

A L^AT_EX Package of utility macros ^{*†}

Arthur Ogawa [‡]

October 9, 2009

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

The distribution point for this work is publish.aps.org/revtex, which contains prebuilt runtime files, documentation, and full source, ready to add to a TDS-compliant T_EX installation.

The `ltxutil` package was commissioned by the American Physical Society and is distributed under the terms of the L^AT_EX Project Public License, the same license under which all the portions of L^AT_EX 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_EX installation equipped with L^AT_EX 2_ε and possibly `pdftex` and Adobe Acrobat Reader or equivalent.

To install, retrieve the distribution, unpack it into a directory on the target computer, and move the file `ltxutil.sty` into a location in your filesystem where it will be found by L^AT_EX.

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

Contents

1	Processing Instructions	2
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
2	Code common to all modules	4

^{*}This file has version number 4.1h, last revised 2009/10/09.

[†]Version 4.1h © 2009 The American Physical Society

[‡]mailto:arthur_ogawa@sbcglobal.net

3	The driver module <code>doc</code>	5
3.1	The Preamble	5
3.1.1	Docstrip and info directives	5
3.2	The “Read Me” File	5
3.3	The Document Body	8
4	Using this package	9
4.1	Invoking the package	9
5	Compatibility with L^AT_EX’s Required Packages	9
5.1	<code>array</code>	10
5.2	<code>longtable</code>	10
6	Implementation of package	10
6.1	Beginning of the <code>package</code> <code>DOCSTRIP</code> module	10
6.2	Banner and beginning of the <code>kernel</code> <code>DOCSTRIP</code> module	10
6.3	Errors and warnings	10
6.4	New Tools	11
6.5	Boolean Control	12
6.6	Begin Document Structure	15
6.7	Class Extensions	17
6.8	Type Tools	18
6.9	Display Math	19
6.10	Footnotes	21
6.11	Floats	26
6.11.1	Usage notes	26
6.11.2	Robustifying fragile commands	27
6.11.3	Preparing for the <code>hyperref</code> package	28
6.11.4	Footnotes within floats, unfloating floats, float font	28
6.11.5	Writing floats out to a file	31
6.12	Counters	35
6.13	Customization of Sections	35
6.14	Patch the <code>tabular</code> and <code>array</code> Environments	39
6.15	Repair other broken parts of L ^A T _E X	61
6.16	Syntax	61
6.17	Auto-indented Contents	61
6.18	Lists	65
6.19	Hypertext capabilities	66
6.20	End of the <code>kernel</code> <code>DOCSTRIP</code> module	67
	Index	68

1 Processing Instructions

The package file `ltxutil.sty` is generated from this file, `ltxutil.dtx`, using the `DOCSTRIP` facility of L^AT_EX via `tex ltxutil.dtx` (Note: do *not* use L^AT_EX for this

task). The typeset documentation that you are now reading is generated from the same file by typesetting it with \LaTeX or `pdftex` via `latex ltxutil.dtx` or `pdflatex ltxutil.dtx`.

1.1 Build Instructions

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

1. Within an otherwise empty directory, typeset `ltxutil.dtx` with \LaTeX or `pdflatex`; you will obtain the typeset documentation you are now reading, along with the file `00readme`.

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

2. Now typeset `ltxutil.dtx` with \TeX (not \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):

- `$TEXMF/tex/latex/revtex/ltxutil.sty`
- `$TEXMF/source/latex/revtex/ltxutil.dtx`
- `$TEXMF/doc/latex/revtex/ltxutil.pdf`

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

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

1.2 Change Log

1.3 Bill of Materials

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

1.3.1 Primary Source

One single file generates all.

```
%ltxutil.dtx
%
```

1.3.2 Generated by latex ltxutil.dtx

Typesetting the source file under pdf_latex generates the readme and the documentation.

```
%00readme ltxutil.pdf
%
```

1.3.3 Generated by tex ltxutil.dtx

Typesetting this file with T_EX generates the package file.

```
%ltxutil.sty
%
```

1.3.4 Auxiliary

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

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

2 Code common to all modules

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

Now we declare that these files can only be used with L^AT_EX 2_ε. An appropriate message is displayed if a different T_EX format is used.

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

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

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

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

```
8 %<*doc|package>
9 [2009/10/09 4.1h utilities package]% \fileversion
10 %</doc|package>
```

3 The driver module doc

This module, consisting of the present section, typesets the programmer’s documentation, generating the `00readme` as required.

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

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

```
11 %<*doc>
```

3.1 The Preamble

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

```
12 \documentclass{ltxdoc}
13 \RequirePackage{ltxdocext}%
14 \let\url\undefined
15 \RequirePackage[colorlinks=true,linkcolor=blue]{hyperref}%
16 \expandafter\ifx\csname package@font\endcsname\@undefined\else
17 \expandafter\RequirePackage\expandafter{\csname package@font\endcsname}%
18 \fi
19 \CodelineIndex\EnableCrossrefs % makeindex -s gind.ist ltxutil
20 \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.

```
21 \setcounter{StandardModuleDepth}{1}
```

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

```
22 \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.

```

23 \begin{filecontents*}{00readme}
24 \title{%
25 A \LaTeX\ Package of utility macros%
26 \thanks{%
27 This file has version number \fileversion,
28 last revised \filedate.%
29 }%
30 \thanks{%
31 Version \fileversion\ \copyright\ 2009 The American Physical Society
32 }%
33 }%
34 \author{%
35 Arthur Ogawa%
36 \thanks{\texttt{mailto:arthur\_ogawa at sbcglobal.net}}%
37 }%
38 %\iffalse
39 % For version number and date,
40 % search on "\fileversion" in the .dtx file,
41 % or see the end of the 00readme file.
42 %\fi
43 \maketitle
44
45 This file embodies the \classname{ltxutil} package,
46 the implementation and its user documentation.
47
48 The distribution point for this work is
49 \url{publish.aps.org/revtex},
50 which contains prebuilt runtime files, documentation, and full source,
51 ready to add to a TDS-compliant \TeX\ installation.
52
53 The \classname{ltxutil} package was commissioned by the American Physical Society
54 and is distributed under the terms of the \LaTeX\ Project Public License,
55 the same license under which all the portions of \LaTeX\ itself are distributed.
56 Please see \url{http://ctan.tug.org/macros/latex/base/lppl.txt} for details.
57
58 To use this document class, you must have a working
59 \TeX\ installation equipped with \LaTeXe\
60 and possibly pdftex and Adobe Acrobat Reader or equivalent.
61
62 To install, retrieve the distribution,
63 unpack it into a directory on the target computer,
64 and move the file \file{ltxutil.sty}
65 into a location in your filesystem where it will be found by \LaTeX.
66
67 To use, read the user documentation \file{ltxutil.pdf}.
68
69 \tableofcontents
70
71 \section{Processing Instructions}
72

```

```

73 The package file \file{ltxutil.sty}
74 is generated from this file, \file{ltxutil.dtx},
75 using the {\sc docstrip} facility of \LaTeX
76 via |tex ltxutil.dtx| (Note: do \emph{not} use \LaTeX\ for this task).
77 The typeset documentation that you are now reading is generated from
78 the same file by typesetting it with \LaTeX\ or pdftex
79 via |latex ltxutil.dtx| or |pdflatex ltxutil.dtx|.
80
81 \subsection{Build Instructions}
82
83 You may bootstrap this suite of files solely from \file{ltxutil.dtx}.
84 Prepare by installing \LaTeXe\ (and either tex or pdftex) on your computer,
85 then carry out the following steps:
86 \begin{enumerate}
87 \item
88 Within an otherwise empty directory,
89 typeset \file{ltxutil.dtx} with \LaTeX\ or pdflatex;
90 you will obtain the typeset documentation you are now reading,
91 along with the file \file{00readme}.
92
93 Note: you will have to run \LaTeX, then
94 \file{makeindex} \texttt{-s gind.ist ltxutil.idx}, then
95 \file{makeindex} \texttt{-s gglo.ist -o ltxutil.gls ltxutil.glo}, then
96 \LaTeX\ again in order to obtain a valid index and table of contents.
97 \item
98 Now typeset \file{ltxutil.dtx} with \TeX (not \LaTeX),
99 thereby generating the package file \file{ltxutil.sty}.
100 \item
101 Install the following files into indicated locations within your
102 TDS-compliant \texttt{texmf} tree (you may need root access):
103 \begin{itemize}
104 \item
105 \file{\$TEXMF/}\file{tex/}\file{latex/}\file{revtex/}\classname{ltxutil.sty}
106 \item
107 \file{\$TEXMF/}\file{source/}\file{latex/}\file{revtex/}\classname{ltxutil.dtx}
108 \item
109 \file{\$TEXMF/}\file{doc/}\file{latex/}\file{revtex/}\classname{ltxutil.pdf}
110 \end{itemize}
111 where \file{\$TEXMF/} stands for \file{texmf-local/}, or some other \texttt{texmf} tree
112 in your installation.
113 \item
114 Run \texttt{mktexlsr} on \file{\$TEXMF/} (you may need root access).
115 \item
116 Build and installation are now complete;
117 now put a \cmd\usepackage\texttt{\{ltxutil\}} in your document preamble!
118 \end{enumerate}
119
120 \subsection{Change Log}
121 \changes{4.0b}{1999/06/20}{A0: Fixed spurious \texttt{CR} and (return) characters in output fil
122 \changes{4.0b}{1999/06/20}{A0: Removed superfluous \cs{def}s, changed to using \cs{floats@sw} a

```

```

123 \changes{4.0b}{1999/06/20}{only execute if there really were floats of the given type}
124 \changes{4.0b}{1999/06/20}{Support the hack with \cs{prepdef}, and delay until \cs{AtBeginDocum
125 \changes{4.0c}{1999/11/13}{(AO, 110) Install hooks for endfloats processing}
126 \changes{4.0c}{1999/11/13}{(AO, 116) Hyperref compatibility}
127 \changes{4.0c}{1999/11/13}{(AO, 130) Interference from array package}
128 \changes{4.0c}{1999/11/13}{*-form mandates pagebreak at each float; only print section head if
129 \changes{4.0d}{2000/04/10}{(AO, 127) Floats placed [h] to allow page breaks}
130 \changes{4.0d}{2000/04/10}{(AO, 174) kernel fix}
131 \changes{4.0d}{2000/05/19}{(AO, 224) Hyperref compatibility.}
132 \changes{4.0d}{2000/05/23}{Allow things to break over pages by setting array@default.}
133 \changes{4.0e}{2000/11/16}{(AO, 221) Remove samepage command from @xfloat@prep: If the float ca
134 \changes{4.0f}{2001/07/13}{(AO, 404) Hyperref compatibility}
135 \changes{4.1a}{2008/01/19}{(AO, 459) do not assume \cs{class@name} is defined}%
136 \changes{4.1a}{2008/01/19}{(AO, 461) Change the csname from \cs{@dotsep} to \cs{!txu@dotsep}. T
137 \changes{4.1a}{2008/01/19}{(AO, 475) I had not properly reproduced the LaTeX macro \cs{eqnarray
138 \changes{4.1a}{2008/01/19}{(AO, 479) Per: Dylan Thurston<dpt at math.harvard.edu>}%
139 \changes{4.1a}{2008/06/30}{(AO) Make \cs{addtocontents} a \cs{!long} \cs{def}; gobble up \cs{foo
140 \changes{4.1a}{2008/06/30}{(AO) Remove code that avoided changes to \cs{@xfootnotemark}}%
141 \changes{4.1a}{2008/06/30}{(AO, 438) Complete rewrite of footnote macros.}
142 \changes{4.1a}{2008/07/07}{\cs{@xfloat@prep} calls \cs{!tx@footnote@pop} to restore the origina
143 \changes{4.1a}{2008/08/12}{\cs{class@documenthook} is the last \cs{AtBeginDocument} token now}
144 \changes{4.1a}{2008/08/12}{Class extension mechanism \cs{@pushfilename!tx} and \cs{@p@pfilename
145 \changes{4.1a}{2008/08/12}{Class extension mechanism \cs{class@extension}, \cs{class@extensionf
146 \changes{4.1a}{2008/08/12}{Get rid of \cs{set@typesize@hook} \cs{set@pica@hook} and the \cs{nor
147 \changes{4.1b}{2008/08/12}{(AO, 487) Support for video figures and the \cs{setfloatlink} comman
148 \changes{4.1b}{2008/08/12}{(AO, 505) try to accommodate \classname{colortbl}.}
149 \changes{4.1b}{2008/08/12}{Acquire \classname{hyperref} savoirre}
150 \changes{4.1b}{2008/08/12}{Default assignment of \cs{float@sw} now, not at \cs{AtBeginDocument}
151 \changes{4.1b}{2008/08/12}{If class option \classoption{lengthcheck} is in effect, log the heig
152 \changes{4.1b}{2008/08/12}{No need to protect against undefined \cs{float@sw}}
153 \changes{4.1b}{2008/08/12}{Patch the array package even later: after all package patches go in.
154 \changes{4.1b}{2008/08/12}{Refine toc processing: provide default.}%
155 \changes{4.1b}{2008/08/12}{Tally and log the height of a float class}
156 \changes{4.1d}{2009/03/27}{(AO, 511) Compatability with lineno.sty's erroneous way of detecting
157 \changes{4.1f}{2009/07/07}{(AO, 515) Hook for setting the font of a footnote}
158 \changes{4.1f}{2009/07/10}{(AO, 518) Tally register overflow when locument is long}
159 \changes{4.1g}{2009/10/06}{(AO, 532) Both arguments of \cs{href} get sanitized}%
160 \changes{4.1g}{2009/10/07}{(AO, 525) Remove phantom paragraph above display math that is given
161 \changes{4.1g}{2009/10/07}{(AO, 539) Use of double-backslash in argument of \cs{section} gives
162
163
164 \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.

