even though at present they are not special. A thought: here is where matrices can be readily accomodated.

```

1158 \def\array@LaTeX{%
1159 \let\@acol\@arrayacol
1160 \let\@classz\@arrayclassz
1161 \let\@classiv\@arrayclassiv
1162 \let\@arraycr
1163 \let\@halignto\@empty
1164 \@tabarray
1165 }%
1166 \def\array@ltx{%
1167 \@ifmode{\}{\@badmath}%
1168 \let\@acoll\@arrayacol
1169 \let\@acolr\@arrayacol
1170 \let\@acol\@arrayacol
1171 \let\@classz\@arrayclassz
1172 \let\@classiv\@arrayclassiv

```

```

1173 \let\\@arraycr
1174 \let\@halignto\@empty
1175 \@tabarray
1176 }%
1177 \def\array@array{%
1178 \col@sep\arraycolsep
1179 \def\d@llarbegin{$}\let\d@llarend\d@llarbegin\gdef\@halignto{}}%
1180 \@tabarray
1181 }
1182 \def\array@array@new{%
1183 \@ifmmode{}{\@badmath$}%
1184 \let\@acoll\@arrayacol
1185 \let\@acolr\@arrayacol
1186 \let\@acol\@arrayacol
1187 \let\col@sep\@undefined
1188 \def\d@llarbegin{$}%
1189 \let\d@llarend\d@llarbegin
1190 \gdef\@halignto{}}%
1191 \@tabarray
1192 }%

```

`\@array` Here we provide old and new versions of `\@array`. The change here is to provide a convenient, flexible, and extensible mechanism for new vertical alignment options.

Instead of testing the optional argument with `\if`, we use a dispatcher based on `\csname`.

We also refrain from using `\ialign`, which would set the `\tabskip` to the wrong value.

Finally, the procedure to set the `\@arstrutbox` is broken out so that it can be patched.

```

1193 \def\@array@LaTeX[#1]#2{%
1194 \if #1\vtop \else \if#1b\vbox \else \vcenter \fi\fi
1195 \bgroup
1196 \setbox\@arstrutbox\hbox{%
1197 \vrule \@height\arraystretch\ht\strutbox
1198 \@depth\arraystretch \dp\strutbox
1199 \@width\z@}%
1200 \mkpream{#2}%
1201 \edef\@preamble{%
1202 \ialign \noexpand\@halignto
1203 \bgroup \@arstrut \@preamble \tabskip\z@skip \cr}%
1204 \let\@startpbox\@startpbox \let\@endpbox\@endpbox
1205 \let\tabularnewline\%
1206 \let\par\@empty
1207 \let\sharp##%
1208 \set@typeset@protect
1209 \lineskip\z@skip\baselineskip\z@skip
1210 \ifhmode \@preamerr\z@ \@@par\fi
1211 \@preamble
1212 }%

```

```

1213 \def\@array@ltx[#1]#2{%
1214 \@nameuse{\@array@align@#1}%
1215 \set@arstrutbox
1216 \mkpream{#2}%
1217 \prepdef\@preamble{%
1218 \tabskip\tabmid@skip
1219 \@arstrut
1220 }%
1221 \appdef\@preamble{%
1222 \tabskip\tabright@skip
1223 \cr
1224 \array@row@pre
1225 }%
1226 % \let\@startpbox\@startpbox
1227 % \let\@endpbox\@endpbox
1228 \let\tabularnewline\%
1229 \let\par\@empty
1230 \let\@sharp##%
1231 \set@typeset@protect
1232 \lineskip\z@skip\baselineskip\z@skip
1233 \tabskip\tableft@skip\relax
1234 \ifhmode \@preamerr\z@ \@@par\fi
1235 \everycr{}%
1236 \expandafter\halign\expandafter\@halignto\expandafter\bgroup\@preamble
1237 }%
1238 %
1239 \def\set@arstrutbox{%
1240 \setbox\@arstrutbox\hbox{%
1241 \vrule \@height\arraystretch\ht\strutbox
1242 \@depth\arraystretch\dp\strutbox
1243 \@width\z@
1244 }%
1245 }%

```

\@array@array

```

1246 \def\@array@array[#1]#2{%
1247 \@tempdima \ht \strutbox
1248 \advance \@tempdima by\extrarowheight
1249 \setbox \@arstrutbox \hbox{\vrule
1250 \@height \arraystretch \@tempdima
1251 \@depth \arraystretch \dp \strutbox
1252 \@width \z@}%
1253 \begingroup
1254 \mkpream{#2}%
1255 \xdef\@preamble{\noexpand \ialign \@halignto
1256 \bgroup \@arstrut \@preamble
1257 \tabskip \z@ \cr}%
1258 \endgroup
1259 \@arrayleft
1260 \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi \fi

```

```

1261 \bgroup
1262 \let \@sharp ##\let \protect \relax
1263 \lineskip \z@
1264 \baselineskip \z@
1265 \m@th
1266 \let\\\@arraycr \let\tabularnewline\\\let\par\@empty \@preamble
1267 }%
1268 \def\@array@array@new[#1]#2{%
1269 \@tempdima\ht\strutbox
1270 \advance\@tempdima by\extrarowheight
1271 \setbox\@arstrutbox\hbox{%
1272 \vrule \@height\arraystretch\@tempdima
1273 \@depth \arraystretch\dp\strutbox
1274 \@width \z@
1275 }%
1276 \begingroup
1277 \@mkpream{#2}%
1278 \xdef\@preamble{\@preamble}%
1279 \endgroup
1280 \prepdef\@preamble{%
1281 \tabskip\tabmid@skip
1282 \@arstrut
1283 }%
1284 \appdef\@preamble{%
1285 \tabskip\tabright@skip
1286 \cr
1287 \array@row@pre
1288 }%
1289 \@arrayleft
1290 \@nameuse{\@array@align@#1}%
1291 \m@th
1292 \let\\\@arraycr
1293 \let\tabularnewline\%
1294 \let\par\@empty
1295 \let\@sharp##%
1296 \set@typeset@protect
1297 \lineskip\z@\baselineskip\z@
1298 \tabskip\tableft@skip
1299 \everycr{}%
1300 \expandafter\halign\expandafter\@halignto\expandafter\bgroup\@preamble
1301 }%

```

`\endarray` Here we provide old and new versions of `\endarray`. The change here is to use a single procedure to close out any array-like structure, namely `\endarray@ltx`. It merely closes out the `\halign`.

```

1302 \def\endarray@LaTeX{%
1303 \crr\egroup\egroup
1304 }%
1305 \def\endarray@ltx{%
1306 \crr\array@row@pst\egroup\egroup

```

```

1307 }%
1308 \def\endarray@array{%
1309 \crr\egroup \egroup \@arrayright \gdef\@preamble{}%
1310 }%
1311 \def\endarray@array@new{%
1312 \crr\array@row@pst\egroup\egroup % Same as \endarray@ltx
1313 \@arrayright
1314 \global\let\@preamble\@empty
1315 }%

\endtabular
1316 \def\endtabular@LaTeX{%
1317 \crr\egroup\egroup $\egroup
1318 }%
1319 \def\endtabular@ltx{%
1320 \endarray
1321 }%
1322 \def\endtabular@array{%
1323 \endarray $\egroup
1324 }%
1325 \def\endtabular@array@new{%
1326 \endarray
1327 }%

endtabular* Here we provide a proper definition for the star-form of \end{endtabular}. It is
one of the enduring curiosities that the LATEX kernel continues to use dangerously
and inappropriately “optimized” definitions for such commands.
1328 \@namedef{endtabular*}{\endtabular}%

\multicolumn
1329 \long\def\multicolumn@LaTeX#1#2#3{%
1330 \multispan{#1}\begingroup
1331 \mkpream{#2}%
1332 \def\@sharp{#3}\set@typeset@protect
1333 \let\@startpbox\@startpbox\let\@endpbox\@endpbox
1334 \arstrut \preamble\hbox{ }\endgroup\ignorespaces
1335 }%
1336 \long\def\multicolumn@ltx#1#2#3{%
1337 \multispan{#1}%
1338 \begingroup
1339 \mkpream{#2}%
1340 \def\@sharp{#3}%
1341 \set@typeset@protect
1342 %\let\@startpbox\@startpbox\let\@endpbox\@endpbox
1343 \arstrut
1344 \preamble
1345 \hbox{ }%
1346 \endgroup
1347 \ignorespaces
1348 }%

```

`\array@align@` Here are the various procedures for the vertical alignment options. The change
`\array@default` from standard L^AT_EX is that we do not go into math mode in every case: only
when required by `\vcenter`. Also, we use `\aftergroup` to close out the boxes
and modes we have started. It requires only that each procedure issue exactly one
unmatched `\bgroup`.

