

# A L<sup>A</sup>T<sub>E</sub>X Package of utility macros <sup>\*†</sup>

Arthur Ogawa <sup>‡</sup>

October 1, 2020

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

The distribution point for this work is [journals.aps.org/revtex](https://journals.aps.org/revtex), which contains prebuilt runtime files, documentation, and full source, ready to add to a TDS-compliant T<sub>E</sub>X installation.

The `ltxutil` package was commissioned by the American Physical Society and is distributed under the terms of the L<sup>A</sup>T<sub>E</sub>X Project Public License 1.3c, the same license under which all the portions of L<sup>A</sup>T<sub>E</sub>X itself are distributed. Please see <http://ctan.tug.org/macros/latex/base/lppl.txt> for details.

To use this document class, you must have a working T<sub>E</sub>X installation equipped with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and possibly pdf<sub>te</sub>x and Adobe Acrobat Reader or equivalent.

To install, retrieve the distribution, unpack it into a directory on the target computer, and move the file `ltxutil.sty` into a location in your filesystem where it will be found by L<sup>A</sup>T<sub>E</sub>X.

To use, read the user documentation `ltxutil.pdf`.

## Contents

<b>1</b>	<b>Processing Instructions</b>	<b>2</b>
1.1	Build Instructions . . . . .	3
1.2	Change Log . . . . .	3
1.3	Bill of Materials . . . . .	3
1.3.1	Primary Source . . . . .	3
1.3.2	Generated by <code>latex ltxutil.dtx</code> . . . . .	4
1.3.3	Generated by <code>tex ltxutil.dtx</code> . . . . .	4
1.3.4	Auxiliary . . . . .	4
<b>2</b>	<b>Code common to all modules</b>	<b>4</b>

---

<sup>\*</sup>This file has version number 4.2d, last revised 2020/09/19.

<sup>†</sup>Version 4.2d © 2019–2020 American Physical Society

<sup>‡</sup>[mailto:arthur\\_ogawa@sbcglobal.net](mailto:arthur_ogawa@sbcglobal.net)

<b>3</b>	<b>The driver module doc</b>	<b>5</b>
3.1	The Preamble	5
3.1.1	Docstrip and info directives	5
3.2	The “Read Me” File	6
3.3	The Document Body	9
<b>4</b>	<b>Using this package</b>	<b>9</b>
4.1	Invoking the package	9
<b>5</b>	<b>Compatibility with L<sup>A</sup>T<sub>E</sub>X’s Required Packages</b>	<b>10</b>
5.1	array	10
5.2	longtable	10
<b>6</b>	<b>Implementation of package</b>	<b>10</b>
6.1	Beginning of the <code>package</code> DOCSTRIP module	10
6.2	Banner and beginning of the <code>kernel</code> DOCSTRIP module	11
6.3	Errors and warnings	11
6.4	New Tools	11
6.5	Boolean Control	13
6.6	Begin Document Structure	15
6.7	Class Extensions	19
6.8	Type Tools	21
6.9	Display Math	22
6.10	Footnotes	24
6.11	Floats	29
6.11.1	Usage notes	29
6.11.2	Robustifying fragile commands	30
6.11.3	Preparing for the <code>hyperref</code> package	31
6.11.4	Footnotes within floats, unfloating floats, float font	31
6.11.5	Writing floats out to a file	34
6.12	Counters	38
6.13	Customization of Sections	38
6.14	Patch the <code>tabular</code> and <code>array</code> Environments	42
6.15	Repair other broken parts of L <sup>A</sup> T <sub>E</sub> X	64
6.16	Syntax	64
6.17	Auto-indented Contents	64
6.18	Lists	68
6.19	Hypertext capabilities	69
6.20	End of the <code>kernel</code> DOCSTRIP module	72
	<b>Index</b>	<b>73</b>

## 1 Processing Instructions

The package file `ltxutil.sty` is generated from this file, `ltxutil.dtx`, using the DOCSTRIP facility of L<sup>A</sup>T<sub>E</sub>X via `tex ltxutil.dtx` (Note: do *not* use L<sup>A</sup>T<sub>E</sub>X for this

task). The typeset documentation that you are now reading is generated from the same file by typesetting it with  $\text{\LaTeX}$  or  $\text{pdf\texttt{tex}}$  via `\texttt{latex ltxutil.dtx}` or `\texttt{pdflatex ltxutil.dtx}`.

## 1.1 Build Instructions

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

1. Within an otherwise empty directory, typeset `ltxutil.dtx` with  $\text{\LaTeX}$  or  $\text{pdf\texttt{latex}}$ ; you will obtain the typeset documentation you are now reading, along with the file `README-LTXUTIL`.

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

2. Now typeset `ltxutil.dtx` with  $\text{\TeX}$ (not  $\text{\LaTeX}$ ), thereby generating the package file `ltxutil.sty`.
3. Install the following files into indicated locations within your TDS-compliant `texmf` tree (you may need root access):

- `$\text{\texttt{TEXMF}}/\text{tex}/\text{latex}/\text{revtex}/\text{ltxutil.sty}`
- `$\text{\texttt{TEXMF}}/\text{source}/\text{latex}/\text{revtex}/\text{ltxutil.dtx}`
- `$\text{\texttt{TEXMF}}/\text{doc}/\text{latex}/\text{revtex}/\text{ltxutil.pdf}`

where `$\text{\texttt{TEXMF}}/$` stands for `texmf-local/`, or some other `texmf` tree in your installation.

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

## 1.2 Change Log

## 1.3 Bill of Materials

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

### 1.3.1 Primary Source

One single file generates all.

```
%ltxutil.dtx
%
```

### 1.3.2 Generated by latex ltxutil.dtx

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

```
%README-LTXUTIL ltxutil.pdf
%
```

### 1.3.3 Generated by tex ltxutil.dtx

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

```
%ltxutil.sty
%
```

### 1.3.4 Auxiliary

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

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

## 2 Code common to all modules

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

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

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

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

```
4 %<kernel&!package&!doc>\typeout{%
5 %<*package|doc>
6 \ProvidesFile{%
7 %</package|doc>
8 %<*kernel|package|doc>
9 ltxutil%
10 %</kernel|package|doc>
11 %<*doc>
12 .dtx%
13 %</doc>
14 %<package>.sty%
15 %<*package|doc>
16 }%
17 %</package|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. Give credit where due.

```
18 %<*doc|package|kernel>
19 %<version>
20 [2020/09/19 4.2d utilities package (portions licensed from W. E. Baxter web at superscript.com
21 %</doc|package|kernel>
22 %<kernel&!package&!doc>}%
```

### 3 The driver module doc

This module, consisting of the present section, typesets the programmer’s documentation, generating the `README-LTXUTIL` 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.

```
23 %<*doc>
```

#### 3.1 The Preamble

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

```
24 \documentclass{ltxdoc}
25 \RequirePackage{ltxdocext}%
26 \let\url\undefined
27 \RequirePackage[colorlinks=true,linkcolor=blue]{hyperref}%
28 \pdfstringdefDisableCommands{%
29   \let\file\relax
30   \let\sc\relax
31 }
32 %\expandafter\ifx\csname package@font\endcsname\@undefined\else
33 % \expandafter\RequirePackage\expandafter{\csname package@font\endcsname}%
34 %\fi
35 \CodelineIndex\EnableCrossrefs % makeindex -s gind.ist ltxutil
36 \RecordChanges % makeindex -s gglo.ist -o ltxutil.gls ltxutil.glo
```

##### 3.1.1 Docstrip and info directives

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

```
37 \setcounter{StandardModuleDepth}{1}
```

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

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

### 3.2 The “Read Me” File

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

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

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

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

```
39 \begin{filecontents*}{README-LTXUTIL}
40 \title{%
41 A \LaTeX\ Package of utility macros%
42 \thanks{%
43 This file has version number \fileversion,
44 last revised \filedate.%
45 }%
46 \thanks{%
47 Version \fileversion\ \copyright\ 2019--2020 American Physical Society
48 }%
49 }%
50 \author{%
51 Arthur Ogawa%
52 \thanks{\texttt{mailto:arthur\_ogawa at sbcglobal.net}}}%
53 }%
54 %\iffalse
55 % For version number and date,
56 % search on "\fileversion" in the .dtx file,
57 % or see the end of the README-LTXUTIL file.
58 %\fi
59 \maketitle
60
61 This file embodies the \classname{ltxutil} package,
62 the implementation and its user documentation.
63
64 The distribution point for this work is
65 \url{journals.aps.org/revtex},
66 which contains prebuilt runtime files, documentation, and full source,
67 ready to add to a TDS-compliant \TeX\ installation.
68
69 The \classname{ltxutil} package was commissioned by the American Physical Society
70 and is distributed under the terms of the \LaTeX\ Project Public License 1.3c,
71 the same license under which all the portions of \LaTeX\ itself are distributed.
72 Please see \url{http://ctan.tug.org/macros/latex/base/lppl.txt} for details.
73
74 To use this document class, you must have a working
75 \TeX\ installation equipped with \LaTeXe\
```

76 and possibly pdftex and Adobe Acrobat Reader or equivalent.  
77  
78 To install, retrieve the distribution,  
79 unpack it into a directory on the target computer,  
80 and move the file \file{ltxutil.sty}  
81 into a location in your filesystem where it will be found by \LaTeX.  
82  
83 To use, read the user documentation \file{ltxutil.pdf}.  
84  
85 \tableofcontents  
86  
87 \section{Processing Instructions}  
88  
89 The package file \file{ltxutil.sty}  
90 is generated from this file, \file{ltxutil.dtx},  
91 using the {\sc docstrip} facility of \LaTeX  
92 via |tex ltxutil.dtx| (Note: do \emph{not} use \LaTeX\ for this task).  
93 The typeset documentation that you are now reading is generated from  
94 the same file by typesetting it with \LaTeX\ or pdftex  
95 via |latex ltxutil.dtx| or |pdflatex ltxutil.dtx|.
96  
97 \subsection{Build Instructions}  
98  
99 You may bootstrap this suite of files solely from \file{ltxutil.dtx}.  
100 Prepare by installing \LaTeXe\ (and either tex or pdftex) on your computer,  
101 then carry out the following steps:  
102 \begin{enumerate}  
103 \item  
104 Within an otherwise empty directory,  
105 typeset \file{ltxutil.dtx} with \LaTeX\ or pdflatex;  
106 you will obtain the typeset documentation you are now reading,  
107 along with the file \file{README-LTXUTIL}.  
108  
109 Note: you will have to run \LaTeX, then  
110 \file{makeindex} \texttt{-s gind.ist ltxutil.idx}, then  
111 \file{makeindex} \texttt{-s gglo.ist -o ltxutil.gls ltxutil.glo}, then  
112 \LaTeX\ again in order to obtain a valid index and table of contents.  
113 \item  
114 Now typeset \file{ltxutil.dtx} with \TeX (not \LaTeX),  
115 thereby generating the package file \file{ltxutil.sty}.  
116 \item  
117 Install the following files into indicated locations within your  
118 TDS-compliant \texttt{texmf} tree (you may need root access):  
119 \begin{itemize}  
120 \item  
121 \file{\$\text{TEXMF}/\text{tex}/\text{latex}/\text{revtex}/\text{classname}{ltxutil.sty}  
122 \item  
123 \file{\$\text{TEXMF}/\text{source}/\text{latex}/\text{revtex}/\text{classname}{ltxutil.dtx}  
124 \item  
125 \file{\$\text{TEXMF}/\text{doc}/\text{latex}/\text{revtex}/\text{classname}{ltxutil.pdf}

```

126 \end{itemize}
127 where \file{$TEXMF/} stands for \file{texmf-local/}, or some other \texttt{texmf} tree
128 in your installation.
129 \item
130 Run \texttt{mktexlsr} on \file{$TEXMF/} (you may need root access).
131 \item
132 Build and installation are now complete;
133 now put a \cmd\usepackage\texttt{\{ltxutil\}} in your document preamble!
134 \end{enumerate}
135
136 \subsection{Change Log}
137 \changes{4.0b}{1999/06/20}{AO: Fixed spurious \texttt{CR} and (return) characters in output fil
138 \changes{4.0b}{1999/06/20}{AO: Removed superfluous \cs{def}s, changed to using \cs{floats@sw} a
139 \changes{4.0b}{1999/06/20}{only execute if there really were floats of the given type}
140 \changes{4.0b}{1999/06/20}{Support the hack with \cs{prepdef}, and delay until \cs{AtBeginDocum
141 \changes{4.0c}{1999/11/13}{(AO, 110) Install hooks for endfloats processing}
142 \changes{4.0c}{1999/11/13}{(AO, 116) Hyperref compatibility}
143 \changes{4.0c}{1999/11/13}{(AO, 130) Interference from array package}
144 \changes{4.0c}{1999/11/13}{*-form mandates pagebreak at each float; only print section head if
145 \changes{4.0d}{2000/04/10}{(AO, 127) Floats placed [h] to allow page breaks}
146 \changes{4.0d}{2000/04/10}{(AO, 174) kernel fix}
147 \changes{4.0d}{2000/05/19}{(AO, 224) Hyperref compatibility.}
148 \changes{4.0d}{2000/05/23}{Allow things to break over pages by setting array@default.}
149 \changes{4.0e}{2000/11/16}{(AO, 221) Remove samepage command from @xfloat@prep: If the float ca
150 \changes{4.0f}{2001/07/13}{(AO, 404) Hyperref compatibility}
151 \changes{4.1a}{2008/01/19}{(AO, 459) do not assume \cs{class@name} is defined}%
152 \changes{4.1a}{2008/01/19}{(AO, 461) Change the csname from \cs{@dotsep} to \cs{ltxu@dotsep}. T
153 \changes{4.1a}{2008/01/19}{(AO, 475) I had not properly reproduced the LaTeX macro \cs{eqnarray
154 \changes{4.1a}{2008/01/19}{(AO, 479) Per: Dylan Thurston<dpt at math.harvard.edu>}%
155 \changes{4.1a}{2008/06/30}{(AO) Make \cs{addtocontents} a \cs{long} \cs{def}; gobble up \cs{foo
156 \changes{4.1a}{2008/06/30}{(AO) Remove code that avoided changes to \cs{@footnotemark}}%
157 \changes{4.1a}{2008/06/30}{(AO, 438) Complete rewrite of footnote macros.}
158 \changes{4.1a}{2008/07/07}{\cs{@xfloat@prep} calls \cs{ltx@footnote@pop} to restore the origina
159 \changes{4.1a}{2008/08/12}{\cs{class@documenthook} is the last \cs{AtBeginDocument} token now}
160 \changes{4.1a}{2008/08/12}{Class extension mechanism \cs{@pushfilename@ltx} and \cs{@p@pfilenam
161 \changes{4.1a}{2008/08/12}{Class extension mechanism \cs{class@extension}, \cs{class@extensionf
162 \changes{4.1a}{2008/08/12}{Get rid of \cs{set@typesize@hook} \cs{set@pica@hook} and the \cs{nor
163 \changes{4.1b}{2008/08/12}{(AO, 487) Support for video figures and the \cs{setfloatlink} comman
164 \changes{4.1b}{2008/08/12}{(AO, 505) try to accommodate \classname{colortbl}.}
165 \changes{4.1b}{2008/08/12}{Acquire \classname{hyperref} saveoire}
166 \changes{4.1b}{2008/08/12}{Default assignment of \cs{float@sw} now, not at \cs{AtBeginDocument}
167 \changes{4.1b}{2008/08/12}{If class option \classoption{lengthcheck} is in effect, log the heig
168 \changes{4.1b}{2008/08/12}{No need to protect against undefined \cs{float@sw}}
169 \changes{4.1b}{2008/08/12}{Patch the array package even later: after all package patches go in.
170 \changes{4.1b}{2008/08/12}{Refine toc processing: provide default.}%
171 \changes{4.1b}{2008/08/12}{Tally and log the height of a float class}
172 \changes{4.1d}{2009/03/27}{(AO, 511) Compatability with lineno.sty's erroneous way of detecting
173 \changes{4.1f}{2009/07/07}{(AO, 515) Hook for setting the font of a footnote}
174 \changes{4.1f}{2009/07/10}{(AO, 518) Tally register overflow when locument is long}
175 \changes{4.1g}{2009/10/06}{(AO, 532) Both arguments of \cs{href} get sanitized}%

```



```

176 \changes{4.1g}{2009/10/07}{(AO, 525) Remove phantom paragraph above display math that is given
177 \changes{4.1g}{2009/10/07}{(AO, 539) Use of double-backslash in argument of \cs{section} gives
178 \changes{4.1n}{2009/12/05}{(AO, 569) Use of \classname{hyperref} interferes with column balanci
179 \changes{4.1n}{2009/12/06}{(AO) Incorporate change to ltmiscen.dtx v1.1i 2000/05/19}%
180 \changes{4.1n}{2009/12/09}{(AO, 569) execute \classname{atveryend}'s \cs{Call@AfterLastShipout}
181 \changes{4.1n}{2009/12/13}{(AO, 574) protect against \classname{lineno.sty}, which forces a vis
182 \changes{4.1n}{2010/01/02}{(AO, 571) Interface \cs{set@footnotewidth} for determining the set w
183 \changes{4.1n}{2010/01/02}{(AO, 571) allow split after last line of footnote}%
184 \changes{4.1n}{2010/01/06}{(AO, 572) title block footnotes numbered independently from body foo
185 \changes{4.1p}{2010/02/24}{(AO, 582) A patch of \classname{hyperref.sty} to provide backward co
186 \changes{4.2a}{2017/11/21}{(MD) Use updated best practice to use https and doi.org}%
187 \changes{4.2a}{2018/12/12}{(MD) Updated name of README file and use standard fonts when typeset
188 \changes{4.2d}{2020/09/19}{(PHO) Adapt \cs{document} and \cs{enddocument} hooks to the 2020-10-
189
190 \end{filecontents*}

```

### 3.3 The Document Body

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

```

191 \begin{document}%
192 \expandafter\DocInput\expandafter{\jobname.dtx}%
193 \end{document}
194 %</doc>

```

## 4 Using this package

Once this package is installed on your filesystem, you can employ it in adding functionality to  $\text{\LaTeX}$  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 at superscript.com) in his SuperScript line of commercial typesetting tools, and which are used here with his permission.

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

```

195 %<*package>
196 \def\package@name{ltxutil}%
197 \expandafter\PackageInfo\expandafter{\package@name}{%
198 Utility macros for \protect\LaTeXe,
199 by A. Ogawa (arthur_ogawa at sbcglobal.net)%
200 }%
201 %</package>

```

## 6.2 Banner and beginning of the kernel DOCSTRIP module

```
202 %<*kernel>
```

## 6.3 Errors and warnings

`\class@err` A few shorthands for Class messages. Your document class should define  
`\class@warn` `\class@name`.

```
\class@info 203 \def\class@err#1{\ClassError{\class@name}{#1}\@eha}%
204 \def\class@warn#1{\ClassWarningNoLine{\class@name}{#1}}%
205 \def\class@info#1{\ClassInfo{\class@name}{#1}}%
206 \def\obsolete@command#1{%
207   \class@warn@end{Command \string#1\space is obsolete.^^JPlease remove from your document}%
208   \global\let#1\@empty
209   #1%
210 }%
211 \def\replace@command#1#2{%
212   \class@warn@end{Command \string#1\space is obsolete;^^JUse \string#2\space instead}%
213   \global\let#1#2%
214   #1%
215 }%
216 \def\replace@environment#1#2{%
217   \class@warn@end{Environment #1 is obsolete;^^JUse #2 instead}%
218   \glet@environment{#1}{#2}%
219   \@nameuse{#1}%
220 }%
221 \def\incompatible@package#1{%
222   \@ifpackageloaded{#1}{%
223     \def\@tempa{I cannot continue. You must remove the \string\usepackage\ statement that caused
224     \ClassError{\class@name}{The #1 package cannot be used with \class@name}%
225     \@tempa\stop
226   }{%
227     \class@info{#1 was not loaded (OK!)}%
228   }%
229 }%
230 \def\class@warn@end#1{%
231   \gapodef\class@enddocumenthook{\class@warn{#1}}%
232 }%
```

Give `\class@name` a meaning if it does not already have one.

```
233 \ifx\undefined\class@name
234   \def\class@name{ltxutil}%
235   \class@warn{You should define the class name before reading in this package. Using default}%
236 \fi
```

## 6.4 New Tools

```
\t@
237 \def\t@{to}%
```

```

\dimen@iii
238 \dimendef\dimen@iii\thr@@

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

\fo@ Analogous to \one, \tw@, and \thr@@.
240 \chardef\fo@=4\relax
241 \chardef\cat@letter=11\relax
242 \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.
243 \def\let@environment#1#2{%
244   \expandafter\let
245   \csname#1\expandafter\endcsname\csname#2\endcsname
246   \expandafter\let
247   \csname end#1\expandafter\endcsname\csname end#2\endcsname
248 }%
249 \def\glet@environment#1#2{%
250   \global\expandafter\let
251   \csname#1\expandafter\endcsname\csname#2\endcsname
252   \global\expandafter\let
253   \csname end#1\expandafter\endcsname\csname end#2\endcsname
254 }%

\tracingplain The command \tracingplain causes TEX'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.
255 \newcommand\tracingplain{%
256   \tracingonline\z@\tracingcommands\z@\tracingstats\z@
257   \tracingpages\z@\tracingoutput\z@\tracinglostchars\@ne
258   \tracingmacros\z@\tracingparagraphs\z@\tracingrestores\z@
259   \showboxbreadth5\showboxdepth3\relax %\errorstopmode
260 }%
261 \newcommand\traceoutput{%
262   \appdef\@resetactivechars{\showoutput}%
263 }%

\say The commands \say and \saythe cause diagnostic messages in the TEX log that
\saythe give the value of a control sequence name or a register respectively.
264 \newcommand\say[1]{\typeout{<\noexpand#1=\meaning#1>}}%
265 \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.
266 \def\fullinterlineskip{\prevdepth\z@}%

```

```

\count@i
\count@ii 267 \countdef\count@i\@ne
           268 \countdef\count@ii\tw@

```

## 6.5 Boolean Control

We introduce just enough of the Boolean calculus for  $\text{\TeX}$ . 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 at 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.

```

```

269 \long\def\prepdef#1#2{%
270   \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
271   \toks@ii{#2}%
272   \edef#1{\the\toks@ii\the\toks@}%
273 }%
274 \long\def\appdef#1#2{%
275   \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
276   \toks@ii{#2}%
277   \edef#1{\the\toks@\the\toks@ii}%
278 }%
279 \long\def\gappdef#1#2{%
280   \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
281   \toks@ii{#2}%
282   \global\edef#1{\the\toks@\the\toks@ii}%
283 }%
284 \long\def\appdef@val#1#2{%
285   \appdef#1{#2}%
286 }%
287 \long\def\appdef@e#1#2{%
288   \expandafter\appdef
289   \expandafter#1%
290   \expandafter{#2}%
291 }%
292 \long\def\appdef@eval#1#2{%
293   \expandafter\appdef@val
294   \expandafter#1%
295   \expandafter{#2}%
296 }%
297 \toksdef\toks@ii=\tw@

```

```

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

```

manufactured by the use of `\csname`.

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

```
298 \long\def\@ifxundefined#1{\@ifx{\undefined#1}}%
299 \long\def\@ifnotrelax#1#2#3{\@ifx{\relax#1}{#3}{#2}}%
300 \long\def\@argswap#1#2{#2#1}%
301 \long\def\@argswap@val#1#2{#2{#1}}%
302 \def\@ifxundefined@cs#1{\expandafter\@ifx\expandafter{\csname#1\endcsname\relax}}%
```

`\rvtx@ifformat@geq` Some changes in the L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> kernel is greater or equal to the argument (the argument should be in the format yyyy-mm-dd).

```
303 \ifx\IfFormatAtLeastTF\undefined
304   \def\rvtx@ifformat@geq{\@ifl@t@r\fmtversion}%
305 \else
306   \let\rvtx@ifformat@geq\IfFormatAtLeastTF
307 \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.

```
308 \def\@boolean#1#2{%
309   \long\def#1{%
310     #2% \if<something>
311     \expandafter\true@sw
312   \else
313     \expandafter\false@sw
314   \fi
315 }%
316 }%
317 \def\@boole@def#1#2{\@boolean{#1}}% Implicit #2
```

`\@booleantrue` The procedures `\@booleantrue` and `\@booleanfalse` are assignment operators for Boolean flags.

```
318 \def\@booleantrue#1{\let#1\true@sw}%
319 \def\@booleanfalse#1{\let#1\false@sw}%
```

`\@ifx` We can now invoke the defining word to create the procedures `\@ifx` and friends.  
`\@ifx@empty` Compatibility Note: earlier versions of this package defined a procedure `\@ifempty`. However, for compatibility with AMSL<sup>A</sup>T<sub>E</sub>X, we must avoid the following three names: `\@ifempty`, `\@xifempty`, and `\@ifnotempty`.

```
\@ifdim 320 \@boole@def\@ifx#1{\ifx#1}%
\@ifeof 321 \@boole@def\@ifx@empty#1{\ifx@empty#1}%
\@ifhbox 322 \@boole@def\@ifempty#1{\if!#1!}%
\@ifhmode 323 %\@boole@def\@ifsw#1{\csname if#1\endcsname}%
\@ifinner 324 \def\@ifsw#1#2{#1\expandafter\true@sw\else\expandafter\false@sw#2}%
\@ifmmode 325 \@boole@def\@ifdim#1{\ifdim#1}%
\@ifnum 326 \@boole@def\@ifeof#1{\ifeof#1}%
```

```
\@ifodd
\@ifvbox
\@ifvmode
\@ifvoid
```

```

327 \@boole@def\@ifhbox#1{\ifhbox#1}%
328 \@boole@def\@ifhmode{\ifhmode}%
329 \@boole@def\@ifinner{\ifinner}%
330 \@boole@def\@ifmmode{\ifmmode}%
331 \@boole@def\@ifnum#1{\ifnum#1}%
332 \@boole@def\@ifodd#1{\ifodd#1}%
333 \@boole@def\@ifvbox#1{\ifvbox#1}%
334 \@boole@def\@ifvmode{\ifvmode}%
335 \@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.

336 \long\def\true@sw#1#2{#1}%
337 \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.