```

165 \begin{document}%
166 \expandafter\DocInput\expandafter{\jobname.dtx}%

```



```
167 \end{document}
168 %</doc>
```

4 Using this package

Once this package is installed on your filesystem, you can employ it in adding functionality to L^AT_EX 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 L^AT_EX command or environment, but not at the document level.

5 Compatibility with L^AT_EX's Required Packages

Certain packages, usually ones written by members of the L^AT_EX Project itself, have been designated “required” and are distributed as part of standard L^AT_EX. These packages have been placed in a privileged position vis á vis the L^AT_EX 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 L^AT_EX's kernel procedures `\@outputdblcol`, `\@startcolumn`, and `\@makecol`.

Hereinafter follows some notes on specific L^AT_EX packages.

5.1 array

This package alters the way tabular environments are done, therefore it could run afoul of the L^AT_EX “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 L^AT_EX 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

```
169 %<*package>
170 \def\package@name{ltxutil}%
171 \expandafter\PackageInfo\expandafter{\package@name}{%
172 Utility macros for \protect\LaTeXe,
173 by A. Ogawa (arthur_ogawa at sbcglobal.net)%
174 }%
175 %</package>
```

6.2 Banner and beginning of the kernel DOCSTRIP module

Credit where due.

```
176 %<*kernel>
177 \typeout{%
178 ltxutil [2009/07/07 4.1f]: portions licensed from W. E. Baxter (web at superscript.com)%
179 }%
```

6.3 Errors and warnings

```
\class@err A few shorthands for Class messages. Your document class should define
\class@warn \class@name.
\class@info 180 \def\class@err#1{\ClassError{\class@name}{#1}\@eha}%
181 \def\class@warn#1{\ClassWarningNoLine{\class@name}{#1}}%
182 \def\class@info#1{\ClassInfo{\class@name}{#1}}%
183 \def\obsolete@command#1{%
184 \class@warn@end{Command \string#1\space is obsolete.^^JPlease remove from your document}%
185 \global\let#1\@empty
```

```

186 #1%
187 }%
188 \def\replace@command#1#2{%
189 \class@warn@end{Command \string#1\space is obsolete;^^JUse \string#2\space instead}%
190 \global\let#1#2%
191 #1%
192 }%
193 \def\replace@environment#1#2{%
194 \class@warn@end{Environment #1 is obsolete;^^JUse #2 instead}%
195 \glet@environment{#1}{#2}%
196 \@nameuse{#1}%
197 }%
198 \def\incompatible@package#1{%
199 \@ifpackageloaded{#1}{%
200 \def\@tempa{I cannot continue. You must remove the \string\usepackage\ statement that caused
201 \ClassError{\class@name}{The #1 package cannot be used with \class@name}%
202 \@tempa\stop
203 }{%
204 \class@info{#1 was not loaded (OK!)}%
205 }%
206 }%
207 \def\class@warn@end#1{%
208 \gappdef\class@enddocumenthook{\class@warn{#1}}%
209 }%

Give \class@name a meaning if it does not already have one.
210 \ifx\undefined\class@name
211 \def\class@name{ltxutil}%
212 \class@warn{You should define the class name before reading in this package. Using default}%
213 \fi

```

6.4 New Tools

```

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

\dimen@iii
215 \dimendef\dimen@iii\thr@@

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

\f@ur Analogous to \@ne, \tw@, and \thr@@.
217 \chardef\f@ur=4\relax
218 \chardef\cat@letter=11\relax
219 \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.

```

220 \def\let@environment#1#2{%
221 \expandafter\let
222 \csname#1\expandafter\endcsname\csname#2\endcsname
223 \expandafter\let
224 \csname end#1\expandafter\endcsname\csname end#2\endcsname
225 }%
226 \def\glet@environment#1#2{%
227 \global\expandafter\let
228 \csname#1\expandafter\endcsname\csname#2\endcsname
229 \global\expandafter\let
230 \csname end#1\expandafter\endcsname\csname end#2\endcsname
231 }%

```

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

```

232 \newcommand\tracingplain{%
233 \tracingonline\z@\tracingcommands\z@\tracingstats\z@
234 \tracingpages\z@\tracingoutput\z@\tracinglostchars\@ne
235 \tracingmacros\z@\tracingparagraphs\z@\tracingrestores\z@
236 \showboxbreadth5\showboxdepth3\relax %\errorstopmode
237 }%
238 \newcommand\traceoutput{%
239 \appdef\@resetactivechars{\showoutput}%
240 }%

```

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

```

241 \newcommand\say[1]{\typeout{<\noexpand#1=\meaning#1>}}%
242 \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.

```

243 \def\fullinterlineskip{\prevdepth\z@}%

```

```

\count@i
\count@ii 244 \countdef\count@i\@ne
245 \countdef\count@ii\tw@

```

6.5 Boolean Control

We introduce just enough of the Boolean calculus for T_EX. Alan Jeffrey was the pioneer here, with an article in TUGboat (Vol. 11, No. 2, page 237). This implementation owes a debt to William Baxter (web 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.
`\gapdef` 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.

```

246 \long\def\prepdef#1#2{%
247 \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
248 \toks@ii{#2}%
249 \edef#1{\the\toks@ii\the\toks@}%
250 }%
251 \long\def\appdef#1#2{%
252 \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
253 \toks@ii{#2}%
254 \edef#1{\the\toks@\the\toks@ii}%
255 }%
256 \long\def\gapdef#1#2{%
257 \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
258 \toks@ii{#2}%
259 \global\edef#1{\the\toks@\the\toks@ii}%
260 }%
261 \long\def\appdef@val#1#2{%
262 \appdef#1{#2}}%
263 }%
264 \long\def\appdef@e#1#2{%
265 \expandafter\appdef
266 \expandafter#1%
267 \expandafter{#2}%
268 }%
269 \long\def\appdef@eval#1#2{%
270 \expandafter\appdef@val
271 \expandafter#1%
272 \expandafter{#2}%
273 }%
274 \toksdef\toks@ii=\tw@

```

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

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

```

275 \long\def\@ifxundefined#1{\@ifx{\undefined#1}}%
276 \long\def\@ifnotrelax#1#2#3{\@ifx{\relax#1}{#3}{#2}}%
277 \long\def\@argswap#1#2{#2#1}%
278 \long\def\@argswap@val#1#2{#2{#1}}%
279 \def\@ifxundefined@cs#1{\expandafter\@ifx\expandafter{\csname#1\endcsname\relax}}%

```

`\@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.

```

280 \def\@boolean#1#2{%
281   \long\def#1{%
282     #2% \if<something>
283     \expandafter\true@sw
284   \else
285     \expandafter\false@sw
286   \fi
287 }%
288 }%
289 \def\@boole@def#1#\@boolean{#1}}% Implicit #2

```

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

```

290 \def\@booleantrue#1{\let#1\true@sw}%
291 \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 AMS \LaTeX , we must avoid the following three names: `\@ifempty`, `\@xifempty`, and `\@ifnotempty`.

```

\@ifcat 292 \@boole@def\@ifx#1{\ifx#1}%
\@ifdim 293 \@boole@def\@ifx@empty#1{\ifx@empty#1}%
\@ifhbox 294 \@boole@def\@if@empty#1{\if!#1!}%
\@ifhmode 295 %\@boole@def\@if@sw#1{\csname if#1\endcsname}%
\@ifinner 296 \def\@if@sw#1#2{#1\expandafter\true@sw\else\expandafter\false@sw#2}%
\@ifmmode 297 \@boole@def\@ifdim#1{\ifdim#1}%
\@ifnum 298 \@boole@def\@ifeof#1{\ifeof#1}%
\@ifodd 299 \@boole@def\@ifhbox#1{\ifhbox#1}%
\@ifvbox 300 \@boole@def\@ifhmode{\ifhmode}%
\@ifvmode 301 \@boole@def\@ifinner{\ifinner}%
\@ifvoid 302 \@boole@def\@ifmmode{\ifmmode}%
303 \@boole@def\@ifnum#1{\ifnum#1}%
304 \@boole@def\@ifodd#1{\ifodd#1}%
305 \@boole@def\@ifvbox#1{\ifvbox#1}%
306 \@boole@def\@ifvmode{\ifvmode}%
307 \@boole@def\@ifvoid#1{\ifvoid#1}%

```

`\true@sw` Note that when a Boolean operator expands, it employs two macros that act as selectors, defined here.

```

308 \long\def\true@sw#1#2{#1}%
309 \long\def\false@sw#1#2{#2}%

```

`\loopuntil` Loop control using the Boolean idiom. Superior to `\loop... \repeat` because these can be nested. The tail of the argument must have a Boolean predicate.

```

310 \long\def\loopuntil#1{#1}{\loopuntil{#1}}%
311 \long\def\loopwhile#1{#1}{\loopwhile{#1}}%

```

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

```
312 \def\@provide#1{%
313 \ifx{\undefined#1}{\true@sw}{\ifx{\relax#1}{\true@sw}{\false@sw}}%
314 {\def#1}{\def\j@nk}%
315 }%
```

6.6 Begin Document Structure

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

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

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.

The facility depends on the stability of this part of L^AT_EX's kernel code (the first token of `\document`), which could change, you see. But considering that L^AT_EX is at this point essentially stagnant once more, we risk it.

`\document` We begin by installing hooks into `\document` that we will manage ourselves. First, we do as `\document` does: end the group begun by `\begin`. Last, we conclude our shenanigans by absorbing the first token of the expansion of `\document`, which we assume to be `\endgroup`.

```
316 \prepdef\document{%
317 \endgroup
318 \document@inithook
319 \true@sw}%
320 }%
```

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

```
321 \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`.

```
322 \appdef\document@inithook{%
323 \AtBeginDocument{\class@documenthook}%
324 }%
325 \AtEndDocument{%
```

```

326 \class@enddocumenthook
327 }%
328 \let\class@documenthook\@empty
329 \let\class@enddocumenthook\@empty

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

330 \def\enddocument{%
331 \@enddocumenthook
332 \@checkend{document}}%
333 \clear@document
334 \check@aux
335 \deadcycles\z@
336 \@@end
337 }%
338 \def\clear@document{\clearpage}%
339 \def\check@aux{\do@check@aux}%
340 \def\do@check@aux{%
341 \if@sw\if@files\fi{%
342 \immediate\closeout\@mainaux
343 \let\@setckpt\@gobbletwo
344 \let\@newl@bel\@testdef
345 \@tempwafalse
346 \makeatletter
347 \input\jobname.aux\relax
348 }{}%
349 \@dofilelist
350 \ifdim{\font@submax >\fontsubfuzz\relax}{%
351 \font@warning{%
352 Size substitutions with differences\MessageBreak
353 up to \font@submax\space have occurred.\@gobbletwo
354 }%
355 }{}%
356 \@defaultsubs
357 \@refundefined
358 \if@sw\if@files\fi{%
359 \ifx{\@multiplelabels\relax}{%
360 \if@sw\if@tempsw\fi{%
361 \@latex@warning@no@line{%
362 Label(s) may have changed.
363 Rerun to get cross-references right
364 }%
365 }{}%
366 }{}%
367 \@multiplelabels

```



```

368 }%
369 }{}%
370 }%

```

6.7 Class Extensions

The L^AT_EX procedure `\@onefilewithoptions` is the vehicle for reading in a L^AT_EX 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 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.