We establish here the default vertical alignment.

```

1349 \def\array@align@t{\leavevmode\vtop\bgroup}%
1350 \def\array@align@b{\leavevmode\vbox\bgroup}%
1351 \def\array@align@c{\leavevmode\@ifmmode{\vcenter\bgroup}{$\vcenter\bgroup\aftergroup$\aftergro
1352 \def\array@align@v{%
1353   \@ifmmode{%
1354     \@badmath
1355     \vcenter\bgroup
1356   }{%
1357     \@ifinner{%
1358       $\vcenter\bgroup\aftergroup$
1359     }{%
1360       \@@par\bgroup
1361     }%
1362   }%
1363 }%
1364 \def\array@default{c}%

```

`\array@row@pre` The procedure `\array@row@rst` reestablishes a default context for an alignment,
`\array@row@pst` so that they can be nested. Any environment or procedure that alters the way
`\array@row@rst` alignments are formatted must patch this procedure to restore from that alteration.
To start things off, we equate `\array@align@v` to `\array@align@c`, because it
does not make sense to do the former in any context other than the MVL or in a
list that will be unboxed onto the MVL.

```

1365 \def\array@row@rst{%
1366   \let\array@align@v\array@align@c
1367 }%
1368 \def\array@row@pre{%
1369 \def\array@row@pst{%

```

`\toprule` Default definitions for `\toprule`, `\colrule`, `\botrule`

```

\colrule 1370 \newcommand\toprule{\tab@rule{\column@font}{\column@fil}{\frstrut}}%
\botrule 1371 \newcommand\colrule{\unskip\lrstrut\\\tab@rule{\body@font}{\frstrut}}%
1372 \newcommand\botrule{\unskip\lrstrut\\\noalign{\hline@rule}{}}%

```

`\hline`

```

1373 \def\hline@LaTeX{%
1374   \noalign{\ifnum0='}\fi\hrule \@height \arrayrulewidth \futurelet
1375     \reserved@a\@xhline
1376 }%
1377 \def\hline@ltx{%
1378   \noalign{%
1379     \ifnum0='}\fi

```

```

1380 \hline@rule
1381 \futurelet\reserved@a\@xhline
1382 % \noalign ended in \@xhline
1383 }%
1384 \def\@xhline@unneeded{%
1385 \say\reserved@a
1386 \ifx\reserved@a\hline
1387 \vskip\doublerulesep
1388 \vskip-\arrayrulewidth
1389 \fi
1390 \ifnum0='{\fi}%
1391 }%
1392 \def\tab@rule#1#2#3{%
1393 \crr
1394 \noalign{%
1395 \hline@rule
1396 \gdef\@arstrut@hook{%
1397 \global\let\@arstrut@hook\@empty
1398 #3%
1399 }%
1400 \gdef\cell@font{#1}%
1401 \gdef\cell@fil{#2}%
1402 }%
1403 }%
1404 \def\column@font{}%
1405 \def\column@fil{}%
1406 \def\body@font{}%
1407 \def\cell@font{}%
1408 \def\frstrut{}%
1409 \def\lrstrut{}%

```

`\@arstrut@hline` The procedure `\@arstrut@hline` is substantially the same as `\@arstrut`, except
`\@arstrut@org` the strut copied in is `\@arstrutbox@hline` instead of `\@arstrutbox`.
`\@arstrut@hook` The procedure `\@arstrut@hook` is redefined in `\tab@rule!`
`\@arstrutbox@hline` The register `\@arstrutbox@hline`.
`\set@arstrutbox` We append to `\set@arstrutbox` the code necessary to set a strut following an
`\hline@rule` `\hline`.

The procedure `\hline@rule` lays down a rule, and changes the meaning of `\@arstrut` so that the next line will be correctly strutted.

The `\@arstrut@hline@cinc` is a kloutch, a magic number.

```

1410 \def\@arstrut@hline{%
1411 \relax
1412 \@ifmmode{\copy}{\unhcopy}\@arstrutbox@hline
1413 \@arstrut@hook
1414 }%
1415 %
1416 \let\@arstrut@org\@arstrut
1417 \def\@arstrut@hook{%
1418 \global\let\@arstrut\@arstrut@org

```

```

1419 }%
1420 %
1421 \newbox\@arstrutbox@hline
1422 \appdef\set@arstrutbox{%
1423   \setbox\@arstrutbox@hline\hbox{%
1424     \setbox\z@\hbox{\$0^{0}_{-}{}}%
1425     \dimen@ht\z@\advance\dimen@\@arstrut@hline@clnc
1426     \@ifdim{\dimen@<\arraystretch\ht\strutbox}{\dimen@=\arraystretch\ht\strutbox}{}%
1427     \vrule \@height\dimen@
1428             \@depth\arraystretch \dp\strutbox
1429             \@width\z@
1430 }%
1431 }%
1432 %
1433 \def\hline@rule{%
1434   \hrule \@height \arrayrulewidth
1435   \global\let\@arstrut\@arstrut@hline
1436 }%
1437 \def\@arstrut@hline@clnc{2\p@}% % Klotch: magic number

\tableleft@skip
1438 \def\tableleft@skip{\z@skip}%
1439 \def\tabmid@skip{\z@skip}\@flushglue
1440 \def\tabright@skip{\z@skip}%
1441 \def\tableleftsep{\tabcolsep}%
1442 \def\tabmidsep{\tabcolsep}%
1443 \def\tabrightsep{\tabcolsep}%
1444 \def\cell@fil{%
1445 \def\pbox@hook{}%

\@arstrut
1446 \appdef\@arstrut{\@arstrut@hook}%
1447 \let\@arstrut@hook\@empty
1448 \def\@addtopreamble{\appdef\@preamble}%

\@mkpream
1449 \def\@mkpream@LaTeX#1{%
1450   \@firstampttrue\@lastchclass6
1451   \let\@preamble\@empty
1452   \let\protect\@unexpandable@protect
1453   \let\@sharp\relax
1454   \let\@startpbox\relax\let\@endpbox\relax
1455   \@expast{#1}%
1456   \expandafter\@tfor \expandafter
1457     \@nextchar \expandafter:\expandafter=\reserved@a\do
1458     {\@testpach\@nextchar
1459     \ifcase \@chclass \@classz \or \@classi \or \@classii \or \@classiii
1460     \or \@classiv \or \@classv \fi\@lastchclass\@chclass}%
1461   \ifcase \@lastchclass \@acol

```



```

1462     \or \or \@preamerr \@ne\or \@preamerr \tw@\or \or \@acol \fi
1463 }%
1464 \def\@mkpream@ltx#1{%
1465   \@firstamptrue
1466   \@lastchclass6
1467   \let\@preamble\@empty
1468   \let\protect\@unexpandable@protect
1469   \let\@sharp\relax
1470 %\let\@startpbox\relax\let\@endpbox\relax
1471   \@expast{#1}%
1472   \expandafter\@tfor\expandafter\@nextchar\expandafter:\expandafter=\reserved@a
1473   \do{%
1474     \expandafter\@testpach\expandafter{\@nextchar}%
1475     \ifcase\@chclass
1476       \@classz
1477     \or
1478       \@classi
1479     \or
1480       \@classii
1481     \or
1482       \@classiii
1483     \or
1484       \@classiv
1485     \or
1486       \@classv
1487     \fi
1488     \@lastchclass\@chclass
1489   }%
1490   \ifcase\@lastchclass
1491     \@acolr % right-hand column
1492   \or
1493   \or
1494     \@preamerr\@ne
1495   \or
1496     \@preamerr\tw@
1497   \or
1498   \or
1499     \@acolr % right-hand column
1500   \fi
1501 }%

```

\insert@column

```

1502 \def\insert@column@array{%
1503   \the@toks \the \@tempcnta
1504   \ignorespaces \@sharp \unskip
1505   \the@toks \the \count@ \relax
1506 }%
1507 \def\insert@column@array@new{%
1508   \the@toks\the\@tempcnta
1509   \array@row@rst\cell@font

```

```

1510 \ignorespaces\@sharp\unskip
1511 \the@toks\the\count@
1512 \relax
1513 }%

```

`\@mkpream@relax` The procedure `\@mkpream@relax` participates in a strange and wonderful method of binding the alignment procedure—but only certain parts thereof.

Here is how it works: in \LaTeX , the `array` package, and in the `longtable` package alike, there is a need to create an alignment preamble (using `\@mkpream`) for use by the upcoming `\halign`. Then, in both `array` and `longtable`, \TeX 's `\edef` is used to ‘compile in place’ that alignment preamble.

In the case of `array`, the operation is done in order to pre-expand the use of `*`, in `longtable`, it is to set the widths of the columns.

Now, during this `\edef`, certain control sequence names must *not* be expanded, and those are robustified by `\@mkpream@relax`.

```

1514 \def\@mkpream@relax{%
1515 \let\tableleftsep\relax
1516 \let\tabmidsep\relax
1517 \let\tabrightsep\relax
1518 \let\array@row@rst\relax
1519 \let\cell@font\relax
1520 \let\@startpbox\relax
1521 }%

```

`\@mkpream`

```

1522 \def\@mkpream@array#1{%
1523 \gdef\@preamble{ }\@lastchclass 4 \@firstampttrue
1524 \let\@sharp\relax \let\@startpbox\relax \let\@endpbox\relax
1525 \@temptokena{#1}\@tempswatrue
1526 \@whilesw\if@tempswa\fi{\@tempswafalse\the\NC@list}%
1527 \count@m@one
1528 \let\the@toks\relax
1529 \prepnext@tok
1530 \expandafter \@tfor \expandafter \@nextchar
1531 \expandafter :\expandafter =\the\@temptokena \do
1532 {\@testpach
1533 \ifcase \@chclass \@classz \or \@classi \or \@classii
1534 \or \save@decl \or \or \@classv \or \@classvi
1535 \or \@classvii \or \@classviii
1536 \or \@classx
1537 \or \@classx \fi
1538 \@lastchclass\@chclass}%
1539 \ifcase\@lastchclass
1540 \@acol \or
1541 \or
1542 \@acol \or
1543 \@preamerr \thr@@ \or
1544 \@preamerr \tw@ \@addtopreamble\@sharp \or
1545 \or

```

```

1546 \else \@preamerr \@ne \fi
1547 \def\the@toks{\the\toks}%
1548 }%
1549 \def\@mkpream@array@new#1{%
1550 \gdef\@preamble{%
1551 \@lastchclass\@f@ur
1552 \@firstamptrue
1553 \let\@sharp\relax
1554 \@mkpream@relax
1555 %\let\@startpbox\relax\let\@endpbox\relax
1556 \@temptokena{#1}\@tempswatrue
1557 \@whilesw@if@tempswa\fi{\@tempswafalse\the\NC@list}%
1558 \count@\m@ne
1559 \let\the@toks\relax
1560 \prepnext@tok
1561 \expandafter\@tfor\expandafter\@nextchar\expandafter:\expandafter=\the\@temptokena
1562 \do{%
1563 \@testpach
1564 \ifcase\@chclass
1565 \@classz
1566 \or
1567 \@classi
1568 \or
1569 \@classii
1570 \or
1571 \save@decl
1572 \or
1573 \or
1574 \@classv
1575 \or
1576 \@classvi
1577 \or
1578 \@classvii
1579 \or
1580 \@classviii
1581 \or
1582 \@classx
1583 \or
1584 \@classx
1585 \fi
1586 \@lastchclass\@chclass
1587 }%
1588 \ifcase\@lastchclass
1589 \@acolr % right-hand column
1590 \or
1591 \or
1592 \@acolr % right-hand column
1593 \or
1594 \@preamerr\thr@@
1595 \or