338 \long\def\loopuntil#1{#1}{\loopuntil{#1}}}%
339 \long\def\loopwhile#1{#1{\loopwhile{#1}}{}}}%

\@provide A defining word that refuses to clobber a prior meaning.

340 \def\@provide#1{%
341 \ifx{\undefined#1}{\true@sw}{\ifx{\relax#1}{\true@sw}{\false@sw}}}%
342 {\def#1}{\def\j@nk}%
343 }%
```

## 6.6 Begin Document Structure

The standard  $\text{\LaTeX}$  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,  $\text{\LaTeX}$  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`.

As it turns out, though, it appears feasible to simply invoke the desired column grid command at `\AtBeginDocument` time, since the MVL has nothing in it at that time that would be problematical.

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

The 2020-10-01  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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).

```

344 \rvtx@ifformat@geq{2020-10-01}%
345   {%
346     \AddToHook{begindocument/before}{\document@inithook}%
347   }{%
348     \prepdef\document{%
349       \endgroup
350       \document@inithook
351       \true@sw}%
352   }%
353 }
```

`\document@inithook` To use, simply `\appdef\document@inithook{your tokens here}`.

```

354 \let\document@inithook\@empty
```

`\class@documenthook` We install the last `\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` instead of `\AtBeginDocument`, and `\appdef \class@enddocumenthook` instead of `\AtEndDocument`.

```

355 \appdef\document@inithook{%
356   \AtBeginDocument{\class@documenthook}%
357 }%
358 \AtEndDocument{%
359   \class@enddocumenthook
360 }%
361 \let\class@documenthook\@empty
362 \let\class@enddocumenthook\@empty
```

`\enddocument` The standard L<sup>A</sup>T<sub>E</sub>X `\end{document}` processing is a potential problem, particularly when the output routine has been changed by `ltxgrid`. We separate out the  
`\check@aux` procedure that checks the auxiliary file at the end of the job so that later it can  
`\do@check@aux` 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<sup>A</sup>T<sub>E</sub>X kernel:

```

363 \rvtx@ifformat@geq{2020-10-01}{%
364   % <definitions for newer LaTeX later>
365 }{%
366   % <definitions for older LaTeX>
367 \def\enddocument{%
```

The following line from `ltxutil.dtxlmiscen.dtx` ‘resets `\AtEndDocument` for latex/3060’.

```

368 \let\AtEndDocument\@firstofone
```



```

369 \@enddocumenthook
370 \@checkend{document}%

```

The `\clear@document` statement ends the current page (we must guarantee no further shipouts), then executes all cleanup procedures that must occur only after the last shipout. Clients will queue up their procedures via `\AfterLastShipout`, if it exists, otherwise by doing `\appdef\clear@document`.

```

371 \clear@document

```

We are very close to ending the T<sub>E</sub>X run, now.

```

372 \check@aux
373 \deadcycles\z@
374 \@@end
375 }%
376 \def\check@aux{\do@check@aux}%
377 \def\do@check@aux{%
378   \if@sw\if@files\fi{%
379     \immediate\closeout\@mainaux
380     \let\@setckpt\@gobbletwo
381     \let\@newl@bel\@testdef
382     \@tempswafalse
383     \makeatletter
384     \input\jobname.aux\relax
385   }{}%
386   \@dofilelist
387   \ifdim\font@submax >\fontsubfuzz\relax}{%
388     \font@warning{%
389       Size substitutions with differences\MessageBreak
390       up to \font@submax\space have occurred.\@gobbletwo
391     }%
392   }{}%
393   \@defaultsubs
394   \@refundefined
395   \if@sw\if@files\fi{%
396     \ifx{\@multiplelabels\relax}{%
397       \if@sw\if@tempswa\fi{%
398         \latex@warning@no@line{%
399           Label(s) may have changed.
400           Rerun to get cross-references right%
401         }%
402       }{}%
403     }{%
404       \@multiplelabels
405     }%
406   }{}%
407 }%
408 }

```

`\rvtx@enddocument@patch` For newer L<sup>A</sup>T<sub>E</sub>X we'll try to be a bit more future-proof (no miracle though). The code for `\enddocument` (in pre-2020-10-01 L<sup>A</sup>T<sub>E</sub>X) 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:

```
409 \rvtx@ifformat@geq{2020-10-01}{%
410   \AddToHook{enddocument}{\rvtx@enddocument@patch{}}%
411 }{}
```

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:

```
412 \protected\long\def\rvtx@enddocument@patch#1#2\@checkend#3{%
413   \begingroup
414     \edef\x{\detokenize{#3}}%
415     \edef\y{\detokenize{document}}%
416     \expandafter\endgroup
417     \ifx\x\y
418       \expandafter\rvtx@enddocument@patch@end
419     \else
420       \expandafter\rvtx@enddocument@patch@more
421     \fi
422     {#1#2}{#3}}
423 \def\rvtx@enddocument@patch@more#1#2{%
424   \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`:

```
425 \long\def\rvtx@enddocument@patch@end#1#2\clearpage#3\endgroup{%
426   \def\do@check@aux{#3\endgroup}%
427 }
```

Then execute the code consumed in the previous step:

```
427 #1%
428 \@checkend{#2}%
429 }
```

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

```
429 \clear@document
430 \check@aux}
431 \def\check@aux{\do@check@aux}%
```

`\clear@document` The procedure `\clear@document` is responsible for flushing out the last page of the document, if not already done. The procedure then executes those procedures that must wait for execution until after the last page is shipped out. Clients of `ltxutil`, such as `ltxgrid` and `revtex4` will queue these procedures up via `\AfterLastShipout`, if it exists, otherwise by doing `\appdef\clear@document`.

The command `\Call@AfterLastShipout` is provided by Heiko Oberdiek’s `atveryend` package. This package is compatible with `ltxutil`.

Note on compatibility with `atveryend`: we arrange for `\Call@AfterLastShipout` to be called from the safety of the output routine, thereby ensuring that all of the procedures queued up by that package’s `\AfterLastShipout` are executed at the right time. We also ensure that `\Call@AfterLastShipout` has a default definition, in case the package was never loaded.

```
432 \def\clear@document{%
433 \clearpage
434 \do@output@cc1v{%
435 \Call@AfterLastShipout
436 }%
437 }%
438 \appdef\class@documenthook{%
439 \providecommand\Call@AfterLastShipout{}%
440 }%
```

## 6.7 Class Extensions

The  $\text{\LaTeX}$  procedure `\@onefilewithoptions` is the vehicle for reading in a  $\text{\LaTeX}$  class or package. The APS RevTeX class implements the use of what are called “substyles”, actually extensions to the class itself. Any document class can do likewise.

`\class@extension` A procedure similar to  $\text{\LaTeX}$ ’s `\@onefilewithoptions`, but as an extension to  
`\class@extensionfile` the current document class.

`\class@ext@hook` Read in the given file as if it were a document class file. Usage: `\class@extensionfile {<class>} \@extension`, where `<class>` is a file (similar to `aps.rtx`) and where `\@extension` is the file extension for `<class>`. For instance, to read in the file `aps.rtx`, do `\class@extensionfile {aps} \substyle@ext`, where the latter has been define to expand to `.rtx`.

Features supported include passing existing class options on to the class extension, `\AtEndOfClass` processing, a stack that restores `\@currname`, `\@currentx`, `\@clsextension`, and the `\catcode` of ‘@’, fall-back to a control sequence name (with leading ‘`rtx@`’) if no file exists.

Note that `\LoadClass` gives one the ability to write a class that calls in another class as a (sort of) module: this scheme is like `\LoadClass`, but turned inside out.

```

441 \def\class@extension#1#2{%
442 \IfFileExists{#1.#2}{%
443 \expandafter\class@extensionfile\csname ver@\@currname.\@current\endcsname{#1}#2%
444 }{%
445 \csname rtx@#1\endcsname
446 }%
447 }%
448 \def\class@extensionfile#1#2#3{%
449 \@pass@options#3\@unusedoptionlist{#2}%
450 \global\let\@unusedoptionlist\@empty
451 \expandafter\class@ext@hook\csname#2.#3-h@k\endcsname#1{#2}#3%
452 }%
453 \def\class@ext@hook#1#2#3#4{%
454 \@pushfilename@ltx
455 \makeatletter
456 \let\CurrentOption\@empty
457 \@reset@options
458 \let#1\@empty
459 \xdef\@currname{#3}%
460 \global\let\@current#4%
461 \global\let\@clsextension\@current
462 \input{#3.#4}%
463 \@ifl@ter#4{#3}#2{%
464 \class@info{Class extension later than: #2}%
465 }{%
466 \class@info{Class extension earlier: #2}%
467 \@@end
468 }%
469 #1%
470 \let#1\@undefined
471 \expandafter\@p@pfilename@ltx\@currnamestack@ltx\@nil
472 \@reset@options
473 }%

```

`\@pushfilename` But! L<sup>A</sup>T<sub>E</sub>X does not provide for a class extension other than `.cls`, there-  
`\@p@pfilename` fore we must extend L<sup>A</sup>T<sub>E</sub>X's file name stack with the file extension of a class extension. This way, procedures like `\ProvidesPackage`, `\OptionNotUsed`, `\ProcessOptions`, `\@reset@options` will still work properly.

```

474 \def\@pushfilename@ltx{%
475 \xdef\@currnamestack@ltx{%
476 {\@currname}%
477 {\@current}%
478 {\@clsextension}%
479 {\the\catcode'\@}%
480 \@currnamestack@ltx
481 }%
482 }%

```

```

483 \def\p@pfilename@ltx#1#2#3#4#5\@nil{%
484 \gdef\@currname{#1}%
485 \gdef\@current{#2}%
486 \gdef\@clsextension{#3}%
487 \catcode'\@#4\relax
488 \gdef\@currnamestack@ltx{#5}%
489 }%
490 \global\let\@currnamestack@ltx\@empty

```

We carefully patch L<sup>A</sup>T<sub>E</sub>X so that the current value of `\@clsextension` can be restored after reading in a class file.

## 6.8 Type Tools

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

```

491 \def\flushing{%
492 \let\\@normalcr
493 \leftskip\z@skip
494 \rightskip\z@skip
495 \@rightskip\z@skip
496 \parfillskip\@flushglue
497 }%

```

`\@centercr` The `\@centercr` command is the replacement for `\@normalcr` when setting type centered or ragged. Normally, the meaning of `\\` is `\@normalcr`, which L<sup>A</sup>T<sub>E</sub>X defines via `\DeclareRobustCommand`. In centered or ragged typesetting, the meaning of `\\` is `\@centercr`, therefore it ought to be defined via `\DeclareRobustCommand` (but unfortunately is not). The fact that it is not is yet another of L<sup>A</sup>T<sub>E</sub>X's early failures that will never get fixed.

The following exemplar fails under L<sup>A</sup>T<sub>E</sub>X version 2005/12/01, package `textcase` 2004/10/07 v0.07:

```

%\documentclass{article}%
%\usepackage[overload]{textcase}
%\begin{document}
%\centering
%\section{\MakeTextUppercase{Section\\title}}
%Text
%\end{document}
%

```

The solution is to promote `\@centercr` to a robust command, just the same as `\\`. We do that here without needing to know the meaning of the command.

```

498 \expandafter\DeclareRobustCommand\expandafter\@centercr\expandafter{\@centercr}%

```

## 6.9 Display Math

`\eqnarray@LaTeX` Team L<sup>A</sup>T<sub>E</sub>X has stated they will never repair Leslie's broken definition of `\eqnarray`. Let us be bold...

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

```

499 \def\eqnarray@LaTeX{%
500   \stepcounter{equation}%
501   \def\@currentlabel{\p@equation\theequation}%
502   \global\@eqnswtrue
503   \m@th
504   \global\@eqcnt\z@
505   \tabskip\@centering
506   \let\\\@eqncr
507   $$\everycr{}\halign to\displaywidth\bgroup
508     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
509     &\global\@eqcnt\@ne\hskip \tw@\arraycolsep \hfil${##}$\hfil
510     &\global\@eqcnt\tw@ \hskip \tw@\arraycolsep
511     $\displaystyle{##}$\hfil\tabskip\@centering
512     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
513     \tabskip\z@skip
514   \cr
515 }
516 \long\def\eqnarray@fleqn@fixed{%
517   \stepcounter{equation}\def\@currentlabel{\p@equation\theequation}%
518   \global\@eqnswtrue\m@th\global\@eqcnt\z@
519   \tabskip\ltx@mathindent
520   \let\=\@eqncr
521   \setlength\abovedisplayskip{\topsep}%
522   \ifvmode\addtolength\abovedisplayskip{\partopsep}\fi
523   \addtolength\abovedisplayskip{\parskip}%
524   \setlength\belowdisplayskip{\abovedisplayskip}%
525   \setlength\belowdisplayshortskip{\abovedisplayskip}%
526   \setlength\abovedisplayshortskip{\abovedisplayskip}%
527   $$$
528   \everycr{}%
529   \halign to\linewidth\bgroup
530     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
531     &\global\@eqcnt\@ne
532     \hskip\tw@\eqncolsep
533     \hfil${}\##{}$}\hfil
534     &\global\@eqcnt\tw@
535     \hskip\tw@\eqncolsep
536     $\displaystyle{##}$\hfil\tabskip\@centering
537     &\global\@eqcnt\thr@@\hb@xt@\z@\bgroup\hss##\egroup
538     \tabskip\z@skip
539   \cr
540 }%
```

```

541 \@ifx{\eqnarray\eqnarray@LaTeX}{%
542 \class@info{Repairing broken LaTeX eqnarray}%
543 \let\eqnarray\eqnarray@fleqn@fixed
544 \newlength\eqncolsep
545 \setlength\eqncolsep\z@
546 \let\eqnarray@LaTeX\relax
547 \let\eqnarray@fleqn@fixed\relax
548 }{}%

```

The macro `\ltx@mathindent` is assigned to the `\tabskip` glue just before the alignment preamble is expanded, the value therefore applying at the left of the first column.

The below value specifies the display math to be set centered, as is common practice. Alternatively, `\tabskip` can be set to a different glue value, accomplishing flush-left display math.

Note that the `ltxutil.dtxfleqn.clo` package provides its own meaning for the `eqnarray` environment, which is also broken. We do not patch that package, however.

Bug note: The `ltxutil.dtxlineno.sty` package detects `ltxutil.dtxfleqn.clo` by testing whether `\mathindent` is defined, instead of using correct  $\text{\LaTeX 2}_{\epsilon}$  means. Even though our `eqnarray` environment is modelled after `ltxutil.dtxfleqn.clo`, we must program defensively here.

```

549 \def\ltx@mathindent{\@centering}%
550 \def\set@eqnarray@skips{}%

```

`\prep@math` If we are in vertical mode when display math mode is entered (via `$$`),  $\text{\TeX}$  will first enter horizontal mode, then display math mode; this results in a phantom paragraph containing a single `\hbox` consisting of the `\parindent` box followed by the `\parskipfillskip` glue. Of course, that `\hbox` is accompanied by `\parskip` glue and `\baselineskip` glue.

The `\prep@math` procedure removes the `\parindent` box, thereby (magically) eliminating the phantom paragraph. The `\prep@math@patch` procedure head-patches the `equation` and `eqnarray` environments to accomplish this removal of the phantom paragraph.

Note that there are three remaining ways to enter display math mode that we do not treat: the `displaymath` environment (equivalent to `\[/\]`), and the primitive the `$$` markup. I refrain from treating the first case because `displaymath` already detects the case where it is entered from vertical mode: I do not wish to engage in the dubious enterprise of attempting to correct a procedure that is ill conceived from the outset. As to the primitive `$$`, there is no help for users who insist upon employing procedural markup in their documents. in their documents.

```

551 \def\prep@math{%
552 \@ifvmode{\everypar{\setbox\z@\lastbox}}{}%
553 }%
554 \def\prep@math@patch{%
555 \prepdef\equation{\prep@math}%
556 \prepdef\eqnarray{\prep@math}%
557 }%

```

A document class may invoke `\prep@math@patch` at any point it wishes to prevent the appearance of the phantom paragraph: it may be a global declaration or a local one.

We fail to patch `\[, \equation`, however.

## 6.10 Footnotes

<code>\footnotemark</code> <code>\footnotetest</code> <code>\ltx@xfootnote</code> <code>\ltx@footmark</code> <code>\ltx@foottext</code> <code>\ltx@make@current@footnote</code>	<p>We repair an error in the L<sup>A</sup>T<sub>E</sub>X kernel (see <code>ltfloat.dtx</code>) involving footnotes. The symptom is that the <code>\footnotemark</code> command does not work properly within a <code>minipage</code> environment. The source of the problem is in the way the <code>\footnotemark</code> and <code>\@xfootnotemark</code> procedures are defined: they do not share the method, used by <code>\footnote</code> and other procedures, that allows a context switch to change the way footnotes behave within a <code>minipage</code> environment. This is a L<sup>A</sup>T<sub>E</sub>X bug of long standing; our fix dates to 1987.</p>
--	---

While we are at it, we rewrite both the `\footnote`, `\footnotemark` and `\footnotetest` procedures, achieving a cleaner separation of syntax and semantics. Note that the `\@footnotetest` procedure is not involved in context switching; `hyperref` will take over that procedure, substituting its own processing around its argument and passing this to `\H@@footnotetest`. We anticipate this, and do our context switching on `\H@@footnotetest`.

The `\@makefnmark` continues as the method of formatting the footnote mark.

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\@footnotetest\@mpfootnotetest
%\let\@makefnmark\@mpmakefnmark
%\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.

Note that L<sup>A</sup>T<sub>E</sub>X initially defines `\@mpfn` as `footnote` and `\thempfn` as `\thefootnote`, so we are initially doing general footnotes.

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

Three user-level command, `\footnote`, `\footnotemark`, and `\footnotetest` are defined (see the L<sup>A</sup>T<sub>E</sub>X manual for user-level details).

<code>\footnote</code>	<p>The first user-level command is <code>\footnote</code>. A simple way to look at this command is to think of it as <code>\footnotemark [<i>&lt;number&gt;</i>] \footnotetest [<i>&lt;number&gt;</i>] {<i>&lt;text&gt;</i>}</code>, where the optional argument is the same in both calls. We also define a syntactical helper procedure <code>\ltx@xfootnote</code>.</p>
------------------------	--



We employ the procedures `\ltx@stp@footproc` and `\ltx@def@footproc`, passing in the procedure to execute, in this case `\ltx@footmark`, which sets the footnote mark. In any case, we end on the procedure `\ltx@foottext`, which sets the footnote text.

```

558 \def\footnote{\@ifnextchar[\ltx@xfootnote\ltx@yfootnote}%
559 \def\ltx@xfootnote[#1]{%
560   \ltx@def@footproc\ltx@footmark[#1]%
561   \expandafter\ltx@foottext\expandafter{\the\csname c@\mpfn\endcsname}%
562 }%
563 \def\ltx@yfootnote{%
564   \ltx@stp@footproc\ltx@footmark
565   \expandafter\ltx@foottext\expandafter{\the\csname c@\mpfn\endcsname}%
566 }%

```

The `\footmark` user-level command is next. Here we use the procedures `\ltx@stp@footproc` and `\ltx@def@footproc` again, but unlike `\footnote`, we do not set the footnote text.

```

567 \def\footnotemark{\@ifnextchar[\ltx@xfootmark\ltx@yfootmark}%
568 \def\ltx@xfootmark[\ltx@def@footproc\ltx@footmark]%
569 \def\ltx@yfootmark{\ltx@stp@footproc\ltx@footmark}%
570 \def\ltx@footmark#1{%
571   \leavevmode
572   \ifhmode\edef\@xsf{\the\spacefactor}\nobreak\fi
573   \begingroup
574     \expandafter\ltx@make@current@footnote\expandafter{\mpfn}{#1}%
575     \expandafter\@argswap@val\expandafter{\Hy@footnote@currentHref}{\hyper@linkstart {link}}%
576     \@makefnmark
577     \hyper@linkend
578   \endgroup
579   \ifhmode\spacefactor\@xsf\fi
580   \relax
581 }%

```

The third user-level command is `\footnotetext`. As with `\footnotemark`, we use the procedures `\ltx@stp@footproc` and `\ltx@def@footproc`, this time passing in the procedure `\ltx@foottext`, which sets the footnote text.

```

582 \def\footnotetext{\@ifnextchar[\ltx@xfoottext\ltx@yfoottext}%
583 \def\ltx@xfoottext{\ltx@def@footproc\ltx@foottext}%
584 \def\ltx@yfoottext{\ltx@stp@footproc\ltx@foottext}%
585 \long\def\ltx@foottext#1#2{%
586   \begingroup
587     \expandafter\ltx@make@current@footnote\expandafter{\mpfn}{#1}%
588     \@footnotetext{#2}%
589   \endgroup
590 }%

```

Here are the definitions of the procedures `\ltx@stp@footproc` and `\ltx@def@footproc`. The `require` argument is the procedure to execute afterwards, and `\ltx@def@footproc` parses a bracket-delimited argument (it is not optional). In each case the given

procedure is executed with an argument prepared for it: the value of the footnote counter.

```

591 \def\ltx@def@footproc#1[#2]{%
592 \begingroup
593   \csname c@\mpfn@endcsname #2\relax
594   \unrestored@protected@xdef\the#fnmark{\thempfn}%
595 \expandafter\endgroup
596 \expandafter#1%
597 \expandafter{\the\csname c@\mpfn@endcsname}%
598 }%
599 \def\ltx@stp@footproc#1{%
600 \expandafter\stepcounter\expandafter{\@mpfn}%
601 \protected@xdef\the#fnmark{\thempfn}%
602 \expandafter#1%
603 \expandafter{\the\csname c@\mpfn@endcsname}%
604 }%

```

Here we provide for our good friend `hyperref` to enter in like a bull in a china shop. If it is not loaded, we do what it would have done, but gentlier and without hypertext functionality.

```

605 \appdef\class@documenthook{%
606 \let\footnote@latex\footnote
607 \@ifpackageloaded{hyperref}{}{}%
608 \let\H@@footnotetext\@footnotetext
609 \def\@footnotetext{\H@@footnotetext}%
610 \let\H@@mpfootnotetext\@mpfootnotetext
611 \def\@mpfootnotetext{\H@@mpfootnotetext}%
612 }%
613 }%

```

In the following, we must use L<sup>A</sup>T<sub>E</sub>X’s rococco equipment in the form of `\protected@edef`, because of the presence of a font switch in the meaning of `\thempfootnote`. But, really, isn’t this a sloppy conflation of semantics and presentation?

```

614 \def\ltx@make@current@footnote#1#2{%
615   \csname c@#1\endcsname#2\relax
616   \protected@edef\Hy@footnote@currentHref{\@currentHref-#1.\csname the#1\endcsname}%
617 }%
618 \def\thempfootnote@latex{\itshape \@alpha \c@mpfootnote }%
619 \def\ltx@thempfootnote{\@alpha \c@mpfootnote}%
620 \@ifx{\thempfootnote\thempfootnote@latex}{}%
621 \class@info{Repairing hyperref-unfriendly LaTeX definition of \string\mpfootnote}%
622 \let\thempfootnote\ltx@thempfootnote
623 }{}%

```

Note on `hyperref` compatibility: In its “Automated L<sup>A</sup>T<sub>E</sub>X hypertext cross-references”, the `hyperref` package alters footnote processing, but it does nothing to address the several issues of concern to us.

The `hyperref` package takes over the `\@mpfootnotetext` and `\@footnotetext` procedures, wrapping the argument in its own code. It also rewrites `\@footnotemark`,

making it a hyperlink, and `\@xfootnotenext`, removing from it all hypertext capabilities.

However, if the `\footnotemark` command has been supplied with an optional argument, `hyperref`'s changes do not apply: it punts in this case.

At the same time, it attempts to turn off its changes during `\maketitle` processing, destroying one of the capabilities we desire.

We make ourself `hyperref` savvy: we re-implement footnote processing, using `hyperref` capabilities if that package has been loaded.

Any other package that rewrites L<sup>A</sup>T<sub>E</sub>X's footnote macros will be incompatible with this package.

Two thoughts about `hyperref`: what for does it define `\realfootnote`? Apparently even SR himself cannot remember.

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<sup>A</sup>T<sub>E</sub>X 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!

<code>\@footnotetext</code> <code>\@mpfootnotetext</code> <code>\@tpfootnotetext</code> <code>\make@footnotetext</code> <code>\set@footnotewidth</code>	<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> <p>Difference from L<sup>A</sup>T<sub>E</sub>X: here we do not set <code>\floatingpenalty</code> to infinity. Doing this must date back to a time when L<sup>A</sup>T<sub>E</sub>X could not accomodate split insertions (footnotes). I cannot think of any other reason to do have done this. At any rate, with the <code>ltxgrid</code> package, split insertions are properly taken care of, so we allow it.</p>
---	---

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.

If we are setting type on multiple page grids, we must still ensure that all footnotes that find their way into the `\footins` insert register are set on the same width. This implies the need for a document to have an "overall" page grid, which determines the set width of all footnotes with the exception of minipage footnotes.

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.

```

624 \def\@makefnmark{%
625   \hbox{%
626     \@textsuperscript{%

```

```

627   \normalfont\itshape\@thefnmark
628 }%
629 }%
630 }%

631 \long\def\@footnotetext{%
632   \insert\footins\bgroup
633   \make@footnotetext
634 }%

635 \long\def\@mpfootnotetext{%
636   \minipagefootnote@pick
637   \make@footnotetext
638 }%

```

Procedure `\make@footnotetext` sets the footnote #1 into type, with the proper font, color, leading, width, and label in effect. It also establishes a strut and null glue at the end of the last paragraph of the footnote; The strut helps compensate for the lack of `\interlineskip` glue between `\inserts`; the glue establishes a feasible `\vsplit` point between footnotes.

Note that in the title block (`ltxfront`), the alternative definition, under the name `\frontmatter@footnotetext`, is used. The only material difference there is the reference to `\frontmatter@makefnmark` instead of `\@makefnmark`.