```

371 \def\class@extension#1#2{%
372 \IfFileExists{#1.#2}{%
373 \expandafter\class@extensionfile\curname ver@\@currname.\@currentx\endcsname{#1}#2%
374 }{%
375 \curname rtx@#1\endcsname
376 }%
377 }%
378 \def\class@extensionfile#1#2#3{%
379 \@passoptions#3\@unusedoptionlist{#2}%
380 \global\let\@unusedoptionlist\@empty
381 \expandafter\class@ext@hook\curname#2.#3-h@k\endcsname#1{#2}#3%
382 }%
383 \def\class@ext@hook#1#2#3#4{%
384 \@pushfilename@ltx
385 \makeatletter
386 \let\CurrentOption\@empty
387 \@resetoptions
388 \let#1\@empty
389 \xdef\@currname{#3}%
390 \global\let\@currentx#4%
391 \global\let\@clsextension\@currentx
392 \input{#3.#4}%
393 \@ifl@ter#4{#3}#2{%
394 \class@info{Class extension later than: #2}%

```

```

395 }{%
396 \class@info{Class extension earlier: #2}%
397 \@@end
398 }%
399 #1%
400 \let#1\@undefined
401 \expandafter\@p@pfilename@ltx\@currnamestack@ltx\@nil
402 \@reset@options
403 }%

```

`\@pushfilename` But! L^AT_EX does not provide for a class extension other than `.cls`, therefore we must extend L^AT_EX'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.

```

404 \def\@pushfilename@ltx{%
405 \xdef\@currnamestack@ltx{%
406   {\@currname}%
407   {\@currentx}%
408   {\@clsextension}%
409   {\the\catcode'\@}%
410   \@currnamestack@ltx
411 }%
412 }%
413 \def\@p@pfilename@ltx#1#2#3#4#5\@nil{%
414 \gdef\@currname{#1}%
415 \gdef\@currentx{#2}%
416 \gdef\@clsextension{#3}%
417 \catcode'\@#4\relax
418 \gdef\@currnamestack@ltx{#5}%
419 }%
420 \global\let\@currnamestack@ltx\@empty

```

We carefully patch L^AT_EX 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`.

```

421 \def\flushing{%
422 \let\\\@normalcr
423 \leftskip\z@skip
424 \rightskip\z@skip
425 \@rightskip\z@skip
426 \parfillskip\@flushglue
427 }%

```

`\@centercr` The `\@centercr` command is the replacement for `\@normalcr` when setting type centered or ragged. Normally, the meaning of `\\` is `\@normalcr`, which

L^AT_EX 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^AT_EX's early failures that will never get fixed.

The following exemplar fails under L^AT_EX 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.

```
428 \expandafter\DeclareRobustCommand\expandafter\@centercr\expandafter{\@centercr}%
```

6.9 Display Math

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

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

```
429 \def\eqnarray@LaTeX{%
430   \stepcounter{equation}%
431   \def\@currentlabel{\p@equation\theequation}%
432   \global\@eqnswtrue
433   \m@th
434   \global\@eqcnt\z@
435   \tabskip\@centering
436   \let\\\@eqnrcr
437   $$\everycr{\halign to\displaywidth\bgroup
438     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnrel
439     &\global\@eqcnt\@ne\hskip \tw@arraycolsep \hfil${##}$\hfil
440     &\global\@eqcnt\tw@ \hskip \tw@arraycolsep
441     $\displaystyle{##}$\hfil\tabskip\@centering
442     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
443     \tabskip\z@skip
444     \cr
445   }
446 \long\def\eqnarray@fleqn@fixed{%
447   \stepcounter{equation}\def\@currentlabel{\p@equation\theequation}%
448   \global\@eqnswtrue\m@th\global\@eqcnt\z@
449   \tabskip\ltx@mathindent
```

```

450 \let\=\@eqnocr
451 \setlength\abovedisplayskip{\topsep}%
452 \ifvmode\addtolength\abovedisplayskip{\partopsep}\fi
453 \addtolength\abovedisplayskip{\parskip}%
454 \setlength\belowdisplayskip{\abovedisplayskip}%
455 \setlength\belowdisplayshortskip{\abovedisplayskip}%
456 \setlength\abovedisplayshortskip{\abovedisplayskip}%
457 $$%
458 \everycr{}%
459 \halign@\linewidth\bgroup
460 \hskip@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
461 &\global\@eqcnt@one
462 \hskip\tw@eqncolsep
463 \hfil${}\##{}$\hfil
464 &\global\@eqcnt\tw@
465 \hskip\tw@eqncolsep
466 $\displaystyle{##}$\hfil\tabskip@centering
467 &\global\@eqcnt\thr@@\hb@xt@\z@\bgroup\hss#\egroup
468 \tabskip\z@skip
469 \cr
470 }%
471 \@ifx{\eqnarray\eqnarray@LaTeX}{%
472 \class@info{Repairing broken LaTeX eqnarray}%
473 \let\eqnarray\eqnarray@fleqn@fixed
474 \newlength\eqncolsep
475 \setlength\eqncolsep\z@
476 \let\eqnarray@LaTeX\relax
477 \let\eqnarray@fleqn@fixed\relax
478 }{}%

```

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 L^AT_EX 2_ε means. Even though our `eqnarray` environment is modelled after `ltxutil.dtxfleqn.clo`, we must program defensively here.

```

479 \def\ltx@mathindent{\@centering}%
480 \def\set@eqnarray@skips{}%

```

`\prep@math` If we are in vertical mode when display math mode is entered (via `$$`), T_EX will
`\prep@math@patch` 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.

```
481 \def\prep@math{%
482   \@ifvmode{\everypar{\setbox\z@\lastbox}}{}}%
483 }%
484 \def\prep@math@patch{%
485   \prepdef\equation{\prep@math}%
486   \prepdef\eqnarray{\prep@math}%
487 }%
```

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

L^AT_EX bug of long standing; our fix dates to 1987.

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\@footnotetext\@mpfootnotetext
%\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^AT_EX 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 `\footnotetext` are defined (see the L^AT_EX manual for user-level details).

`\footnote` The first user-level command is `\footnote`. A simple way to look at this command is to think of it as `\footnotemark [number] \footnotetext [number] {text}`, where the optional argument is the same in both calls. We also define a syntactical helper procedure `\ltx@xfootnote`.

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.

```

488 \def\footnote{\@ifnextchar[\ltx@xfootnote\ltx@yfootnote}%
489 \def\ltx@xfootnote[#1]{%
490 \ltx@def@footproc\ltx@footmark[#1]%
491 \expandafter\ltx@foottext\expandafter{\the\csname c@\@mpfn\endcsname}%
492 }%
493 \def\ltx@yfootnote{%
494 \ltx@stp@footproc\ltx@footmark
495 \expandafter\ltx@foottext\expandafter{\the\csname c@\@mpfn\endcsname}%
496 }%

```

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.

```

497 \def\footnotemark{\@ifnextchar[\ltx@xfootmark\ltx@yfootmark}%
498 \def\ltx@xfootmark{\ltx@def@footproc\ltx@footmark}%
499 \def\ltx@yfootmark{\ltx@stp@footproc\ltx@footmark}%
500 \def\ltx@footmark#1{%
501 \leavevmode
502 \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
503 \begingroup
504 \expandafter\ltx@make@current@footnote\expandafter{\@mpfn-#1}%
505 \expandafter\@argswap@val\expandafter{\Hy@footnote@currentHref}{\hyper@linkstart {link}}%
506 \@makefnmark
507 \hyper@linkend

```

```

508 \endgroup
509 \ifhmode\spacefactor\@x@sf\fi
510 \relax
511 }%

```

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.

```

512 \def\footnotetext{\@ifnextchar[\ltx@xfoottext\ltx@yfoottext]}%
513 \def\ltx@xfoottext{\ltx@def@footproc\ltx@foottext}%
514 \def\ltx@yfoottext{\ltx@stp@footproc\ltx@foottext}%
515 \long\def\ltx@foottext#1#2{%
516 \begingroup
517 \expandafter\ltx@make@current@footnote\expandafter{\@mpfn}{#1}%
518 \@footnotetext{#2}%
519 \endgroup
520 }%

```

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.

```

521 \def\ltx@def@footproc#1[#2]{%
522 \begingroup
523 \csname c@\@mpfn\endcsname #2\relax
524 \unrestored@protected@xdef\@thefnmark{\thempfn}%
525 \expandafter\endgroup
526 \expandafter#1%
527 \expandafter{\the\csname c@\@mpfn\endcsname}%
528 }%
529 \def\ltx@stp@footproc#1{%
530 \expandafter\stepcounter\expandafter{\@mpfn}%
531 \protected@xdef\@thefnmark{\thempfn}%
532 \expandafter#1%
533 \expandafter{\the\csname c@\@mpfn\endcsname}%
534 }%

```

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 gentler and without hypertext functionality.

```

535 \appdef\class@documenthook{%
536 \let\footnote@latex\footnote
537 \@ifpackageloaded{hyperref}{}{%
538 \let\H@@footnotetext\@footnotetext
539 \def\@footnotetext{\H@@footnotetext}%
540 \let\H@@mpfootnotetext\@mpfootnotetext
541 \def\@mpfootnotetext{\H@@mpfootnotetext}%
542 }%
543 }%

```

In the following, we must use L^AT_EX’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?

```

544 \def\ltx@make@current@footnote#1#2{%
545   \csname c@#1\endcsname#2\relax
546   \protected@edef\Hy@footnote@currentHref{\@currentHref-#1.\csname the#1\endcsname}%
547 }%
548 \def\thempfootnote@latex{\itshape \@alph \@c@mpfootnote }%
549 \def\ltx@thempfootnote{\@alph@c@mpfootnote}%
550 \@ifx{\thempfootnote\thempfootnote@latex}{%
551   \class@info{Repairing hyperref-unfriendly LaTeX definition of \string\mpfootnote}%
552   \let\thempfootnote\ltx@thempfootnote
553 }{}%

```

Note on `hyperref` compatibility: In its “Automated L^AT_EX 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^AT_EX’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^AT_EX hypertext cross-references” does not do any of this:

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

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

```

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

```


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

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

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

```
554 \def\@makefnmark{%
555   \hbox{%
556     \@textsuperscript{%
557       \normalfont\itshape\@thefnmark
558     }%
559   }%
560 }
```

```
561 \long\def\@footnotetext{%
562   \insert\footins\bgroup
563   \make@footnotetext
564 }
```

```
565 \long\def\@mpfootnotetext{%
566   \minipagefootnote@pick
567   \make@footnotetext
568 }
```

```
569 \long\def\make@footnotetext#1{%
570   \set@footnotefont
```

As noted above, we do *not* do `\floatingpenalty \@MM`.

```
571   \set@footnotewidth
572   \@parboxrestore
573   \protected@edef\@currentlabel{%
574     \csname p@footnote\endcsname\@thefnmark
575   }%
576   \color@begingroup
577   \@makefntext{%
578     \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox
579   }%
580   \color@endgroup
581   \minipagefootnote@drop
582 }
```

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

```

583 \def\set@footnotefont{%
584   \reset@font\footnotesize
585   \interlinepenalty\interfootnotelinepenalty
586   \splittopskip\footnotesep
587   \splitmaxdepth\dp\strutbox
588 }%

```

`\set@footnotewidth` is the procedure for setting the width of a footnote.