```

```

1596 \@preamerr\tw@\@addtopreamble\@sharp
1597 \or
1598 \or
1599 \else
1600 \@preamerr\@ne
1601 \fi
1602 \def\the@toks{\the\toks}%
1603 }%

\@addamp
1604 \def\@addamp@LaTeX{%
1605 \if@firstamp\@firstampfalse\else\edef\@preamble{\@preamble &}\fi
1606 }%
1607 \def\@addamp@ltx{%
1608 \if@firstamp\@firstampfalse\else\@addtopreamble{&}\fi
1609 }%

\@arrayacol
1610 \def\@arrayacol@LaTeX{%
1611 \edef\@preamble{\@preamble \hskip \arraycolsep}%
1612 }%
1613 \def\@arrayacol@ltx{%
1614 \@addtopreamble{\hskip\arraycolsep}%
1615 }%

\@tabacol
1616 \def\@tabacol1{%
1617 \@addtopreamble{\hskip\tableftsep\relax}%
1618 }%
1619 \def\@tabacol@LaTeX{%
1620 \edef\@preamble{\@preamble \hskip \tabcolsep}%
1621 }%
1622 \def\@tabacol@ltx{%
1623 \@addtopreamble{\hskip\abmidsep\relax}%
1624 }%
1625 \def\@tabacolr{%
1626 \@addtopreamble{\hskip\abrightsep\relax}%
1627 }%

\@arrayclassz
1628 \def\@arrayclassz@LaTeX{%
1629 \ifcase \@lastchclass \@acolampacol \or \@ampacol \or
1630 \or \or \@addamp \or
1631 \@acolampacol \or \@firstampfalse \@acol \fi
1632 \edef\@preamble{\@preamble
1633 \ifcase \@chnum
1634 \hfil$\relax\@sharp$\hfil \or $\relax\@sharp$\hfil
1635 \or \hfil$\relax\@sharp$\fi}%
1636 }%

```

```

1637 \def\@arrayclassz@ltx{%
1638 \ifcase\@lastchclass
1639 \@acolampacol
1640 \or
1641 \@ampacol
1642 \or
1643 \or
1644 \or
1645 \@addamp
1646 \or
1647 \@acolampacol
1648 \or
1649 \@firstampfalse\@acoll
1650 \fi
1651 \ifcase\@chnum
1652 \@addtopreamble{%
1653 \hfil\array@row@rst$\relax\@sharp$\hfil
1654 }%
1655 \or
1656 \@addtopreamble{%
1657 \array@row@rst$\relax\@sharp$\hfil
1658 }%
1659 \or
1660 \@addtopreamble{%
1661 \hfil\array@row@rst$\relax\@sharp$%
1662 }%
1663 \fi
1664 }%

```

\@tabclassz

```

1665 \def\@tabclassz@LaTeX{%
1666 \ifcase\@lastchclass
1667 \@acolampacol
1668 \or
1669 \@ampacol
1670 \or
1671 \or
1672 \or
1673 \@addamp
1674 \or
1675 \@acolampacol
1676 \or
1677 \@firstampfalse\@acol
1678 \fi
1679 \edef\@preamble{%
1680 \@preamble{%
1681 \ifcase\@chnum
1682 \hfil\ignorespaces\@sharp\unskip\hfil
1683 \or
1684 \hskip1sp\ignorespaces\@sharp\unskip\hfil

```

```

1685     \or
1686     \hfil\hskip1sp\ignorespaces\@sharp\unskip
1687     \fi}}%
1688 }%
1689 \def\@tabclassz@ltx{%
1690 \ifcase\@lastchclass
1691 \@acolampacol
1692 \or
1693 \@ampacol
1694 \or
1695 \or
1696 \or
1697 \@addamp
1698 \or
1699 \@acolampacol
1700 \or
1701 \@firstampfalse\@acoll
1702 \fi
1703 \ifcase\@chnum
1704 \@addtopreamble{%
1705   {\hfil\array@row@rst\cell@font\ignorespaces\@sharp\unskip\hfil}%
1706   }%
1707 \or
1708 \@addtopreamble{%
1709   {\cell@fil\hskip1sp\array@row@rst\cell@font\ignorespaces\@sharp\unskip\hfil}%
1710   }%
1711 \or
1712 \@addtopreamble{%
1713   {\hfil\hskip1sp\array@row@rst\cell@font\ignorespaces\@sharp\unskip\cell@fil}%
1714   }%
1715 \fi
1716 }%

\@tabclassiv
1717 \def\@tabclassiv@LaTeX{%
1718 \@addtopreamble\@nextchar
1719 }%
1720 \def\@tabclassiv@ltx{%
1721 \expandafter\@addtopreamble\expandafter{\@nextchar}%
1722 }%

\@arrayclassiv
1723 \def\@arrayclassiv@LaTeX{%
1724 \@addtopreamble{\@nextchar$}%
1725 }%
1726 \def\@arrayclassiv@ltx{%
1727 \expandafter\@addtopreamble\expandafter{\expandafter$\@nextchar$}%
1728 }%

\@classv

```

```

1729 \def\@classv@LaTeX{%
1730 \@addtopreamble{\@startpbox{\@nextchar}\ignorespaces
1731 \@sharp\@endpbox}%
1732 }%
1733 \def\@classv@ltx{%
1734 \expandafter\@addtopreamble
1735 \expandafter{%
1736 \expandafter \@startpbox
1737 \expandafter {\@nextchar}%
1738 \pbox@hook\array@row@rst\cell@font\ignorespaces\@sharp\@endpbox
1739 }%
1740 }%

```

\@classx

```

1741 \def\@classx@array{%
1742 \ifcase \@lastchclass
1743 \@acolampacol \or
1744 \@addamp \@acol \or
1745 \@acolampacol \or
1746 \or
1747 \@acol \@firstampfalse \or
1748 \@addamp
1749 \fi
1750 }%
1751 \def\@classx@array@new{%
1752 \ifcase \@lastchclass
1753 \@acolampacol
1754 \or
1755 \@addamp \@acol
1756 \or
1757 \@acolampacol
1758 \or
1759 \or
1760 \@firstampfalse\@acoll
1761 \or
1762 \@addamp
1763 \fi
1764 }%

```

6.14 Repair other broken parts of L^AT_EX

\@xbitor Expansion part has extraneous space token. Removed.

```

1765 \def\@xbitor@LaTeX #1{\@tempcntb \count#1
1766 \ifnum \@tempcnta =\z@
1767 \else
1768 \divide\@tempcntb\@tempcnta
1769 \ifodd\@tempcntb \@testtrue\fi
1770 \fi}%
1771 \def\@xbitor@ltx#1{%

```

```

1772 \@tempcntb\count#1%
1773 \@ifnum{\@tempcnta=\z@}{-}{%
1774   \divide\@tempcntb\@tempcnta
1775   \@ifodd\@tempcntb{\@testtrue}{-}{%
1776   }%
1777 }%
1778 \@ifx{\@xbitor\@xbitor@LaTeX}{%
1779   \class@info{Repairing broken LaTeX \string\@xbitor}%
1780 }%
1781   \class@info{Unrecognized LaTeX \string\@xbitor. Please update this document class! (Proceedin
1782 }%
1783 \let\@xbitor\@xbitor@ltx

```

6.15 Syntax

`\gobble@opt@one` The `\gobble@opt@one` command eats up an optional argument and one required argument.

```
1784 \newcommand*\gobble@opt@one [2] [] {}%
```

6.16 Auto-indented Contents

Facility to automatically determine the proper indentation of the TOC entries.

Note on `hyperref` compatibility: We must respect that `\contentslinenow` has a 4th argument. So, instead of trying to override the meaning of `\contentsline`, we use the aux file to remember max values from one run to the next.

In this respect, this package retains compatibility with `hyperref`.

`\starttoc` Install hooks at beginning and end of the TOC processing.

```

1785 \def\starttoc#1{%
1786   \begingroup
1787   \toc@pre
1788   \makeatletter
1789   \input{\jobname.#1}%
1790   \if@filesw
1791     \expandafter\newwrite\csname tf@#1\endcsname
1792     \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1793   \fi
1794   \@nobreakfalse
1795   \toc@post
1796   \endgroup
1797 }%
1798 \def\toc@pre{}%
1799 \def\toc@post{}%

```

`\toc@font` Interface for setting the formatting characteristics of this part of the TOC.

Note: `\toc@font` is the common font for all auto-sizing toc commands, although this, too, could become a dispatcher.

```

1800 \def\toc@font{}{\footnotesize\rmfamily}%
1801 \def\@dotsep{\z@}{5.5pt}%

```


`\l@section` Interface for determining which TOC elements are automatically indented.

All of the `\l@...` commands simply go through the bottleproc `\l@sections`. The calling convention is to pass the name of self and the name of parent. If you want to exclude any of these from the indentation scheme, simply leave the `\l@...` command undefined.

Note that the parent of “section” is nil, so we have to define a stub.

```
%\def\l@section{%
% \l@sections{}{section}% Implicit #3#4
}%
%\def\tocleft@{\z@}%
%\def\l@subsection{%
% \l@sections{section}{subsection}% Implicit #3#4
}%
%\def\l@subsubsection{%
% \l@sections{subsection}{subsubsection}% Implicit #3#4
}%
%\def\l@paragraph{%
% \l@sections{subsubsection}{paragraph}% Implicit #3#4
}%
%\def\l@subparagraph#1#2{%
% \l@sections{paragraph}{subparagraph}% Implicit #3#4
}%
%
```

Glom some `\dimen` registers.

```
1802 \let\tocdim@section \leftmargini
1803 \let\tocdim@subsection \leftmarginii
1804 \let\tocdim@subsubsection \leftmarginiii
1805 \let\tocdim@paragraph \leftmarginiv
1806 \let\tocdim@appendix \leftmarginv
1807 \let\tocdim@pagenum \leftmarginvi
```

`\toc@pre@auto` We patch `\@starttoc` to: 1) before TOC processing, initialize the max registers
`\toc@post@auto` and set the needed dimensions from the values stored in the auxiliary file, and 2)
after TOC processing, store out those max register values into the auxiliary file.