Dependency note: the `\@makefnmark` procedure is used to further process the footnote text and to execute the `\@makefnmark` procedure to produce the footnote mark. The definition of the former is customarily found in the document class (hereunder that of `ltxutil.dtxarticle.cls`), the latter in `ltxutil.dtxlatex.ltx`. They are as follows:

```

%\newcommand\@makefnmark[1]{%
% \parindent 1em\noindent
% \hb@xt@1.8em{\hss\@makefnmark}%
% #1%
%}%
%\def\@makefnmark{%
% \hbox{\@textsuperscript{\normalfont\@thefnmark}}%
%}%
%

```

```

639 \long\def\make@footnotetext#1{%
640   \set@footnotefont

```

As noted above, we do *not* do `\floatingpenalty \@MM`, as in standard L<sup>A</sup>T<sub>E</sub>X.

```

641   \set@footnotewidth
642   \@parboxrestore
643   \protected@edef\@currentlabel{%

```

Note that we employ `\@mpfn` as a level of redirection for the footnotecounter.

```

644   \csname p@\@mpfn@endcsname\@thefnmark
645 }%
646   \color@begingroup

```

```

647 \makeifntext{%
648 \rule{z@footnotesep}{\ignorespaces#1%

```

The following strut and glue are for spacing and splitting, as mentioned above.

```

649 \finalstrut\strutbox\vadjust{\vskip{z@skip}}%
650 }%
651 \color@endgroup
652 \minipagefootnote@drop
653 }%

```

`\set@footnotefont` is the procedure for setting the font of a footnote. Other aspects of the environment may be set using this hook.

```

654 \def\set@footnotefont{%
655 \reset@font\footnotesize
656 \interlinepenalty\interfootnotelinepenalty
657 \splittopskip\footnotesep
658 \splitmaxdepth\dp\strutbox
659 }%

```

`\set@footnotewidth` is the procedure for setting the width of a footnote. The default page grid, a single, full-width column, sets footnotes on the width of the text.

```

660 \def\set@footnotewidth{\set@footnotewidth@one}%

```

## 6.11 Floats

### 6.11.1 Usage notes

We extend the L<sup>A</sup>T<sub>E</sub>X 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 unfloats. 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.

```
\appdef\class@documenthook{\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.

### 6.11.2 Robustifying fragile commands

Certain of L<sup>A</sup>T<sub>E</sub>X'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
\robustify@contents 661 \def\robustify@contents{%
662 \let \label \@gobble
663 \let \index \@gobble
664 \let \glossary \@gobble
665 \let \footnote \@gobble
666 \def\({\string\}%
667 \def\){\string\}}%
668 \def\{\{\string\}%
669 }%
670 \long\def\addtocontents#1#2{%
671 \protected@write\@auxout{\robustify@contents}{\string \@writefile {#1}{#2}}%
672 }%
```

### 6.11.3 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<sup>A</sup>T<sub>E</sub>X. 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.

Finally, we wrap L<sup>A</sup>T<sub>E</sub>X's `\contentsline` command with code to detect the case where the expected procedure is not defined, and we give it a syntax with no semantics.

We switch over to this new definition only after `hyperref` has loaded.

```

673 \def\addcontentsline#1#2#3{%
674   \addtocontents{#1}{%
675     \protect\contentsline{#2}{#3}{\thepage}{}%
676   }%
677 }%
678 \def\label#1{%
679   \@bsphack
680   \protected@write\@auxout{}{%
681     \string\newlabel{#1}{\@currentlabel}{\thepage}{\@currentlabel}{\@currentlabel}}%
682   }%
683   \@esphack
684 }%
685 \def\ltx@contentsline#1{%
686   \expandafter\@ifnotrelax\csname l@#1\endcsname{}{%
687     \expandafter\let\csname l@#1\endcsname\@gobbletwo
688   }%
689   \contentsline@latex{#1}%
690 }%
691 \appdef\document@inithook{%
692   \let\contentsline@latex\contentsline
693   \let\contentsline\ltx@contentsline
694 }%
```

### 6.11.4 Footnotes within floats, unfloating floats, float font

`\caption` DPC: Er a bit of a hack, but seems best way of supporting normal L<sup>A</sup>T<sub>E</sub>X syntax at this point: If a caption is used below a table, then put out the footnotes before the caption.

```

695 \appdef\class@documenthook{%
696   \prepdef\caption{\minipagefootnote@here}%
697 }%
```

Note on `hyperref` compatibility: this change to the `\caption` command is compatible with the “Automated L<sup>A</sup>T<sub>E</sub>X 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 698 \def\minipagefootnote@init{%
\minipagefootnote@drop 699 \setbox\@mpfootins\box\voidb@x
700 }%
701 \def\minipagefootnote@pick{%
702 \global\setbox\@mpfootins\vbox\bgroup
703 \unvbox\@mpfootins
704 }%
705 \def\minipagefootnote@drop{%
706 \egroup
707 }%
708 \def\minipagefootnote@here{%
709 \par
710 \@ifvoid\@mpfootins{}{%
711 \vskip\skip\@mpfootins
712 \fullinterlineskip
713 \@ifinner{%
714 \vtop{\unvcopy\@mpfootins}%
715 {\setbox\z@\lastbox}%
716 }{}%
717 \unvbox\@mpfootins
718 }%
719 }%
720 \def\minipagefootnote@foot{%
721 \@ifvoid\@mpfootins{}{%
722 \insert\footins\bgroup\unvbox\@mpfootins\egroup
723 }%
724 }%
725 \def\endminipage{%
726 \par
727 \unskip
728 \minipagefootnote@here
729 \@minipagefalse %% added 24 May 89

```



```

730 \color@endgroup
731 \egroup
732 \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}%
733 }%

```

`\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 assignment of this Boolean is to be overridden by the document class in response to user-selected options.

```

734 \@booleantrue\floats@sw

```

`\@xfloat` The float start-code is redefined to set up footnotes in the style of minipage. Also, `\@mpmakefntext` 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.

```

735 \let\@xfloat@LaTeX\@xfloat
736 \def\@xfloat#1[#2]{%
737   \@xfloat@prep
738   \@nameuse{fp@proc#2}%
739   \floats@sw{\@xfloat@LaTeX{#1}[#2]}\@xfloat@anchored{#1}[]}%
740 }%
741 \def\@xfloat@prep{%
742   \ltx@footnote@pop
743   \def\@mpfn{mpfootnote}%
744   \def\thempfn{\thempfootnote}%
745   \c@mpfootnote\z@
746   \let\H@@footnotetext\H@@mpfootnotetext
747 }%
748 \let\ltx@footnote@pop\@empty
749 \def\@xfloat@anchored#1[#2]{%
750   \def\@capytype{#1}%
751   \begin@float@pagebreak
752   \let\end@float\end@float@anchored
753   \let\end@dblfloat\end@float@anchored
754     \hsize\columnwidth
755     \@parboxrestore
756     \@floatboxreset
757   \minipagefootnote@init
758 }%
759 \def\end@float@anchored{%
760   \minipagefootnote@here
761   \par\vskip\z@skip
762   \par
763   \end@float@pagebreak
764 }%
765 \def\begin@float@pagebreak{\par\addvspace\intextsep}%
766 \def\end@float@pagebreak{\par\addvspace\intextsep}%
767 \def\@mpmakefntext#1{%
768   \parindent=1em

```

```

769 \noindent
770 \hb@xt@1em{\hss\@makefnmark}%
771 #1%
772 }%

```

### 6.11.5 Writing floats out to a file

`\do@if@floats` The procedure `\do@if@floats` should be executed at `\class@documenthook` time: it 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.

```

773 \def\do@if@floats#1#2{%
774 \floats@sw}{}%

```

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

```

775 \expandafter\newwrite
776 \csname#1write\endcsname
777 \expandafter\def
778 \csname#1@stream\endcsname{\jobname#2}%
779 \expandafter\immediate
780 \expandafter\openout
781 \csname#1write\endcsname
782 \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`.

```

783 \ifxundefined\@float@LaTeX{%
784 \let\@float@LaTeX\@float
785 \let\@dblfloat@LaTeX\@dblfloat
786 \let\@float\write@float
787 \let\@dblfloat\write@floats
788 }{%
789 \let@environment{#1@float}{#1}%
790 \let@environment{#1@floats}{#1*}%
791 \ifxundefined@cs{#1@write}{}%
792 \let@environment{#1}{#1@write}%
793 }%
794 }%
795 }%

```

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

Here, we make use of the `\floats@sw` Boolean to select the non-floating type of processing.

```

796 \def\triggerpar{\leavevmode\@par}%
797 \def\oneapage{\def\begin@float@pagebreak{\newpage}\def\end@float@pagebreak{\newpage}}%
798 \def\print@float#1#2{%

```

```

799 \lengthcheck@sw{%
800 \total@float{#1}%
801 }{%
802 \ifxundefined@cs{#1write}{}{%
803 \begingroup
804 \@booleanfalse\floats@sw
805 #2%
806 \raggedbottom
807 \def\array@default{v}% floats must
808 \let\@float\@float@LaTeX
809 \let\@dblfloat\@dblfloat@LaTeX
810 \let\trigger@float@par\triggerpar
811 \let@environment{#1}{#1@float}%
812 \let@environment{#1*}{#1@floats}%
813 \expandafter\prepdef\csname#1\endcsname{\trigger@float@par}%
814 \expandafter\prepdef\csname#1*\endcsname{\trigger@float@par}%
815 \@namedef{fps@#1}{h!}%
816 \expandafter\immediate
817 \expandafter\closeout
818 \csname#1write\endcsname
819 \everypar{%
820 \global\let\trigger@float@par\relax
821 \global\everypar{}\setbox\z@\lastbox
822 \ifxundefined@cs{#1sname}{}{%
823 \begin@float@pagebreak
824 \expandafter\section
825 \expandafter*%
826 \expandafter{%
827 \csname#1sname\endcsname
828 }%
829 }%
830 }%
831 \input{\csname#1@stream\endcsname}%
832 \endgroup
833 \global\expandafter\let\csname#1write\endcsname\relax
834 }%
835 }%

```

`\tally@float` If we are tallying column inches, `\tally@float` tallies a contribution to `\ftype@`  
`\total@float` `\@capytype`, depending upon the width of `\@currbox`. In effect, each float class is  
tallied in two sections, one for narrow, one for wide floats.

If statistics are wanted, `\total@float` logs the tally for the given float class.  
The quantity `\@twopowerfourteen` is  $2^{14}$ , `\@twopowertwo` is  $2^2$ .

```

836 \chardef\@xvi=16\relax
837 \mathchardef\@twopowerfourteen="4000
838 \mathchardef\@twopowertwo="4
839 \def\tally@float#1{%
840 \begingroup

```

We strip all but the least significant 5 bits from `\count \@currbox`, and put them

into \@tempcnta. We then subtract 16 from \count \@currbox (unless this would make it negative), effectively reversing the process carried out in \@float.

```

841 \count\@tempcnta\count\@currbox
842 \divide\@tempcnta\@xxxii
843 \multiply\@tempcnta\@xxxii
844 \advance\count\@currbox-\@tempcnta
845 \divide\@tempcnta\@xxxii
846 \ifnum{\count\@currbox>\@xvi}{%
847 \advance\count\@currbox-\@xvi\@booleantrue\@temp@sw
848 }{%
849 \@booleanfalse\@temp@sw
850 }%

```

If so desired, we log the characteristics of this float object: float class and float placement parameters, height, depth, and width.

```

851 \show@box@size@sw{%
852 \class@info{Float #1
853 (\the\@tempcnta)[\@temp@sw{16+}{}\the\count\@currbox]^^J%
854 (\the\ht\@currbox+\the\dp\@currbox)X\the\wd\@currbox
855 }%
856 }{}%
857 \endgroup

```

Here we tally the height of this float object.

```

858 \expandafter\let
859 \expandafter\@tempa
860 \csname fbox@\csname ftype@#1\endcsname\endcsname
861 \ifnotrelax\@tempa{%
862 \ifhbox\@tempa{%
863 \setbox\@tempboxa\vbox{\unvcopy\@currbox\hrule}%
864 \dimen@ht\@tempboxa
865 \divide\dimen@\@twopowerfourteen
866 \ifdim{\wd\@tempboxa<\textwidth}{%
867 \advance\dimen@ht\@tempa
868 \global\ht\@tempa\dimen@
869 }{%
870 \advance\dimen@dp\@tempa
871 \global\dp\@tempa\dimen@
872 }%
873 }{}%
874 }{}%
875 }%
876 \def\total@float#1{%
877 \expandafter\let
878 \expandafter\@tempa
879 \csname fbox@\csname ftype@#1\endcsname\endcsname
880 \ifnotrelax\@tempa{%
881 \ifhbox\@tempa{%
882 \@tempdima\the\ht\@tempa\divide\@tempdima\@twopowertwo\@tempcnta\@tempdima
883 \@tempdimb\the\dp\@tempa\divide\@tempdimb\@twopowertwo\@tempcntb\@tempdimb

```

```

884 \class@info{Total #1: Column(\the\@tempcnta pt), Page(\the\@tempcnta pt)}%
885 }{}%
886 }{}%
887 }%

```

`\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 user-level environment, therefore it is properly not defined with `\newenvironment`.  
`\write@@float`

```

888 \def\write@float#1{\write@@float{#1}{#1}}%
889 \def\endwrite@float{\@Esphack}%
890 \def\write@floats#1{\write@@float{#1*}{#1}}%
891 \def\endwrite@floats{\@Esphack}%

```

`\write@@float`

```

892 \def\write@@float#1#2{%
893 \ifhmode
894 \@@bsphack
895 \fi
896 \chardef\@tempc\csname#2write\endcsname
897 \toks@{\begin{#1}}%
898 \def\@tempb{#1}%
899 \expandafter\let\csname end#1\endcsname\endwrite@float
900 \catcode'\^M\active
901 \@makeother\{\@makeother\}\@makeother\%
902 \write@floatline
903 }%

```

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

```

\@write@floatline
\float@end@tag
904 \begingroup
905 \catcode'\[the\catcode'\{\catcode'\}\the\catcode'\}\@makeother\{\@makeother\}%
906 \gdef\float@end@tag#1\end{#2}#3\nul[%
907 \def\@tempa{#2}%
908 \@ifx[\@tempa\@tempb][\end{#2}][\write@floatline]%
909 ]%
910 \obeylines%
911 \gdef\write@floatline#1^M[%
912 \begingroup%
913 \newlinechar'\^M%
914 \toks@\expandafter[the\toks@#1]\immediate\write\@tempc[the\toks@]%
915 \endgroup%
916 \toks@[]%
917 \float@end@tag#1\end{}\nul%
918 ]%
919 \endgroup

```

## 6.12 Counters

The following definitions override those of the L<sup>A</sup>T<sub>E</sub>X kernel, providing for a greater range of inputs.

```

920 \def\@alph#1{\ifcase#1\or a\or b\or c\or d\else\@ialph{#1}\fi}
921 \def\@ialph#1{\ifcase#1\or \or \or \or \or e\or f\or g\or h\or i\or j\or
922   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
923   y\or z\or aa\or bb\or cc\or dd\or ee\or ff\or gg\or hh\or ii\or jj\or
924   kk\or ll\or mm\or nn\or oo\or pp\or qq\or rr\or ss\or tt\or uu\or
925   vv\or ww\or xx\or yy\or zz\else\@ctrerr\fi}

```

## 6.13 Customization of Sections

Patch the standard L<sup>A</sup>T<sub>E</sub>X 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 `\@secntformat`.
- 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.

```

926 \def\@startsection#1#2#3#4#5#6{%
927   \@startsection@hook
928   \if@noskipsec \leavevmode \fi
929   \par
930   \@tempskipa #4\relax
931   \@afterindenttrue
932   \ifdim \@tempskipa <\z@
933     \@tempskipa -\@tempskipa \@afterindentfalse
934   \fi
935   \if@nobreak
936     \everypar{}\fi

```

```

937 \else
938 \addpenalty\@secpenalty\advspace\@tempskipa
939 \fi
940 \ifstar
941 {\@dblarg{\@ssect@ltx{#1}{#2}{#3}{#4}{#5}{#6}}}%
942 {\@dblarg{\@sect@ltx {#1}{#2}{#3}{#4}{#5}{#6}}}%
943 }%
944 \def\@startsection@hook{%

```

`\@sect` When defining `\@svsec`, do not expand `\@seccntformat`. 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<sup>A</sup>T<sub>E</sub>X procedure.

```

945 \class@info{Repairing broken LaTeX \string\@sect}%
946 \def\@sect@ltx#1#2#3#4#5#6[#7]#8{%
947 \ifnum{#2}>\c@secnumdepth{%
948 \def\H@svsec{\phantomsection}%
949 \let\@svsec\@empty
950 }{%
951 \H@refstepcounter{#1}%
952 \def\H@svsec{%
953 \phantomsection
954 }%
955 \protected@edef\@svsec{{#1}}%
956 \@ifundefined{@#1cntformat}{%
957 \prepdef\@svsec\@seccntformat
958 }{%
959 \expandafter\prepdef
960 \expandafter\@svsec
961 \csname @#1cntformat\endcsname
962 }%
963 }%
964 \@tempskipa #5\relax
965 \@ifdim{\@tempskipa>\z@}{%
966 \begingroup
967 \interlinepenalty \@M
968 #6{%
969 \@ifundefined{@hangfrom@#1}{\@hang@from}{\csname @hangfrom@#1\endcsname}%
970 {\hskip#3\relax\H@svsec}{\@svsec}{#8}%
971 }%
972 \@@par
973 \endgroup

```

```

974 \ifundefined{#1mark}{\@gobble}{\csname #1mark\endcsname}{#7}%
975 \addcontentsline{toc}{#1}{%
976 \ifnum{#2}>\c@secnumdepth}{%
977 \protect\numberline{%
978 }{%
979 \protect\numberline{\csname the#1\endcsname}%
980 }%
981 #8}%
982 }{%
983 \def\@svsechd{%
984 #6{%
985 \ifundefined{@runin@to@#1}{\@runin@to}{\csname @runin@to@#1\endcsname}%
986 {\hskip#3\relax\H@svsec}{\@svsec}{#8}%
987 }%
988 \ifundefined{#1mark}{\@gobble}{\csname #1mark\endcsname}{#7}%
989 \addcontentsline{toc}{#1}{%
990 \ifnum{#2}>\c@secnumdepth}{%
991 \protect\numberline{%
992 }{%
993 \protect\numberline{\csname the#1\endcsname}%
994 }%
995 #8}%
996 }%
997 }%
998 \@xsect{#5}%
999 }%
1000 \def\@hang@from#1#2#3{\@hangfrom{#1#2}{#3}}%
1001 \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`.

```

1002 \def\@ssect@ltx#1#2#3#4#5#6[#7]#8{%
    Removed \def\@currentlabelname{#8}
1003 \def\H@svsec{\phantomsection}%
1004 \@tempskipa #5\relax
1005 \ifdim{\@tempskipa}>\z@{%
1006 \begingroup
1007 \interlinepenalty \@M
1008 #6{%
1009 \ifundefined{@hangfroms@#1}{\@hang@froms}{\csname @hangfroms@#1\endcsname}%
    Removed {\hskip#3\relax\H@svsec}{\Sectionformat{#8}{#1}}

```



```

1010      {\hskip#3\relax\H@svsec}{#8}%
1011      }%
1012      \@@par
1013      \endgroup
1014      \@ifundefined{#1smark}{\@gobble}{\csname #1smark\endcsname}{#7}%
1015      \addcontentsline{toc}{#1}{\protect\numberline{#8}%
1016      }{%
1017      \def\@svsechd{%
1018      #6%
1019      \@ifundefined{@runin@tos@#1}{\@runin@tos}{\csname @runin@tos@#1\endcsname}%
1020      Removed {\hskip#3\relax\H@svsec}{\Sectionformat{#8}{#1}}
1021      {\hskip#3\relax\H@svsec}{#8}%
1022      }%
1023      \@ifundefined{#1smark}{\@gobble}{\csname #1smark\endcsname}{#7}%
1024      \addcontentsline{toc}{#1}{\protect\numberline{#8}%
1025      }%
1026      \xsect{#5}%
1027      }%
1028      \def\@hang@froms#1#2{#1#2}%
1029      \def\@runin@tos #1#2{#1#2}%

```

`\init@hyperref` 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 type-setting 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 `.aux` file will not 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.

```

1030 \def\init@hyperref{%
1031 \providecommand\phantomsection{}%
1032 \providecommand\hyper@makecurrent[1]{}%
1033 \providecommand\Hy@raisedlink[1]{}%
1034 \providecommand\hyper@anchorstart[1]{}%
1035 \providecommand\hyper@anchorend{}%
1036 \providecommand\hyper@linkstart[2]{}%
1037 \providecommand\hyper@linkend{}%
1038 \providecommand\@currentHref{}%
1039 }%
1040 \let\H@refstepcounter\refstepcounter
1041 \appdef\document@inithook{%
1042 \init@hyperref
1043 }%

```

`\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<sup>A</sup>T<sub>E</sub>X default.

```
1044 \def\sec@upcase#1{\relax{#1}}%
```

## 6.14 Patch the tabular and array Environments

`\endtabular` We headpatch the begin processing and tailpatch the end processing of the `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<sup>A</sup>T<sub>E</sub>X `array` and `tabular` environments. I know of none that do so.

```
1045 \appdef\document@inithook{%
1046   \@ifpackageloaded{array}{\switch@array}{\switch@tabular}}%
1047 \prepdef\endtabular{\endtabular@hook}%
1048 \@provide\endtabular@hook}%
1049 \prepdef\endarray{\endarray@hook}%
1050 \@provide\endarray@hook}%
1051 \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.

```
1052 \prepdef\@tabular{\tabular@hook}%
1053 \@provide\tabular@hook}%
1054 }%
```

`\switch@tabular` The two procedures `\switch@tabular` and `\switch@array` apply needed patches to the various tabular procedures, the former applying to the L<sup>A</sup>T<sub>E</sub>X kernel, the latter to the required `array` package (and to the number of other required packages that load it).

```
1055 \def\switch@tabular{%
1056   \let\@array@sw\@array@sw@array
1057   \@ifx{\@array\@array@LaTeX}{%
1058     \@ifx{\multicolumn\multicolumn@LaTeX}{%
1059       \@ifx{\@tabular\@tabular@LaTeX}{%
1060         \@ifx{\@tabarray\@tabarray@LaTeX}{%
1061           \@ifx{\array\array@LaTeX}{%
1062             \@ifx{\endarray\endarray@LaTeX}{%
1063               \@ifx{\endtabular\endtabular@LaTeX}{%
1064                 \@ifx{\@mkpream\@mkpream@LaTeX}{%
1065                   \@ifx{\@addamp\@addamp@LaTeX}{%
```

```

1066 \@ifx{\@arrayacol\@arrayacol@LaTeX}{%
1067 \@ifx{\@tabacol\@tabacol@LaTeX}{%
1068 \@ifx{\@arrayclassz\@arrayclassz@LaTeX}{%
1069 \@ifx{\@tabclassiv\@tabclassiv@LaTeX}{%
1070 \@ifx{\@arrayclassiv\@arrayclassiv@LaTeX}{%
1071 \@ifx{\@tabclassz\@tabclassz@LaTeX}{%
1072 \@ifx{\@classv\@classv@LaTeX}{%
1073 \@ifx{\hline\hline@LaTeX}{%
1074 \@ifx{\@tabularcr\@tabularcr@LaTeX}{%
1075 \@ifx{\@xtabularcr\@xtabularcr@LaTeX}{%
1076 \@ifx{\@xargarraycr\@xargarraycr@LaTeX}{%
1077 \@ifx{\@yargarraycr\@yargarraycr@LaTeX}{%
1078 \true@sw
1079 }{%
1080 \false@sw
1081 }%
1082 }{%
1083 \false@sw
1084 }%
1085 }{%
1086 \false@sw
1087 }%
1088 }{%
1089 \false@sw
1090 }%
1091 }{%
1092 \false@sw
1093 }%
1094 }{%
1095 \false@sw
1096 }%
1097 }{%
1098 \false@sw
1099 }%
1100 }{%
1101 \false@sw
1102 }%
1103 }{%
1104 \false@sw
1105 }%
1106 }{%
1107 \false@sw
1108 }%
1109 }{%
1110 \false@sw
1111 }%
1112 }{%
1113 \false@sw
1114 }%
1115 }{%

```

```

1116         \false@sw
1117     }%
1118     }{%
1119         \false@sw
1120     }%
1121     }{%
1122         \false@sw
1123     }%
1124     }{%
1125         \false@sw
1126     }%
1127     }{%
1128         \false@sw
1129     }%
1130     }{%
1131         \false@sw
1132     }%
1133     }{%
1134         \false@sw
1135     }%
1136     }{%
1137         \false@sw
1138     }%
1139     }{%
1140         \false@sw
1141     }%
1142     {%
1143     \class@info{Patching LaTeX tabular.}%
1144     }{%
1145     \class@info{Unrecognized LaTeX tabular. Please update this document class! (Proceeding with f
1146     }%
1147     \let\@array\@array@ltx
1148     \let\multicolumn\multicolumn@ltx
1149     \let\@tabular\@tabular@ltx
1150     \let\@tabarray\@tabarray@ltx
1151     \let\array\array@ltx
1152     \let\endarray\endarray@ltx
1153     \let\endtabular\endtabular@ltx
1154     \let\@mkpream\@mkpream@ltx
1155     \let\@addamp\@addamp@ltx
1156     \let\@arrayacol\@arrayacol@ltx
1157     \let\@tabacol\@tabacol@ltx
1158     \let\@arrayclassz\@arrayclassz@ltx
1159     \let\@tabclassiv\@tabclassiv@ltx
1160     \let\@arrayclassiv\@arrayclassiv@ltx
1161     \let\@tabclassz\@tabclassz@ltx
1162     \let\@classv\@classv@ltx
1163     \let\hline\hline@ltx
1164     \let\@tabularcr\@tabularcr@ltx
1165     \let\@xtabularcr\@xtabularcr@ltx