```

589 \def\set@footnotewidth{%
590   \hsize\columnwidth
591   \linewidth\hsize
592 }%
593 \def\set@footnotewidth@ii{%
594   \hsize\textwidth
595   \advance\hsize\columnsep
596   \divide\hsize\tw@
597   \advance\hsize-\columnsep
598   \linewidth\hsize
599 }%

```

6.11 Floats

6.11.1 Usage notes

We extend the L^AT_EX kernel for three purposes:

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

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

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

```

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

```

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

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

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

```
\addtocontents
\robustify@contents 600 \def\robustify@contents{%
601 \let \label \@gobble
602 \let \index \@gobble
603 \let \glossary \@gobble
604 \let\footnote \@gobble
605 \def\({\string\}%
606 \def\)\{string\})%
607 \def\{\{string\}\}%
608 }%
609 \long\def\addtocontents#1#2{%
610 \protected@write\@auxout{\robustify@contents}{\string \@writefile {#1}{#2}}%
611 }%
```

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^AT_EX. We fix things up by always writing that fourth argument and by supplying a `\contentsline` command that can read them.

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

Finally, we wrap L^AT_EX'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.

```
612 \def\addcontentsline#1#2#3{%
613   \addtocontents{#1}{%
614     \protect\contentsline{#2}{#3}{\thepage}{}%
615   }%
616 }%
617 \def\label#1{%
618   \@bsphack
619   \protected@write\@auxout{}{%
620     \string\newlabel{#1}{\@currentlabel}{\thepage}{}{}}%
621   }%
622   \@esphack
623 }%
624 \def\ltx@contentsline#1{%
625   \expandafter\@ifnotrelax\csname l@#1\endcsname{}{%
626     \expandafter\let\csname l@#1\endcsname\@gobbletwo
627   }%
628   \contentsline@latex{#1}%
629 }%
630 \appdef\document@inithook{%
631   \let\contentsline@latex\contentsline
632   \let\contentsline\ltx@contentsline
633 }%
```

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^AT_EX syntax at this point: If a caption is used below a table, then put out the footnotes before the caption.

```
634 \appdef\class@documenthook{%
635   \prepdef\caption{\minipagefootnote@here}%
636 }%
```

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

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

```

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

```

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

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

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

```

\minipagefootnote@pick 637 \def\minipagefootnote@init{%
\minipagefootnote@drop 638 \setbox\@mpfootins\box\voidb@x
639 }%
640 \def\minipagefootnote@pick{%
641 \global\setbox\@mpfootins\vbox\bgroup
642 \unvbox\@mpfootins
643 }%
644 \def\minipagefootnote@drop{%
645 \egroup
646 }%
647 \def\minipagefootnote@here{%
648 \par
649 \@ifvoid\@mpfootins{}{%
650 \vskip\skip\@mpfootins
651 \fullinterlineskip
652 \@ifinner{%
653 \vtop{\unvcopy\@mpfootins}%
654 {\setbox\z@\lastbox}%
655 }{}%
656 \unvbox\@mpfootins
657 }%
658 }%
659 \def\minipagefootnote@foot{%
660 \@ifvoid\@mpfootins{}{%
661 \insert\footins\bgroup\unvbox\@mpfootins\egroup
662 }%
663 }%
664 \def\endminipage{%
665 \par
666 \unskip
667 \minipagefootnote@here
668 \@minipagefalse %% added 24 May 89

```

```

669 \color@endgroup
670 \egroup
671 \expandafter\@iiparbox\@mpargs{\unvbox\@tempboxa}%
672 }%

```

`\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.

```
673 \@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.

```

674 \let\@xfloat@LaTeX\@xfloat
675 \def\@xfloat#1[#2]{%
676   \@xfloat@prep
677   \@nameuse{fp@proc#2}%
678   \floats@sw{\@xfloat@LaTeX{#1}[#2]}{\@xfloat@anchored{#1}[]}%
679 }%
680 \def\@xfloat@prep{%
681   \ltx@footnote@pop
682   \def\@mpfn{mpfootnote}%
683   \def\thempfn{\thempfootnote}%
684   \c@mpfootnote\z@
685   \let\H@footnotetext\H@mpfootnotetext
686 }%
687 \let\ltx@footnote@pop\@empty
688 \def\@xfloat@anchored#1[#2]{%
689   \def\@capttype{#1}%
690   \begin@float@pagebreak
691   \let\end@float\end@float@anchored
692   \let\end@dblfloat\end@float@anchored
693     \hsize\columnwidth
694     \@parboxrestore
695     \floatboxreset
696   \minipagefootnote@init
697 }%
698 \def\end@float@anchored{%
699   \minipagefootnote@here
700   \par\vskip\z@skip
701   \par
702   \end@float@pagebreak
703 }%
704 \def\begin@float@pagebreak{\par\addvspace\intextsep}%
705 \def\end@float@pagebreak{\par\addvspace\intextsep}%
706 \def\@mpmakefntext#1{%
707   \parindent=1em

```

```

708 \noindent
709 \hb@xt@1em{\hss\@makefnmark}%
710 #1%
711 }%

```

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.

```

712 \def\do@if@floats#1#2{%
713 \floats@sw}{-%

```

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

```

714 \expandafter\newwrite
715           \csname#1write\endcsname
716 \expandafter\def
717           \csname#1@stream\endcsname{\jobname#2}%
718 \expandafter\immediate
719 \expandafter\openout
720           \csname#1write\endcsname
721           \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`.

```

722 \@ifxundefined\@float@LaTeX{%
723 \let\@float@LaTeX\@float
724 \let\@dblfloat@LaTeX\@dblfloat
725 \let\@float\write@float
726 \let\@dblfloat\write@floats
727 }{%
728 \let@environment{#1@float}{#1}%
729 \let@environment{#1@floats}{#1*}%
730 \@ifxundefined@cs{#1@write}{}{-%
731 \let@environment{#1}{#1@write}%
732 }%
733 }%
734 }%

```

`\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.

```

735 \def\triggerpar{\leavevmode\@par}%
736 \def\onepage{\def\begin@float@pagebreak{\newpage}\def\end@float@pagebreak{\newpage}}%
737 \def\print@float#1#2{%

```

```

738 \lengthcheck@sw{%
739 \total@float{#1}%
740 }{%
741 \@ifundefined@cs{#1write}{}{%
742 \begingroup
743 \@booleanfalse\floats@sw
744 #2%
745 \raggedbottom
746 \def\array@default{v}% floats must
747 \let\@float\@float@LaTeX
748 \let\@dblfloat\@dblfloat@LaTeX
749 \let\trigger@float@par\triggerpar
750 \let@environment{#1}{#1@float}%
751 \let@environment{#1*}{#1@floats}%
752 \expandafter\prepdef\csname#1\endcsname{\trigger@float@par}%
753 \expandafter\prepdef\csname#1*\endcsname{\trigger@float@par}%
754 \@namedef{fps@#1}{h!}%
755 \expandafter\immediate
756 \expandafter\closeout
757 \csname#1write\endcsname
758 \everypar{%
759 \global\let\trigger@float@par\relax
760 \global\everypar{}\setbox\z@\lastbox
761 \@ifundefined@cs{#1sname}{}{%
762 \begin@float@pagebreak
763 \expandafter\section
764 \expandafter*%
765 \expandafter{%
766 \csname#1sname\endcsname
767 }%
768 }%
769 }%
770 \input{\csname#1@stream\endcsname}%
771 \endgroup
772 \global\expandafter\let\csname#1write\endcsname\relax
773 }%
774 }%

```

`\tally@float` If we are tallying column inches, `\tally@float` tallies a contribution to `\ftype@`
`\total@float` `\@cuptype`, 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 .

```

775 \chardef\@xvi=16\relax
776 \mathchardef\@twopowerfourteen="4000
777 \mathchardef\@twopowertwo="4
778 \def\tally@float#1{%
779 \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.

```

780 \@tempcnta\count\@currbox
781 \divide\@tempcnta\@xxxii
782 \multiply\@tempcnta\@xxxii
783 \advance\count\@currbox-\@tempcnta
784 \divide\@tempcnta\@xxxii
785 \@ifnum{\count\@currbox>\@xvi}{%
786   \advance\count\@currbox-\@xvi\@booleantrue\@temp@sw
787 }{%
788   \@booleanfalse\@temp@sw
789 }%
```

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

```

790 \show@box@size@sw{%
791   \class@info{Float #1
792     (\the\@tempcnta)[\@temp@sw{16+}{}\the\count\@currbox]^^J%
793     (\the\ht\@currbox+\the\dp\@currbox)X\the\wd\@currbox
794   }%
795 }-}%
796 \endgroup
```

Here we tally the height of this float object.

```

797 \expandafter\let
798 \expandafter\@tempa
799   \csname fbox@\csname ftype@#1\endcsname\endcsname
800 \@ifnotrelax\@tempa{%
801   \@ifhbox\@tempa{%
802     \setbox\@tempboxa\vbox{\unvcopy\@currbox\hrule}%
803     \dimen@ht\@tempboxa
804     \divide\dimen@\@twopowerfourteen
805     \@ifdim{\wd\@tempboxa<\textwidth}{%
806       \advance\dimen@ht\@tempa
807       \global\ht\@tempa\dimen@
808     }{%
809       \advance\dimen@dp\@tempa
810       \global\dp\@tempa\dimen@
811     }%
812   }-}%
813 }-}%
814 }%
815 \def\total@float#1{%
816 \expandafter\let
817 \expandafter\@tempa
818   \csname fbox@\csname ftype@#1\endcsname\endcsname
819 \@ifnotrelax\@tempa{%
820   \@ifhbox\@tempa{%
821     \@tempdima\the\ht\@tempa\divide\@tempdima\@twopowertwo\@tempcnta\@tempdima
822     \@tempdimb\the\dp\@tempa\divide\@tempdimb\@twopowertwo\@tempcntb\@tempdimb
```

```

823   \class@info{Total #1: Column(\the\@tempcnta pt), Page(\the\@tempcnta pt)}%
824   }{}%
825 }{}%
826 }%

```

`\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`.

```

827 \def\write@float#1{\write@@float{#1}{#1}}%
828 \def\endwrite@float{\@Esphack}%
829 \def\write@floats#1{\write@@float{#1*}{#1}}%
830 \def\endwrite@floats{\@Esphack}%

```

`\write@@float`

```

831 \def\write@@float#1#2{%
832   \ifhmode
833     \@bsphack
834   \fi
835   \chardef\@tempc\csname#2write\endcsname
836   \toks@{\begin{#1}}%
837   \def\@tempb{#1}%
838   \expandafter\let\csname end#1\endcsname\endwrite@float
839   \catcode'\^M\active
840   \@makeother{\@makeother\}\@makeother\%
841   \write@floatline
842 }%

```

`\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.

```

843 \begingroup
844 \catcode'\[the\catcode'\{\catcode'\}\the\catcode'\}\@makeother\{\@makeother\}%
845 \gdef\float@end@tag#1\end{#2}#3\nul[%
846   \def\@tempa{#2}%
847   \@ifx[\@tempa\@tempb][\end{#2}][\write@floatline]%
848 ]%
849 \obeylines%
850 \gdef\write@floatline#1^M[%
851   \begingroup%
852   \newlinechar'\^M%
853   \toks@\expandafter[the\toks@#1]\immediate\write\@tempc[the\toks@]%
854   \endgroup%
855   \toks@[]%
856   \float@end@tag#1\end{\}\@nul%
857 ]%
858 \endgroup

```

6.12 Counters

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

```
859 \def\@alph#1{\ifcase#1\or a\or b\or c\or d\else\@ialph{#1}\fi}
860 \def\@ialph#1{\ifcase#1\or \or \or \or \or e\or f\or g\or h\or i\or j\or
861 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
862 y\or z\or aa\or bb\or cc\or dd\or ee\or ff\or gg\or hh\or ii\or jj\or
863 kk\or ll\or mm\or nn\or oo\or pp\or qq\or rr\or ss\or tt\or uu\or
864 vv\or ww\or xx\or yy\or zz\else\@ctrerr\fi}
```

6.13 Customization of Sections

Patch the standard L^AT_EX sectioning procedure to:

- Allow a sectioning command to trigger the title page, or more generally to recognize that it is the first object in the document, so we headpatch `\@startsection`.
- Allow a tail command in `#6` to uppercase the title, so we retain DPC's braces.
- Allow each type of sectioning command to format its number differently, so we generalize `\@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.

```
865 \def\@startsection#1#2#3#4#5#6{%
866 \@startsection@hook
867 \if@noskipsec \leavevmode \fi
868 \par
869 \@tempskipa #4\relax
870 \@afterindenttrue
871 \ifdim \@tempskipa <\z@
872 \@tempskipa -\@tempskipa \@afterindentfalse
873 \fi
874 \if@nobreak
875 \everypar{ }%
```

```

876 \else
877 \addpenalty\@secpenalty\addvspace\@tempskipa
878 \fi
879 \@ifstar
880 {\@dblarg{\@sect@ltx{#1}{#2}{#3}{#4}{#5}{#6}}}%
881 {\@dblarg{\@sect@ltx {#1}{#2}{#3}{#4}{#5}{#6}}}%
882 }%
883 \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^AT_EX procedure.