Note that the font is set here: all other TOC entries must override these font settings.

To activate this override of the standard \LaTeX processing, the substyle does:

`\let\toc@pre\toc@pre@auto` and `\let\toc@post\toc@post@auto`.

```
1808 \def\toc@pre@auto{%
1809 \toc@@font
1810 \@tempdima\z@
1811 \toc@setindent\@tempdima{section}%
1812 \toc@setindent\@tempdima{subsection}%
1813 \toc@setindent\@tempdima{subsubsection}%
1814 \toc@setindent\@tempdima{paragraph}%
1815 \toc@letdimen{appendix}%
1816 \toc@letdimen{pagenum}%
```

```

1817 }%
1818 \def\toc@post@auto{%
1819   \if@filesw
1820     \begingroup
1821       \toc@writedimen{section}%
1822       \toc@writedimen{subsection}%
1823       \toc@writedimen{subsubsection}%
1824       \toc@writedimen{paragraph}%
1825       \toc@writedimen{appendix}%
1826       \toc@writedimen{pagenum}%
1827     \endgroup
1828   \fi
1829 }%

```

`\toc@setindent`

```

1830 \def\toc@setindent#1#2{%
1831   \csname tocdim@#2\endcsname\tocdim@min\relax
1832   \@ifundefined{tocmax@#2}{\@namedef{tocmax@#2}{\z@}}{}%
1833   \advance#1\@nameuse{tocmax@#2}\relax
1834   \expandafter\edef\csname tocleft@#2\endcsname{\the#1}%
1835 }%

```

`\toc@letdimen`

```

1836 \def\toc@letdimen#1{%
1837   \csname tocdim@#1\endcsname\tocdim@min\relax
1838   \@ifundefined{tocmax@#1}{\@namedef{tocmax@#1}{\z@}}{}%
1839   \expandafter\let\csname tocleft@#1\expandafter\endcsname\csname tocmax@#1\endcsname
1840 }%

```

`\toc@writedimen`

```

1841 \def\toc@writedimen#1{%
1842   \immediate\write\@auxout{%
1843     \gdef\expandafter\string\csname tocmax@#1\endcsname{%
1844       \expandafter\the\csname tocdim@#1\endcsname
1845     }%
1846   }%
1847 }%

```

`\l@sections` The procedure for formatting the indented TOC entries. We use control sequence names such as `\tocmax@section` and `\tocleft@section`, the former being written to the auxiliary file and the latter only defined for the duration of the TOC processing.

Note that the assignment of `\box\z@` must endure over the invocation of #3.

```

1848 \def\l@sections#1#2#3#4{%
1849   % #1 - superior section
1850   % #2 - this section
1851   % #3 - content, including possible \numberline
1852   % #4 - page number
1853   \begingroup

```

```

1854 \everypar{}%
1855 \set@tocdim@pagenum{#4}%
1856 \global\@tempdima\csname tocdim@#2\endcsname
1857 \leftskip\csname tocleft@#2\endcsname\relax
1858 \dimen@\csname tocleft@#1\endcsname\relax
1859 \parindent-\leftskip\advance\parindent\dimen@
1860 \rightskip\tocleft@pagenum plus 1fil\relax
1861 \skip@\parfillskip\parfillskip\z@
1862 \let\numberline\numberline@@sections
1863 \@nameuse{1@f@#2}%
1864 \ignorespaces#3\unskip\nobreak\hskip\skip@
1865 \hb@xt@\rightskip{\hfil\unhbox\@tempboxa}\hskip-\rightskip\hskip\z@skip
1866 \par
1867 \expandafter\aftergroup\csname tocdim@#2\endcsname\expandafter
1868 \endgroup\the\@tempdima\relax
1869 }%
1870 \def\set@tocdim@pagenum#1{%
1871 \setbox\@tempboxa\hbox{\ignorespaces#1}%
1872 \@ifdim{\tocdim@pagenum<\wd\z@}{\global\tocdim@pagenum\wd\z@}{}%
1873 }%

```

`\numberline@@sections` The bottleproc for all `\numberline` processing in indented TOC entries. The first argument is self.

We use `\@tempdima` to pass a value around (via global assignment) because `\numberline` executes inside a group if the `hyperref` package is loaded. Would that it were not so!

```

1874 \def\numberline@@sections#1{%
1875 \leavevmode\hb@xt@-\parindent{%
1876 \hfil
1877 \@ifempty{#1}{}%
1878 \setbox\z@\hbox{#1.\kern\@dotsep}%
1879 \@ifdim{\@tempdima<\wd\z@}{\global\@tempdima\wd\z@}{}%
1880 \unhbox\z@
1881 }%
1882 }%
1883 \ignorespaces
1884 }%
1885 \def\tocdim@min{\z@}%

```

6.17 Lists

`\list` Using `\parshape` to implement lists was always suspect (can you get behind `\parshape@ne`?) and we now see that it was a mistake all along. Why? Because `\parshape`, like `\hangindent`, achieves its effect via “shifting” the `\hboxes` in a paragraph instead of using `\leftskip` and `\parindent`, which is robust during column balancing.

We introduce the alternative method with a hook into the \LaTeX kernel procedure `\list`, which is the implementation of all lists.

```

1886 \def\list#1#2{%
1887   \ifnum \@listdepth >5\relax
1888     \@toodeep
1889   \else
1890     \global\advance\@listdepth\@ne
1891   \fi
1892   \rightmargin\z@
1893   \listparindent\z@
1894   \itemindent\z@
1895   \csname @list\romannumeral\the\@listdepth\endcsname
1896   \def\@itemlabel{#1}%
1897   \let\makelabel\@mklab
1898   \@nmbbrlistfalse
1899   #2\relax
1900   \@trivlist
1901   \parskip\parsep
1902   \set@listindent
1903   \ignorespaces
1904 }%
1905 \def\set@listindent@parshape{%
1906   \parindent\listparindent
1907   \advance\@totalleftmargin\leftmargin
1908   \advance\linewidth-\rightmargin
1909   \advance\linewidth-\leftmargin
1910   \parshape\@ne\@totalleftmargin\linewidth
1911 }%
1912 \def\set@listindent@{%
1913   \parindent\listparindent
1914   \advance\@totalleftmargin\leftmargin
1915   \advance\rightskip\rightmargin
1916   \advance\leftskip\@totalleftmargin
1917 }%
1918 \let\set@listindent\set@listindent@parshape

```

6.18 End of the ltxutil DOCSTRIP module

Here ends the module.

```
1919 %</ltxutil-krn>
```

Here ends the programmer's documentation.

Index

Symbols	
<code>\%</code>	711
<code>\(</code>	527
<code>\)</code>	528
<code>.dtx</code>	1, 4–6
<code>.ins</code>	4
<code>.tfm</code>	1
<code>.vf</code>	1
<code>.vpl</code>	1
<code>\@array</code>	1041, 1102, 1105
<code>\@end</code>	340
<code>\@endpbox</code>	1204, 1227, 1333, 1342
<code>\@par</code>	661, 783, 825, 1210, 1234, 1360
<code>\@startpbox</code>	1204, 1226, 1333, 1342
<code>\@Esphack</code>	699, 701
<code>\@M</code>	40
<code>\@MM</code>	500
<code>\@acol</code>	1063, 1072, 1089, 1159, 1170, 1186, 1461, 1462, 1540, 1542, 1631, 1677, 1744, 1747, 1755
<code>\@acolampacol</code>	1629, 1631, 1639, 1647, 1667, 1675, 1691, 1699, 1743, 1745, 1753, 1757
<code>\@acoll</code>	1070, 1087, 1168, 1184, 1649, 1701, 1760
<code>\@acolr</code>	1071, 1088, 1169, 1185, 1491, 1499, 1589, 1592
<code>\@addamp</code>	877, 967, 1604, 1630, 1645, 1673, 1697, 1744, 1748, 1755, 1762
<code>\@addamp@LaTeX</code>	877, 1604
<code>\@addamp@ltx</code>	967, 1607
<code>\@addtopreamble</code>	1448, 1544, 1596, 1608, 1614, 1617, 1623, 1626, 1652, 1656, 1660, 1704, 1708, 1712, 1718, 1721, 1724, 1727, 1730, 1734
<code>\@afterindentfalse</code>	743
<code>\@afterindenttrue</code>	741
<code>\@ampacol</code>	1629, 1641, 1669, 1693
<code>\@argarraycr</code>	1131, 1135
<code>\@argswap</code>	13
<code>\@argswap@val</code>	255
<code>\@argswap@val</code>	13
<code>\@argswap@val</code>	255
<code>\@argtabularcr</code>	1113, 1114
<code>\@array</code>	869, 959, 983, 1040, 1041, 1096, 1099, 1193
<code>\@array@LaTeX</code>	869, 1193
<code>\@array@align@</code>	1349
<code>\@array@align@b</code>	1350
<code>\@array@align@c</code>	46
<code>\@array@align@c</code>	1351, 1366
<code>\@array@align@t</code>	1349
<code>\@array@align@v</code>	46
<code>\@array@align@v</code>	1352, 1366
<code>\@array@array</code>	983, 1246
<code>\@array@array@new</code>	1040, 1268
<code>\@array@ltx</code>	959, 1213
<code>\@array@sw</code>	38
<code>\@array@sw</code>	868, 982, 1058
<code>\@array@sw@LaTeX</code>	982, 1058
<code>\@array@sw@array</code>	868, 1059
<code>\@arrayacol</code>	878, 968, 1049, 1159, 1168–1170, 1184–1186, 1610
<code>\@arrayacol@LaTeX</code>	878, 1610
<code>\@arrayacol@ltx</code>	968, 1049, 1613
<code>\@arrayclassiv</code>	882, 972, 1161, 1172, 1723
<code>\@arrayclassiv@LaTeX</code>	882, 1723
<code>\@arrayclassiv@ltx</code>	972, 1726
<code>\@arrayclassz</code>	880, 970, 1160, 1171, 1628
<code>\@arrayclassz@LaTeX</code>	880, 1628
<code>\@arrayclassz@ltx</code>	970, 1637
<code>\@arraycr</code>	40
<code>\@arraycr</code>	992, 1053, 1107, 1162, 1173, 1266, 1292
<code>\@arraycr@array</code>	992, 1119
<code>\@arraycr@new</code>	1053, 1124