```

```

1166 \let\@xargarraycr\@xargarraycr\ltx
1167 \let\@yargarraycr\@yargarraycr\ltx
1168 }%

1169 \def\switch@array{%
1170 \ifpackageloaded{colortbl}{\let\switch@array@info\colortbl@message}{\let\switch@array@info\ar
1171 \let\@array@sw\@array@sw@LaTeX
1172 \ifx{\@array\@array@array}{}%
1173 \ifx{\@tabular\@tabular@array}{}%
1174 \ifx{\@tabarray\@tabarray@array}{}%
1175 \ifx{\array\array@array}{}%
1176 \ifx{\endarray\endarray@array}{}%
1177 \ifx{\endtabular\endtabular@array}{}%
1178 \ifx{\@mkpream\@mkpream@array}{}%
1179 \ifx{\@classx\@classx@array}{}%
1180 \ifx{\insert@column\insert@column@array}{}%
1181 \ifx{\@arraycr\@arraycr@array}{}%
1182 \ifx{\@xarraycr\@xarraycr@array}{}%
1183 \ifx{\@xargarraycr\@xargarraycr@array}{}%
1184 \ifx{\@yargarraycr\@yargarraycr@array}{}%
1185 \true@sw
1186 }{%
1187 \false@sw
1188 }%
1189 }{%
1190 \false@sw
1191 }%
1192 }{%
1193 \false@sw
1194 }%
1195 }{%
1196 \false@sw
1197 }%
1198 }{%
1199 \false@sw
1200 }%
1201 }{%
1202 \false@sw
1203 }%
1204 }{%
1205 \false@sw
1206 }%
1207 }{%
1208 \false@sw
1209 }%
1210 }{%
1211 \false@sw
1212 }%
1213 }{%
1214 \false@sw

```

```

1215 }%
1216 }{%
1217 \false@sw
1218 }%
1219 }{%
1220 \false@sw
1221 }%
1222 }{%
1223 \false@sw
1224 }{%
1225 \class@info{Patching array package.}%
1226 }{%
1227 \switch@array@info
1228 }%
1229 \let\@array \@array@array@new
1230 \let\@@array \@array % Così fan tutti
1231 \let\@tabular \@tabular@array@new
1232 \let\@tabarray \@tabarray@array@new
1233 \let\@array \@array@array@new
1234 \let\endarray \endarray@array@new
1235 \let\endtabular\endtabular@array@new
1236 \let\@mkpream \@mkpream@array@new
1237 \let\@classx \@classx@array@new
1238 \let\@arrayacol\@arrayacol@ltx
1239 \let\@tabacol \@tabacol@ltx
1240 \let\insert@column\insert@column@array@new
1241 \expandafter\let\csname endtabular*\endcsname\endtabular % Così fan tutti
1242 \let\@arraycr \@arraycr@new
1243 \let\@xarraycr \@xarraycr@new
1244 \let\@xargarraycr\@xargarraycr@new
1245 \let\@yargarraycr\@yargarraycr@new
1246 }%
1247 \def\array@message{%
1248 \class@info{Unrecognized array package. Please update this document class! (Proceeding with fi
1249 }%
1250 \def\colortbl@message{%
1251 \class@info{colortbl package is loaded. (Proceeding with fingers crossed.)}%
1252 }%

```

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

```

1253 \def\@array@sw@LaTeX{\@ifx{\@tabularcr}}%
1254 \def\@array@sw@array{\@ifx{\dollarbegin\begin group}}%

```

**\@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).

Compatibility note: I had done `\let \col@sep \undefined` here, but this was not compatible with `colortbl`. I have removed that statement.

```

1255 \def\@tabular@LaTeX{%
1256 \leavevmode
1257 \hbox\bgroup$%
1258 \let\@acol\@tabacol
1259 \let\@classz\@tabclassz
1260 \let\@classiv\@tabclassiv
1261 \let\\\@tabularcr
1262 \@tabarray
1263 }%
1264 \def\@tabular@ltx{%
1265 \let\@acoll\@tabacoll
1266 \let\@acolr\@tabacolr
1267 \let\@acol\@tabacol
1268 \let\@classz\@tabclassz
1269 \let\@classiv\@tabclassiv
1270 \let\\\@tabularcr
1271 \@tabarray
1272 }%
1273 \def\@tabular@array{%
1274 \leavevmode
1275 \hbox\bgroup$%
1276 \col@sep\tabcolsep
1277 \let\d@llarbegin\beginngroup
1278 \let\d@llarend\endgroup
1279 \@tabarray
1280 }%
1281 \def\@tabular@array@new{%
1282 \let\@acoll\@tabacoll
1283 \let\@acolr\@tabacolr
1284 \let\@acol\@tabacol
1285 \sepundefined
1286 \let\d@llarbegin\beginngroup
1287 \let\d@llarend\endgroup
1288 }%

```

`\@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<sup>A</sup>T<sub>E</sub>X. Under some circumstances, we want to change this to, say, 'v'.

FIXME: must decouple `array` and `tabular`. Done (it seems).

Note on `colortbl`: this package head-patches `\@tabarray` with its own command `\CTstart`, and tails onto `\endarray` with `\CTend`. It fortuitously does the former at `\AtBeginDocument` time, and, fortuitously, we do not patch `\endarray`, which it overwrites.

```

1289 \def\@tabarray@LaTeX{%
1290 \m@th\@ifnextchar[\@array{\@array[c]}%
1291 }%
1292 \def\@tabarray@ltx{%

```

```

1293 \m@th\@ifnextchar[\@array{\expandafter\@array\expandafter[\array@default]]}%
1294 }%
1295 \def\@tabarray@array{%
1296 \@ifnextchar[{\@array}{\@array[c]]}%
1297 }%
1298 \def\@tabarray@array@new{%
1299 \@ifnextchar[{\@array}{\expandafter\@array\expandafter[\array@default]]}%
1300 }%

\@tabularcr We provide for the \ command within tabular to provide control over page
\@tbpen breaking, just the same as that of eqnarray.
\@tabularcr The count register \intertabularlinepenalty is similar to \interdisplaylinepenalty:
\@xtabularcr it is the penalty associated with each row of a tabular. When it is set to \@M, the
\@xargarraycr tabular will cleave together.
\@yargarraycr The count register \@tbpen is similar to \@eqpen: it memorizes the penalty
\@arraycr 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.
1301 \newcount\intertabularlinepenalty
1302 \intertabularlinepenalty=100
1303 \newcount\@tbpen
1304 \appdef\samepage{\intertabularlinepenalty\@M}%
1305 \def\@tabularcr@LaTeX{\ifnum 0='}\fi \ifstar \@xtabularcr \@xtabularcr}%
1306 \def\@tabularcr\ltx{\ifnum 0='}\fi \ifstar {\global \@tbpen \@M \@xtabularcr }\global \@tbpen
1307 \def\@xtabularcr@LaTeX{\@ifnextchar [\@argtabularcr {\ifnum 0='{ \fi }\cr }}%
1308 \def\@xtabularcr\ltx{\@ifnextchar [\@argtabularcr {\ifnum 0='{ \fi }\cr \noalign {\penalty \@tbpen
1309 \def\@xargarraycr@LaTeX#1{\@tempdima #1\advance \@tempdima \dp \@arstrutbox \vrule \@height \z@ \
1310 \def\@xargarraycr\ltx#1{\@tempdima #1\advance \@tempdima \dp \@arstrutbox \vrule \@height \z@ \
1311 \def\@yargarraycr@LaTeX#1{\cr \noalign {\vskip #1}}}%
1312 \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.
1313 \def\@arraycr@array{%
1314 \relax
1315 \iffalse{\fi\ifnum 0='}\fi
1316 \ifstar \@xarraycr \@xarraycr
1317 }%
1318 \def\@arraycr@new{%
1319 \relax

```



```

1320 \iffalse{\fi\ifnum 0='}\fi
1321 \@ifstar {\global \@tbpen \@M \@xarraycr }{\global \@tbpen \intertabularlinepenalty \@xarraycr
1322 }%
1323 \def\@xarraycr@array{%
1324 \@ifnextchar [%]
1325 \@argarraycr {\ifnum 0='}\fi\cr}%
1326 }%
1327 \def\@xarraycr@new{%
1328 \@ifnextchar [%]
1329 \@argarraycr {\ifnum 0='{ }\fi\cr \noalign {\penalty \@tbpen }}%
1330 }%
1331 \def\@xargarraycr@array#1{%
1332 \unskip
1333 \@tempdima #1\advance\@tempdima \dp\@arstrutbox
1334 \vrule \@depth\@tempdima \@width\z@
1335 \cr
1336 }%
1337 \def\@xargarraycr@new#1{%
1338 \unskip
1339 \@tempdima #1\advance\@tempdima \dp\@arstrutbox
1340 \vrule \@depth\@tempdima \@width\z@
1341 \cr
1342 \noalign {\penalty \@tbpen }%
1343 }%
1344 \def\@yargarraycr@array#1{%
1345 \cr
1346 \noalign{\vskip #1}%
1347 }%
1348 \def\@yargarraycr@new#1{%
1349 \cr
1350 \noalign{\penalty \@tbpen \vskip #1}%
1351 }%

```

`\array` We provide old and new versions of the `\array` procedure for both L<sup>A</sup>T<sub>E</sub>X 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.

```

1352 \def\array@LaTeX{%
1353 \let\@acol\@arrayacol
1354 \let\@classz\@arrayclassz
1355 \let\@classiv\@arrayclassiv
1356 \let\\\@arraycr
1357 \let\@halignto\@empty
1358 \@tabarray
1359 }%
1360 \def\array@ltx{%
1361 \@ifmmode{}\{\@badmath$\}%
1362 \let\@acoll\@arrayacol
1363 \let\@acolr\@arrayacol
1364 \let\@acol\@arrayacol

```

```

1365 \let\@classz\@arrayclassz
1366 \let\@classiv\@arrayclassiv
1367 \let\@arraycr
1368 \let\@halignto\@empty
1369 \@tabarray
1370 }%
1371 \def\array@array{%
1372 \col@sep\arraycolsep
1373 \def\d@llarbegin{$}\let\d@llarend\d@llarbegin\gdef\@halignto{}%
1374 \@tabarray
1375 }
1376 \def\array@array@new{%
1377 \@ifmode{}{\@badmath$}%
1378 \let\@acoll\@arrayacol
1379 \let\@acolr\@arrayacol
1380 \let\@acol\@arrayacol
    Removed: \let\col@sep\@undefined
1381 \def\d@llarbegin{$}%
1382 \let\d@llarend\d@llarbegin
1383 \gdef\@halignto{}%
1384 \@tabarray
1385 }%

```

`\@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.

```

1386 \def\@array@LaTeX[#1]#2{%
1387 \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi\fi
1388 \bgroup
1389 \setbox\@arstrutbox\hbox{%
1390 \vrule \@height\arraystretch\ht\strutbox
1391 \quad \@depth\arraystretch\dp\strutbox
1392 \quad \@width\z@}%
1393 \mkpream{#2}%
1394 \edef\@preamble{%
1395 \ialign \noexpand\@halignto
1396 \bgroup \@arstrut \@preamble \tabskip\z@skip \cr}%
1397 \let\@startpbox\@startpbox \let\@endpbox\@endpbox
1398 \let\tabularnewline\%
1399 \let\par\@empty
1400 \let\@sharp##%
1401 \set@typeset@protect
1402 \lineskip\z@skip\baselineskip\z@skip

```

```

1403 \ifhmode \@preamerr\z@ \@par\fi
1404 \@preamble
1405 }%
1406 \def\@array@ltx[#1]#2{%
1407 \@nameuse{\@array@align@#1}%
1408 \set@arstrutbox
1409 \@mkpream{#2}%
1410 \prepdef\@preamble{%
1411 \tabskip\tabmid@skip
1412 \@arstrut
1413 }%
1414 \appdef\@preamble{%
1415 \tabskip\tabright@skip
1416 \cr
1417 \array@row@pre
1418 }%
1419 % \let\@startpbox\@@startpbox
1420 % \let\@endpbox\@@endpbox
1421 \let\tabularnewline\%
1422 \let\par\@empty
1423 \let\@sharp##%
1424 \set@typeset@protect
1425 \lineskip\z@skip\baselineskip\z@skip
1426 \tabskip\tableft@skip\relax
1427 \ifhmode \@preamerr\z@ \@par\fi
1428 \everycr{}%
1429 \expandafter\halign\expandafter\@halignto\expandafter\bgroup\@preamble
1430 }%
1431 %
1432 \def\set@arstrutbox{%
1433 \setbox\@arstrutbox\hbox{%
1434 \vrule \@height\arraystretch\ht\strutbox
1435 \quad \@depth\arraystretch\dp\strutbox
1436 \quad \@width\z@
1437 }%
1438 }%

```

\@array@array

```

1439 \def\@array@array[#1]#2{%
1440 \@tempdima \ht \strutbox
1441 \advance \@tempdima by\extrarowheight
1442 \setbox \@arstrutbox \hbox{\vrule
1443 \quad \@height \arraystretch \@tempdima
1444 \quad \@depth \arraystretch \dp \strutbox
1445 \quad \@width \z@}%
1446 \beginngroup
1447 \@mkpream{#2}%
1448 \xdef\@preamble{\noexpand \ialign \@halignto
1449 \quad \bgroup \@arstrut \@preamble
1450 \quad \tabskip \z@ \cr}%

```

```

1451 \endgroup
1452 \@arrayleft
1453 \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi \fi
1454 \bgroup
1455 \let \@sharp ##\let \protect \relax
1456 \lineskip \z@
1457 \baselineskip \z@
1458 \m@th
1459 \let\\\@arraycr \let\tabularnewline\\\let\par\@empty \@preamble
1460 }%
1461 \def\@array@array@new[#1]#2{%
1462   \@tempdima\ht\strutbox
1463   \advance\@tempdima by\extrarowheight
1464   \setbox\@arstrutbox\hbox{%
1465     \vrule \@height\arraystretch\@tempdima
1466     \@depth \arraystretch\dp\strutbox
1467     \@width \z@
1468   }%
1469   \begingroup
1470     \mkpream{#2}%
1471     \xdef\@preamble{\@preamble}%
1472   \endgroup
1473   \prepdef\@preamble{%
1474     \tabskip\tabmid@skip
1475     \@arstrut
1476   }%
1477   \appdef\@preamble{%
1478     \tabskip\tabright@skip
1479     \cr
1480     \array@row@pre
1481   }%
1482   \@arrayleft
1483   \@nameuse{\@array@align@#1}%
1484   \m@th
1485   \let\\\@arraycr
1486   \let\tabularnewline\\%
1487   \let\par\@empty
1488   \let\@sharp##%
1489   \set@typeset@protect
1490   \lineskip\z@\baselineskip\z@
1491   \tabskip\tableft@skip
1492   \everycr{}%
1493   \expandafter\halign\expandafter\@halignto\expandafter\bgroup\@preamble
1494 }%

```

`\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`.

```

1495 \def\endarray@LaTeX{%
1496   \crr\egroup\egroup

```

```

1497 }%
1498 \def\endarray@ltx{%
1499 \crr\array@row@pst\egroup\egroup
1500 }%
1501 \def\endarray@array{%
1502 \crr \egroup \egroup \@arrayright \gdef\@preamble{}%
1503 }%
1504 \def\endarray@array@new{%
1505 \crr\array@row@pst\egroup\egroup % Same as \endarray@ltx
1506 \@arrayright
1507 \global\let\@preamble\@empty
1508 }%

\endtabular
1509 \def\endtabular@LaTeX{%
1510 \crr\egroup\egroup $\egroup
1511 }%
1512 \def\endtabular@ltx{%
1513 \endarray
1514 }%
1515 \def\endtabular@array{%
1516 \endarray $\egroup
1517 }%
1518 \def\endtabular@array@new{%
1519 \endarray
1520 }%

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.
1521 \@namedef{endtabular*}{\endtabular}%

\multicolumn
1522 \long\def\multicolumn@LaTeX#1#2#3{%
1523 \multispan{#1}\begingroup
1524 \mkpream{#2}%
1525 \def\@sharp{#3}\set@typeset@protect
1526 \let\@startpbox\@startpbox\let\@endpbox\@endpbox
1527 \carstrut \preamble\hbox{}\endgroup\ignorespaces
1528 }%
1529 \long\def\multicolumn@ltx#1#2#3{%
1530 \multispan{#1}%
1531 \begingroup
1532 \mkpream{#2}%
1533 \def\@sharp{#3}%
1534 \set@typeset@protect
1535 %\let\@startpbox\@startpbox\let\@endpbox\@endpbox
1536 \carstrut
1537 \preamble

```

```

1538 \hbox{}%
1539 \endgroup
1540 \ignorespaces
1541 }%

\array@align@ Here are the various procedures for the vertical alignment options. The change
\array@default from standard LATEX 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.
1542 \def\array@align@t{\leavevmode\top\bgroup}%
1543 \def\array@align@b{\leavevmode\bottom\bgroup}%
1544 \def\array@align@c{\leavevmode\@ifmmode{\vcenter\bgroup}{$\vcenter\bgroup\aftergroup$\aftergroup}%
1545 \def\array@align@v{%
1546 \@ifmmode{%
1547 \badmath
1548 \vcenter\bgroup
1549 }{%
1550 \@ifinner{%
1551 $\vcenter\bgroup\aftergroup$
1552 }{%
1553 \@@par\bgroup
1554 }%
1555 }%
1556 }%
1557 \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.

1558 \def\array@row@rst{%
1559 \let\array@align@v\array@align@c
1560 }%
1561 \def\array@row@pre{%
1562 \def\array@row@pst{%

\toprule Default definitions for \toprule, \colrule, \botrule
\colrule 1563 \newcommand\toprule{\tab@rule{\column@font}{\column@fil}{\frstrut}}%
\botrule 1564 \newcommand\colrule{\unskip\lrstrut\\\tab@rule{\body@font}{\frstrut}}%
1565 \newcommand\botrule{\unskip\lrstrut\\\noalign{\hline@rule}{}}%

\hline
1566 \def\hline@LaTeX{%
1567 \noalign{\ifnum0=‘}\fi\hrule \@height \arrayrulewidth \futurelet

```

```

1568 \reserved@a\@xhline
1569 }%
1570 \def\hline<ltx{%
1571 \noalign{%
1572 \ifnum0='}\fi
1573 \hline@rule
1574 \futurelet\reserved@a\@xhline
1575 % \noalign ended in \@xhline
1576 }%
1577 \def\@xhline@unneeded{%
1578 \say\reserved@a
1579 \ifx\reserved@a\hline
1580 \vskip\doublerulesep
1581 \vskip-\arrayrulewidth
1582 \fi
1583 \ifnum0='{ \fi}%
1584 }%
1585 \def\tab@rule#1#2#3{%
1586 \crr
1587 \noalign{%
1588 \hline@rule
1589 \gdef\@arstrut@hook{%
1590 \global\let\@arstrut@hook\@empty
1591 #3%
1592 }%
1593 \gdef\cell@font{#1}%
1594 \gdef\cell@fil{#2}%
1595 }%
1596 }%
1597 \def\column@font{}%
1598 \def\column@fil{}%
1599 \def\body@font{}%
1600 \def\cell@font{}%
1601 \def\frstrut{}%
1602 \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.

```

1603 \def\@arstrut@hline{%
1604 \relax
1605 \@ifmmode{\copy}{\unhcopy}\@arstrutbox@hline
1606 \@arstrut@hook

```

```

1607 }%
1608 %
1609 \let\@arstrut@org\@arstrut
1610 \def\@arstrut@hook{%
1611   \global\let\@arstrut\@arstrut@org
1612 }%
1613 %
1614 \newbox\@arstrutbox@hline
1615 \appdef\set@arstrutbox{%
1616   \setbox\@arstrutbox@hline\hbox{%
1617     \setbox\z@\hbox{$0^{0}_{-}$}%
1618     \dimen@ \ht\z@ \advance\dimen@\@arstrut@hline@clnc
1619     \@ifdim{\dimen@ < \arraystretch\ht\strutbox}{\dimen@ = \arraystretch\ht\strutbox}{}%
1620     \vrule \@height\dimen@
1621             \@depth\arraystretch \dp\strutbox
1622             \@width\z@
1623   }%
1624 }%
1625 %
1626 \def\hline@rule{%
1627   \hrule \@height \arrayrulewidth
1628   \global\let\@arstrut\@arstrut@hline
1629 }%
1630 \def\@arstrut@hline@clnc{2\p@}% % Klotch: magic number

\tableleft@skip
1631 \def\tableleft@skip{\z@skip}%
1632 \def\tabmid@skip{\z@skip}%\@flushglue
1633 \def\tabright@skip{\z@skip}%
1634 \def\tableleftsep{\tabcolsep}%
1635 \def\tabmidsep{\tabcolsep}%
1636 \def\tabrightsep{\tabcolsep}%
1637 \def\cell@fil{}%
1638 \def\pbox@hook{}%

\@arstrut
1639 \appdef\@arstrut{\@arstrut@hook}%
1640 \let\@arstrut@hook\@empty
1641 \def\@addtopreamble{\appdef\@preamble}%

\@mkpream
1642 \def\@mkpream@LaTeX#1{%
1643   \@firstampttrue\@lastchclass6
1644   \let\@preamble\@empty
1645   \let\protect\@unexpandable@protect
1646   \let\@sharp\relax
1647   \let\@startpbox\relax\let\@endpbox\relax
1648   \@expast{#1}%
1649   \expandafter\@tfor \expandafter

```



```

1650 \nextchar \expandafter:\expandafter=\reserved@a\do
1651 {\testpach\nextchar
1652 \ifcase \@chclass \@classz \or \@classi \or \@classii \or \@classiii
1653 \or \@classiv \or \@classv \fi\@lastchclass\@chclass}%
1654 \ifcase \@lastchclass \@acol
1655 \or \or \@preamerr \@ne\or \@preamerr \tw@\or \or \@acol \fi
1656 }%
1657 \def\@mkpream@ltx#1{%
1658 \@firstamptrue
1659 \@lastchclass6
1660 \let\@preamble\@empty
1661 \let\protect\@unexpandable@protect
1662 \let\@sharp\relax
1663 %\let\@startpbox\relax\let\@endpbox\relax
1664 \@expast{#1}%
1665 \expandafter\@tfor\expandafter\nextchar\expandafter:\expandafter=\reserved@a
1666 \do{%
1667 \expandafter\@testpach\expandafter{\nextchar}%
1668 \ifcase\@chclass
1669 \@classz
1670 \or
1671 \@classi
1672 \or
1673 \@classii
1674 \or
1675 \@classiii
1676 \or
1677 \@classiv
1678 \or
1679 \@classv
1680 \fi
1681 \@lastchclass\@chclass
1682 }%
1683 \ifcase\@lastchclass
1684 \@acolr % right-hand column
1685 \or
1686 \or
1687 \@preamerr\@ne
1688 \or
1689 \@preamerr\tw@
1690 \or
1691 \or
1692 \@acolr % right-hand column
1693 \fi
1694 }%

```

\insert@column

```

1695 \def\insert@column@array{%
1696 \the@toks \the \@tempcnta
1697 \ignorespaces \@sharp \unskip

```

```

1698 \the@toks \the \count@ \relax
1699 }%
1700 \def\insert@column@array@new{%
1701 \the@toks\the\@tempcnta
1702 \array@row@rst\cell@font
1703 \ignorespaces\sharp\unskip
1704 \the@toks\the\count@
1705 \relax
1706 }%

```

`\@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 L<sup>A</sup>T<sub>E</sub>X, 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`, T<sub>E</sub>X'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`.

```

1707 \def\@mkpream@relax{%
1708 \let\tableleftsep \relax
1709 \let\tabmidsep \relax
1710 \let\tabrightsep \relax
1711 \let\array@row@rst\relax
1712 \let\cell@font \relax
1713 \let\@startpbox \relax
1714 }%

```

`\@mkpream` We insert `\@mkpream@relax` at the head of the procedure. The robustifying of `\@startpbox` and `\@endpbox` is taken over by this mechanism. We also invoke `\@acolr` instead of `\@acol` when a right-hand column is at hand.

Note on `colortbl`: this package head-patches `\@mkpream` to robustify a number of its commands during the construction of the alignment preamble. The best we can do is to supplement the `\@mkpream@relax` procedure to perform this action.

```

1715 \def\@mkpream@array#1{%
1716 \gdef\@preamble{}\@lastchclass 4 \@firstampttrue
1717 \let\sharp\relax \let\@startpbox\relax \let\@endpbox\relax
1718 \@temptokena{#1}\@tempwattrue
1719 \@whilesw@if@tempwa\fi{\@tempwafalse\the\NC@list}%
1720 \count@\m@ne
1721 \let\the@toks\relax
1722 \prepnext@tok
1723 \expandafter \@tfor \expandafter \@nextchar
1724 \expandafter :\expandafter =\the\@temptokena \do
1725 {\@testpach
1726 \ifcase \@chclass \@classz \or \@classi \or \@classii
1727 \or \save@decl \or \or \@classv \or \@classvi

```

```

1728     \or \@classvii \or \@classviii
1729     \or \@classx
1730     \or \@classx \fi
1731     \@lastchclass\@chclass}%
1732     \ifcase\@lastchclass
1733     \@acol \or
1734     \or
1735     \@acol \or
1736     \@preamerr \thr@@ \or
1737     \@preamerr \tw@ \@addtopreamble\@sharp \or
1738     \or
1739     \else \@preamerr \@ne \fi
1740     \def\the@toks{\the\toks}%
1741 }%
1742 \def\@mkpream@array@new#1{%
1743 \gdef\@preamble{}%
1744 \@lastchclass\@f@ur
1745 \@firststamptrue
1746 \let\@sharp\relax
1747 \@mkpream@relax
1748 %\let\@startpbox\relax\let\@endpbox\relax
1749 \@temptokena{#1}\@tempswatrue
1750 \@whilesw\if@tempswa\fi{\@tempswafalse\the\NC@list}%
1751 \count@m@ne
1752 \let\the@toks\relax
1753 \prepnext@tok
1754 \expandafter\@tfor\expandafter\@nextchar\expandafter:\expandafter=\the\@temptokena
1755 \do{%
1756 \@testpach
1757 \ifcase\@chclass
1758     \@classz
1759     \or
1760     \@classi
1761     \or
1762     \@classii
1763     \or
1764     \save@decl
1765     \or
1766     \or
1767     \@classv
1768     \or
1769     \@classvi
1770     \or
1771     \@classvii
1772     \or
1773     \@classviii
1774     \or
1775     \@classx
1776     \or
1777     \@classx

```

```

1778 \fi
1779 \@lastchclass\@chclass
1780 }%
1781 \ifcase\@lastchclass
1782 \@acolr % right-hand column
1783 \or
1784 \or
1785 \@acolr % right-hand column
1786 \or
1787 \@preamerr\thr@@
1788 \or
1789 \@preamerr\tw@\@addtopreamble\@sharp
1790 \or
1791 \or
1792 \else
1793 \@preamerr\@ne
1794 \fi
1795 \def\the@toks{\the\toks}%
1796 }%

```

`\@mkpream@relax` David P. Carlisle's `colortbl` package headpatches `\@mkpream` in place during package loading, so it does not know whom it is working on. Let us try to accommodate this package by doing what it would have liked to have done.

Note: it would be far better to break out this mechanism in the `array` package.