```

884 \class@info{Repairing broken LateX \string\@sect}%
885 \def\@sect@ltx#1#2#3#4#5#6[#7]#8{%
886 \ifnum{#2}>\c@secnumdepth}{%
887 \def\H@svsec{\phantomsection}%
888 \let\@svsec\@empty
889 }{%
890 \H@refstepcounter{#1}%
891 \def\H@svsec{%
892 \phantomsection
893 }%
894 \protected@edef\@svsec{{#1}}%
895 \@ifundefined{@#1cntformat}{%
896 \prepdef\@svsec\@seccntformat
897 }{%
898 \expandafter\prepdef
899 \expandafter\@svsec
900 \csname @#1cntformat\endcsname
901 }%
902 }%
903 \@tempskipa #5\relax
904 \@ifdim{\@tempskipa>\z@}{%
905 \begingroup
906 \interlinepenalty \@M
907 #6{%
908 \@ifundefined{@hangfrom@#1}{\@hang@from}{\csname @hangfrom@#1\endcsname}}%
909 {\hskip#3\relax\H@svsec}{\@svsec}{#8}%
910 }%
911 \@@par
912 \endgroup

```

```

913 \@ifundefined{#1mark}{\@gobble}{\csname #1mark\endcsname}{#7}%
914 \addcontentsline{toc}{#1}{%
915 \@ifnum{#2}>\c@secnumdepth}{%
916 \protect\numberline{}}%
917 }{%
918 \protect\numberline{\csname the#1\endcsname}%
919 }%
920 #8}%
921 }{%
922 \def\@svsechd{%
923 #6{%
924 \@ifundefined{@runin@to@#1}{\@runin@to}{\csname @runin@to@#1\endcsname}%
925 {\hskip#3\relax\H@svsec}{\@svsec}{#8}%
926 }%
927 \@ifundefined{#1mark}{\@gobble}{\csname #1mark\endcsname}{#7}%
928 \addcontentsline{toc}{#1}{%
929 \@ifnum{#2}>\c@secnumdepth}{%
930 \protect\numberline{}}%
931 }{%
932 \protect\numberline{\csname the#1\endcsname}%
933 }%
934 #8}%
935 }%
936 }%
937 \@xsect{#5}%
938 }%
939 \def\@hang@from#1#2#3{\@hangfrom{#1#2}#3}%
940 \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`.

```

941 \def\@ssect@ltx#1#2#3#4#5#6[#7]#8{%
  Removed \def\@currentlabelname{#8}
942 \def\H@svsec{\phantomsection}%
943 \@tempskipa #5\relax
944 \@ifdim{\@tempskipa}>\z@}{%
945 \begingroup
946 \interlinepenalty \@M
947 #6{%
948 \@ifundefined{@hangfroms@#1}{\@hangfroms}{\csname @hangfroms@#1\endcsname}%
  Removed {\hskip#3\relax\H@svsec}{\Sectionformat{#8}{#1}}

```

```

949     {\hskip#3\relax\H@svsec}{#8}%
950     }%
951     \@@par
952     \endgroup
953     \@ifundefined{#1smark}{\@gobble}{\csname #1smark\endcsname}{#7}%
954     \addcontentsline{toc}{#1}{\protect\numberline{#8}}%
955     }{%
956     \def\@svsechd{%
957         #6%
958         \@ifundefined{@runin@tos@#1}{\@runin@tos}{\csname @runin@tos@#1\endcsname}%
959         Removed {\hskip#3\relax\H@svsec}{\Sectionformat{#8}{#1}}
960     }%
961     \@ifundefined{#1smark}{\@gobble}{\csname #1smark\endcsname}{#7}%
962     \addcontentsline{toc}{#1}{\protect\numberline{#8}}%
963     }%
964     }%
965     \@xsect{#5}%
966     }%
967     \def\@hang@froms#1#2{#1#2}%
968     \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.

```

969 \def\init@hyperref{%
970 \providecommand\phantomsection{}%
971 \providecommand\hyper@makecurrent[1]{}%
972 \providecommand\Hy@raisedlink[1]{}%
973 \providecommand\hyper@anchorstart[1]{}%
974 \providecommand\hyper@anchorend{}%
975 \providecommand\hyper@linkstart[2]{}%
976 \providecommand\hyper@linkend{}%
977 \providecommand\@currentHref{}%
978 }%
979 \let\H@refstepcounter\refstepcounter
980 \appdef\document@inithook{%
981 \init@hyperref
982 }%

```

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

The following setting gives the L^AT_EX default.

```
983 \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 `\endarray` `\endarray` tabular and array environments. A document class can define these hooks as needed.

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

```
984 \apptdef\document@inithook{%
985   \ifpackageloaded{array}{\switch@array}{\switch@tabular}%
986   \prepdef\endtabular{\endtabular@hook}%
987   \@provide\endtabular@hook}%
988 \prepdef\endarray{\endarray@hook}%
989 \@provide\endarray@hook}%
990 \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.

```
991 \prepdef\@tabular{\tabular@hook}%
992 \@provide\@tabular@hook}%
993 }%
```

`\switch@tabular` The two procedures `\switch@tabular` and `\switch@array` apply needed patches to the various tabular procedures, the former applying to the L^AT_EX kernel, the latter to the required `array` package (and to the number of other required packages that load it).

```
994 \def\switch@tabular{%
995   \let\@array@sw\@array@sw@array
996   \@ifx{\@array\@array@LaTeX}{%
997     \@ifx{\multicolumn\multicolumn@LaTeX}{%
998       \@ifx{\@tabular\@tabular@LaTeX}{%
999         \@ifx{\@tabarray\@tabarray@LaTeX}{%
1000          \@ifx{\array\array@LaTeX}{%
1001            \@ifx{\endarray\endarray@LaTeX}{%
1002              \@ifx{\endtabular\endtabular@LaTeX}{%
1003                \@ifx{\@mkpream\@mkpream@LaTeX}{%
1004                  \@ifx{\@addamp\@addamp@LaTeX}{%
```

```

1005 \@ifx{\@arrayacol\@arrayacol@LaTeX}{%
1006 \@ifx{\@tabacol\@tabacol@LaTeX}{%
1007 \@ifx{\@arrayclassz\@arrayclassz@LaTeX}{%
1008 \@ifx{\@tabclassiv\@tabclassiv@LaTeX}{%
1009 \@ifx{\@arrayclassiv\@arrayclassiv@LaTeX}{%
1010 \@ifx{\@tabclassz\@tabclassz@LaTeX}{%
1011 \@ifx{\@classv\@classv@LaTeX}{%
1012 \@ifx{\hline\hline@LaTeX}{%
1013 \@ifx{\@tabularcr\@tabularcr@LaTeX}{%
1014 \@ifx{\@xtabularcr\@xtabularcr@LaTeX}{%
1015 \@ifx{\@xargarraycr\@xargarraycr@LaTeX}{%
1016 \@ifx{\@yargarraycr\@yargarraycr@LaTeX}{%
1017 \true@sw
1018 }{%
1019 \false@sw
1020 }%
1021 }{%
1022 \false@sw
1023 }%
1024 }{%
1025 \false@sw
1026 }%
1027 }{%
1028 \false@sw
1029 }%
1030 }{%
1031 \false@sw
1032 }%
1033 }{%
1034 \false@sw
1035 }%
1036 }{%
1037 \false@sw
1038 }%
1039 }{%
1040 \false@sw
1041 }%
1042 }{%
1043 \false@sw
1044 }%
1045 }{%
1046 \false@sw
1047 }%
1048 }{%
1049 \false@sw
1050 }%
1051 }{%
1052 \false@sw
1053 }%
1054 }{%

```



```

1055     \false@sw
1056     }%
1057     }{%
1058     \false@sw
1059     }%
1060     }{%
1061     \false@sw
1062     }%
1063     }{%
1064     \false@sw
1065     }%
1066     }{%
1067     \false@sw
1068     }%
1069     }{%
1070     \false@sw
1071     }%
1072     }{%
1073     \false@sw
1074     }%
1075     }{%
1076     \false@sw
1077     }%
1078     }{%
1079     \false@sw
1080     }%
1081     {%
1082     \class@info{Patching LaTeX tabular.}%
1083     }{%
1084     \class@info{Unrecognized LaTeX tabular. Please update this document class! (Proceeding with f
1085     }%
1086     \let\@array\@array@ltx
1087     \let\multicolumn\multicolumn@ltx
1088     \let\@tabular\@tabular@ltx
1089     \let\@tabarray\@tabarray@ltx
1090     \let\array\array@ltx
1091     \let\endarray\endarray@ltx
1092     \let\endtabular\endtabular@ltx
1093     \let\@mkpream\@mkpream@ltx
1094     \let\@addamp\@addamp@ltx
1095     \let\@arrayacol\@arrayacol@ltx
1096     \let\@tabacol\@tabacol@ltx
1097     \let\@arrayclassz\@arrayclassz@ltx
1098     \let\@tabclassiv\@tabclassiv@ltx
1099     \let\@arrayclassiv\@arrayclassiv@ltx
1100     \let\@tabclassz\@tabclassz@ltx
1101     \let\@classv\@classv@ltx
1102     \let\hline\hline@ltx
1103     \let\@tabularcr\@tabularcr@ltx
1104     \let\@xtabularcr\@xtabularcr@ltx

```

```

1105 \let\@xargarraycr\@xargarraycr@ltx
1106 \let\@yargarraycr\@yargarraycr@ltx
1107 }%

1108 \def\switch@array{%
1109 \@ifpackageloaded{colortbl}{\let\switch@array@info\colortbl@message}{\let\switch@array@info\ar
1110 \let\@array@sw\@array@sw@LaTeX
1111 \@ifx{\@array\@array@array}{%
1112 \@ifx{\@tabular\@tabular@array}{%
1113 \@ifx{\@tabarray\@tabarray@array}{%
1114 \@ifx{\@array\array@array}{%
1115 \@ifx{\endarray\endarray@array}{%
1116 \@ifx{\endtabular\endtabular@array}{%
1117 \@ifx{\@mkpream\@mkpream@array}{%
1118 \@ifx{\@classx\@classx@array}{%
1119 \@ifx{\insert@column\insert@column@array}{%
1120 \@ifx{\@arraycr\@arraycr@array}{%
1121 \@ifx{\@xarraycr\@xarraycr@array}{%
1122 \@ifx{\@xargarraycr\@xargarraycr@array}{%
1123 \@ifx{\@yargarraycr\@yargarraycr@array}{%
1124 \true@sw
1125 }{%
1126 \false@sw
1127 }%
1128 }{%
1129 \false@sw
1130 }%
1131 }{%
1132 \false@sw
1133 }%
1134 }{%
1135 \false@sw
1136 }%
1137 }{%
1138 \false@sw
1139 }%
1140 }{%
1141 \false@sw
1142 }%
1143 }{%
1144 \false@sw
1145 }%
1146 }{%
1147 \false@sw
1148 }%
1149 }{%
1150 \false@sw
1151 }%
1152 }{%
1153 \false@sw

```

```

1154 }%
1155 }{%
1156 \false@sw
1157 }%
1158 }{%
1159 \false@sw
1160 }%
1161 }{%
1162 \false@sw
1163 }{%
1164 \class@info{Patching array package.}%
1165 }{%
1166 \switch@array@info
1167 }%
1168 \let\@array \@array@array@new
1169 \let\@@array \@array % Così fan tutti
1170 \let\@tabular \@tabular@array@new
1171 \let\@tabarray \@tabarray@array@new
1172 \let\array \@array@array@new
1173 \let\endarray \endarray@array@new
1174 \let\endtabular\endtabular@array@new
1175 \let\@mkpream \@mkpream@array@new
1176 \let\@classx \@classx@array@new
1177 \let\@arrayacol\@arrayacol@ltx
1178 \let\@tabacol \@tabacol@ltx
1179 \let\insert@column\insert@column@array@new
1180 \expandafter\let\csname endtabular*\endcsname\endtabular % Così fan tutti
1181 \let\@arraycr \@arraycr@new
1182 \let\@xarraycr \@xarraycr@new
1183 \let\@xargarraycr\@xargarraycr@new
1184 \let\@yargarraycr\@yargarraycr@new
1185 }%
1186 \def\array@message{%
1187 \class@info{Unrecognized array package. Please update this document class! (Proceeding with fi
1188 }%
1189 \def\colortbl@message{%
1190 \class@info{colortbl package is loaded. (Proceeding with fingers crossed.)}%
1191 }%

```

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