<code>\@arrayleft</code>	1259, 1289	<code>\@classv@LaTeX</code>	884, 1729
<code>\@arrayright</code>	1309, 1313	<code>\@classv@ltx</code>	974, 1733
<code>\@arstrut</code>	47	<code>\@classvi</code>	1534, 1576
<code>\@arstrut</code>	1203, 1219, 1256, 1282, 1334, 1343, 1416, 1418, 1435, 1446	<code>\@classvii</code>	1535, 1578
<code>\@arstrut@hline</code>	47	<code>\@classviii</code>	1535, 1580
<code>\@arstrut@hline</code>	1410	<code>\@classx</code>	990, 1048, 1536, 1537, 1582, 1584, 1741
<code>\@arstrut@hline@cinc</code>	47	<code>\@classx@array</code>	990, 1741
<code>\@arstrut@hline@cinc</code>	1425, 1437	<code>\@classx@array@new</code>	1048, 1751
<code>\@arstrut@hook</code>	47	<code>\@classz</code>	1064, 1073, 1160, 1171, 1459, 1476, 1533, 1565
<code>\@arstrut@hook</code>	1396, 1397, 1410, 1446, 1447	<code>\@ctrerr</code>	735
<code>\@arstrut@org</code>	1410	<code>\@currentlabel</code>	409, 425, 503, 543
<code>\@arstrutbox</code>	42, 47	<code>\@currentlabelname</code>	32
<code>\@arstrutbox</code>	1115, 1116, 1139, 1145, 1196, 1240, 1249, 1271	<code>\@currentlabelname</code>	814
<code>\@arstrutbox@hline</code>	47	<code>\@dblarg</code>	751, 752
<code>\@arstrutbox@hline</code>	1410	<code>\@dblfloat</code>	650, 652, 671
<code>\@auxout</code>	525, 542, 1842	<code>\@dblfloat@LaTeX</code>	650, 671
<code>\@boole@def</code>	13	<code>\@defaultsubs</code>	360
<code>\@boole@def</code>	265, 277–280, 282–292	<code>\@depth</code>	1115, 1116, 1140, 1146, 1198, 1242, 1251, 1273, 1428
<code>\@boolean</code>	13	<code>\@dofilelist</code>	353
<code>\@boolean</code>	265	<code>\@eha</code>	161
<code>\@booleanfalse</code>	14	<code>\@enddocumenthook</code>	335
<code>\@booleanfalse</code>	275, 607, 608, 666	<code>\@endpbox</code>	1204, 1227, 1333, 1342, 1454, 1470, 1524, 1555, 1731, 1738
<code>\@booleantrue</code>	14	<code>\@eqcnt</code>	412, 417, 418, 420, 426, 439, 442, 445
<code>\@booleantrue</code>	275, 590, 638	<code>\@eqncr</code>	40
<code>\@bsphack</code>	541, 704	<code>\@eqncr</code>	414, 428
<code>\@caption</code>	25	<code>\@eqnset</code>	416, 438
<code>\@captive</code>	610	<code>\@eqnswtrue</code>	410, 426
<code>\@centering</code>	413, 416, 419, 438, 444, 457	<code>\@eqpen</code>	40
<code>\@chclass</code>	1459, 1460, 1475, 1488, 1533, 1538, 1564, 1586	<code>\@esphack</code>	545
<code>\@checkend</code>	336, 379, 391, 395	<code>\@expast</code>	1455, 1471
<code>\@chnum</code>	1633, 1651, 1681, 1703	<code>\@finalstrut</code>	508
<code>\@classi</code>	1459, 1478, 1533, 1567	<code>\@firststampfalse</code>	1605, 1608, 1631, 1649, 1677, 1701, 1747, 1760
<code>\@classii</code>	1459, 1480, 1533, 1569	<code>\@firststamptrue</code>	1450, 1465, 1523, 1552
<code>\@classiii</code>	1459, 1482	<code>\@firstoftwo</code>	846
<code>\@classiv</code>	1065, 1074, 1161, 1172, 1460, 1484	<code>\@float</code>	23, 28
<code>\@classv</code>	884, 974, 1460, 1486, 1534, 1574, 1729	<code>\@float@LaTeX</code>	648, 649, 670
		<code>\@floatboxreset</code>	618

<code>\@flushglue</code>	405, 1439	<code>\@ifvmode</code>	277
<code>\@font@warning</code>	355	<code>\@ifvoid</code>	277, 562, 573
<code>\@footnotemark</code>	20, 21	<code>\@ifx</code>	13, 14
<code>\@footnotemark</code>	483	<code>\@ifx</code>	255, 256, 259, 277, 298,
<code>\@footnotetext</code>	20, 21		363, 449, 718, 846, 869–889,
<code>\@footnotetext</code>	460, 467, 487, 599		983–995, 1058, 1059, 1778
<code>\@gobble@opt@one</code>	56	<code>\@ifx@empty</code>	277
<code>\@gobble@opt@one</code>	1784	<code>\@ifxundefined</code>	13
<code>\@halignto</code>	1163, 1174, 1179, 1190,	<code>\@ifxundefined</code>	192, 227, 232,
	1202, 1236, 1255, 1300		237, 255, 482, 590, 638, 648
<code>\@hang@from</code>	780, 811	<code>\@ifxundefined@cs</code>	259, 656, 664,
<code>\@hang@froms</code>	821, 842		684
<code>\@hangfrom</code>	30–32	<code>\@iiparbox</code>	584
<code>\@hangfrom</code>	811	<code>\@itemlabel</code>	1896
<code>\@hangfrom@section</code>	31	<code>\@lastchclass</code>	
<code>\@height</code>	1115, 1116, 1197, 1241,		1450, 1460, 1461, 1466,
	1250, 1272, 1374, 1427, 1434		1488, 1490, 1523, 1538, 1539,
<code>\@ialph</code>	730, 731		1551, 1586, 1588, 1629, 1638,
<code>\@if@empty</code>	277, 1877		1666, 1690, 1742, 1752
<code>\@if@sw</code>	280, 281, 345, 362, 364	<code>\@latex@warning@no@line</code>	365
<code>\@ifcat</code>	277	<code>\@listdepth</code>	1887, 1890, 1895
<code>\@ifdim</code>	277, 354, 776, 817, 1426,	<code>\@mainaux</code>	16
	1872, 1879	<code>\@mainaux</code>	346
<code>\@ifempty</code>	14	<code>\@makecol</code>	9
<code>\@ifeof</code>	277	<code>\@makefnmark</code>	634
<code>\@ifhbox</code>	277	<code>\@makefntext</code>	507, 601
<code>\@ifhmode</code>	277	<code>\@makeother</code>	711, 715
<code>\@ifinner</code>	277, 565, 1357	<code>\@minipagefalse</code>	581
<code>\@ifl@t@r</code>	261	<code>\@mklab</code>	1897
<code>\@ifmmode</code>	277, 1167, 1183, 1351,	<code>\@mkpream</code>	50
	1353, 1412	<code>\@mkpream</code>	876, 966,
<code>\@ifnextchar</code>	460,		989, 1047, 1200, 1216, 1254,
	463, 1096, 1099, 1102, 1105,		1277, 1331, 1339, 1449, 1522
	1113, 1114, 1130, 1134	<code>\@mkpream@LaTeX</code>	876, 1449
<code>\@ifnotempty</code>	14	<code>\@mkpream@array</code>	989, 1522
<code>\@ifnotrelax</code>	13	<code>\@mkpream@array@new</code>	1047, 1549
<code>\@ifnotrelax</code>	255	<code>\@mkpream@ltx</code>	966, 1464
<code>\@ifnum</code>	277, 758, 787, 801, 1773	<code>\@mkpream@relax</code>	50
<code>\@ifodd</code>	277, 1775	<code>\@mkpream@relax</code>	1514, 1554
<code>\@ifpackageloaded</code>	180, 858	<code>\@mpargs</code>	584
<code>\@ifstar</code>	750, 1111, 1112, 1122,	<code>\@mpfn</code>	471, 477, 595
	1127	<code>\@mpfootins</code>	25
<code>\@ifundefined</code>	21, 767, 780, 785,	<code>\@mpfootins</code>	551, 554, 555, 562,
	796, 799, 821, 827, 832, 836,		563, 566, 569, 573, 574
	1832, 1838	<code>\@mpfootnotetext</code>	21
<code>\@ifvbox</code>	277	<code>\@mpfootnotetext</code>	487, 599, 600