```

1797 \appdef\@mkpream@relax{%
1798 \let\CT@setup \relax
1799 \let\CT@color \relax
1800 \let\CT@do@color \relax
1801 \let\color \relax
1802 \let\CT@column@color\relax
1803 \let\CT@row@color \relax
1804 \let\CT@cell@color \relax
1805 }%

```

`\@addamp`

```

1806 \def\@addamp@LaTeX{%
1807 \if@firstamp\@firstampfalse\else\edef\@preamble{\@preamble &}\fi
1808 }%
1809 \def\@addamp@ltx{%
1810 \if@firstamp\@firstampfalse\else\@addtopreamble{&}\fi
1811 }%

```

`\@arrayacol`

```

1812 \def\@arrayacol@LaTeX{%
1813 \edef\@preamble{\@preamble \hskip \arraycolsep}%
1814 }%
1815 \def\@arrayacol@ltx{%
1816 \@addtopreamble{\hskip\arraycolsep}%
1817 }%

```

```

\@tabacol
1818 \def\@tabacoll{%
1819 \addtopreamble{\hskip\tableftsep\relax}%
1820 }%
1821 \def\@tabacol@LaTeX{%
1822 \edef\@preamble{\@preamble \hskip \tabcolsep}%
1823 }%
1824 \def\@tabacol@ltx{%
1825 \addtopreamble{\hskip\abmidsep\relax}%
1826 }%
1827 \def\@tabacolr{%
1828 \addtopreamble{\hskip\abrightsep\relax}%
1829 }%

\@arrayclassz
1830 \def\@arrayclassz@LaTeX{%
1831 \ifcase \@lastchclass \@acolampacol \or \@ampacol \or
1832 \or \or \@addamp \or
1833 \@acolampacol \or \@firstampfalse \@acol \fi
1834 \edef\@preamble{\@preamble
1835 \ifcase \@chnum
1836 \hfil$\relax\@sharp$\hfil \or $\relax\@sharp$\hfil
1837 \or \hfil$\relax\@sharp$\fi}%
1838 }%
1839 \def\@arrayclassz@ltx{%
1840 \ifcase \@lastchclass
1841 \@acolampacol
1842 \or
1843 \@ampacol
1844 \or
1845 \or
1846 \or
1847 \@addamp
1848 \or
1849 \@acolampacol
1850 \or
1851 \@firstampfalse\@acoll
1852 \fi
1853 \ifcase \@chnum
1854 \addtopreamble{%
1855 \hfil\array@row@rst$\relax\@sharp$\hfil
1856 }%
1857 \or
1858 \addtopreamble{%
1859 \array@row@rst$\relax\@sharp$\hfil
1860 }%
1861 \or
1862 \addtopreamble{%
1863 \hfil\array@row@rst$\relax\@sharp$%
1864 }%

```

```

1865 \fi
1866 }%

\@tabclassz
1867 \def\@tabclassz@LaTeX{%
1868 \ifcase\@lastchclass
1869 \acolampacol
1870 \or
1871 \ampacol
1872 \or
1873 \or
1874 \or
1875 \addamp
1876 \or
1877 \acolampacol
1878 \or
1879 \@firstampfalse\acol
1880 \fi
1881 \edef\@preamble{%
1882 \preamble{%
1883 \ifcase\@chnum
1884 \hfil\ignorespaces\@sharp\unskip\hfil
1885 \or
1886 \hskip1sp\ignorespaces\@sharp\unskip\hfil
1887 \or
1888 \hfil\hskip1sp\ignorespaces\@sharp\unskip
1889 \fi}}%
1890 }%
1891 \def\@tabclassz@ltx{%
1892 \ifcase\@lastchclass
1893 \acolampacol
1894 \or
1895 \ampacol
1896 \or
1897 \or
1898 \or
1899 \addamp
1900 \or
1901 \acolampacol
1902 \or
1903 \@firstampfalse\acoll
1904 \fi
1905 \ifcase\@chnum
1906 \@addtopreamble{%
1907 {\hfil\array@row@rst\cell@font\ignorespaces\@sharp\unskip\hfil}%
1908 }%
1909 \or
1910 \@addtopreamble{%
1911 {\cell@fil\hskip1sp\array@row@rst\cell@font\ignorespaces\@sharp\unskip\hfil}%
1912 }%

```

```

1913 \or
1914 \@addtopreamble{%
1915     {\hfil\hskip1sp\array@row@rst\cell@font\ignorespaces\@sharp\unskip\cell@fil}%
1916 }%
1917 \fi
1918 }%

\@tabclassiv
1919 \def\@tabclassiv@LaTeX{%
1920 \@addtopreamble\@nextchar
1921 }%
1922 \def\@tabclassiv@ltx{%
1923 \expandafter\@addtopreamble\expandafter{\@nextchar}%
1924 }%

\@arrayclassiv
1925 \def\@arrayclassiv@LaTeX{%
1926 \@addtopreamble{$\@nextchar$}%
1927 }%
1928 \def\@arrayclassiv@ltx{%
1929 \expandafter\@addtopreamble\expandafter{\expandafter$\@nextchar$}%
1930 }%

\@classv
1931 \def\@classv@LaTeX{%
1932 \@addtopreamble{\@startpbox{\@nextchar}\ignorespaces
1933 \@sharp\@endpbox}%
1934 }%
1935 \def\@classv@ltx{%
1936 \expandafter\@addtopreamble
1937 \expandafter{%
1938 \expandafter \@startpbox
1939 \expandafter {\@nextchar}%
1940 \pbox@hook\array@row@rst\cell@font\ignorespaces\@sharp\@endpbox
1941 }%
1942 }%

\@classx
1943 \def\@classx@array{%
1944 \ifcase \@lastchclass
1945 \@acolampacol \or
1946 \@addamp \@acol \or
1947 \@acolampacol \or
1948 \or
1949 \@acol \@firstampfalse \or
1950 \@addamp
1951 \fi
1952 }%
1953 \def\@classx@array@new{%

```

```

1954 \ifcase \@lastchclass
1955   \@acolampacol
1956 \or
1957   \@addamp \@acol
1958 \or
1959   \@acolampacol
1960 \or
1961 \or
1962   \@firstampfalse \@acoll
1963 \or
1964   \@addamp
1965 \fi
1966 }%

```

## 6.15 Repair other broken parts of L<sup>A</sup>T<sub>E</sub>X

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

```

1967 \def\@xbitor@LaTeX #1{\@tempcntb \count#1
1968   \ifnum \@tempcnta =\z@
1969   \else
1970     \divide \@tempcntb \@tempcnta
1971     \ifodd \@tempcntb \@testtrue \fi
1972   \fi}%
1973 \def\@xbitor@ltx#1{%
1974   \@tempcntb \count#1 \relax
1975   \@ifnum {\@tempcnta =\z@} {} {%
1976     \divide \@tempcntb \@tempcnta
1977     \@ifodd \@tempcntb {\@testtrue} {}%
1978   }%
1979 }%
1980 \@ifx {\@xbitor \@xbitor@LaTeX} {%
1981   \class@info{Repairing broken LaTeX \string \@xbitor}%
1982 }{%
1983   \class@info{Unrecognized LaTeX \string \@xbitor. Please update this document class! (Proceedin
1984 }%
1985 \let \@xbitor \@xbitor@ltx

```

## 6.16 Syntax

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

```

1986 \newcommand* \@gobble@opt@one [2] [] {}%

```

## 6.17 Auto-indented Contents

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

Note on `hyperref` compatibility: We must respect that `\contentsline` now has a fourth 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.

```
1987 \def\@starttoc#1{%
1988   \begingroup
1989   \toc@pre
1990   \makeatletter
1991   \@input{\jobname.#1}%
1992   \if@filesw
1993     \expandafter\newwrite\csname tf@#1\endcsname
1994     \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1995   \fi
1996   \@nobreakfalse
1997   \toc@post
1998 \endgroup
1999 }%
2000 \def\toc@pre{%
2001 \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.

```
2002 \def\toc@@font{%
2003 \def\ltxu@dotsep{z}%
```

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

All of the `\l@...` commands simply go through the utility procedure `\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.

```
2004 \let\tocdim@section \leftmargini
2005 \let\tocdim@subsection \leftmarginii
2006 \let\tocdim@subsubsection \leftmarginiii
```

```

2007 \let\tocdim@paragraph \leftmarginiv
2008 \let\tocdim@appendix \leftmarginv
2009 \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.

2010 \def\toc@pre@auto{%
2011 \toc@font
2012 \@tempdima\z@
2013 \toc@setindent\@tempdima{section}%
2014 \toc@setindent\@tempdima{subsection}%
2015 \toc@setindent\@tempdima{subsubsection}%
2016 \toc@setindent\@tempdima{paragraph}%
2017 \toc@letdimen{appendix}%
2018 \toc@letdimen{pagenum}%
2019 }%
2020 \def\toc@post@auto{%
2021 \if@filesw
2022 \begingroup
2023 \toc@writedimen{section}%
2024 \toc@writedimen{subsection}%
2025 \toc@writedimen{subsubsection}%
2026 \toc@writedimen{paragraph}%
2027 \toc@writedimen{appendix}%
2028 \toc@writedimen{pagenum}%
2029 \endgroup
2030 \fi
2031 }%

\toc@setindent
2032 \def\toc@setindent#1#2{%
2033 \csname tocdim@#2\endcsname\tocdim@min\relax
2034 \@ifundefined{tocmax@#2}{\@namedef{tocmax@#2}{\z@}}{}%
2035 \advance#1\@nameuse{tocmax@#2}\relax
2036 \expandafter\edef\csname tocleft@#2\endcsname{\the#1}%
2037 }%

\toc@letdimen
2038 \def\toc@letdimen#1{%
2039 \csname tocdim@#1\endcsname\tocdim@min\relax
2040 \@ifundefined{tocmax@#1}{\@namedef{tocmax@#1}{\z@}}{}%
2041 \expandafter\let\csname tocleft@#1\expandafter\endcsname\csname tocmax@#1\endcsname
2042 }%

```

`\toc@writedimen`

```

2043 \def\toc@writedimen#1{%
2044 \immediate\write\@auxout{%
2045 \gdef\expandafter\string\csname tocmax@#1\endcsname{%
2046 \expandafter\the\csname tocdim@#1\endcsname
2047 }%
2048 }%
2049 }%

```

`\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\@tempboxa` by `\set@tocdim@pagenum` must endure over the invocation of `#3`: it contains the page number which will be set just before the `\par`.

The arguments:

- `#1` superior section
- `#2` this section
- `#3` content, including possible `\numberline`
- `#4` page number

```

2050 \def\l@sections#1#2#3#4{%
2051 \begingroup
2052 \everypar{}%
2053 \set@tocdim@pagenum\@tempboxa{#4}%
2054 \global\@tempdima\csname tocdim@#2\endcsname
2055 \leftskip\csname toclef@#2\endcsname\relax
2056 \dimen@\csname toclef@#1\endcsname\relax
2057 \parindent-\leftskip\advance\parindent\dimen@
2058 \rightskip\toclef@pagenum plus 1fil\relax
2059 \skip@\parfillskip\parfillskip\z@
2060 \let\numberline\numberline@sections
2061 \@nameuse{l@f@#2}%
2062 \ignorespaces#3\unskip\nobreak\hskip\skip@
2063 \hb@xt@\rightskip{\hfil\unhbox\@tempboxa}\hskip-\rightskip\hskip\z@skip

```

By side effect, set the value of, e.g., `\tocdim@section`.

Note that the `\par` must not be executed before the value of `\@tempdima` is expanded (outside the current group). Otherwise, the `lineno.sty` package may interfere (it unfortunately does a global assignment of `\@tempdima`).

```

2064 \expandafter\par
2065 \expandafter\aftergroup\csname tocdim@#2%
2066 \expandafter\endcsname
2067 \expandafter\endgroup
2068 \the\@tempdima\relax
2069 }%

```

In the call to `\set@tocdim@pagenum`, I am now exposing the use of the particular box register.

```
2070 \def\set@tocdim@pagenum#1#2{%
2071   \setbox#1\hbox{\ignorespaces#2}%
2072   \@ifdim{\tocdim@pagenum<\wd#1}{\global\tocdim@pagenum\wd#1}{}%
2073 }%
```

`\numberline@@sections` The utility procedure 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!

```
2074 \def\numberline@@sections#1{%
2075   \leavevmode\hb@xt@-\parindent{%
2076     \hfil
2077     \@ifempty{#1}{}{%
2078       \setbox\z@\hbox{#1.\kern\ltxu@dotsep}%
2079       \@ifdim{\@tempdima<\wd\z@}{\global\@tempdima\wd\z@}{}%
2080       \unhbox\z@
2081     }%
2082   }%
2083   \ignorespaces
2084 }%
2085 \def\tocdim@min{\z@}%
```

## 6.18 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 L<sup>A</sup>T<sub>E</sub>X kernel procedure `\list`, which is the implementation of all lists.

```
2086 \def\list#1#2{%
2087   \ifnum \@listdepth >5\relax
2088     \@toodeep
2089   \else
2090     \global\advance\@listdepth\@ne
2091   \fi
2092   \rightmargin\z@
2093   \listparindent\z@
2094   \itemindent\z@
2095   \csname @list\romannumeral\the\@listdepth\endcsname
2096   \def\@itemlabel{#1}%
2097   \let\makelabel\@mklab
2098   \@nmblistfalse
2099   #2\relax
```

```

2100 \@trivlist
2101 \parskip\parsep
2102 \set@listindent
2103 \ignorespaces
2104 }%
2105 \def\set@listindent@parshape{%
2106 \parindent\listparindent
2107 \advance\@totalleftmargin\leftmargin
2108 \advance\linewidth-\rightmargin
2109 \advance\linewidth-\leftmargin
2110 \parshape\@ne\@totalleftmargin\linewidth
2111 }%
2112 \def\set@listindent@{%
2113 \parindent\listparindent
2114 \advance\@totalleftmargin\leftmargin
2115 \advance\rightskip\rightmargin
2116 \advance\leftskip\@totalleftmargin
2117 }%
2118 \let\set@listindent\set@listindent@parshape

```

## 6.19 Hypertext capabilities

`\href` We provide support for the `\href`, `\url`, and `\doi` commands. Packages, like `hyperref`, may override these definitions and provide better semantics.

```

\URL@prefix 2119 \providecommand\href[0]{\begingroup\@sanitize@url\@href}%
\doi 2120 \def\@href#1{\@startlink{#1}\endgroup\@@href}%
\doibase 2121 \def\@@href#1{#1\@endlink}%
2122 \providecommand \url [0]{\begingroup\@sanitize@url \@url }%
2123 \def \@url #1{\endgroup\@href {#1}{\URL@prefix#1}}%
2124 \providecommand \URL@prefix [0]{URL }%
2125 \providecommand\doi[0]{\begingroup\@sanitize@url\@doi}%
2126 \def\@doi#1{\endgroup\@startlink{\doibase#1}doi:\discretionary {}{}{}#1\@endlink }%
2127 %changes{4.2a}{2017/11/21}{(MD) Use updated best practice to use https and doi.org}%
2128 \providecommand \doibase [0]{https://doi.org/}%
2129 \providecommand \@sanitize@url[0]{\chardef\cat@space\the\catcode'\@sanitize\catcode'\@cat@sp

```

`\@startlink` How we define `\@startlink` and `\@endlink` will depend on whether we are running under PDFLATEX. If so, and if PDF output is requested, then we use its primitives to implement hypertext, breaking out the link attributes in `\pdfstartlink@attr` and using the `hyperref` defaults; `\pdfstartlink@attr` can be redefined by a client package. Otherwise we fall back the HyperTeX standard and leave things to the DVI translator.

A class or package that wishes to employ hypertext capabilities should execute the `\hypertext@enable@ltx` procedure.

```

2130 \def\@startlink#1{%
2131 \def\@endlink{}%
2132 \@ifundefined \pdfoutput {\true@sw}{\ifnum{\z@=\pdfoutput}{\true@sw}{\false@sw}}%
2133 {%

```

```

2134 \def\@@startlink@hypertext#1{\leavevmode\special{html:<a href="#1">}}%
2135 \def\@@endlink@hypertext{\special{html:</a>}}%
2136 }{%
2137 \def\@@startlink@hypertext#1{%
2138   \leavevmode
2139   \pdfstartlink\pdfstartlink@attr
2140     user{/Subtype/Link/A<</Type/Action/S/URI/URI(#1)>>}}%
2141   \relax
2142 }%
2143 \def\@@endlink@hypertext{\pdfendlink}%
2144 \def\pdfstartlink@attr{attr{/Border[0 0 1 ]/H/I/C[0 1 1]}}%
2145 }%
2146 \def\hypertext@enable@ltx{%
2147   \let\@@startlink\@@startlink@hypertext
2148   \let\@@endlink\@@endlink@hypertext
2149 }%

```

`\href` The `\href` command of `hyperref` was extend somewhere between versions 6.75r and 6.80e. We apply a repair to the earlier version (if present) so that it works like the later version.

The issue is the presence of whitespace, either following the `\href` token or following the first argument's closing brace character.

```

2150 \def\href@Hy{\hyper@normalise \href@ }%
2151 \def\href@Hy@ltx{\@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}}%
2152 \def\Hy@href#{\hyper@normalise\href@}%
2153 \begingroup
2154   \endlinechar=-1 %
2155   \catcode'\^A=14 %
2156   \catcode'\^M\active
2157   \catcode'\%\active
2158   \catcode'\#\active
2159   \catcode'\_\active
2160   \catcode'\$\active
2161   \catcode'\&\active
2162   \gdef\hyper@normalise@ltx{^A
2163     \begingroup
2164       \catcode'\^M\active
2165       \def^M{ }^A
2166       \catcode'\%\active
2167       \let%\@percentchar
2168       \let%\@percentchar
2169       \catcode'\#\active
2170       \def#{\hyper@hash}^A
2171       \def\#{\hyper@hash}^A
2172       \@makeother\&^A
2173       \edef&{\string&}^A
2174       \edef&{\string&}^A
2175       \edef\textunderscore{\string_}^A
2176       \let\_ \textunderscore

```

```

2177 \catcode'\_ \active
2178 \let_\textunderscore
2179 \let~\hyper@tilde
2180 \let\~\hyper@tilde
2181 \let\textasciitilde\hyper@tilde
2182 \let\\\@backslashchar
2183 \edef$\string$^^A
2184 \Hy@safe@activetrue
2185 \hyper@n@rmalise
2186 }^^A
2187 \catcode'\#=6 ^^A
2188 \gdef\Hy@ActiveCarriageReturn@ltx{^^M}^^A
2189 \gdef\hyper@n@rmalise@ltx#1#2{^^A
2190 \def\Hy@tempa{#2}^^A
2191 \ifx\Hy@tempa\Hy@ActiveCarriageReturn
2192 \Hy@ReturnAfterElseFi{^^A
2193 \hyper@@normalise{#1}^^A
2194 }^^A
2195 \else
2196 \Hy@ReturnAfterFi{^^A
2197 \hyper@@normalise{#1}{#2}^^A
2198 }^^A
2199 \fi
2200 }^^A
2201 \gdef\hyper@@normalise@ltx#1#2{^^A
2202 \edef\Hy@tempa{^^A
2203 \endgroup
2204 \noexpand#1{\Hy@RemovePercentCr#2%^^M\@nil}^^A
2205 }^^A
2206 \Hy@tempa
2207 }^^A
2208 \gdef\Hy@RemovePercentCr@ltx#1%^^M#2\@nil{^^A
2209 #1^^A
2210 \ifx\limits#2\limits
2211 \else
2212 \Hy@ReturnAfterFi{^^A
2213 \Hy@RemovePercentCr #2\@nil
2214 }^^A
2215 \fi
2216 }^^A
2217 \endgroup
2218 \def\switch@hyperref@href{%
2219 \expandafter\@ifx\expandafter{\csname href \endcsname\href@Hy}{
2220 \class@info{Repairing hyperref 6.75r \string\href}%
2221 \let\hyper@normalise\hyper@normalise@ltx
2222 \let\hyper@@normalise\hyper@@normalise@ltx
2223 \let\hyper@n@rmalise\hyper@n@rmalise@ltx
2224 \let\Hy@ActiveCarriageReturn\Hy@ActiveCarriageReturn@ltx
2225 \let\Hy@RemovePercentCr\Hy@RemovePercentCr@ltx
2226 \let\href\href@Hy@ltx

```

```

2227 }{}%
2228 }%
2229 \appdef\document@inithook{\switch@hyperref@href}%

\typeout We make the \typeout procedure of LATEX be \long, because sometimes we are
         talking about \par.
2230 \def\typeout@org#1{%
2231   \begingroup
2232   \set@display@protect
2233   \immediate\write\@unused{#1}%
2234   \endgroup
2235 }%
2236 \long\def\typeout@ltx#1{%
2237   \begingroup
2238   \set@display@protect
2239   \immediate\write\@unused{#1}%
2240   \endgroup
2241 }%
2242 \@ifx{\typeout\typeout@org}{%
2243   \let\typeout\typeout@ltx
2244   \true@sw
2245 }{%
2246   \rvtx@ifformat@geq{2020-10-01}%
2247   {\true@sw}{\false@sw}%
2248 }%
2249 {\class@info{Making \string\typeout\space \string\long}}%
2250 {}%

```

## 6.20 End of the kernel DOCSTRIP module

Here ends the module.

```
2251 %</kernel>
```



# Index

## Symbols

<code>\#</code> .....	2158, 2169, 2171, 2187
<code>\\$</code> .....	2160
<code>\$TEXMF/</code> .....	<a href="#">3</a>
<code>\%</code> .....	901, 2157, 2166, 2168
<code>\&amp;</code> .....	2161, 2172, 2174
<code>\(</code> .....	666
<code>\)</code> .....	667
<code>.aux</code> .....	<a href="#">41</a>
<code>.cls</code> .....	<a href="#">20</a>
<code>.dtx</code> .....	<a href="#">5</a>
<code>.rtx</code> .....	<a href="#">19</a>
<code>\@</code> .....	479, 487
<code>\@array</code> .....	1230, 1296, 1299
<code>\@end</code> .....	374, 467
<code>\@endlink</code> .....	<a href="#">69</a>
<code>\@endlink</code> ....	2121, 2126, <a href="#">2130</a>
<code>\@endlink@hypertext</code> ...	2135, 2143, 2148
<code>\@endpbox</code> .....	1397, 1420, 1526, 1535
<code>\@href</code> .....	2120, 2121
<code>\@par</code> .....	796, 972, 1012, 1403, 1427, 1553
<code>\@startlink</code> .....	<a href="#">69</a>
<code>\@startlink</code> ..	2120, 2126, <a href="#">2130</a>
<code>\@startlink@hypertext</code> ..	2134, 2137, 2147
<code>\@startpbox</code> ..	1397, 1419, 1526, 1535
<code>\@Esphack</code> .....	889, 891
<code>\@M</code> .....	<a href="#">48</a>
<code>\@MM</code> .....	<a href="#">28</a>
<code>\@acol</code> .....	<a href="#">58</a>
<code>\@acol</code> .....	1258, 1267, 1284, 1353, 1364, 1380, 1654, 1655, 1733, 1735, 1833, 1879, 1946, 1949, 1957
<code>\@acolampacol</code> .....	.. 1831, 1833, 1841, 1849, 1869, 1877, 1893, 1901, 1945, 1947, 1955, 1959
<code>\@acoll</code> ..	1265, 1282, 1362, 1378, 1851, 1903, 1962
<code>\@acolr</code> .....	<a href="#">58</a>
<code>\@acolr</code> ..	1266, 1283, 1363, 1379, 1684, 1692, 1782, 1785
<code>\@addamp</code> ..	1065, 1155, <a href="#">1806</a> , 1832, 1847, 1875, 1899, 1946, 1950, 1957, 1964
<code>\@addamp@LaTeX</code> .....	1065, 1806
<code>\@addamp@ltx</code> .....	1155, 1809
<code>\@addtopreamble</code> .....	.. 1641, 1737, 1789, 1810, 1816, 1819, 1825, 1828, 1854, 1858, 1862, 1906, 1910, 1914, 1920, 1923, 1926, 1929, 1932, 1936
<code>\@afterindentfalse</code> .....	<a href="#">933</a>
<code>\@afterindenttrue</code> .....	<a href="#">931</a>
<code>\@ampacol</code> .....	1831, 1843, 1871, 1895
<code>\@argarraycr</code> .....	1325, 1329
<code>\@argswap</code> .....	<a href="#">14</a>
<code>\@argswap</code> .....	<a href="#">298</a>
<code>\@argswap@val</code> .....	<a href="#">14</a>
<code>\@argswap@val</code> .....	<a href="#">298</a> , 575
<code>\@argtabularcr</code> .....	1307, 1308
<code>\@array</code> .....	<a href="#">50</a>
<code>\@array</code> ..	1057, 1147, 1172, 1229, 1230, 1290, 1293, <a href="#">1386</a>
<code>\@array@LaTeX</code> .....	1057, 1386
<code>\@array@align@</code> .....	<a href="#">1542</a>
<code>\@array@align@b</code> .....	1543
<code>\@array@align@c</code> .....	<a href="#">54</a>
<code>\@array@align@c</code> ....	1544, 1559
<code>\@array@align@t</code> .....	1542
<code>\@array@align@v</code> .....	<a href="#">54</a>
<code>\@array@align@v</code> ....	1545, 1559
<code>\@array@array</code> .....	1172, <a href="#">1439</a>
<code>\@array@array@new</code> ..	1229, 1461
<code>\@array@ltx</code> .....	1147, 1406
<code>\@array@sw</code> .....	<a href="#">46</a>
<code>\@array@sw</code> ....	1056, 1171, <a href="#">1253</a>
<code>\@array@sw@LaTeX</code> ...	1171, 1253
<code>\@array@sw@array</code> ...	1056, 1254
<code>\@arrayacol</code> .....	.. 1066, 1156, 1238, 1353, 1362–1364, 1378–1380, <a href="#">1812</a>