```

1192 \def\@array@sw@LaTeX{\@ifx{\@tabularcr}}%
1193 \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.

```

1194 \def\@tabular@LaTeX{%
1195 \leavevmode
1196 \hbox\bgroup$%
1197 \let\@acol\@tabacol
1198 \let\@classz\@tabclassz
1199 \let\@classiv\@tabclassiv
1200 \let\\\@tabularcr
1201 \@tabarray
1202 }%
1203 \def\@tabular@ltx{%
1204 \let\@acoll\@tabacoll
1205 \let\@acolr\@tabacolr
1206 \let\@acol\@tabacol
1207 \let\@classz\@tabclassz
1208 \let\@classiv\@tabclassiv
1209 \let\\\@tabularcr
1210 \@tabarray
1211 }%
1212 \def\@tabular@array{%
1213 \leavevmode
1214 \hbox\bgroup$%
1215 \col@sep\tabcolsep
1216 \let\d@llarbegin\beginngroup
1217 \let\d@llarend\endngroup
1218 \@tabarray
1219 }%
1220 \def\@tabular@array@new{%
1221 \let\@acoll\@tabacoll
1222 \let\@acolr\@tabacolr
1223 \let\@acol\@tabacol
1224 \sepundefined
1225 \let\d@llarbegin\beginngroup
1226 \let\d@llarend\endngroup
1227 \@tabarray
1228 }%

```

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

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

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

```

1228 \def\@tabarray@LaTeX{%
1229 \m@th\@ifnextchar[\@array{\@array[c]}%
1230 }%
1231 \def\@tabarray@ltx{%

```

```

1232 \m@th@ifnextchar[\@array{\expandafter\@array\expandafter[\array@default]}%
1233 }%
1234 \def\@tabarray@array{%
1235 \ifnextchar[{\@array}{\@array[c]}%
1236 }%
1237 \def\@tabarray@array@new{%
1238 \ifnextchar[{\@array}{\expandafter\@array\expandafter[\array@default]}%
1239 }%

```

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

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

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

```

1240 \newcount\intertabularlinepenalty
1241 \intertabularlinepenalty=100
1242 \newcount\@tbpen
1243 \apptdef\samepage{\intertabularlinepenalty\@M}%
1244 \def\@tabularcr@LaTeX{\ifnum 0='}\fi \ifstar \@xtabularcr \@xtabularcr}%
1245 \def\@tabularcr<ltx{\ifnum 0='}\fi \ifstar {\global \@tbpen \@M \@xtabularcr }{\global \@tbpen \@M}
1246 \def\@xtabularcr@LaTeX{\ifnextchar [\@argtabularcr {\ifnum 0='{ \fi } \cr }}%
1247 \def\@xtabularcr<ltx{\ifnextchar [\@argtabularcr {\ifnum 0='{ \fi } \cr \noalign {\penalty \@tbpen}
1248 \def\@xargarraycr@LaTeX#1{\@tempdima #1\advance \@tempdima \dp \@arstrutbox \vrule \@height \z@ \width \@width}
1249 \def\@xargarraycr<ltx#1{\@tempdima #1\advance \@tempdima \dp \@arstrutbox \vrule \@height \z@ \width \@width}
1250 \def\@yargarraycr@LaTeX#1{\cr \noalign {\vskip #1}}%
1251 \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.

```

1252 \def\@arraycr@array{%
1253 \relax
1254 \iffalse{\fi\ifnum 0='}\fi
1255 \@ifstar \@xarraycr \@xarraycr
1256 }%
1257 \def\@arraycr@new{%
1258 \relax

```

```

1259 \iffalse{\fi\ifnum 0='}\fi
1260 \@ifstar {\global \@tbpen \@M \@xarraycr }{\global \@tbpen \intertabularlinepenalty \@xarraycr
1261 }%
1262 \def\@xarraycr@array{%
1263 \@ifnextchar [%]
1264 \@argarraycr {\ifnum 0='{}\fi\cr}%
1265 }%
1266 \def\@xarraycr@new{%
1267 \@ifnextchar [%]
1268 \@argarraycr {\ifnum 0='{}\fi\cr \noalign {\penalty \@tbpen }}%
1269 }%
1270 \def\@xargarraycr@array#1{%
1271 \unskip
1272 \@tempdima #1\advance\@tempdima \dp\@arstrutbox
1273 \vrule \@depth\@tempdima \@width\z@
1274 \cr
1275 }%
1276 \def\@xargarraycr@new#1{%
1277 \unskip
1278 \@tempdima #1\advance\@tempdima \dp\@arstrutbox
1279 \vrule \@depth\@tempdima \@width\z@
1280 \cr
1281 \noalign {\penalty \@tbpen }}%
1282 }%
1283 \def\@yargarraycr@array#1{%
1284 \cr
1285 \noalign{\vskip #1}%
1286 }%
1287 \def\@yargarraycr@new#1{%
1288 \cr
1289 \noalign{\penalty \@tbpen \vskip #1}%
1290 }%

```

`\array` We provide old and new versions of the `\array` procedure for both \LaTeX 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.

```

1291 \def\array@LaTeX{%
1292 \let\@acol\@arrayacol
1293 \let\@classz\@arrayclassz
1294 \let\@classiv\@arrayclassiv
1295 \let\@arraycr
1296 \let\@halignto\@empty
1297 \let\@tabarray
1298 }%
1299 \def\array@ltx{%
1300 \@ifmmode{\@badmath}%
1301 \let\@acoll\@arrayacol
1302 \let\@acolr\@arrayacol
1303 \let\@acol\@arrayacol

```

```

1304 \let\@classz\@arrayclassz
1305 \let\@classiv\@arrayclassiv
1306 \let\\\@arraycr
1307 \let\@halignto\@empty
1308 \@tabarray
1309 }%
1310 \def\array@array{%
1311 \col@sep\arraycolsep
1312 \def\d@llarbegin{$}\let\d@llarend\d@llarbegin\gdef\@halignto{}%
1313 \@tabarray
1314 }
1315 \def\array@array@new{%
1316 \@ifmode{}{\@badmath$}%
1317 \let\@acoll\@arrayacol
1318 \let\@acolr\@arrayacol
1319 \let\@acol\@arrayacol
    Removed: \let\col@sep\@undefined
1320 \def\d@llarbegin{$}%
1321 \let\d@llarend\d@llarbegin
1322 \gdef\@halignto{}%
1323 \@tabarray
1324 }%

```

`\@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.

```

1325 \def\@array@LaTeX[#1]#2{%
1326 \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi\fi
1327 \bgroup
1328 \setbox\@arstrutbox\hbox{%
1329 \vrule \@height\arraystretch\ht\strutbox
1330 \quad \@depth\arraystretch \dp\strutbox
1331 \quad \@width\z@}%
1332 \@mkpream{#2}%
1333 \edef\@preamble{%
1334 \ialign \noexpand\@halignto
1335 \bgroup \@arstrut \@preamble \tabskip\z@skip \cr}%
1336 \let\@startpbox\@startpbox \let\@endpbox\@endpbox
1337 \let\tabularnewline\%
1338 \let\par\@empty
1339 \let\@sharp##%
1340 \set@typeset@protect
1341 \lineskip\z@skip\baselineskip\z@skip

```

```

1342 \ifhmode \@preamerr\z@ \@par\fi
1343 \@preamble
1344 }%
1345 \def\@array@ltx[#1]#2{%
1346 \@nameuse{\@array@align@#1}%
1347 \set@arstrutbox
1348 \@mkpream{#2}%
1349 \prepdef\@preamble{%
1350 \tabskip\@tabmid@skip
1351 \@arstrut
1352 }%
1353 \appdef\@preamble{%
1354 \tabskip\@tabright@skip
1355 \cr
1356 \array@row@pre
1357 }%
1358 % \let\@startpbox\@startpbox
1359 % \let\@endpbox\@endpbox
1360 \let\@tabularnewline\@%
1361 \let\@par\@empty
1362 \let\@sharp##%
1363 \set@typeset@protect
1364 \lineskip\z@skip\baselineskip\z@skip
1365 \tabskip\@tableft@skip\relax
1366 \ifhmode \@preamerr\z@ \@par\fi
1367 \everycr{}%
1368 \expandafter\@halign\expandafter\@halignto\expandafter\@bgroup\@preamble
1369 }%
1370 %
1371 \def\set@arstrutbox{%
1372 \setbox\@arstrutbox\hbox{%
1373 \vrule \@height\arraystretch\ht\strutbox
1374 \quad \@depth\arraystretch\dp\strutbox
1375 \quad \@width\z@
1376 }%
1377 }%

```

\@array@array

```

1378 \def\@array@array[#1]#2{%
1379 \@tempdima \ht \strutbox
1380 \advance \@tempdima by\extrarowheight
1381 \setbox \@arstrutbox \hbox{\vrule
1382 \quad \@height \arraystretch \@tempdima
1383 \quad \@depth \arraystretch \dp \strutbox
1384 \quad \@width \z@}%
1385 \beginngroup
1386 \@mkpream{#2}%
1387 \xdef\@preamble{\noexpand \ialign \@halignto
1388 \quad \bgroup \@arstrut \@preamble
1389 \quad \tabskip \z@ \cr}%

```



```

1390 \endgroup
1391 \@arrayleft
1392 \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi \fi
1393 \bgroup
1394 \let \@sharp ##\let \protect \relax
1395 \lineskip \z@
1396 \baselineskip \z@
1397 \m@th
1398 \let\\\@arraycr \let\tabularnewline\\\let\par\@empty \@preamble
1399 }%
1400 \def\@array@array@new[#1]#2{%
1401 \@tempdima\ht\strutbox
1402 \advance\@tempdima by\extrarowheight
1403 \setbox\@arstrutbox\hbox{%
1404 \vrule \@height\arraystretch\@tempdima
1405 \@depth \arraystretch\dp\strutbox
1406 \@width \z@
1407 }%
1408 \begingroup
1409 \mkpream{#2}%
1410 \xdef\@preamble{\@preamble}%
1411 \endgroup
1412 \prepdef\@preamble{%
1413 \tabskip\@tabmid@skip
1414 \@arstrut
1415 }%
1416 \appdef\@preamble{%
1417 \tabskip\@tabright@skip
1418 \cr
1419 \array@row@pre
1420 }%
1421 \@arrayleft
1422 \@nameuse{\@array@align@#1}%
1423 \m@th
1424 \let\\\@arraycr
1425 \let\tabularnewline\%
1426 \let\par\@empty
1427 \let\@sharp##%
1428 \set@typeset@protect
1429 \lineskip\z@\baselineskip\z@
1430 \tabskip\@tableft@skip
1431 \everycr{}%
1432 \expandafter\halign\expandafter\@halignto\expandafter\bgroup\@preamble
1433 }%

```

`\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`.