<code>\@mpmakefntext</code>	586	<code>\@ssect</code>	813
<code>\@multiplelabels</code>	363, 371	<code>\@ssect@ltx</code>	751, 813
<code>\@namedef</code> .	677, 1328, 1832, 1838	<code>\@startcolumn</code>	9
<code>\@nameuse</code> .	177, 589, 1214, 1290, 1833, 1863	<code>\@startpbox</code>	
<code>\@ne</code>	11, 59	..	1204, 1226, 1333, 1342, 1454, 1470, 1520, 1524, 1555, 1730, 1736
<code>\@newl@bel</code>	348	<code>\@startsection</code>	30, 33
<code>\@nextchar</code>	1457, 1458, 1472, 1474, 1530, 1561, 1718, 1721, 1724, 1727, 1730, 1737	<code>\@startsection</code>	736
<code>\@nmbriestfalse</code>	1898	<code>\@startsection@hook</code> ..	737, 754
<code>\@nobreakfalse</code>	1794	<code>\@starttoc</code>	57
<code>\@normalcr</code>	401	<code>\@starttoc</code>	1785
<code>\@nul</code>	716, 727	<code>\@svsec</code>	31
<code>\@outputdblcol</code>	9	<code>\@svsec</code> ..	760, 766, 768, 771, 781, 797
<code>\@parboxrestore</code>	502, 617	<code>\@svsechd</code>	794, 830
<code>\@preamble</code>	1201, 1203, 1211, 1217, 1221, 1236, 1255, 1256, 1266, 1278, 1280, 1284, 1300, 1309, 1314, 1334, 1344, 1448, 1451, 1467, 1523, 1550, 1605, 1611, 1620, 1632, 1679, 1680	<code>\@tabacol</code> .	879, 969, 1050, 1063, 1072, 1089, 1616
<code>\@preamerr</code>	1210, 1234, 1462, 1494, 1496, 1543, 1544, 1546, 1594, 1596, 1600	<code>\@tabacol@LaTeX</code>	879, 1619
<code>\@provide</code>	297, 860, 862, 865	<code>\@tabacol@ltx</code> ..	969, 1050, 1622
<code>\@refundefined</code>	361	<code>\@tabacoll</code>	1070, 1087, 1616
<code>\@resetactivechars</code>	219	<code>\@tabacolr</code>	1071, 1088, 1625
<code>\@rightskip</code>	404	<code>\@tabarray</code>	39
<code>\@runin@to</code>	31	<code>\@tabarray</code>	872, 962, 985, 1043, 1067, 1076, 1084, 1093, 1095, 1164, 1175, 1180, 1191
<code>\@runin@to</code>	796, 812	<code>\@tabarray@LaTeX</code>	872, 1095
<code>\@runin@tos</code>	832, 843	<code>\@tabarray@array</code>	985, 1101
<code>\@secntformat</code>	30, 31	<code>\@tabarray@array@new</code>	1043, 1104
<code>\@secntformat</code>	768	<code>\@tabarray@ltx</code>	962, 1098
<code>\@secpenalty</code>	748	<code>\@tabclassiv</code>	881, 971, 1065, 1074, 1717
<code>\@sect</code>	30, 31	<code>\@tabclassiv@LaTeX</code> ..	881, 1717
<code>\@sect</code>	755	<code>\@tabclassiv@ltx</code>	971, 1720
<code>\@sect@ltx</code>	752, 757	<code>\@tabclassz</code>	883, 973, 1064, 1073, 1665
<code>\@setckpt</code>	347	<code>\@tabclassz@LaTeX</code> ...	883, 1665
<code>\@sharp</code> ..	1207, 1230, 1262, 1295, 1332, 1340, 1453, 1469, 1504, 1510, 1524, 1544, 1553, 1596, 1634, 1635, 1653, 1657, 1661, 1682, 1684, 1686, 1705, 1709, 1713, 1731, 1738	<code>\@tabclassz@ltx</code>	973, 1689
<code>\@ssect</code>	30, 31, 33	<code>\@tabular</code>	39
		<code>\@tabular</code> ...	864, 871, 961, 984, 1042, 1060
		<code>\@tabular@LaTeX</code>	871, 1060
		<code>\@tabular@array</code>	984, 1078
		<code>\@tabular@array@new</code>	1042, 1086
		<code>\@tabular@ltx</code>	961, 1069

<code>\@tabularcr</code>	40	<code>\xarraycr@array</code>	993, 1129
<code>\@tabularcr</code>	886, 976, 1058, 1066,		<code>\xarraycr@new</code>	1054, 1133
	1075, 1107		<code>\xbitor</code>	1765
<code>\@tabularcr@LaTeX</code>	40	<code>\xbitor@LaTeX</code>	1765, 1778
<code>\@tabularcr@LaTeX</code>	... 886, 1111		<code>\xbitor@ltx</code>	1771, 1783
<code>\@tabularcr@ltx</code>	976, 1112	<code>\xfloat</code>	586
<code>\@tbpen</code>	40	<code>\xfloat@LaTeX</code>	586, 591
<code>\@tbpen</code>	1107	<code>\xfloat@anchored</code>	591, 609
<code>\@tempa</code>	181, 183, 717, 718	<code>\xfloat@prep</code>	588, 593
<code>\@tempb</code>	708, 718	<code>\xfootnote</code>	459
<code>\@tempc</code>	706, 724	<code>\xfootnotemark</code>	20 , 21
<code>\@tempcntb</code>	1765, 1768, 1769, 1772,		<code>\xfootnotemark</code>	459
	1774, 1775		<code>\xfootnotemark@ltx</code>	..	469, 485
<code>\@tempdima</code>	59	<code>\xfootnotenext</code>	26
<code>\@tempskipa</code>	. 740, 742, 743, 748,		<code>\xhline</code>	1375, 1381, 1382
	775, 776, 816, 817		<code>\xhline@unneeded</code>	1384
<code>\@temptokena</code>	. 1525, 1531, 1556,		<code>\xifempty</code>	14
	1561		<code>\xsect</code>	809, 840
<code>\@testdef</code>	348	<code>\xctabularcr</code>	40
<code>\@testpach</code>	1458, 1474, 1532, 1563		<code>\xctabularcr</code>	... 887, 977, 1107	
<code>\@testtrue</code>	1769, 1775	<code>\xctabularcr@LaTeX</code>	..	887, 1113
<code>\@tfor</code>	... 1456, 1472, 1530, 1561		<code>\xctabularcr@ltx</code>	977, 1114
<code>\@thefnmark</code>	472, 478, 504	<code>\@yargarraycr</code>	40
<code>\@toodeep</code>	1888	<code>\@yargarraycr</code>	889, 979, 995,
<code>\@totalleftmargin</code>	. 1907, 1910,			1056, 1107	
	1914, 1916		<code>\@yargarraycr@LaTeX</code>	. 889, 1117	
<code>\@tpfootnotetext</code>	487	<code>\@yargarraycr@array</code>	. 995, 1150	
<code>\@trivlist</code>	1900	<code>\@yargarraycr@ltx</code>	... 979, 1118	
<code>\@undefined</code>	... 846, 1090, 1187		<code>\@yargarraycr@new</code>	..	1056, 1154
<code>\@unexpandable@protect</code>	.. 1452,		<code>\@yfootnote</code>	459
	1468		<code>\[</code>	715
<code>\@whilesw</code>	1526, 1557	<code>\]</code>	40
<code>\@width</code>	.. 1115, 1116, 1140, 1146,		<code>\{</code>	711, 715
	1199, 1243, 1252, 1274, 1429		<code>\}</code>	711, 715
<code>\@write@floatline</code>	29	<code>\]</code>	715
<code>\@write@floatline</code>	714	<code>\^</code>	710, 723
<code>\@writefile</code>	530	<code>00readme.txt</code>	3 , 4
<code>\@xargarraycr</code>	40			
<code>\@xargarraycr</code> 888, 978, 994,				
	1055, 1107				
<code>\@xargarraycr@LaTeX</code>	. 888, 1115		<code>_</code>	.. 69, 93, 94, 98, 117, 123, 128,	
<code>\@xargarraycr@array</code>	. 994, 1137			134, 135, 181	
<code>\@xargarraycr@ltx</code>	... 978, 1116				
<code>\@xargarraycr@new</code>	.. 1055, 1143		A		
<code>\@xarraycr</code>	40	<code>\abovedisplayshortskip</code>	... 434	
<code>\@xarraycr</code> 993, 1054, 1107		<code>\abovedisplayskip</code> 429–434	
			<code>acronym document class</code>	1

<code>\class@name</code>	10	
<code>\class@name</code> ..	161–163, 182, 192	
<code>\class@warn</code>	<u>161</u>	
<code>\class@warn@end</code> ..	165, 170, 175, 188	836, 1052, 1791, 1792, 1831, 1834, 1837, 1839, 1843, 1844, 1856–1858, 1867, 1895
D		
<i>(class customization commands)</i> place- holder	9	<code>\dollarbegin</code> . 1059, 1082, 1091, 1179, 1188, 1189
<code>\ClassError</code>	161, 182	<code>\dollarend</code> 1083, 1092, 1179, 1189
<code>classes.dtx</code>	23	<code>\deadcycles</code>
<code>classes.dtx</code> document class ..	28	339
<code>\ClassInfo</code>	163	<code>\detokenize</code>
<code>\classname</code>	85, 92, 140	381, 382
<code>\ClassWarningNoLine</code>	162	<code>\dimen</code>
<code>\clear@document</code>	17, 18	57
<code>\clear@document</code> ..	337, 342, 396, 399	<code>\dimen@</code> ..
<code>\clearpage</code>	17, 18	1425–1427, 1858, 1859
<code>\clearpage</code>	342, 392, 399	<code>\dimen@iii</code>
<code>\closeout</code>	346, 679	195
<code>\col@sep</code> .	1081, 1090, 1178, 1187	<code>\dimendef</code>
<code>\color@begingroup</code>	506	195
<code>\color@endgroup</code>	510, 582	<code>\displaystyle</code> .
<code>\colrule</code>	46	416, 419, 438, 444
<code>\colrule</code>	<u>1370</u>	<code>\displaywidth</code>
<code>\column@fil</code>	1370, 1405	415
<code>\column@font</code>	1370, 1404	<code>\do</code>
<code>\column@sep</code>	519, 521	1457, 1473, 1531, 1562
<code>\columnwidth</code>	22	<code>\do@check@aux</code>
<code>\columnwidth</code>	514, 616	17, 18
<code>\contentsline</code>	24, 56	<code>\do@check@aux</code>
<code>\contentsline</code>	<u>532</u>	330, 393, 398
<code>\copy</code>	1412	<code>\do@if@floats</code>
<code>\count</code>	1765, 1772	23, 27
<code>\count@</code> ..	1505, 1511, 1527, 1558	<code>\do@if@floats</code>
<code>\count@i</code>	<u>224</u>	<u>637</u>
<code>\count@ii</code>	<u>224</u>	doc
<code>\countdef</code>	224, 225	4, 6
<code>\crrc</code> 1303, 1306, 1309, 1312, 1317, 1393		<code>\DocInput</code>
<code>\csname</code>	13, 42	8
<code>\csname</code>	22, 202, 204, 208, 210, 259, 280, 471, 504, 538, 641, 643, 646, 647, 675, 676, 680, 689, 693, 695, 706, 709, 772, 780, 785, 790, 796, 799, 804, 821, 827, 832,	<code>\document</code>
		15
		<code>\document</code>
		301
		document class
		acronym
		1
		acronym.sty
		3
		array 9, 10, 34, 35, 38, 40, 41, 50
		classes.dtx
		28
		ftnright
		9
		hyperref ..
		18, 21, 24, 25, 32, 33, 56, 59
		hyperref _□ package
		2, 24
		longtable
		10, 50
		ltxdoc
		1, 5, 8
		ltxdocext
		1
		ltxdocext.sty
		3
		ltxgrid
		10, 16, 21
		ltxutil
		9
		document environment
		4, 16
		<code>\doublerulesep</code>
		1387
		<code>\dp</code> .
		499, 1115, 1116, 1139, 1145, 1198, 1242, 1251, 1273, 1428