<code>\@arrayacol@LaTeX</code> . . .	1066, 1812	<code>\@bsphack</code> . . . . .	679, 894
<code>\@arrayacol@ltx</code> . . .	1156, 1238, 1815	<code>\@caption</code> . . . . .	<a href="#">31</a>
<code>\@arrayclassiv</code> . . .	1070, 1160, 1355, 1366, <a href="#">1925</a>	<code>\@capttype</code> . . . . .	<a href="#">35</a>
<code>\@arrayclassiv@LaTeX</code> . . .	1070, 1925	<code>\@capttype</code> . . . . .	750
<code>\@arrayclassiv@ltx</code> . . .	1160, 1928	<code>\@centercr</code> . . . . .	<a href="#">21</a>
<code>\@arrayclassz</code> . . .	1068, 1158, 1354, 1365, <a href="#">1830</a>	<code>\@centercr</code> . . . . .	<a href="#">498</a>
<code>\@arrayclassz@LaTeX</code> . . .	1068, 1830	<code>\@centering</code> . . . . .	505, 508, 511, 530, 536, 549
<code>\@arrayclassz@ltx</code> . . .	1158, 1839	<code>\@chclass</code> . . . . .	1652, 1653, 1668, 1681, 1726, 1731, 1757, 1779
<code>\@arraycr</code> . . . . .	<a href="#">48</a>	<code>\@checkend</code> . . . . .	370, 412, 424, 428
<code>\@arraycr</code> . . . . .	1181, 1242, <a href="#">1301</a> , 1356, 1367, 1459, 1485	<code>\@chnum</code> . . . . .	1835, 1853, 1883, 1905
<code>\@arraycr@array</code> . . . . .	1181, 1313	<code>\@classi</code> . . . . .	1652, 1671, 1726, 1760
<code>\@arraycr@new</code> . . . . .	1242, 1318	<code>\@classii</code> . . . . .	1652, 1673, 1726, 1762
<code>\@arrayleft</code> . . . . .	1452, 1482	<code>\@classiii</code> . . . . .	1652, 1675
<code>\@arrayright</code> . . . . .	1502, 1506	<code>\@classiv</code> . . . . .	1260, 1269, 1355, 1366, 1653, 1677
<code>\@arstrut</code> . . . . .	<a href="#">55</a>	<code>\@classv</code> . . . . .	1072, 1162, 1653, 1679, 1727, 1767, <a href="#">1931</a>
<code>\@arstrut</code> . . . . .	1396, 1412, 1449, 1475, 1527, 1536, 1609, 1611, 1628, <a href="#">1639</a>	<code>\@classv@LaTeX</code> . . . . .	1072, 1931
<code>\@arstrut@hline</code> . . . . .	<a href="#">55</a>	<code>\@classv@ltx</code> . . . . .	1162, 1935
<code>\@arstrut@hline</code> . . . . .	<a href="#">1603</a>	<code>\@classvi</code> . . . . .	1727, 1769
<code>\@arstrut@hline@clnc</code> . . . . .	<a href="#">55</a>	<code>\@classvii</code> . . . . .	1728, 1771
<code>\@arstrut@hline@clnc</code> . . . . .	1618, 1630	<code>\@classviii</code> . . . . .	1728, 1773
<code>\@arstrut@hook</code> . . . . .	<a href="#">55</a>	<code>\@classx</code> . . . . .	1179, 1237, 1729, 1730, 1775, 1777, <a href="#">1943</a>
<code>\@arstrut@hook</code> . . . . .	1589, 1590, <a href="#">1603</a> , 1639, 1640	<code>\@classx@array</code> . . . . .	1179, 1943
<code>\@arstrut@org</code> . . . . .	<a href="#">1603</a>	<code>\@classx@array@new</code> . . . . .	1237, 1953
<code>\@arstrutbox</code> . . . . .	<a href="#">50</a> , <a href="#">55</a>	<code>\@classz</code> . . . . .	1259, 1268, 1354, 1365, 1652, 1669, 1726, 1758
<code>\@arstrutbox</code> . . . . .	. . . . .	<code>\@clsextension</code> . . . . .	<a href="#">19</a> , <a href="#">21</a>
. . . . .	1309, 1310, 1333, 1339, 1389, 1433, 1442, 1464	<code>\@clsextension</code> . . . . .	461, 478, 486
<code>\@arstrutbox@hline</code> . . . . .	<a href="#">55</a>	<code>\@ctrerr</code> . . . . .	925
<code>\@arstrutbox@hline</code> . . . . .	<a href="#">1603</a>	<code>\@currbox</code> . . . . .	<a href="#">35</a> , <a href="#">36</a>
<code>\@auxout</code> . . . . .	671, 680, 2044	<code>\@currbox</code> . . . . .	841, 844, 846, 847, 853, 854, 863
<code>\@backslashchar</code> . . . . .	2182	<code>\@currentHref</code> . . . . .	616, 1038
<code>\@boole@def</code> . . . . .	<a href="#">14</a>	<code>\@currentlabel</code> . . . . .	501, 517, 643, 681
<code>\@boole@def</code> . . . . .	<a href="#">308</a> , 320–323, 325–335	<code>\@currentlabelname</code> . . . . .	<a href="#">40</a>
<code>\@boolean</code> . . . . .	<a href="#">14</a>	<code>\@currentxt</code> . . . . .	<a href="#">19</a>
<code>\@boolean</code> . . . . .	<a href="#">308</a>	<code>\@currentxt</code> . . . . .	443, 460, 461, 477, 485
<code>\@booleanfalse</code> . . . . .	<a href="#">14</a>	<code>\@currname</code> . . . . .	<a href="#">19</a>
<code>\@booleanfalse</code> . . . . .	<a href="#">318</a> , 804, 849	<code>\@currname</code> . . . . .	443, 459, 476, 484
<code>\@booleantrue</code> . . . . .	<a href="#">14</a>	<code>\@currnamestack@ltx</code> . . . . .	471, 475, 480, 488, 490
<code>\@booleantrue</code> . . . . .	<a href="#">318</a> , 734, 847	<code>\@dblarg</code> . . . . .	941, 942

<code>\@dblfloat</code> . . . . .	785, 787, 809	<code>\@hangfrom@section</code> . . . . .	<a href="#">39</a>
<code>\@dblfloat@LaTeX</code> . . . . .	785, 809	<code>\@height</code> . . . . .	1309, 1310, 1390, 1434, 1443, 1465, 1567, 1620, 1627
<code>\@defaultsubs</code> . . . . .	393	<code>\@href</code> . . . . .	2119, 2120, 2123
<code>\@depth</code> . . . . .	1309, 1310, 1334, 1340, 1391, 1435, 1444, 1466, 1621	<code>\@ialph</code> . . . . .	920, 921
<code>\@dofilelist</code> . . . . .	386	<code>\@ifempty</code> . . . . .	<a href="#">320</a> , 2077
<code>\@doi</code> . . . . .	2125, 2126	<code>\@if@sw</code> . . . . .	323, 324, 378, 395, 397
<code>\@eha</code> . . . . .	203	<code>\@ifcat</code> . . . . .	<a href="#">320</a>
<code>\@enddocumenthook</code> . . . . .	369	<code>\@ifdim</code> . . . . .	<a href="#">320</a> , 387, 866, 965, 1005, 1619, 2072, 2079
<code>\@endpbox</code> . . . . .	<a href="#">58</a>	<code>\@ifempty</code> . . . . .	<a href="#">14</a>
<code>\@endpbox</code> 1397, 1420, 1526, 1535, 1647, 1663, 1717, 1748, 1933, 1940		<code>\@ifeof</code> . . . . .	<a href="#">320</a>
<code>\@eqcnt</code> . . . . .	504, 509, 510, 512, 518, 531, 534, 537	<code>\@ifhbox</code> . . . . .	<a href="#">320</a> , 862, 881
<code>\@eqncr</code> . . . . .	<a href="#">48</a>	<code>\@ifhmode</code> . . . . .	<a href="#">320</a>
<code>\@eqncr</code> . . . . .	506, 520	<code>\@ifinner</code> . . . . .	<a href="#">320</a> , 713, 1550
<code>\@eqnset</code> . . . . .	508, 530	<code>\@ifl@t@r</code> . . . . .	304
<code>\@eqnswtrue</code> . . . . .	502, 518	<code>\@ifl@ter</code> . . . . .	463
<code>\@eqpen</code> . . . . .	<a href="#">48</a>	<code>\@ifmmode</code> . . . . .	<a href="#">320</a> , 1361, 1377, 1544, 1546, 1605
<code>\@esphack</code> . . . . .	683	<code>\@ifnextchar</code> . . . . .	558, 567, 582, 1290, 1293, 1296, 1299, 1307, 1308, 1324, 1328, 2151
<code>\@expast</code> . . . . .	1648, 1664	<code>\@ifnotempty</code> . . . . .	<a href="#">14</a>
<code>\@extension</code> . . . . .	<a href="#">19</a>	<code>\@ifnotrelax</code> . . . . .	<a href="#">13</a>
<code>\@finalstrut</code> . . . . .	649	<code>\@ifnotrelax</code> . . . . .	<a href="#">298</a> , 686, 861, 880
<code>\@firstampfalse</code> . . . . .		<code>\@ifnum</code> . . . . .	<a href="#">320</a> , 846, 947, 976, 990, 1975, 2132
. . . . .	1807, 1810, 1833, 1851, 1879, 1903, 1949, 1962	<code>\@ifodd</code> . . . . .	<a href="#">320</a> , 1977
<code>\@firstamptrue</code> 1643, 1658, 1716, 1745		<code>\@ifpackageloaded</code> . . . . .	222, 607, 1046, 1170
<code>\@firstofone</code> . . . . .	368	<code>\@ifstar</code> . . . . .	940, 1305, 1306, 1316, 1321
<code>\@float</code> . . . . .	<a href="#">30</a> , <a href="#">34</a> , <a href="#">36</a>	<code>\@ifundefined</code> . . . . .	956, 969, 974, 985, 988, 1009, 1014, 1019, 1022, 2034, 2040
<code>\@float@LaTeX</code> . . . . .	783, 784, 808	<code>\@ifvbox</code> . . . . .	<a href="#">320</a>
<code>\@floatboxreset</code> . . . . .	756	<code>\@ifvmode</code> . . . . .	<a href="#">320</a> , 552
<code>\@flushglue</code> . . . . .	496, 1632	<code>\@ifvoid</code> . . . . .	<a href="#">320</a> , 710, 721
<code>\@font@warning</code> . . . . .	388	<code>\@ifx</code> . . . . .	<a href="#">13</a> , <a href="#">14</a>
<code>\@footnotemark</code> . . . . .	<a href="#">26</a>	<code>\@ifx</code> . . . . .	298, 299, 302, <a href="#">320</a> , 341, 396, 541, 620, 908, 1057–1077, 1172–1184, 1253, 1254, 1980, 2219, 2242
<code>\@footnotetext</code> . . . . .	<a href="#">24</a> , <a href="#">26</a> , <a href="#">27</a>	<code>\@ifx@empty</code> . . . . .	<a href="#">320</a>
<code>\@footnotetext</code> 588, 608, 609, <a href="#">624</a>		<code>\@ifxundefined</code> . . . . .	<a href="#">13</a>
<code>\@gobble@opt@one</code> . . . . .	<a href="#">64</a>		
<code>\@gobble@opt@one</code> . . . . .	<a href="#">1986</a>		
<code>\@halignto</code> 1357, 1368, 1373, 1383, 1395, 1429, 1448, 1493			
<code>\@hang@from</code> . . . . .	969, 1000		
<code>\@hang@froms</code> . . . . .	1009, 1028		
<code>\@hangfrom</code> . . . . .	<a href="#">38–40</a>		
<code>\@hangfrom</code> . . . . .	1000		

<code>\@ifxundefined</code> . . . 270, 275, 280, 298, 783, 2132	<code>\@nameuse</code> . 219, 738, 1407, 1483, 2035, 2061
<code>\@ifxundefined@cs</code> 302, 791, 802, 822	<code>\@ne</code> . . . . . 12, 68
<code>\@iiiparbox</code> . . . . . 732	<code>\@newl@bel</code> . . . . . 381
<code>\@itemlabel</code> . . . . . 2096	<code>\@nextchar</code> 1650, 1651, 1665, 1667, 1723, 1754, 1920, 1923, 1926, 1929, 1932, 1939
<code>\@lastchclass</code> . . . . . . . 1643, 1653, 1654, 1659, 1681, 1683, 1716, 1731, 1732, 1744, 1779, 1781, 1831, 1840, 1868, 1892, 1944, 1954	<code>\@nmbrlistfalse</code> . . . . . 2098
<code>\@latex@warning@no@line</code> . . 398	<code>\@nobreakfalse</code> . . . . . 1996
<code>\@listdepth</code> . . . 2087, 2090, 2095	<code>\@normalcr</code> . . . . . 21
<code>\@mainaux</code> . . . . . 16	<code>\@normalcr</code> . . . . . 492
<code>\@mainaux</code> . . . . . 379	<code>\@nul</code> . . . . . 906, 917
<code>\@makecol</code> . . . . . 10	<code>\@onefilewithoptions</code> . . . . 19
<code>\@makefnmark</code> . . . . . 24, 28	<code>\@outputdblcol</code> . . . . . 10
<code>\@makefnmark</code> . . . . 576, 624, 770	<code>\@p@pfilename</code> . . . . . 474
<code>\@makefntext</code> . . . . . 28	<code>\@p@pfilename@ltx</code> . . . . 471, 483
<code>\@makefntext</code> . . . . . 647	<code>\@parboxrestore</code> . . . . . 642, 755
<code>\@makeoother</code> . . . . 901, 905, 2172	<code>\@pass@options</code> . . . . . 449
<code>\@minipagefalse</code> . . . . . 729	<code>\@percentchar</code> . . . . . 2167, 2168
<code>\@mklab</code> . . . . . 2097	<code>\@preamble</code> 1394, 1396, 1404, 1410, 1414, 1429, 1448, 1449, 1459, 1471, 1473, 1477, 1493, 1502, 1507, 1527, 1537, 1641, 1644, 1660, 1716, 1743, 1807, 1813, 1822, 1834, 1881, 1882
<code>\@mkpream</code> . . . . . 58, 60	<code>\@preamerr</code> 1403, 1427, 1655, 1687, 1689, 1736, 1737, 1739, 1787, 1789, 1793
<code>\@mkpream</code> 1064, 1154, 1178, 1236, 1393, 1409, 1447, 1470, 1524, 1532, 1642, 1715	<code>\@provide</code> . 340, 1048, 1050, 1053
<code>\@mkpream@LaTeX</code> . . . . 1064, 1642	<code>\@pushfilename</code> . . . . . 474
<code>\@mkpream@array</code> . . . . 1178, 1715	<code>\@pushfilename@ltx</code> . . . 454, 474
<code>\@mkpream@array@new</code> 1236, 1742	<code>\@refundefined</code> . . . . . 394
<code>\@mkpream@ltx</code> . . . . . 1154, 1657	<code>\@reset@options</code> . . . . . 20
<code>\@mkpream@relax</code> . . . . . 58	<code>\@reset@options</code> . . . . . 457, 472
<code>\@mkpream@relax</code> 1707, 1747, 1797	<code>\@resetactivechars</code> . . . . . 262
<code>\@mpargs</code> . . . . . 732	<code>\@rightskip</code> . . . . . 495
<code>\@mpfn</code> . . . . . 24, 28	<code>\@runin@to</code> . . . . . 39
<code>\@mpfn</code> . . . 561, 565, 574, 587, 593, 597, 600, 603, 644, 743	<code>\@runin@to</code> . . . . . 985, 1001
<code>\@mpfootins</code> . . . . . 32	<code>\@runin@tos</code> . . . . . 1019, 1029
<code>\@mpfootins</code> . 699, 702, 703, 710, 711, 714, 717, 721, 722	<code>\@sanitize</code> . . . . . 2129
<code>\@mpfootnotetext</code> . . . . . 26, 27	<code>\@sanitize@url</code> 2119, 2122, 2125, 2129
<code>\@mpfootnotetext</code> . 610, 611, 624	<code>\@secCNTformat</code> . . . . . 38, 39
<code>\@mpmakefntext</code> . . . . . 735	<code>\@secCNTformat</code> . . . . . 957
<code>\@multiplelabels</code> . . . . 396, 404	<code>\@secpenalty</code> . . . . . 938
<code>\@namedef</code> . 815, 1521, 2034, 2040	<code>\@sect</code> . . . . . 38

<code>\@sect</code> .....	945	<code>\@tabclassz</code> ..	1071, 1161, 1259,
<code>\@sect@ltx</code> .....	942, 946		1268, <u>1867</u>
<code>\@setckpt</code> .....	380	<code>\@tabclassz@LaTeX</code> ..	1071, 1867
<code>\@sharp</code> ..	1400, 1423, 1455, 1488,	<code>\@tabclassz@ltx</code> ....	1161, 1891
	1525, 1533, 1646, 1662, 1697,	<code>\@tabular</code> .....	<u>46</u>
	1703, 1717, 1737, 1746, 1789,	<code>\@tabular</code>	1052, 1059, 1149, 1173,
	1836, 1837, 1855, 1859, 1863,		1231, <u>1255</u>
	1884, 1886, 1888, 1907, 1911,	<code>\@tabular@LaTeX</code> ....	1059, 1255
	1915, 1933, 1940	<code>\@tabular@array</code> ....	1173, 1273
<code>\@ssect</code> .....	<u>38</u> , <u>40</u>	<code>\@tabular@array@new</code>	1231, 1281
<code>\@ssect</code> .....	<u>1002</u>	<code>\@tabular@ltx</code> .....	1149, 1264
<code>\@ssect@ltx</code> .....	941, 1002	<code>\@tabularcr</code> .....	<u>48</u>
<code>\@startcolumn</code> .....	<u>10</u>	<code>\@tabularcr</code> ..	1074, 1164, 1253,
<code>\@startpbox</code> .....	<u>58</u>		1261, 1270, <u>1301</u>
<code>\@startpbox</code> .....		<code>\@tabularcr@LaTeX</code> .....	<u>48</u>
	.. 1397, 1419, 1526, 1535,	<code>\@tabularcr@LaTeX</code> ..	1074, 1305
	1647, 1663, 1713, 1717, 1748,	<code>\@tabularcr@ltx</code> ....	1164, 1306
	1932, 1938	<code>\@tbpopen</code> .....	<u>48</u>
<code>\@startsection</code> .....	<u>38</u> , <u>40</u>	<code>\@tbpopen</code> .....	<u>1301</u>
<code>\@startsection</code> .....	<u>926</u>	<code>\@temp@sw</code> .....	847, 849, 853
<code>\@startsection@hook</code> ..	927, 944	<code>\@tempa</code> .....	223, 225, 859, 861,
<code>\@starttoc</code> .....	<u>66</u>		862, 867, 868, 870, 871, 878,
<code>\@starttoc</code> .....	<u>1987</u>		880–883, 907, 908
<code>\@svsec</code> .....	<u>39</u>	<code>\@tempb</code> .....	898, 908
<code>\@svsec</code> ..	949, 955, 957, 960, 970,	<code>\@tempboxa</code> .....	<u>67</u>
	986	<code>\@tempc</code> .....	896, 914
<code>\@svsechd</code> .....	983, 1017	<code>\@tempcnta</code> .....	<u>36</u>
<code>\@tabacol</code>	1067, 1157, 1239, 1258,	<code>\@tempcntb</code>	883, 1967, 1970, 1971,
	1267, 1284, <u>1818</u>		1974, 1976, 1977
<code>\@tabacol@LaTeX</code> ....	1067, 1821	<code>\@tempdima</code> .....	<u>67</u> , <u>68</u>
<code>\@tabacol@ltx</code> ..	1157, 1239, 1824	<code>\@tempskipa</code> ..	930, 932, 933, 938,
<code>\@tabacoll</code> ....	1265, 1282, 1818		964, 965, 1004, 1005
<code>\@tabacolr</code> ....	1266, 1283, 1827	<code>\@temptokena</code> ..	1718, 1724, 1749,
<code>\@tabarray</code> .....	<u>47</u>		1754
<code>\@tabarray</code>	1060, 1150, 1174, 1232,	<code>\@testdef</code> .....	381
	1262, 1271, 1279, 1287, <u>1289</u> ,	<code>\@testpach</code>	1651, 1667, 1725, 1756
	1358, 1369, 1374, 1384	<code>\@testtrue</code> .....	1971, 1977
<code>\@tabarray@LaTeX</code> ...	1060, 1289	<code>\@textsuperscript</code> .....	626
<code>\@tabarray@array</code> ...	1174, 1295	<code>\@tfor</code> ...	1649, 1665, 1723, 1754
<code>\@tabarray@array@new</code>	1232, 1298	<code>\@thefnmark</code> ..	594, 601, 627, 644
<code>\@tabarray@ltx</code> .....	1150, 1292	<code>\@toodeep</code> .....	2088
<code>\@tabclassiv</code> ..	1069, 1159, 1260,	<code>\@totalleftmargin</code> ..	2107, 2110,
	1269, <u>1919</u>		2114, 2116
<code>\@tabclassiv@LaTeX</code> ..	1069, 1919	<code>\@tpfootnotetext</code> .....	<u>624</u>
<code>\@tabclassiv@ltx</code> ...	1159, 1922	<code>\@trivlist</code> .....	2100
		<code>\@twopowerfourteen</code> .....	<u>35</u>

<code>\@twopowerfourteen</code> . . .	837, 865	<code>\@xxxii</code> . . . . .	842, 843, 845
<code>\@twopowertwo</code> . . . . .	<a href="#">35</a>	<code>\@yargarraycr</code> . . . . .	<a href="#">48</a>
<code>\@twopowertwo</code> . . . . .	838, 882, 883	<code>\@yargarraycr</code> .	1077, 1167, 1184,
<code>\@undefined</code> . . . . .	<a href="#">46</a>	1245, <a href="#">1301</a>	
<code>\@undefined</code> . . . . .	32, 470	<code>\@yargarraycr@LaTeX</code>	1077, 1311
<code>\@unexpandable@protect</code> . .	1645,	<code>\@yargarraycr@array</code>	1184, 1344
1661		<code>\@yargarraycr@ltx</code> . .	1167, 1312
<code>\@unused</code> . . . . .	2233, 2239	<code>\@yargarraycr@new</code> . .	1245, 1348
<code>\@unusedoptionlist</code> . . .	449, 450	<code>\[</code> . . . . .	<a href="#">23</a> , <a href="#">24</a>
<code>\@url</code> . . . . .	2122, 2123	<code>\[</code> . . . . .	905
<code>\@whilesw</code> . . . . .	1719, 1750	<code>\backslash</code> . . . . .	<a href="#">21</a> , <a href="#">48</a>
<code>\@width</code> . . . . .	1309, 1310, 1334, 1340,	<code>\{</code> . . . . .	133, 901, 905
1392, 1436, 1445, 1467, 1622		<code>\}</code> . . . . .	133, 901, 905
<code>\@write@floatline</code> . . . . .	<a href="#">37</a>	<code>\]</code> . . . . .	<a href="#">23</a>
<code>\@write@floatline</code> . . . . .	<a href="#">904</a>	<code>\]</code> . . . . .	905
<code>\@writefile</code> . . . . .	671	<code>\^</code> . . . . .	900, 913, 2155, 2156, 2164
<code>\@x@sf</code> . . . . .	572, 579	<code>\_</code> . . . . .	52, 2159, 2176, 2177
<code>\@xargarraycr</code> . . . . .	<a href="#">48</a>	<code>\~</code> . . . . .	2180
<code>\@xargarraycr</code> .	1076, 1166, 1183,		
1244, <a href="#">1301</a>		<code>\_</code> . . . . .	41, 47, 67, 70, 71, 75, 92, 94,
<code>\@xargarraycr@LaTeX</code>	1076, 1309	100, 105, 112, 223, 2129	
<code>\@xargarraycr@array</code>	1183, 1331		
<code>\@xargarraycr@ltx</code> . .	1166, 1310		
<code>\@xargarraycr@new</code> . .	1244, 1337		
<code>\@xarraycr</code> . . . . .	<a href="#">48</a>		
<code>\@xarraycr</code> . . . . .	1182, 1243, <a href="#">1301</a>		
<code>\@xarraycr@array</code> . . .	1182, 1323		
<code>\@xarraycr@new</code> . . . . .	1243, 1327		
<code>\@xbitor</code> . . . . .	<a href="#">1967</a>		
<code>\@xbitor@LaTeX</code> . . . . .	1967, 1980		
<code>\@xbitor@ltx</code> . . . . .	1973, 1985		
<code>\@xfloat</code> . . . . .	<a href="#">735</a>		
<code>\@xfloat@LaTeX</code> . . . . .	735, 739		
<code>\@xfloat@anchored</code> . . . .	739, 749		
<code>\@xfloat@prep</code> . . . . .	737, 741		
<code>\@xfootnotemark</code> . . . . .	<a href="#">24</a>		
<code>\@xfootnotenext</code> . . . . .	<a href="#">27</a>		
<code>\@xhline</code> . . . . .	1568, 1574, 1575		
<code>\@xhline@unneeded</code> . . . . .	1577		
<code>\@xifempty</code> . . . . .	<a href="#">14</a>		
<code>\@xsect</code> . . . . .	998, 1026		
<code>\@xtabularcr</code> . . . . .	<a href="#">48</a>		
<code>\@xtabularcr</code> . . . . .	1075, 1165, <a href="#">1301</a>		
<code>\@xtabularcr@LaTeX</code> .	1075, 1307		
<code>\@xtabularcr@ltx</code> . . .	1165, 1308		
<code>\@xvi</code> . . . . .	836, 846, 847		

  