```

1434 \def\endarray@LaTeX{%
1435 \crr\egroup\egroup

```

```

1436 }%
1437 \def\endarray@ltx{%
1438 \crr\array@row@pst\egroup\egroup
1439 }%
1440 \def\endarray@array{%
1441 \crr\egroup\egroup \@arrayright \gdef\@preamble{%
1442 }%
1443 \def\endarray@array@new{%
1444 \crr\array@row@pst\egroup\egroup % Same as \endarray@ltx
1445 \@arrayright
1446 \global\let\@preamble\@empty
1447 }%

```

`\endtabular`

```

1448 \def\endtabular@LaTeX{%
1449 \crr\egroup\egroup $\egroup
1450 }%
1451 \def\endtabular@ltx{%
1452 \endarray
1453 }%
1454 \def\endtabular@array{%
1455 \endarray $\egroup
1456 }%
1457 \def\endtabular@array@new{%
1458 \endarray
1459 }%

```

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

```

1460 \@namedef{endtabular*}{\endtabular}%

```

`\multicolumn`

```

1461 \long\def\multicolumn@LaTeX#1#2#3{%
1462 \multispan{#1}\begingroup
1463 \mkpream{#2}%
1464 \def\@sharp{#3}\set@typeset@protect
1465 \let\@startpbox\@startpbox\let\@endpbox\@endpbox
1466 \carstrut \preamble\hbox{ }\endgroup\ignorespaces
1467 }%
1468 \long\def\multicolumn@ltx#1#2#3{%
1469 \multispan{#1}%
1470 \begingroup
1471 \mkpream{#2}%
1472 \def\@sharp{#3}%
1473 \set@typeset@protect
1474 %\let\@startpbox\@startpbox\let\@endpbox\@endpbox
1475 \carstrut
1476 \preamble

```

```

1477 \hbox{ }%
1478 \endgroup
1479 \ignorespaces
1480 }%

```

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

We establish here the default vertical alignment.

```

1481 \def\array@align@t{\leavevmode\vtop\bgroup}%
1482 \def\array@align@b{\leavevmode\vbox\bgroup}%
1483 \def\array@align@c{\leavevmode\@ifmmode{\vcenter\bgroup}{\vcenter\bgroup\aftergroup$\aftergro
1484 \def\array@align@v{%
1485 \@ifmmode{%
1486 \@badmath
1487 \vcenter\bgroup
1488 }{%
1489 \@ifinner{%
1490 $\vcenter\bgroup\aftergroup$
1491 }{%
1492 \@@par\bgroup
1493 }%
1494 }%
1495 }%
1496 \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.

```

1497 \def\array@row@rst{%
1498 \let\array@align@v\array@align@c
1499 }%
1500 \def\array@row@pre{%
1501 \def\array@row@pst{%

```

```

\toprule Default definitions for \toprule, \colrule, \botrule
\colrule 1502 \newcommand\toprule{\tab@rule{\column@font}{\column@fil}{\frstrut}}%
\botrule 1503 \newcommand\colrule{\unskip\lrstrut\\\tab@rule{\body@font}{\frstrut}}%
1504 \newcommand\botrule{\unskip\lrstrut\\\noalign{\hline@rule}{}}%

```

`\hline`

```

1505 \def\hline@LaTeX{%
1506 \noalign{\ifnum0=‘}\fi\hrule \@height \arrayrulewidth \futurelet

```

```

1507   \reserved@a\@xhline
1508 }%
1509 \def\hline<ltx{%
1510   \noalign{%
1511     \ifnum0='}\fi
1512     \hline@rule
1513     \futurelet\reserved@a\@xhline
1514     % \noalign ended in \@xhline
1515 }%
1516 \def\@xhline@unneeded{%
1517   \say\reserved@a
1518   \ifx\reserved@a\hline
1519     \vskip\doublerulesep
1520     \vskip-\arrayrulewidth
1521   \fi
1522   \ifnum0='{}\fi}%
1523 }%
1524 \def\tab@rule#1#2#3{%
1525   \crr
1526   \noalign{%
1527     \hline@rule
1528     \gdef\@arstrut@hook{%
1529       \global\let\@arstrut@hook\@empty
1530       #3%
1531     }%
1532     \gdef\cell@font{#1}%
1533     \gdef\cell@fil{#2}%
1534   }%
1535 }%
1536 \def\column@font{}%
1537 \def\column@fil{}%
1538 \def\body@font{}%
1539 \def\cell@font{}%
1540 \def\frstrut{}%
1541 \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.

```

1542 \def\@arstrut@hline{%
1543   \relax
1544   \@ifmode{\copy}{\unhcopy}\@arstrutbox@hline
1545   \@arstrut@hook

```

```

1546 }%
1547 %
1548 \let\@arstrut@org\@arstrut
1549 \def\@arstrut@hook{%
1550 \global\let\@arstrut\@arstrut@org
1551 }%
1552 %
1553 \newbox\@arstrutbox@hline
1554 \appdef\set@arstrutbox{%
1555 \setbox\@arstrutbox@hline\hbox{%
1556 \setbox\z@\hbox{\$0^{0}_-{\$}}%
1557 \dimen@ht\z@\advance\dimen@\@arstrut@hline@clnc
1558 \@ifdim{\dimen@<\arraystretch\ht\strutbox}{\dimen@=\arraystretch\ht\strutbox}{}%
1559 \vrule \@height\dimen@
1560 \qquad \qquad \qquad \@depth\arraystretch \dp\strutbox
1561 \qquad \qquad \qquad \@width\z@
1562 }%
1563 }%
1564 %
1565 \def\hline@rule{%
1566 \hrule \@height \arrayrulewidth
1567 \global\let\@arstrut\@arstrut@hline
1568 }%
1569 \def\@arstrut@hline@clnc{2\p@}% % Klotch: magic number

\tableleft@skip
1570 \def\tableleft@skip{\z@skip}%
1571 \def\tabmid@skip{\z@skip}%\@flushglue
1572 \def\tabright@skip{\z@skip}%
1573 \def\tableleftsep{\tabcolsep}%
1574 \def\tabmidsep{\tabcolsep}%
1575 \def\tabrightsep{\tabcolsep}%
1576 \def\cell@fil{}%
1577 \def\pbox@hook{}%

\@arstrut
1578 \appdef\@arstrut{\@arstrut@hook}%
1579 \let\@arstrut@hook\@empty
1580 \def\@addtopreamble{\appdef\@preamble}%

\@mkpream
1581 \def\@mkpream@LaTeX#1{%
1582 \qquad \qquad \qquad \@firstampttrue\@lastchclass6
1583 \qquad \qquad \qquad \let\@preamble\@empty
1584 \qquad \qquad \qquad \let\protect\@unexpandable@protect
1585 \qquad \qquad \qquad \let\@sharp\relax
1586 \qquad \qquad \qquad \let\@startpbox\relax\let\@endpbox\relax
1587 \qquad \qquad \qquad \@expast{#1}%
1588 \qquad \qquad \qquad \expandafter\@tfor \expandafter

```

```

1589 \nextchar \expandafter:\expandafter=\reserved@a\do
1590 {\testpach\nextchar
1591 \ifcase \@chclass \@classz \or \@classi \or \@classii \or \@classiii
1592 \or \@classiv \or \@classv \fi\@lastchclass\@chclass}%
1593 \ifcase \@lastchclass \@acol
1594 \or \or \@preamerr \@ne\or \@preamerr \tw@\or \or \@acol \fi
1595 }%
1596 \def\@mkpream@ltx#1{%
1597 \@firstamptrue
1598 \@lastchclass6
1599 \let\@preamble\@empty
1600 \let\protect\@unexpandable@protect
1601 \let\@sharp\relax
1602 %\let\@startpbox\relax\let\@endpbox\relax
1603 \@expast{#1}%
1604 \expandafter\@tfor\expandafter\nextchar\expandafter:\expandafter=\reserved@a
1605 \do{%
1606 \expandafter\testpach\expandafter{\nextchar}%
1607 \ifcase\@chclass
1608 \@classz
1609 \or
1610 \@classi
1611 \or
1612 \@classii
1613 \or
1614 \@classiii
1615 \or
1616 \@classiv
1617 \or
1618 \@classv
1619 \fi
1620 \@lastchclass\@chclass
1621 }%
1622 \ifcase\@lastchclass
1623 \@acolr % right-hand column
1624 \or
1625 \or
1626 \@preamerr\@ne
1627 \or
1628 \@preamerr\tw@
1629 \or
1630 \or
1631 \@acolr % right-hand column
1632 \fi
1633 }%

```

\insert@column

```

1634 \def\insert@column@array{%
1635 \the@toks \the \@tempcnta
1636 \ignorespaces \@sharp \unskip

```

```

1637   \the@toks \the \count@ \relax
1638 }%
1639 \def\insert@column@array@new{%
1640   \the@toks\the\@tempcnta
1641   \array@row@rst\cell@font
1642   \ignorespaces\@sharp\unskip
1643   \the@toks\the\count@
1644   \relax
1645 }%

```

`\@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^AT_EX, 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_EX'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`.

```

1646 \def\@mkpream@relax{%
1647   \let\tableleftsep \relax
1648   \let\abmidsep \relax
1649   \let\abrightsep \relax
1650   \let\array@row@rst\relax
1651   \let\cell@font \relax
1652   \let\@startpbox \relax
1653 }%

```

`\@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.

```

1654 \def\@mkpream@array#1{%
1655   \gdef\@preamble{}\@lastchclass 4 \@firstampttrue
1656   \let\@sharp\relax \let\@startpbox\relax \let\@endpbox\relax
1657   \@temptokena{#1}\@tempwattrue
1658   \@whilesw\if@tempswa\fi{\@tempswafalse\the\NC@list}%
1659   \count@m@ne
1660   \let\the@toks\relax
1661   \prepnext@tok
1662   \expandafter \tfor \expandafter \@nextchar
1663   \expandafter :\expandafter =\the\@temptokena \do
1664   {\@testpach
1665   \ifcase \@chclass \@classz \or \@classi \or \@classii
1666     \or \save@decl \or \or \@classv \or \@classvi

```

```

1667     \or \@classvii \or \@classviii
1668     \or \@classx
1669     \or \@classx \fi
1670 \lastchclass\@chclass}%
1671 \ifcase\@lastchclass
1672 \@acol \or
1673 \or
1674 \@acol \or
1675 \@preamerr \thr@@ \or
1676 \@preamerr \tw@ \@addtopreamble\@sharp \or
1677 \or
1678 \else \@preamerr \@ne \fi
1679 \def\the@toks{\the\toks}%
1680 }%
1681 \def\@mkpream@array@new#1{%
1682 \gdef\@preamble{}%
1683 \@lastchclass\@f@ur
1684 \@firststamptrue
1685 \let\@sharp\relax
1686 \@mkpream@relax
1687 %\let\@startpbox\relax\let\@endpbox\relax
1688 \@temptokena{#1}\@tempswatrue
1689 \@whiles\if@tempswa\fi{\@tempswafalse\the\NC@list}%
1690 \count@m@ne
1691 \let\the@toks\relax
1692 \prepnext@tok
1693 \expandafter\@tfor\expandafter\@nextchar\expandafter:\expandafter=\the\@temptokena
1694 \do{%
1695 \@testpach
1696 \ifcase\@chclass
1697 \@classz
1698 \or
1699 \@classi
1700 \or
1701 \@classii
1702 \or
1703 \save@decl
1704 \or
1705 \or
1706 \@classv
1707 \or
1708 \@classvi
1709 \or
1710 \@classvii
1711 \or
1712 \@classviii
1713 \or
1714 \@classx
1715 \or
1716 \@classx

```



```

1717 \fi
1718 \@lastchclass\@chclass
1719 }%
1720 \ifcase\@lastchclass
1721 \@acolr % right-hand column
1722 \or
1723 \or
1724 \@acolr % right-hand column
1725 \or
1726 \@preamerr\thr@@
1727 \or
1728 \@preamerr\tw@\@addtopreamble\@sharp
1729 \or
1730 \or
1731 \else
1732 \@preamerr\@ne
1733 \fi
1734 \def\the@toks{\the\toks}%
1735 }%

```

`\@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.