E

`\edef` 50
`\edef` 229, 234, 239,
381, 382, 1201, 1605, 1611,
1620, 1632, 1679, 1834
`\end` 29
`\end@float@anchored` . 613, 614,
622
`\end@float@pagebreak` 627, 630,
662
`\endarray` 44
`\endarray` 857, 874, 964, 987, 1045,
1302, 1320, 1323, 1326
`\endarray@array` 987, 1308
`\endarray@array@new` 1045, 1311
`\endarray@hook` 861, 862
`\endarray@LaTeX` 874, 1302
`\endarray@ltx` 44
`\endarray@ltx` .. 964, 1305, 1312
`\endbatchfile` 65
`\endcsname` 22, 202,
204, 208, 210, 259, 280, 471,
504, 538, 641, 643, 646, 647,
675, 676, 680, 689, 693, 695,
706, 709, 772, 780, 785, 790,
796, 799, 804, 821, 827, 832,
836, 1052, 1791, 1792, 1831,
1834, 1837, 1839, 1843, 1844,
1856–1858, 1867, 1895
`\enddocument` 17, 18
`\enddocument` 330
`\endgroup` 15, 17
`\endminipage` 577
`\endpreamble` 44
`\endtabular` . 857, 875, 965, 988,
1046, 1052, 1316, 1328
endtabular environment 45
`\endtabular*` 1328
`\endtabular@array` ... 988, 1322
`\endtabular@array@new` ... 1046,
1325
`\endtabular@hook` 859, 860
`\endtabular@LaTeX` ... 875, 1316
`\endtabular@ltx` 965, 1319
`\endwrite@float` 699, 709
`\endwrite@floats` 701

environment

array 2, 34, 39
document 4, 16
endtabular 45
eqnarray 18, 40
figure 23
figure@write 23, 28
longtable 23, 29
minipage 20, 23
tabular 2, 34, 39, 40
`\eqnarray` 18
`\eqnarray` 449, 451
eqnarray environment ... 18, 40
`\eqnarray@fleqn@fixed` 407
`\eqnarray@LaTeX` 407
`\eqncolsep` ... 440, 443, 452, 453
`\errorstopmode` 216
`\extrarowheight` 1248, 1270

F

`\f@ur` 197, 1551
`\false@sw` 270, 276,
281, 293, 298, 892, 895, 898,
901, 904, 907, 910, 913, 916,
919, 922, 925, 928, 931, 934,
937, 940, 943, 946, 949, 952,
998, 1001, 1004, 1007, 1010,
1013, 1016, 1019, 1022, 1025,
1028, 1031, 1034
figure environment 23
figure@write environment 23, 28
`\figuresname` 23
file
.dtx 1, 4–6
.ins 4
.tfm 1
.vf 1
.vpl 1
00readme.txt 3, 4
acrofont.sty 1–3
classes.dtx 23
doc 4, 6
ltxfloat.dtx 20
ltxdocext.dtx 1–3
ltxdocext.ins 3
ltxdocext.pdf 1

ltxdocext.sty	1–3
ltxutil	2, 10, 60
ltxutil.drv	5
ltxutil.ins	5
ltxutil.sty	5
makeindex	3
nameref.sty	32
pdfmark.def	32
zpsynocmr	1
zptmnochr	1
zptmnochr	1
zpcznochr	1
\file	17, 47, 48, 103, 106, 112, 113, 122, 128, 131, 132, 134, 137, 138
float, argument	29
\float@end@tag	29
\float@end@tag	714
\floatingpenalty	21
\floatingpenalty	500
\floats@sw	26, 27
\floats@sw	586, 590, 591, 607, 608, 638, 639, 666
\flushing	400
\fmtversion	261
\font@submax	354, 357
\font@subfuzz	354
\footins	488, 574
\footnote	20, 23
\footnote	459, 594, 605
\footnote@latex	594, 605
\footnotemark	20
\footnotemark	459
\footnotesep	498, 508
\footnotesize	496, 1800
\fp@proc@H	608
\fp@proc@h	607
\from	47, 49
\frustrut	1370, 1371, 1408
ftnright document class	9
\fullinterlineskip	223, 564
\futurelet	1374, 1381
G	
\gappdef	189, 226
\generate	46
\GetFileInfo	26
\glet@environment	176, 200
\glossary	526
H	
\H@@footnotemark	474, 479, 482, 483
\H@@footnotetext	600
\H@refstepcounter	31
\H@refstepcounter	762, 855
\H@svsec	759, 763, 781, 797, 815, 822, 823, 833, 834
\halign	44, 50
\halign@	196, 437
\hangindent	59
\hb@xt@	420, 445, 634, 1865, 1875
\hbox	59
\hline	47
\hline	885, 975, 1373
\hline@LaTeX	885, 1373
\hline@ltx	975, 1377
\hline@rule	47
\hline@rule	1372, 1380, 1395, 1410
<i><hooks and bookkeeping></i> placeholder	17
\hrule	12
\hsize	15
\ht	1197, 1241, 1247, 1269, 1425, 1426
\Hy@raisedlink	849
\hyper@anchor	847
\hyper@anchorend	33
\hyper@anchorend	851
\hyper@anchorstart	33
\hyper@anchorstart	850
\hyper@last	33
\hyper@last	848
\hyper@linkend	33
\hyper@linkend	853
\hyper@linkstart	33
\hyper@linkstart	852
\hyperanchor	33
hyperref document class	18, 21, 24, 25, 32, 33, 56, 59

hyperref _␣ package document class	\itemindent	1894
..... 2, 24		
I	J	
\ialign	\j@nk	299
..... 42		
\ialign	K	
..... 1202, 1255	\keepsilent	45
\if		
..... 42	L	
\if	\l@	57
..... 267, 279, 1194, 1260	\l@@sections	57
\if@filesw	\l@@sections	1848
.. 345, 362, 1790, 1819	\l@section	1802
\if@firststamp	\label	526, 540
..... 1605, 1608	\lastbox	567, 683
\if@nobreak	\LaTeX	69, 93, 94, 104, 114, 117,
..... 745	128, 134, 135, 142	
\if@noskipsec	\LaTeXe	98, 123, 153
..... 738	\leftmargin	1907, 1909, 1914
\ifeof	\leftmargini	1802
..... 283	\leftmarginii	1803
\iffalse	\leftmarginiii	1804
..... 1121, 1126	\leftmarginiv	1805
\IfFormatAtLeastTF	\leftmarginv	1806
... 260, 263	\leftmarginvi	1807
\ifhbox	\leftskip	59
..... 284	\leftskip	402, 1857, 1859, 1916
\ifinner	\let	57
..... 286	\let@environment	11
\ifmmode	\let@environment	200, 654, 655,
..... 287	657, 673, 674	
\ifodd	\lineskip	1209, 1232, 1263, 1297
..... 289, 1769	\linewidth	437, 515, 522,
\ifToplevel	1908–1910	
..... 52	\list	59
\ifvbox	\list	1886
..... 290	\listparindent	1893, 1906, 1913
\ifvoid	longtable document class	10, 50
..... 292	longtable environment	23, 29
\ignorespaces	\loop	14
.....	\loopuntil	295
... 508, 1334, 1347, 1504,	\loopwhile	295
1510, 1682, 1684, 1686, 1705,	\lrstrut	1371, 1372, 1409
1709, 1713, 1730, 1738, 1864,	\LT@array	10
1871, 1883, 1903	ltfloat.dtx	20
\immediate	ltxdoc document class	1, 5, 8
.. 346, 644, 678, 724,		
1792, 1842		
\incompatible@package		
.... 179		
\index		
..... 526		
\init@documenthook		
.. 304, 312,		
319, 844		
\insert		
..... 488, 574		
\insert@column		
.. 991, 1051, 1502		
\insert@column@array		
991, 1502		
\insert@column@array@new		
1051,		
1507		
\interdisplaylinepenalty		
.. 40		
\interfootnotelinepenalty		
497		
\interlinepenalty		
497, 778, 819		
\intertabularlinepenalty		
.. 40		
\intertabularlinepenalty		
1107,		
1108, 1110, 1112, 1127		
\intextsep		
..... 629, 630		
\item		
..... 126, 136, 139		

ltxdocext document class	1	\newenvironment	29	
ltxdocext.dtx	1–3	\newlabel	24	
ltxdocext.ins	3	\newlabel	543	
ltxdocext.pdf	1	\newlinechar	723	
ltxdocext.sty	1–3	\newwrite	640, 1791	
ltxdocext.sty document class	3	\noalign	1114, 1116–1118, 1135, 1148, 1152, 1156, 1372, 1374, 1378, 1382, 1394	
ltxgrid document class	10, 16, 21	\noexpand	221, 222, 1202, 1255	
ltxutil	2, 10, 60	\normalsize	306, 314	
ltxutil document class	9	\numberline	59	
ltxutil.drv	5	\numberline@sections	1862, 1874	
ltxutil.ins	5			
ltxutil.sty	5			
M				
\make@footnotetext	21	O		
\make@footnotetext	487	\obeylines	720	
\makeatletter	20, 350, 1788	\obsolete@command	164	
\makeatother	23	\oneapage	662	
makeindex	3	\openout	645, 1792	
\makelabel	1897	\other	199	
\maketitle	21	P		
\maketitle	83	\p@equation	409, 425	
\mark	24	\package@name	151, 152	
\mathindent	427, 457	\PackageInfo	152	
\meaning	221	\pagegrid@col	620	
\MessageBreak	356	\parindent	59	
minipage environment	20, 23	\parindent	632, 1859, 1875, 1906, 1913	
\minipagefootnote@drop	21	\parsep	1901	
\minipagefootnote@drop	511, 550	\parshape	59	
\minipagefootnote@foot	550	\parshape	1910	
\minipagefootnote@here	25	\parskip	431, 1901	
\minipagefootnote@here	548, 550, 623	\partopsep	430	
\minipagefootnote@init	550, 619	\pbox@hook	1445, 1738	
\minipagefootnote@pick	492, 550	pdfmark.def	32	
\Msg	53–59, 61–63	\pdfstringdefDisableCommands	16	
\multicolumn	870, 960, 1329	\phantomsection	33	
\multicolumn@LaTeX	870, 1329	\phantomsection	759, 764, 815, 845	
\multicolumn@ltx	960, 1336	placeholder		
\multispan	1330, 1337	<class customization commands>		
N			 9
nameref.sty	32	<hooks and bookkeeping>	17	
\NC@list	1526, 1557			
\newbox	1421			
\newcount	1107, 1109			