<code>\abovedisplayshortskip</code> . . .	526
<code>\abovedisplayskip</code> . . . .	521–526
<code>\active</code> . . . . .	900, 2156–2161, 2164,
2166, 2169, 2177	
<code>\addcontentsline</code> . . . . .	<a href="#">40</a>
<code>\addcontentsline</code> .	<a href="#">673</a> , 975, 989,
1015, 1023	
<code>\addtocontents</code> . . . . .	<a href="#">661</a> , 674
<code>\AddToHook</code> . . . . .	346, 410
<code>\aftergroup</code> . . . . .	<a href="#">54</a>
<code>\aftergroup</code> . . . . .	1544, 1551, 2065
<code>\AfterLastShipout</code> . . . . .	<a href="#">17</a> , <a href="#">19</a>
<code>\appdef</code> . . . . .	<a href="#">13</a> , <a href="#">16</a> , <a href="#">17</a> , <a href="#">19</a> , <a href="#">30</a>
<code>\appdef</code> . . . . .	262,
269, 355, 438, 605, 691,	
695, 1041, 1045, 1304, 1414,	
1477, 1615, 1639, 1641, 1797,	
2229	
<code>\appdef@e</code> . . . . .	287
<code>\appdef@eval</code> . . . . .	292
<code>\appdef@val</code> . . . . .	284, 293
<code>aps.rtx</code> . . . . .	<a href="#">19</a>

argument  
  class ..... 19  
  float ..... 37  
  text ..... 24, 42  
  your\_tokens\_here ..... 16  
argument, optional  
  number ..... 24  
\array ..... 49  
\array ... 1061, 1151, 1175, 1233,  
  1352  
array document class . 10, 42, 46,  
  48, 49, 58, 60  
array environment .... 2, 42, 47  
\array@array ..... 1175, 1371  
\array@array@new ... 1233, 1376  
\array@default . 807, 1293, 1299,  
  1542  
\array@hook ..... 1051  
\array@LaTeX ..... 1061, 1352  
\array@ltx ..... 1151, 1360  
\array@message ..... 1170, 1247  
\array@row@pre . 1417, 1480, 1558  
\array@row@pst . 1499, 1505, 1558  
\array@row@rst ..... 54  
\array@row@rst .....  
  .. 1558, 1702, 1711, 1855,  
  1859, 1863, 1907, 1911, 1915,  
  1940  
\arraycolsep 509, 510, 1372, 1813,  
  1816  
\arrayrulewidth 1567, 1581, 1627  
\arraystretch .....  
  .. 1390, 1391, 1434, 1435,  
  1443, 1444, 1465, 1466, 1619,  
  1621  
\AtBeginDocument . 15, 16, 30, 47  
\AtBeginDocument ..... 356  
\AtEndDocument ..... 16  
\AtEndDocument ..... 358, 368  
\AtEndOfClass ..... 19  
atveryend document class 19, 92,  
  93  
\author ..... 50

## B

\baselineskip ..... 12, 23

\baselineskip . 1402, 1425, 1457,  
  1490  
\baselineskip ..... 12  
\begin ..... 15  
\begin@float@pagebreak ... 751,  
  765, 797, 823  
\belowdisplayshortskip ... 525  
\belowdisplayskip ..... 524  
\bgroup ..... 27, 54  
\body@font ..... 1564, 1599  
\botrule ..... 54  
\botrule ..... 1563  
\box ..... 67

## C

\c@mpfootnote .... 618, 619, 745  
\c@secnumdepth ... 947, 976, 990  
\Call@AfterLastShipout .... 19  
\Call@AfterLastShipout 435, 439  
\caption ..... 31  
\caption ..... 695  
\cat@letter ..... 241  
\cat@space ..... 2129  
\catcode ..... 19  
\catcode ..... 479, 487, 900,  
  905, 2129, 2155–2161, 2164,  
  2166, 2169, 2177, 2187  
\cell@fil 1594, 1637, 1911, 1915  
\cell@font 1593, 1600, 1702, 1712,  
  1907, 1911, 1915, 1940  
\centering ..... 21  
\changes ..... 137–188  
\chardef 240–242, 836, 896, 2129  
\check@aux ..... 19  
\check@aux ..... 363  
⟨class⟩ placeholder ..... 19  
class, argument ..... 19  
\class@documenthook ... 16, 34  
\class@documenthook . 355, 438,  
  605, 695  
\class@enddocumenthook . 16, 30  
\class@enddocumenthook 231, 355  
\class@err ..... 203  
\class@ext@hook ..... 441  
\class@extension ..... 441  
\class@extensionfile ..... 19

<code>\class@extensionfile</code> . . . .	441	<code>\copyright</code> . . . . .	47
<code>\class@info</code> . . . . .	203, 464, 466, 542, 621, 852, 884, 945, 1143, 1145, 1225, 1248, 1251, 1981, 1983, 2220, 2249	<code>\count</code> . . . . .	35, 36
<code>\class@name</code> . . . . .	11	<code>\count</code> . . . 841, 844, 846, 847, 853, 1967, 1974	
<code>\class@name</code> . 203–205, 224, 233, 234		<code>\count@</code> . . 1698, 1704, 1720, 1751	
<code>\class@warn</code> . . . . .	203, 235	<code>\count@i</code> . . . . .	267
<code>\class@warn@end</code> . . 207, 212, 217, 230		<code>\count@ii</code> . . . . .	267
<i>(class customization commands)</i> place- holder . . . . .	9	<code>\countdef</code> . . . . .	267, 268
<code>\ClassError</code> . . . . .	203, 224	<code>\crrcr</code> 1496, 1499, 1502, 1505, 1510, 1586	
<code>classes.dtx</code> . . . . .	30	<code>\cs</code> . 137, 138, 140, 151–153, 155, 156, 158–163, 166, 168, 175, 177, 180–182, 188	
<code>classes.dtx</code> document class . .	34	<code>\csname</code> . . . . .	14, 50
<code>\ClassInfo</code> . . . . .	205	<code>\csname</code> . . . 32, 33, 245, 247, 251, 253, 302, 323, 443, 445, 451, 561, 565, 593, 597, 603, 615, 616, 644, 686, 687, 776, 778, 781, 782, 813, 814, 818, 827, 831, 833, 860, 879, 896, 899, 961, 969, 974, 979, 985, 988, 993, 1009, 1014, 1019, 1022, 1241, 1993, 1994, 2033, 2036, 2039, 2041, 2045, 2046, 2054– 2056, 2065, 2095, 2219	
<code>\classname</code> . . . . .	61, 69, 121, 123, 125, 140, 164, 165, 177, 178, 180, 181, 185	<code>\CT@cell@color</code> . . . . .	1804
<code>\classoption</code> . . . . .	167	<code>\CT@color</code> . . . . .	1799
<code>\ClassWarningNoLine</code> . . . .	204	<code>\CT@column@color</code> . . . . .	1802
<code>\clear@document</code> . . . . .	17–19	<code>\CT@do@color</code> . . . . .	1800
<code>\clear@document</code> . . 371, 429, 432		<code>\CT@end</code> . . . . .	47
<code>\clearpage</code> . . . . .	18, 19	<code>\CT@row@color</code> . . . . .	1803
<code>\clearpage</code> . . . . .	425, 433	<code>\CT@setup</code> . . . . .	1798
<code>\closeout</code> . . . . .	379, 817	<code>\CT@start</code> . . . . .	47
<code>\cmd</code> . . . . .	133	<code>\CurrentOption</code> . . . . .	456
<code>\col@sep</code> . . . . .	46		
<code>\col@sep</code> . . . . .	1276, 1372		
<code>\color</code> . . . . .	1801		
<code>\color@begingroup</code> . . . . .	646		
<code>\color@endgroup</code> . . . . .	651, 730		
<code>colortbl</code> document class . 46, 47, 58, 60, 91, 92			
<code>\colortbl@message</code> . . 1170, 1250			
<code>\colrule</code> . . . . .	54		
<code>\colrule</code> . . . . .	1563		
<code>\column@fil</code> . . . . .	1563, 1598		
<code>\column@font</code> . . . . .	1563, 1597		
<code>\columnwidth</code> . . . . .	27		
<code>\columnwidth</code> . . . . .	754		
<code>\contentsline</code> . . . . .	31, 64, 65		
<code>\contentsline</code> . . . . 675, 692, 693			
<code>\contentsline@latex</code> . . 689, 692			
<code>\copy</code> . . . . .	1605		

## D

<code>\dollarbegin</code> . 1254, 1277, 1285, 1373, 1381, 1382	
<code>\dollarend</code> 1278, 1286, 1373, 1382	
<code>\deadcycles</code> . . . . .	373
<code>\DeclareRobustCommand</code> . . . .	21
<code>\DeclareRobustCommand</code> . . . .	498
<code>\detokenize</code> . . . . .	414, 415
<code>\dimen</code> . . . . .	65
<code>\dimen@</code> . . 864, 865, 867, 868, 870, 871, 1618–1620, 2056, 2057	



<code>\dimen@iii</code> .....	238	<code>\doi</code> .....	2119
<code>\dimendef</code> .....	238	<code>\doibase</code> .....	2119
<code>\discretionary</code> .....	2126	<code>\doublerulesep</code> .....	1580
<code>displaymath</code> environment .....	23	<code>\dp</code> .....	658, 854, 870, 871, 883, 1309, 1310, 1333, 1339, 1391, 1435, 1444, 1466, 1621
<code>\displaystyle</code> ..	508, 511, 530, 536		
<code>\displaywidth</code> .....	507		
<code>\do</code> .....	1650, 1666, 1724, 1755		
<code>\do@check@aux</code> .....	18		
<code>\do@check@aux</code> .....	363		
<code>\do@if@floats</code> .....	30, 34		
<code>\do@if@floats</code> .....	773		
<code>\do@output@cclv</code> .....	434		
<code>doc</code> .....	5		
<code>doc/</code> .....	3		
<code>\DocInput</code> .....	9		
<code>\document</code> .....	15, 16		
<code>\document</code> .....	344		
document class			
<code>array</code> .....	10, 42, 46, 48, 49, 58, 60		
<code>atveryend</code> .....	19, 92, 93		
<code>classes.dtx</code> .....	34		
<code>colortbl</code> .....	46, 47, 58, 60, 91, 92		
<code>ftnright</code> .....	10		
<code>hyperref</code> .....	22, 24, 26, 27, 31, 40, 41, 64, 65, 68–70, 90, 92, 93		
<code>hyperref<sub>L</sub>package</code> .....	2, 31		
<code>hyperref.sty</code> .....	93		
<code>lineno.sty</code> .....	67, 93		
<code>longtable</code> .....	10, 58		
<code>ltxdoc</code> .....	5, 9		
<code>ltxfront</code> .....	28		
<code>ltxgrid</code> .....	10, 16, 19, 27		
<code>ltxutil</code> .....	1, 10, 19		
<code>ltxutil.dtx</code> .....	3		
<code>ltxutil.pdf</code> .....	3		
<code>ltxutil.sty</code> .....	3		
<code>revtex4</code> .....	19		
<code>textcase</code> .....	21, 92		
document class option			
<code>hypertext</code> .....	92		
<code>lengthcheck</code> .....	92		
document environment .....	5, 16		
<code>\document@inithook</code> .....	16		
<code>\document@inithook</code> .....	346, 350, 354, 355, 691, 1041, 1045, 2229		
<code>\doi</code> .....	69		
		<b>E</b>	
		<code>\edef</code> .....	58
		<code>\edef</code> .....	272, 277, 282, 414, 415, 572, 1394, 1807, 1813, 1822, 1834, 1881, 2036, 2173–2175, 2183, 2202
		<code>\emph</code> .....	92
		<code>\end</code> .....	37
		<code>\end@float@anchored</code> ..	752, 753, 759
		<code>\end@float@pagebreak</code> ..	763, 766, 797
		<code>\endarray</code> .....	47, 52
		<code>\endarray</code> .....	1045, 1062, 1152, 1176, 1234, 1495, 1513, 1516, 1519
		<code>\endarray@array</code> .....	1176, 1501
		<code>\endarray@array@new</code> ..	1234, 1504
		<code>\endarray@hook</code> .....	1049, 1050
		<code>\endarray@LaTeX</code> .....	1062, 1495
		<code>\endarray@ltx</code> .....	52
		<code>\endarray@ltx</code> ..	1152, 1498, 1505
		<code>\endcsname</code> ..	32, 33, 245, 247, 251, 253, 302, 323, 443, 445, 451, 561, 565, 593, 597, 603, 615, 616, 644, 686, 687, 776, 778, 781, 782, 813, 814, 818, 827, 831, 833, 860, 879, 896, 899, 961, 969, 974, 979, 985, 988, 993, 1009, 1014, 1019, 1022, 1241, 1993, 1994, 2033, 2036, 2039, 2041, 2045, 2046, 2054– 2056, 2066, 2095, 2219
		<code>\enddocument</code> .....	17, 18
		<code>\enddocument</code> .....	363
		<code>\endgroup</code> .....	16, 18
		<code>\endlinechar</code> .....	2154
		<code>\endminipage</code> .....	725
		<code>\endtabular</code> .....	.. 1045, 1063, 1153, 1177,

1235, 1241, [1509](#), 1521  
 endtabular environment . . . . . [53](#)  
 \endtabular\* . . . . . [1521](#)  
 \endtabular@array . . . 1177, 1515  
 \endtabular@array@new . . . 1235,  
 1518  
 \endtabular@hook . . . 1047, 1048  
 \endtabular@LaTeX . . . 1063, 1509  
 \endtabular@ltx . . . . 1153, 1512  
 \endwrite@float . . . . . 889, 899  
 \endwrite@floats . . . . . 891  
 environment  
     array . . . . . [2](#), [42](#), [47](#)  
     displaymath . . . . . [23](#)  
     document . . . . . [5](#), [16](#)  
     endtabular . . . . . [53](#)  
     eqnarray . . . . . [22](#), [23](#), [48](#)  
     equation . . . . . [23](#)  
     figure . . . . . [29](#)  
     figure@write . . . . . [29](#), [34](#)  
     longtable . . . . . [30](#), [37](#)  
     minipage . . . . . [24](#), [29](#)  
     tabular . . . . . [2](#), [42](#), [47](#), [48](#)  
 \eqnarray . . . . . [22](#)  
 \eqnarray . . . . . 541, 543, 556  
 eqnarray environment . . [22](#), [23](#), [48](#)  
 \eqnarray@fleqn@fixed . . . [499](#)  
 \eqnarray@LaTeX . . . . . [499](#)  
 \eqncolsep . . . 532, 535, 544, 545  
 \equation . . . . . [24](#)  
 \equation . . . . . 555  
 equation environment . . . . . [23](#)  
 \errorstopmode . . . . . 259  
 \extrarowheight . . . . 1441, 1463

## F

\f@ur . . . . . 240, 1744  
 \false@sw . . . 313, 319, 324, [336](#),  
 341, 1080, 1083, 1086, 1089,  
 1092, 1095, 1098, 1101, 1104,  
 1107, 1110, 1113, 1116, 1119,  
 1122, 1125, 1128, 1131, 1134,  
 1137, 1140, 1187, 1190, 1193,  
 1196, 1199, 1202, 1205, 1208,  
 1211, 1214, 1217, 1220, 1223,  
 2132, 2247

figure environment . . . . . [29](#)  
 figure@write environment . [29](#), [34](#)  
 \figuresname . . . . . [30](#)  
 file  
     \$TEXMF/ . . . . . [3](#)  
     .aux . . . . . [41](#)  
     .cls . . . . . [20](#)  
     .dtx . . . . . [5](#)  
     .rtx . . . . . [19](#)  
     aps.rtx . . . . . [19](#)  
     classes.dtx . . . . . [30](#)  
     doc . . . . . [5](#)  
     doc/ . . . . . [3](#)  
     kernel . . . . . [2](#), [11](#), [72](#)  
     latex/ . . . . . [3](#)  
     ltxfloat.dtx . . . . . [24](#)  
     ltxutil.dtx . . . . . [2](#), [3](#)  
     ltxutil.pdf . . . . . [1](#)  
     ltxutil.sty . . . . . [1–3](#)  
     makeindex . . . . . [3](#)  
     nameref.sty . . . . . [40](#)  
     package . . . . . [2](#), [10](#)  
     pdfmark.def . . . . . [40](#)  
     README-LTXUTIL . . . . . [3](#), [5](#)  
     revtex/ . . . . . [3](#)  
     source/ . . . . . [3](#)  
     tex/ . . . . . [3](#)  
     texmf-local/ . . . . . [3](#)  
 \file . . . . . 29, 80, 83, 89, 90,  
 99, 105, 107, 110, 111, 114,  
 115, 121, 123, 125, 127, 130  
 float, argument . . . . . [37](#)  
 \float@end@tag . . . . . [37](#)  
 \float@end@tag . . . . . [904](#)  
 \floatingpenalty . . . . . [27](#), [28](#)  
 \floats@sw . . . . . [33](#), [34](#)  
 \floats@sw . . . [734](#), 739, 774, 804  
 \flushing . . . . . [491](#)  
 \fmtversion . . . . . 304  
 \font@submax . . . . . 387, 390  
 \fontsubfuzz . . . . . 387  
 \footins . . . . . [27](#)  
 \footins . . . . . 632, 722  
 \footmark . . . . . [25](#)  
 \footnote . . . . . [24](#), [25](#), [29](#)  
 \footnote . . . . . [558](#), 665

`\footnote@latex` ..... 606  
`\footnotemark` ..... [24](#), [25](#), [27](#)  
`\footnotemark` ..... [558](#)  
`\footnotesep` ..... 648, 657  
`\footnotesize` ..... 655  
`\footnotetext` ..... [558](#)  
`\footnotetext` ..... [24](#), [25](#)  
`\footnotetext` ..... 582  
`\frontmatter@footnotetext` . [28](#)  
`\frontmatter@makefntext` ... [28](#)  
`\firststrut` ..... 1563, 1564, 1601  
`ftnright` document class .... [10](#)  
`\ftype@` ..... [35](#)  
`\fullinterlineskip` ... [266](#), 712  
`\futurelet` ..... 1567, 1574

## G

`\gappdef` ..... 231, [269](#)  
`\GetFileInfo` ..... 38  
`\glet@environment` .... 218, [243](#)  
`\glossary` ..... 664

## H

`\H@@footnotetext` ..... [24](#)  
`\H@@footnotetext` . 608, 609, 746  
`\H@@mpfootnotetext` 610, 611, 746  
`\H@refstepcounter` ..... [39](#)  
`\H@refstepcounter` ... 951, 1040  
`\H@svsec` 948, 952, 970, 986, 1003,  
1010, 1020  
`\halign` ..... [52](#), [58](#)  
`\halign@` ..... [239](#), 529  
`\hangindent` ..... [68](#)  
`\hb@xt@` 512, 537, 770, 2063, 2075  
`\hbox` ..... [23](#), [68](#)  
`\hline` ..... [55](#)  
`\hline` ..... 1073, 1163, [1566](#)  
`\hline@LaTeX` ..... 1073, 1566  
`\hline@ltx` ..... 1163, 1570  
`\hline@rule` ..... [55](#)  
`\hline@rule` .. 1565, 1573, 1588,  
[1603](#)  
*<hooks and bookkeeping>* placeholder  
..... [18](#)  
`\href` ..... [69](#), [70](#)  
`\href` ..... [2119](#), [2150](#)

`\href@` ..... 2150–2152  
`\href@Hy` ..... 2150, 2219  
`\href@Hy@ltx` ..... 2151, 2226  
`\hrule` ..... [12](#)  
`\hsize` ..... [15](#)  
`\ht` ..... 854, 864, 867,  
868, 882, 1390, 1434, 1440,  
1462, 1618, 1619  
`\Hy@ActiveCarriageReturn` 2191,  
2224  
`\Hy@ActiveCarriageReturn@ltx`  
..... 2188, 2224  
`\Hy@footnote@currentHref` . 575,  
616  
`\Hy@chref` ..... 2151, 2152  
`\Hy@raisedlink` ..... 1033  
`\Hy@RemovePercentCr` 2204, 2213,  
2225  
`\Hy@RemovePercentCr@ltx` . 2208,  
2225  
`\Hy@ReturnAfterElseFi` ... 2192  
`\Hy@ReturnAfterFi` .. 2196, 2212  
`\Hy@safe@activetrue` ... 2184  
`\Hy@tempa` 2190, 2191, 2202, 2206  
`\hyper@@normalise` . 2193, 2197,  
2222  
`\hyper@@normalise@ltx` ... 2201,  
2222  
`\hyper@anchorend` ..... [41](#)  
`\hyper@anchorend` ..... 1035  
`\hyper@anchorstart` ..... [41](#)  
`\hyper@anchorstart` ..... 1034  
`\hyper@hash` ..... 2170, 2171  
`\hyper@last` ..... [41](#)  
`\hyper@linkend` ..... [41](#)  
`\hyper@linkend` ..... 577, 1037  
`\hyper@linkstart` ..... [41](#)  
`\hyper@linkstart` .... 575, 1036  
`\hyper@makecurrent` ..... 1032  
`\hyper@n@rmalise` ... 2185, 2223  
`\hyper@n@rmalise@ltx` 2189, 2223  
`\hyper@normalise` ... 2150–2152,  
2221  
`\hyper@normalise@ltx` 2162, 2221  
`\hyper@tilde` ..... 2179–2181  
`\hyperanchor` ..... [41](#)

hyperref document class . 22, 24,  
26, 27, 31, 40, 41, 64, 65,  
68–70, 90, 92, 93  
hyperref<sub>l</sub> package document class  
..... 2, 31  
hyperref.sty document class . 93  
hypertext document class option 92  
\hypertext@enable@ltx ..... 69  
\hypertext@enable@ltx ... 2130

## I

\ialign ..... 50  
\align ..... 1395, 1448  
\if ..... 50  
\if ..... 310, 322, 1387, 1453  
\if@filesw . 378, 395, 1992, 2021  
\if@firstamp ..... 1807, 1810  
\if@nobreak ..... 935  
\if@noskipsec ..... 928  
\ifeof ..... 326  
\iffalse ..... 54, 1315, 1320  
\IfFileExists ..... 442  
\IfFormatAtLeastTF ... 303, 306  
\ifhbox ..... 327  
\ifinner ..... 329  
\ifmmode ..... 330  
\ifodd ..... 332, 1971  
\ifvbox ..... 333  
\ifvoid ..... 335  
\ignorespaces .....  
... 648, 1527, 1540, 1697,  
1703, 1884, 1886, 1888, 1907,  
1911, 1915, 1932, 1940, 2062,  
2071, 2083, 2103  
\immediate .. 379, 779, 816, 914,  
1994, 2044, 2233, 2239  
\incompatible@package .... 221  
\index ..... 663  
\init@hyperref ..... 1030  
\insert ..... 28  
\insert ..... 632, 722  
\insert@column . 1180, 1240, 1695  
\insert@column@array 1180, 1695  
\insert@column@array@new 1240,  
1700  
\interdisplaylinepenalty .. 48

\interfootnotelinepenalty 656  
\interlinepenalty 656, 967, 1007  
\interlineskip ..... 28  
\intertabularlinepenalty .. 48  
\intertabularlinepenalty 1301,  
1302, 1304, 1306, 1321  
\intextsep ..... 765, 766  
\item 103, 113, 116, 120, 122, 124,  
129, 131  
\itemindent ..... 2094  
\itshape ..... 618, 627

## J

\j@nk ..... 342

## K

kernel ..... 2, 11, 72

## L

\l@ ..... 65  
\l@sections ..... 65  
\l@sections ..... 2050  
\l@section ..... 2004  
\label ..... 662, 673  
\lastbox ..... 552, 715, 821  
\LaTeX . 41, 70, 71, 81, 91, 92, 94,  
105, 109, 112, 114, 188  
latex/ ..... 3  
\LaTeXe ..... 75, 100, 198  
\leftmargin ... 2107, 2109, 2114  
\leftmargini ..... 2004  
\leftmarginii ..... 2005  
\leftmarginiii ..... 2006  
\leftmarginiv ..... 2007  
\leftmarginv ..... 2008  
\leftmarginvi ..... 2009  
\leftskip ..... 68  
\leftskip . 493, 2055, 2057, 2116  
lengthcheck document class option  
..... 92  
\lengthcheck@sw ..... 799  
\let ..... 46, 66  
\let@environment ..... 12  
\let@environment . 243, 789, 790,  
792, 811, 812  
\limits ..... 2210

lineno.sty document class [67, 93](#)  
 \lineskip [1402, 1425, 1456, 1490](#)  
 \linewidth [529, 2108–2110](#)  
 \list [68](#)  
 \list [2086](#)  
 \listparindent [2093, 2106, 2113](#)  
 \LoadClass [20](#)  
 \long [72](#)  
 longtable document class [10, 58](#)  
 longtable environment [30, 37](#)  
 \loop [15](#)  
 \loopuntil [338](#)  
 \loopwhile [338](#)  
 \lrstrut [1564, 1565, 1602](#)  
 \LT@array [10](#)  
 ltxfloat.dtx [24](#)  
 \ltx@contentsline [673](#)  
 \ltx@def@footproc [25](#)  
 \ltx@def@footproc [560, 568, 583, 591](#)  
 \ltx@footmark [25](#)  
 \ltx@footmark [558](#)  
 \ltx@footnote@pop [742, 748](#)  
 \ltx@foottext [25](#)  
 \ltx@foottext [558](#)  
 \ltx@make@current@footnote [558](#)  
 \ltx@mathindent [23](#)  
 \ltx@mathindent [519, 549](#)  
 \ltx@stp@footproc [25](#)  
 \ltx@stp@footproc [564, 569, 584, 599](#)  
 \ltx@thempfootnote [619, 622](#)  
 \ltx@xfootmark [567, 568](#)  
 \ltx@xfootnote [24](#)  
 \ltx@xfootnote [558](#)  
 \ltx@xfoottext [582, 583](#)  
 \ltx@yfootmark [567, 569](#)  
 \ltx@yfootnote [558, 563](#)  
 \ltx@yfoottext [582, 584](#)  
 ltxdoc document class [5, 9](#)  
 ltxfront document class [28](#)  
 ltxgrid document class [10, 16, 19, 27](#)  
 \ltxu@dotsep [2003, 2078](#)  
 ltxutil document class [1, 10, 19](#)  
 ltxutil.dtx [2, 3](#)

ltxutil.dtx document class [3](#)  
 ltxutil.pdf [1](#)  
 ltxutil.pdf document class [3](#)  
 ltxutil.sty [1–3](#)  
 ltxutil.sty document class [3](#)