<i>(read main .aux and final checks)</i>	\rvtx@ifformat@geq .. 260, 301,
..... 17	330, 376
<i>(your document here)</i> 9	
\preamble 31	
\prepdef 12	
\prepdef 226, 310, 548, 675,	
676, 768, 770, 859, 861, 864,	
1217, 1280	
\prepnxt@tok 1529, 1560	
\prevdepth 12	
\prevdepth 223	
\print@float 28	
\print@float 661	
\printfigures 24	
\protected 379	
\protected@edef 503, 766	
\protected@write 525, 542	
\protected@xdef 478	
\providecommand .. 845, 847–853,	
863	
\ProvidesFile 4, 6	
R	
\raggedleft 18	
\raggedright 18	
<i>(read main .aux and final checks)</i> place-	
holder 17	
\realfootnote 21	
\refstepcounter 31	
\relax 31, 34	
\repeat 14	
\replace@command 169	
\replace@environment 174	
\RequirePackage 13, 15, 22	
\reserved@a .. 1375, 1381, 1385,	
1386, 1457, 1472	
\reset@font 496	
\romannumeral 1895	
\rule 508	
\rvtx@enddocument@patch .. 376	
\rvtx@enddocument@patch@end .	
..... 385, 392	
\rvtx@enddocument@patch@more	
..... 387, 390	
\rvtx@ifformat@geq 13	
	S
	\samepage 40
	\samepage 602, 1110
	\save@decl 1534, 1571
	\say 12
	\say 221, 1385
	\saythe 12
	\saythe 221
	\sc 18, 114
	\sec@upcase 34
	\sec@upcase 856
	\section 110, 686
	\Sectionformat 32
	\Sectionformat ... 822, 833, 846
	\set@arstrutbox 47
	\set@arstrutbox 1215, 1239, 1410
	\set@eqnarray@skips 458
	\set@footnotewidth 22
	\set@footnotewidth 487
	\set@footnotewidth@ii 22
	\set@footnotewidth@ii 517
	\set@listindent 1902, 1918
	\set@listindent@ 1912
	\set@listindent@parshape 1905,
	1918
	\set@pica@hook 16
	\set@pica@hook ... 307, 315, 328
	\set@tocdim@pagenum 1855, 1870
	\set@typeset@protect ... 1208,
	1231, 1296, 1332, 1341
	\set@typesize@hook 16
	\set@typesize@hook 305, 313, 328
	\shipout 12
	\showboxbreadth 216
	\showboxdepth 216
	\showoutput 219
	\skip@ 1861, 1864
	\splitmaxdepth 499
	\splittopskip 498
	\stepcounter 408, 425, 477
	\stop 183
	\StopEventually 6

<code>\string</code> .. 165, 170, 181, 527–530, 543, 756, 1779, 1781, 1843	<code>\thr@@</code> .. 11
<code>\strutbox</code> .. 499, 508, 1197, 1198, 1241, 1242, 1247, 1251, 1269, 1273, 1426, 1428	<code>\thr@@</code> . 195, 420, 445, 1543, 1594
<code>\subsection</code> .. 120	<code>\title</code> .. 68
<code>\switch@array</code> .. 34, 40	<code>\toc@font</code> .. 56
<code>\switch@array</code> .. 858, 867	<code>\toc@font</code> .. 1800, 1809
<code>\switch@tabular</code> .. 34, 40	<code>\toc@letdimen</code> .. 1815, 1816, 1836
<code>\switch@tabular</code> .. 858, 867	<code>\toc@post</code> .. 57
	<code>\toc@post</code> .. 1795, 1799
	<code>\toc@post@auto</code> .. 57
	<code>\toc@post@auto</code> .. 1808
	<code>\toc@pre</code> .. 57
	<code>\toc@pre</code> .. 1787, 1798
	<code>\toc@pre@auto</code> .. 57
	<code>\toc@pre@auto</code> .. 1808
	<code>\toc@setindent</code> . 1811–1814, 1830
	<code>\toc@writedimen</code> 1821–1826, 1841
	<code>\tocdim@appendix</code> .. 1806
	<code>\tocdim@min</code> .. 1831, 1837, 1885
	<code>\tocdim@pagenum</code> .. 1807, 1872
	<code>\tocdim@paragraph</code> .. 1805
	<code>\tocdim@section</code> .. 1802
	<code>\tocdim@subsection</code> .. 1803
	<code>\tocdim@subsubsection</code> .. 1804
	<code>\tocleft@pagenum</code> .. 1860
	<code>\tocleft@section</code> .. 58
	<code>\tocmax@section</code> .. 58
	<code>\toks</code> .. 1547, 1602
	<code>\toks@</code> .. 227, 229, 232, 234, 237, 239, 707, 724, 726
	<code>\toks@ii</code> .. 12
	<code>\toks@ii</code> . 228, 229, 233, 234, 238, 239, 254
	<code>\toksdef</code> .. 254
	<code>\toprule</code> .. 46
	<code>\toprule</code> .. 1370
	<code>\topsep</code> .. 429
	<code>\traceoutput</code> .. 12
	<code>\traceoutput</code> .. 218
	<code>\tracingall</code> .. 12
	<code>\tracingcommands</code> .. 213
	<code>\tracinglostchars</code> .. 214
	<code>\tracingmacros</code> .. 215
	<code>\tracingonline</code> .. 213
	<code>\tracingoutput</code> .. 12
	<code>\tracingoutput</code> .. 214
	<code>\tracingpages</code> .. 214

T

<code>\t@</code> .. 194, 196, 415
<code>\tab@rule</code> .. 47
<code>\tab@rule</code> .. 1370, 1371, 1392
<code>\tabcolsep</code> 1081, 1441–1443, 1620
<code>\tableleft@skip</code> .. 1233, 1298, 1438
<code>\tableleftsep</code> .. 1441, 1515, 1617
<code>\tableofcontents</code> .. 108
<code>\tabmid@skip</code> .. 1218, 1281, 1439
<code>\tabmidsep</code> .. 1442, 1516, 1623
<code>\tabright@skip</code> . 1222, 1285, 1440
<code>\tabrightsep</code> .. 1443, 1517, 1626
<code>\tabskip</code> .. 42
<code>\tabskip</code> .. 413, 416, 419, 421, 427, 438, 444, 446, 1203, 1218, 1222, 1233, 1257, 1281, 1285, 1298
<code>\tabular</code> .. 34
tabular environment 2, 34, 39, 40
<code>\tabular@hook</code> .. 864, 865
<code>\tabularnewline</code> 1205, 1228, 1266, 1293
<code>\TeX</code> .. 98
text, argument .. 34
<code>\texttt</code> .. 80
<code>\textwidth</code> .. 518
<code>\thanks</code> .. 70
<code>\the@toks</code> 1503, 1505, 1508, 1511, 1528, 1547, 1559, 1602
<code>\theequation</code> .. 409, 425
<code>\thefootnote</code> .. 597
<code>\thempfn</code> .. 472, 478, 596
<code>\thempfootnote</code> .. 596, 597
<code>\thepage</code> .. 534, 543

<code>\tracingparagraphs</code>	215	<code>\xdef</code>	1255, 1278
<code>\tracingplain</code>	12	Y	
<code>\tracingplain</code>	212	<code>\y</code>	382, 384
<code>\tracingrestores</code>	215	<i>\your document here</i> placeholder 9	
<code>\tracingstats</code>	213	Z	
<code>\trigger@float@par</code> ..	672, 675, 676, 682	<code>\z@</code>	58
<code>\triggerpar</code>	661, 672	<code>\z@skip</code> ..	402–404, 416, 421, 438, 446, 624, 1203, 1209, 1232, 1438–1440, 1865
<code>\true@sw</code> .	268, 275, 281, 293 , 298, 316, 890, 996	<code>zpsynocmr</code>	1
<code>\tw@</code>	11	<code>zptmnochr</code>	1
U			
<code>\unhbox</code>	1865, 1880	<code>zptmnochr</code>	1
<code>\unhcopy</code>	1412	<code>zptmnochr</code>	1
<code>\unrestored@protected@xdef</code>	472	<code>zpcnocmry</code>	1
<code>\unskip</code> 579, 1138, 1144, 1371, 1372, 1504, 1510, 1682, 1684, 1686, 1705, 1709, 1713, 1864			
<code>\unvbox</code>	555, 569, 574, 584		
<code>\unvcopy</code>	566		
<code>\uppercase</code>	34		
<code>\url</code>	14, 89, 95		
V			
<code>\vbox</code> ..	554, 612, 1194, 1260, 1350		
<code>\vcenter</code>	46		
<code>\vcenter</code> .	1194, 1260, 1351, 1355, 1358		
<code>\voidb@x</code>	551		
<code>\vrule</code> 1115, 1116, 1140, 1146, 1197, 1241, 1249, 1272, 1427			
<code>\vsize</code>	15		
<code>\vtop</code>	566, 1194, 1260, 1349		
W			
<code>\write</code>	724, 1842		
<code>\write@float</code>	698 , 702		
<code>\write@float</code>	23 , 28 , 29		
<code>\write@float</code>	651, 698		
<code>\write@floatline</code>	29		
<code>\write@floatline</code>	712, 714		
<code>\write@floats</code>	652, 698		
X			
<code>\x</code>	381, 384		