## M

\make@footnotetext [27, 28](#)  
 \make@footnotetext [624](#)  
 \makeatletter [383, 455, 1990](#)  
 makeindex [3](#)  
 \makelabel [2097](#)  
 \maketitle [27](#)  
 \maketitle [59](#)  
 \mark [30](#)  
 \mathchardef [837, 838](#)  
 \mathindent [23](#)  
 \meaning [264](#)  
 \MessageBreak [389](#)  
 minipage environment [24, 29](#)  
 \minipagefootnote@drop [27](#)  
 \minipagefootnote@drop [652, 698](#)  
 \minipagefootnote@foot [698](#)  
 \minipagefootnote@here [32](#)  
 \minipagefootnote@here [696, 698, 760](#)  
 \minipagefootnote@init [698, 757](#)  
 \minipagefootnote@pick [636, 698](#)  
 \mpfootnote [621](#)  
 \multicolumn [1058, 1148, 1522](#)  
 \multicolumn@LaTeX [1058, 1522](#)  
 \multicolumn@ltx [1148, 1529](#)  
 \multiply [843](#)  
 \multispan [1523, 1530](#)

## N

nameref.sty [40](#)  
 \NC@list [1719, 1750](#)  
 \newbox [1614](#)  
 \newcount [1301, 1303](#)  
 \newenvironment [37](#)  
 \newlabel [31](#)  
 \newlabel [681](#)  
 \newlinechar [913](#)  
 \newwrite [775, 1993](#)

`\noalign` . 1308, 1310–1312, 1329,  
1342, 1346, 1350, 1565, 1567,  
1571, 1575, 1587  
`\noexpand` . 264, 265, 1395, 1448,  
2204  
`\normalfont` . . . . . 627  
number, optional argument . . . 24  
`\numberline` . . . . . 67, 68  
`\numberline@sections` . . . 2060,  
2074

## O

`\obeylines` . . . . . 910  
`\obsolete@command` . . . . . 206  
`\oneapage` . . . . . 797  
`\openout` . . . . . 780, 1994  
`\OptionNotUsed` . . . . . 20  
`\other` . . . . . 242

## P

`\p@equation` . . . . . 501, 517  
package . . . . . 2, 10  
`\package@name` . . . . . 196, 197  
`\PackageInfo` . . . . . 197  
`\par` . . . . . 67, 72  
`\parindent` . . . . . 23, 68  
`\parindent` 768, 2057, 2075, 2106,  
2113  
`\parsep` . . . . . 2101  
`\parshape` . . . . . 68  
`\parshape` . . . . . 2110  
`\parskip` . . . . . 23  
`\parskip` . . . . . 523, 2101  
`\parskipfillskip` . . . . . 23  
`\partopsep` . . . . . 522  
`\pbox@hook` . . . . . 1638, 1940  
`\pdfendlink` . . . . . 2143  
`pdfmark.def` . . . . . 40  
`\pdfoutput` . . . . . 2132  
`\pdfstartlink` . . . . . 2139  
`\pdfstartlink@attr` . . . . . 69  
`\pdfstartlink@attr` . . . . . 2130  
`\pdfstringdefDisableCommands`  
. . . . . 28  
`\phantomsection` . . . . . 40

`\phantomsection` . 948, 953, 1003,  
1031  
placeholder  
  `\langle class \rangle` . . . . . 19  
  `\langle class customization commands \rangle`  
  . . . . . 9  
  `\langle hooks and bookkeeping \rangle` . . 18  
  `\langle read main .aux and final checks \rangle`  
  . . . . . 18  
  `\langle your document here \rangle` . . . . 9  
`\prep@math` . . . . . 23  
`\prep@math` . . . . . 551  
`\prep@math@patch` . . . . . 23, 24  
`\prep@math@patch` . . . . . 551  
`\prepdef` . . . . . 13  
`\prepdef` . . . . . 269,  
348, 555, 556, 696, 813, 814,  
957, 959, 1047, 1049, 1052,  
1410, 1473

`\prepnext@tok` . . . . . 1722, 1753  
`\prevdepth` . . . . . 12  
`\prevdepth` . . . . . 266  
`\print@float` . . . . . 34  
`\print@float` . . . . . 796  
`\printfigures` . . . . . 30  
`\ProcessOptions` . . . . . 20  
`\protected` . . . . . 412  
`\protected@edef` . . . . . 26  
`\protected@edef` . . 616, 643, 955  
`\protected@write` . . . . 671, 680  
`\protected@xdef` . . . . . 601  
`\providecommand` . . . . .  
. . . 439, 1031–1038, 1051,  
2119, 2122, 2124, 2125, 2128,  
2129

`\ProvidesFile` . . . . . 6  
`\ProvidesPackage` . . . . . 20

## R

`\raggedleft` . . . . . 21  
`\raggedright` . . . . . 21  
`\langle read main .aux and final checks \rangle` place-  
holder . . . . . 18  
README-LTXUTIL . . . . . 3, 5  
`\realfootnote` . . . . . 27  
`\RecordChanges` . . . . . 36

`\refstepcounter` ..... 39  
`\relax` ..... 39, 42  
`\repeat` ..... 15  
`\replace@command` ..... 211  
`\replace@environment` .... 216  
`\RequirePackage` ..... 25, 27, 33  
`\reserved@a` .. 1568, 1574, 1578,  
1579, 1650, 1665  
`\reset@font` ..... 655  
`revtex/` ..... 3  
`revtex4` document class ..... 19  
`\robustify@contents` ..... 661  
`\romannumeral` ..... 2095  
`\rule` ..... 648  
`\rvtx@enddocument@patch` .. 409  
`\rvtx@enddocument@patch@end` .  
..... 418, 425  
`\rvtx@enddocument@patch@more`  
..... 420, 423  
`\rvtx@ifformat@geq` ..... 14  
`\rvtx@ifformat@geq` .. 303, 344,  
363, 409, 2246

## S

`\samepage` ..... 48  
`\samepage` ..... 1304  
`\save@decl` ..... 1727, 1764  
`\say` ..... 12  
`\say` ..... 264, 1578  
`\saythe` ..... 12  
`\saythe` ..... 264  
`\sc` ..... 30, 91  
`\sec@upcase` ..... 42  
`\sec@upcase` ..... 1044  
`\section` ..... 87, 824  
`\Sectionformat` ..... 40  
`\set@arstrutbox` ..... 55  
`\set@arstrutbox` 1408, 1432, 1603  
`\set@display@protect` 2232, 2238  
`\set@eqnarray@skips` ..... 550  
`\set@footnotefont` ..... 29  
`\set@footnotefont` .... 640, 654  
`\set@footnotewidth` .... 27, 29  
`\set@footnotewidth` ..... 624  
`\set@footnotewidth@one` ... 660  
`\set@listindent` .... 2102, 2118

`\set@listindent@` ..... 2112  
`\set@listindent@parshape` 2105,  
2118  
`\set@tocdim@pagenum` ... 67, 68  
`\set@tocdim@pagenum` 2053, 2070  
`\set@typeset@protect` ... 1401,  
1424, 1489, 1525, 1534  
`\shipout` ..... 12  
`\show@box@size@sw` ..... 851  
`\showboxbreadth` ..... 259  
`\showboxdepth` ..... 259  
`\showoutput` ..... 262  
`\skip@` ..... 2059, 2062  
`source/` ..... 3  
`\spacefactor` ..... 572, 579  
`\special` ..... 2134, 2135  
`\splitmaxdepth` ..... 658  
`\splittopskip` ..... 657  
`\stepcounter` .... 500, 517, 600  
`\stop` ..... 225  
`\StopEventually` ..... 6  
`\string` ..... 207,  
212, 223, 621, 666–668, 671,  
681, 945, 1981, 1983, 2045,  
2173–2175, 2183, 2220, 2249  
`\strutbox` ..... 649,  
658, 1390, 1391, 1434, 1435,  
1440, 1444, 1462, 1466, 1619,  
1621  
`\subsection` ..... 97, 136  
`\substyle@ext` ..... 19  
`\switch@array` ..... 42, 48  
`\switch@array` ..... 1046, 1055  
`\switch@array@info` . 1170, 1227  
`\switch@hyperref@href` ... 2218,  
2229  
`\switch@tabular` ..... 42, 48  
`\switch@tabular` .... 1046, 1055

## T

`\t@` ..... 237, 239  
`\tab@rule` ..... 55  
`\tab@rule` ..... 1563, 1564, 1585  
`\tabcolsep` 1276, 1634–1636, 1822  
`\tableft@skip` .. 1426, 1491, 1631  
`\tableftsep` ... 1634, 1708, 1819

<code>\tableofcontents</code>	85	<code>\toc@post@auto</code>	66
<code>\tabmid@skip</code>	1411, 1474, 1632	<code>\toc@post@auto</code>	2010
<code>\tabmidsep</code>	1635, 1709, 1825	<code>\toc@pre</code>	66
<code>\tabright@skip</code>	1415, 1478, 1633	<code>\toc@pre</code>	1989, 2000
<code>\tabrightsep</code>	1636, 1710, 1828	<code>\toc@pre@auto</code>	66
<code>\tabskip</code>	23, 50	<code>\toc@pre@auto</code>	2010
<code>\tabskip</code>	505, 508, 511, 513, 519, 530, 536, 538, 1396, 1411, 1415, 1426, 1450, 1474, 1478, 1491	<code>\toc@setindent</code>	2013–2016, 2032
<code>\tabular</code>	42	<code>\toc@writedimen</code>	2023–2028, 2043
tabular environment	2, 42, 47, 48	<code>\tocdim@appendix</code>	2008
<code>\tabular@hook</code>	1052, 1053	<code>\tocdim@min</code>	2033, 2039, 2085
<code>\tabularnewline</code>	1398, 1421, 1459, 1486	<code>\tocdim@pagenum</code>	2009, 2072
<code>\tally@float</code>	35	<code>\tocdim@paragraph</code>	2007
<code>\tally@float</code>	836	<code>\tocdim@section</code>	67
<code>\TeX</code>	67, 75, 114, 185	<code>\tocdim@section</code>	2004
<code>tex/</code>	3	<code>\tocdim@subsection</code>	2005
<code>texmf-local/</code>	3	<code>\tocdim@subsubsection</code>	2006
text, argument	24, 42	<code>\tocleft@pagenum</code>	2058
<code>\textasciitilde</code>	2181	<code>\tocleft@section</code>	67
textcase document class	21, 92	<code>\tocmax@section</code>	67
<code>\texttt</code>	52, 110, 111, 118, 127, 130, 133, 137	<code>\toks</code>	1740, 1795
<code>\textunderscore</code>	2175, 2176, 2178	<code>\toks@</code>	270, 272, 275, 277, 280, 282, 897, 914, 916
<code>\textwidth</code>	866	<code>\toks@ii</code>	13
<code>\thanks</code>	42, 46, 52	<code>\toks@ii</code>	271, 272, 276, 277, 281, 282, 297
<code>\the@toks</code>	1696, 1698, 1701, 1704, 1721, 1740, 1752, 1795	<code>\toksdef</code>	297
<code>\theequation</code>	501, 517	<code>\toprule</code>	54
<code>\thefootnote</code>	24	<code>\toprule</code>	1563
<code>\thempfn</code>	24	<code>\topsep</code>	521
<code>\thempfn</code>	594, 601, 744	<code>\total@float</code>	35
<code>\thempfootnote</code>	26	<code>\total@float</code>	800, 836
<code>\thempfootnote</code>	620, 622, 744	<code>\traceoutput</code>	12
<code>\thempfootnote@latex</code>	618, 620	<code>\traceoutput</code>	261
<code>\thepage</code>	675, 681	<code>\tracingall</code>	12
<code>\thr@@</code>	12	<code>\tracingcommands</code>	256
<code>\thr@@</code>	238, 512, 537, 1736, 1787	<code>\tracinglostchars</code>	257
<code>\title</code>	40	<code>\tracingmacros</code>	258
<code>\toc@font</code>	65	<code>\tracingonline</code>	256
<code>\toc@font</code>	2002, 2011	<code>\tracingoutput</code>	12
<code>\toc@letdimen</code>	2017, 2018, 2038	<code>\tracingoutput</code>	257
<code>\toc@post</code>	66	<code>\tracingpages</code>	257
<code>\toc@post</code>	1997, 2001	<code>\tracingparagraphs</code>	258
		<code>\tracingplain</code>	12
		<code>\tracingplain</code>	255
		<code>\tracingrestores</code>	258
		<code>\tracingstats</code>	256



`\trigger@float@par` .. 810, 813,  
     814, 820  
`\triggerpar` ..... 796, 810  
`\true@sw` .... 311, 318, 324, [336](#),  
     341, 351, 1078, 1185, 2132,  
     2244, 2247  
`\tw@` ..... [12](#)  
`\typeout` ..... [72](#)  
`\typeout` ..... [2230](#)  
`\typeout@ltx` ..... 2236, 2243  
`\typeout@org` ..... 2230, 2242

## U

`\unhbox` ..... 2063, 2080  
`\unhcopy` ..... 1605  
`\unrestored@protected@xdef` 594  
`\unskip` 727, 1332, 1338, 1564, 1565,  
     1697, 1703, 1884, 1886, 1888,  
     1907, 1911, 1915, 2062  
`\unvbox` ..... 703, 717, 722, 732  
`\unvcopy` ..... 714, 863  
`\uppercase` ..... [42](#)  
`\url` ..... [69](#)  
`\url` ..... 26, 65, 72, [2119](#)  
`\URL@prefix` ..... [2119](#)  
`\usepackage` ..... [3](#)

## V

`\vadjust` ..... 649  
`\vbox` .. 702, 863, 1387, 1453, 1543  
`\vcenter` ..... [54](#)  
`\vcenter` . 1387, 1453, 1544, 1548,  
     1551  
`\voidb@x` ..... 699  
`\vrule` 1309, 1310, 1334, 1340, 1390,  
     1434, 1442, 1465, 1620  
`\vsize` ..... [15](#)  
`\vsplit` ..... [28](#)  
`\vtop` ..... 714, 1387, 1453, 1542

## W

`\write` .... 914, 2044, 2233, 2239  
`\write@@float` ..... [888](#), [892](#)  
`\write@float` ..... [30](#), [34](#), [37](#)  
`\write@float` ..... 786, [888](#)  
`\write@floatline` ..... [37](#)

`\write@floatline` ..... 902, [904](#)  
`\write@floats` ..... 787, [888](#)

## X

`\x` ..... 414, 417  
`\xdef` ..... 459, 475, 1448, 1471

## Y

`\y` ..... 415, 417  
`your_\tokens_\here`, argument . [16](#)  
`\your document here` placeholder [9](#)

## Z

`\z@skip` .. 493–495, 508, 513, 530,  
     538, 649, 761, 1396, 1402,  
     1425, 1631–1633, 2063

## Change History

4.0b	
\@mpmakefnctext: AO: Removed superfluous \defs, changed to using \floats@sw as the flag. Also stopped using DPC's \if@twocolumn flag: using \floats@sw instead. Also added \par\vskip\z@skip after the \minipagefootnotes so that the float box would have zero depth like the kernel one. . . . .	33
\caption: Support the hack with \prepdef, and delay until \AtBeginDocument time, since hyperref clobbers \caption. .	31
\print@float: only execute if there really were floats of the given type . . . . .	34
\write@@float: AO: Fixed spurious CR and (return) characters in output file. Also, if the document did not have the \endfigure on a line of its own, the macro wouldn't work. Fixed. . . . .	37
General: AO: Fixed spurious CR and (return) characters in output file. Also, if the document did not have the \endfigure on a line of its own, the macro wouldn't work. Fixed. . . . .	3
AO: Removed superfluous \defs, changed to using \floats@sw as the flag. Also stopped using DPC's \if@twocolumn flag: using \floats@sw instead. Also added \par\vskip\z@skip after the \minipagefootnotes so that the float box would have zero depth like the kernel one. . . . .	3
only execute if there really were floats of the given type . . . . .	3
Support the hack with \prepdef, and delay until	
\AtBeginDocument time, since hyperref clobbers \caption. . .	3
4.0c	
\@mpmakefnctext: (AO, 110) Install hooks for endfloats processing	33
\@ssect: (AO, 116) Hyperref compatibility . . . . .	40
\endarray: (AO, 130) Interference from array package . . . . .	42
\print@float: *-form mandates pagebreak at each float; only print section head if there is something there. . . . .	34
General: (AO, 110) Install hooks for endfloats processing . . . . .	3
(AO, 116) Hyperref compatibility . . . . .	3
(AO, 130) Interference from array package . . . . .	3
*-form mandates pagebreak at each float; only print section head if there is something there. . . . .	3
4.0d	
\@mpmakefnctext: (AO, 127) Floats placed [h] to allow page breaks	33
(AO, 224) Hyperref compatibility. . . . .	33
\print@float: Allow things to break over pages by setting array@default. . . . .	34
General: (AO, 127) Floats placed [h] to allow page breaks . . . . .	3
(AO, 174) kernel fix . . . . .	3, 24
(AO, 224) Hyperref compatibility. . . . .	3
Allow things to break over pages by setting array@default. . . . .	3
4.0e	
\@mpmakefnctext: (AO, 221) Remove samepage command from @xfloat@prep: If the float can break over pages, we want better control. . . . .	33
General: (AO, 221) Remove samepage command from @xfloat@prep: If the float can	

break over pages, we want better control. ....	3	understood in mu. (What we wanted was a dimension.) ...	65
4.0f		General: (AO) Make	
\@ssect: (AO, 404) Hyperref compatibility .....	40	\addtocontents a \long \def; gobble up \footnote .....	3
General: (AO, 404) Hyperref compatibility .....	3	(AO) Remove code that avoided changes to \@xfootnotemark ..	3
4.1a		(AO, 438) Complete rewrite of footnote macros. ....	3
\@mpmakefntext: \@xfloat@prep calls \ltx@footnote@pop to restore the original \ltx@footmark and \ltx@foottext procedures, in case footnote processing has switched. ....	33	(AO, 459) do not assume \class@name is defined ...	3, 11
\@p@pfilename: Class extension mechanism \@pushfilename@ltx and \@p@pfilename@ltx. ....	20	(AO, 461) Change the csname from \@dotsep to \ltxu@dotsep. The former is understood in mu. (What we wanted was a dimension.) ....	3
\@class@enddocumenthook: \@class@documenthook is the last \@AtBeginDocument token now .....	16	(AO, 475) I had not properly reproduced the LaTeX macro \eqnarray. ....	3
\document: Get rid of \set@typesize@hook \set@pica@hook and the \normalsize directive .....	16	\@xfloat@prep calls \ltx@footnote@pop to restore the original \ltx@footmark and \ltx@foottext procedures, in case footnote processing has switched. ....	3
\eqnarray@f@leqn@fixed: (AO, 475) I had not properly reproduced the LaTeX macro \eqnarray. ....	22	\@class@documenthook is the last \@AtBeginDocument token now .	3
\footnote: (AO) Remove code that avoided changes to \@xfootnotemark .....	27	Class extension mechanism \@pushfilename@ltx and \@p@pfilename@ltx. ....	3
\ltx@make@current@footnote: (AO, 438) Complete rewrite of footnote macros. ....	24	Class extension mechanism \@class@extension, \@class@extensionfile, and \@class@ext@hook. ....	3, 19
\numberline@@sections: (AO, 461) Change the csname from \@dotsep to \ltxu@dotsep. The former is understood in mu. (What we wanted was a dimension.) .....	68	Get rid of \set@typesize@hook \set@pica@hook and the \normalsize directive .....	3
\robustify@contents: (AO) Make \addtocontents a \long \def; gobble up \footnote .....	30	4.1b	
\toc@font: (AO, 461) Change the csname from \@dotsep to \ltxu@dotsep. The former is		\@mkpream: (AO, 505) try to accommodate colortbl. ....	58
		\@mkpream@relax: (AO, 505) try to accommodate colortbl. ...	60
		\@mpmakefntext: No need to protect against undefined \float@sw .....	33
		\@tabarray: (AO, 505) try to accommodate colortbl. ....	47
		\@tabular: (AO, 505) try to accommodate colortbl. ....	46

<code>\array:</code> (AO, 505) try to accommodate <code>colortbl</code> . . . . .	50	General: (AO, 511) Compatability with <code>lineno.sty</code> 's erroneous way of detecting <code>fleqn.clo</code> . . . . .	3
<code>\do@if@floats:</code> No need to protect against undefined <code>\float@sw</code> . . . . .	34	4.1f	
<code>\doibase:</code> (AO, 487) Support for video figures and the <code>\setfloatlink</code> command . . . . .	69	<code>\set@footnotewidth:</code> (AO, 515) Hook for setting the font of a footnote . . . . .	29
<code>\endarray:</code> Patch the array package even later: after all package patches go in. . . . .	42	<code>\total@float:</code> (AO, 518) Tally register overflow when locument is long . . . . .	35
<code>\floats@sw:</code> Default assignment of <code>\float@sw</code> now, not at <code>\AtBeginDocument</code> time. . . . .	33	General: (AO, 515) Hook for setting the font of a footnote . . . . .	3
<code>\init@hyperref:</code> Acquire <code>hyperref</code> savoirre . . . . .	41	(AO, 518) Tally register overflow when locument is long . . . . .	3
4.1g			
<code>\ltx@contentsline:</code> Refine toc processing: provide default. . . . .	31	<code>\doibase:</code> (AO, 532) Both arguments of <code>\href</code> get sanitized . . . . .	69
<code>\print@float:</code> If class option <code>lengthcheck</code> is in effect, log the height of this float class. . . . .	34	General: (AO, 525) Remove phantom paragraph above display math that is given in vertical mode . . . . .	3, 23
<code>\switch@array:</code> (AO, 505) Try to accommodate <code>colortbl</code> . . . . .	45	(AO, 532) Both arguments of <code>\href</code> get sanitized . . . . .	3
<code>\total@float:</code> Tally and log the height of a float class . . . . .	35	(AO, 539) Use of double-backslash in argument of <code>\section</code> gives error. The <code>textcase</code> package is involved. . . . .	3, 21
General: (AO, 487) Support for video figures and the <code>\setfloatlink</code> command . . . . .	3		
(AO, 505) try to accommodate <code>colortbl</code> . . . . .	3		
Acquire <code>hyperref</code> savoirre . . . . .	3	4.1j	
Default assignment of <code>\float@sw</code> now, not at <code>\AtBeginDocument</code> time. . . . .	3	<code>\doibase:</code> (AO, 545) Provide definition for <code>\doi</code> that does hypertext . . . . .	69
If class option <code>lengthcheck</code> is in effect, log the height of this float class. . . . .	3	<code>\hypertext@enable@ltx:</code> (AO, 545) hypertext capabilities off by default; enable with <code>hypertext</code> . . . . .	69
No need to protect against undefined <code>\float@sw</code> . . . . .	3	4.1n	
Patch the array package even later: after all package patches go in. . . . .	3	<code>\clear@document:</code> (AO, 569) Use of <code>hyperref</code> interferes with column balancing of last page . . . . .	19
Refine toc processing: provide default. . . . .	3	(AO, 569) execute <code>atveryend</code> 's <code>\Call@AfterLastShipout</code> at the proper time . . . . .	19
Tally and log the height of a float class . . . . .	3	<code>\do@check@aux:</code> (AO) Incorporate change to <code>ltxmiscn.dtx</code> v1.1i 2000/05/19 . . . . .	16
4.1d		(AO, 569) Use of <code>hyperref</code> interferes with column balancing of last page . . . . .	16
<code>\eqnarray@fleqn@fixed:</code> (AO, 511) Compatability with <code>lineno.sty</code> 's erroneous way of detecting <code>fleqn.clo</code> . . . . .	23		

<code>\l@sections:</code> (AO, 574) protect against <code>lineno.sty</code> , which forces a visit to the output routine, which appears to destroy the value of	<code>lineno.sty</code> , which forces a visit to the output routine, which appears to destroy the value of <code>\@tempdima</code> . . . . . 3
<code>\@tempdima</code> . . . . . 67	4.1p
<code>\set@footnotewidth:</code> (AO, 571) Interface <code>\set@footnotewidth</code> for determining the set width of footnotes . . . . . 29	<code>\href:</code> (AO, 582) A patch of <code>hyperref.sty</code> to provide backward compatibility to T <sub>E</sub> XLive 2007's version 6.75r . . 70
(AO, 571) allow split after last line of footnote . . . . . 29	General: (AO, 582) A patch of <code>hyperref.sty</code> to provide backward compatibility to T <sub>E</sub> XLive 2007's version 6.75r . . 3
(AO, 572) title block footnotes numbered independently from body footnotes . . . . . 28	4.2a
General: (AO) Incorporate change to <code>ltniscen.dtx</code> v1.1i 2000/05/19 . . . . . 3	General: (MD) Updated name of README file and use standard fonts when typesetting . . . . . 3
(AO, 569) Use of <code>hyperref</code> interferes with column balancing of last page . . . . . 3	(MD) Use updated best practice to use <code>https</code> and <code>doi.org</code> . . . . 3
(AO, 569) execute <code>atveryend</code> 's <code>\Call@AfterLastShipout</code> at the proper time . . . . . 3	4.2d
(AO, 571) Interface <code>\set@footnotewidth</code> for determining the set width of footnotes . . . . . 3	<code>\do@check@aux:</code> (PHO) Only redefine <code>\enddocument</code> in older versions. . . . . 16
(AO, 571) allow split after last line of footnote . . . . . 3	(PHO) Patch <code>\enddocument</code> at runtime in newer versions. . . . 17
(AO, 572) title block footnotes numbered independently from body footnotes . . . . . 3	<code>\document:</code> (PHO) Use L <sup>A</sup> T <sub>E</sub> X's hook management system, if possible. . . . . 16
(AO, 574) protect against	<code>\rvtx@ifformat@geq:</code> (PHO) Add <code>\rvtx@ifformat@geq</code> . . . . . 14
	General: (PHO) Adapt <code>\document</code> and <code>\enddocument</code> hooks to the 2020-10-01 L <sup>A</sup> T <sub>E</sub> X release. . . . . 3