```

1736 \appdef\@mkpream@relax{%
1737 \let\CT@setup \relax
1738 \let\CT@color \relax
1739 \let\CT@do@color \relax
1740 \let\color \relax
1741 \let\CT@column@color\relax
1742 \let\CT@row@color \relax
1743 \let\CT@cell@color \relax
1744 }%

```

`\@addamp`

```

1745 \def\@addamp@LaTeX{%
1746 \if@firstamp\@firstampfalse\else\edef\@preamble{\@preamble &}\fi
1747 }%
1748 \def\@addamp@ltx{%
1749 \if@firstamp\@firstampfalse\else\@addtopreamble{&}\fi
1750 }%

```

`\@arrayacol`

```

1751 \def\@arrayacol@LaTeX{%
1752 \edef\@preamble{\@preamble \hskip \arraycolsep}%
1753 }%
1754 \def\@arrayacol@ltx{%
1755 \@addtopreamble{\hskip\arraycolsep}%
1756 }%

```

\@tabacol

```
1757 \def\@tabacoll{%
1758 \@addtopreamble{\hskip\tableftsep\relax}%
1759 }%
1760 \def\@tabacol@LaTeX{%
1761 \edef\@preamble{\@preamble \hskip \tabcolsep}%
1762 }%
1763 \def\@tabacol@ltx{%
1764 \@addtopreamble{\hskip\tabmidsep\relax}%
1765 }%
1766 \def\@tabacolr{%
1767 \@addtopreamble{\hskip\tabrightsep\relax}%
1768 }%
```

\@arrayclassz

```
1769 \def\@arrayclassz@LaTeX{%
1770 \ifcase \@lastchclass \@acolampacol \or \@ampacol \or
1771 \or \or \@addamp \or
1772 \@acolampacol \or \@firstampfalse \@acol \fi
1773 \edef\@preamble{\@preamble
1774 \ifcase \@chnum
1775 \hfil$\relax\@sharp$\hfil \or $\relax\@sharp$\hfil
1776 \or \hfil$\relax\@sharp$\fi}%
1777 }%
1778 \def\@arrayclassz@ltx{%
1779 \ifcase\@lastchclass
1780 \@acolampacol
1781 \or
1782 \@ampacol
1783 \or
1784 \or
1785 \or
1786 \@addamp
1787 \or
1788 \@acolampacol
1789 \or
1790 \@firstampfalse\@acoll
1791 \fi
1792 \ifcase\@chnum
1793 \@addtopreamble{%
1794 \hfil\array@row@rst$\relax\@sharp$\hfil
1795 }%
1796 \or
1797 \@addtopreamble{%
1798 \array@row@rst$\relax\@sharp$\hfil
1799 }%
1800 \or
1801 \@addtopreamble{%
1802 \hfil\array@row@rst$\relax\@sharp$%
1803 }%
```

```

1804 \fi
1805 }%

\@tabclassz
1806 \def\@tabclassz@LaTeX{%
1807 \ifcase\@lastchclass
1808 \acolampacol
1809 \or
1810 \ampacol
1811 \or
1812 \or
1813 \or
1814 \addamp
1815 \or
1816 \acolampacol
1817 \or
1818 \@firstampfalse\acol
1819 \fi
1820 \edef\@preamble{%
1821 \@preamble{%
1822 \ifcase\@chnum
1823 \hfil\ignorespaces\@sharp\unskip\hfil
1824 \or
1825 \hskip1sp\ignorespaces\@sharp\unskip\hfil
1826 \or
1827 \hfil\hskip1sp\ignorespaces\@sharp\unskip
1828 \fi}}%
1829 }%
1830 \def\@tabclassz@ltx{%
1831 \ifcase\@lastchclass
1832 \acolampacol
1833 \or
1834 \ampacol
1835 \or
1836 \or
1837 \or
1838 \addamp
1839 \or
1840 \acolampacol
1841 \or
1842 \@firstampfalse\acoll
1843 \fi
1844 \ifcase\@chnum
1845 \@addtopreamble{%
1846 {\hfil\array@row@rst\cell@font\ignorespaces\@sharp\unskip\hfil}%
1847 }%
1848 \or
1849 \@addtopreamble{%
1850 {\cell@fil\hskip1sp\array@row@rst\cell@font\ignorespaces\@sharp\unskip\hfil}%
1851 }%

```

```

1852 \or
1853 \@addtopreamble{%
1854   {\hfil\hskip1sp\array@row@rst\cell@font\ignorespaces\@sharp\unskip\cell@fil}%
1855 }%
1856 \fi
1857 }%

\@tabclassiv
1858 \def\@tabclassiv@LaTeX{%
1859 \@addtopreamble\@nextchar
1860 }%
1861 \def\@tabclassiv@ltx{%
1862 \expandafter\@addtopreamble\expandafter{\@nextchar}%
1863 }%

\@arrayclassiv
1864 \def\@arrayclassiv@LaTeX{%
1865 \@addtopreamble{\@nextchar}%
1866 }%
1867 \def\@arrayclassiv@ltx{%
1868 \expandafter\@addtopreamble\expandafter{\expandafter\@nextchar}%
1869 }%

\@classv
1870 \def\@classv@LaTeX{%
1871 \@addtopreamble{\@startpbox{\@nextchar}\ignorespaces
1872 \@sharp\@endpbox}%
1873 }%
1874 \def\@classv@ltx{%
1875 \expandafter\@addtopreamble
1876 \expandafter{%
1877 \expandafter \@startpbox
1878 \expandafter {\@nextchar}%
1879 \pbox@hook\array@row@rst\cell@font\ignorespaces\@sharp\@endpbox
1880 }%
1881 }%

\@classx
1882 \def\@classx@array{%
1883 \ifcase \@lastchclass
1884 \@acolampacol \or
1885 \@addamp \@acol \or
1886 \@acolampacol \or
1887 \or
1888 \@acol \@firstampfalse \or
1889 \@addamp
1890 \fi
1891 }%
1892 \def\@classx@array@new{%

```

```

1893 \ifcase \@lastchclass
1894 \@acolampacol
1895 \or
1896 \@addamp \@acol
1897 \or
1898 \@acolampacol
1899 \or
1900 \or
1901 \@firstampfalse\@acoll
1902 \or
1903 \@addamp
1904 \fi
1905 }%

```

6.15 Repair other broken parts of L^AT_EX

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

```

1906 \def\@xbitor@LaTeX #1{\@tempcntb \count#1
1907     \ifnum \@tempcnta =\z@
1908     \else
1909         \divide\@tempcntb\@tempcnta
1910         \ifodd\@tempcntb \@testtrue\fi
1911     \fi}%
1912 \def\@xbitor@ltx#1{%
1913     \@tempcntb\count#1\relax
1914     \@ifnum{\@tempcnta=\z@}{-}{%
1915         \divide\@tempcntb\@tempcnta
1916         \ifodd\@tempcntb{\@testtrue}{-}}%
1917 }%
1918 }%
1919 \@ifx{\@xbitor\@xbitor@LaTeX}{%
1920     \class@info{Repairing broken LaTeX \string\@xbitor}%
1921 }{%
1922     \class@info{Unrecognized LaTeX \string\@xbitor. Please update this document class! (Proceedin
1923 }%
1924 \let\@xbitor\@xbitor@ltx

```

6.16 Syntax

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

```

1925 \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.

```

1926 \def\starttoc#1{%
1927   \begingroup
1928   \toc@pre
1929   \makeatletter
1930   \@input{\jobname.#1}%
1931   \if@filesw
1932     \expandafter\newwrite\csname tf@#1\endcsname
1933     \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1934   \fi
1935   \@nobreakfalse
1936   \toc@post
1937 \endgroup
1938 }%
1939 \def\toc@pre{}%
1940 \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.

```

1941 \def\toc@@font{}%
1942 \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.

```

1943 \let\tocdim@section      \leftmargini
1944 \let\tocdim@subsection   \leftmarginii
1945 \let\tocdim@subsubsection \leftmarginiii
```

```

1946 \let\tocdim@paragraph \leftmarginiv
1947 \let\tocdim@appendix \leftmarginv
1948 \let\tocdim@pagenum \leftmarginvi

```

`\toc@pre@auto` We patch `\@starttoc` to: 1) before TOC processing, initialize the max registers and set the needed dimensions from the values stored in the auxiliary file, and 2) `\toc@post@auto` 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 L^AT_EX processing, the substyle does:

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

```

1949 \def\toc@pre@auto{%
1950 \toc@@font
1951 \@tempdima\z@
1952 \toc@setindent\@tempdima{section}%
1953 \toc@setindent\@tempdima{subsection}%
1954 \toc@setindent\@tempdima{subsubsection}%
1955 \toc@setindent\@tempdima{paragraph}%
1956 \toc@letdimen{appendix}%
1957 \toc@letdimen{pagenum}%
1958 }%
1959 \def\toc@post@auto{%
1960 \if@filesw
1961 \begingroup
1962 \toc@writedimen{section}%
1963 \toc@writedimen{subsection}%
1964 \toc@writedimen{subsubsection}%
1965 \toc@writedimen{paragraph}%
1966 \toc@writedimen{appendix}%
1967 \toc@writedimen{pagenum}%
1968 \endgroup
1969 \fi
1970 }%

```

`\toc@setindent`

```

1971 \def\toc@setindent#1#2{%
1972 \csname tocdim@#2\endcsname\tocdim@min\relax
1973 \@ifundefined{tocmax@#2}{\@namedef{tocmax@#2}{\z@}}{}%
1974 \advance#1\@nameuse{tocmax@#2}\relax
1975 \expandafter\edef\csname tocleft@#2\endcsname{\the#1}%
1976 }%

```

`\toc@letdimen`

```

1977 \def\toc@letdimen#1{%
1978 \csname tocdim@#1\endcsname\tocdim@min\relax
1979 \@ifundefined{tocmax@#1}{\@namedef{tocmax@#1}{\z@}}{}%
1980 \expandafter\let\csname tocleft@#1\expandafter\endcsname\csname tocmax@#1\endcsname
1981 }%

```

`\toc@writedimen`

```
1982 \def\toc@writedimen#1{%
1983 \immediate\write\@auxout{%
1984 \gdef\expandafter\string\csname tocmax@#1\endcsname{%
1985 \expandafter\the\csname tocdim@#1\endcsname
1986 }%
1987 }%
1988 }%
```

`\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

```
1989 \def\l@sections#1#2#3#4{%
1990 \begingroup
1991 \everypar{}%
1992 \set@tocdim@pagenum\@tempboxa{#4}%
1993 \global\@tempdima\csname tocdim@#2\endcsname
1994 \leftskip\csname tocleft@#2\endcsname\relax
1995 \dimen@\csname tocleft@#1\endcsname\relax
1996 \parindent-\leftskip\advance\parindent\dimen@
1997 \rightskip\tocleft@pagenum plus 1fil\relax
1998 \skip@\parfillskip\parfillskip\z@
1999 \let\numberline\numberline@sections
2000 \@nameuse{1@f@#2}%
2001 \ignorespaces#3\unskip\nobreak\hskip\skip@
2002 \hb@xt@\rightskip{\hfil\unhbox\@tempboxa}\hskip-\rightskip\hskip\z@skip
2003 \par
2004 \expandafter\aftergroup\csname tocdim@#2\endcsname\expandafter
2005 \endgroup\the\@tempdima\relax
2006 }%
```

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

```
2007 \def\set@tocdim@pagenum#1#2{%
2008 \setbox#1\hbox{\ignorespaces#2}%
2009 \@ifdim{\tocdim@pagenum<\wd#1}{\global\tocdim@pagenum\wd#1}{}%
2010 }%
```


`\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!

```

2011 \def\numberline@@sections#1{%
2012 \leavevmode\hb@xt@-\parindent{%
2013 \hfil
2014 \@ifempty{#1}{}{%
2015 \setbox\z@\hbox{#1.\kern\ltxu@dotsep}%
2016 \@ifdim{\@tempdima<\wd\z@}{\global\@tempdima\wd\z@}{}%
2017 \unhbox\z@
2018 }%
2019 }%
2020 \ignorespaces
2021 }%
2022 \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^AT_EX kernel procedure `\list`, which is the implementation of all lists.

```

2023 \def\list#1#2{%
2024 \ifnum \@listdepth >5\relax
2025 \toodeep
2026 \else
2027 \global\advance\@listdepth\@ne
2028 \fi
2029 \rightmargin\z@
2030 \listparindent\z@
2031 \itemindent\z@
2032 \csname @list\romannumeral\the\@listdepth\endcsname
2033 \def\@itemlabel{#1}%
2034 \let\makelabel\@mklab
2035 \@nmblistfalse
2036 #2\relax
2037 \@trivlist
2038 \parskip\parsep
2039 \set@listindent
2040 \ignorespaces
2041 }%
2042 \def\set@listindent@parshape{%
2043 \parindent\listparindent

```

```

2044 \advance\@totalleftmargin\leftmargin
2045 \advance\linewidth-\rightmargin
2046 \advance\linewidth-\leftmargin
2047 \parshape\@ne\@totalleftmargin\linewidth
2048 }%
2049 \def\set@listindent@{%
2050 \parindent\listparindent
2051 \advance\@totalleftmargin\leftmargin
2052 \advance\rightskip\rightmargin
2053 \advance\leftskip\@totalleftmargin
2054 }%
2055 \let\set@listindent\set@listindent@parshape

```

6.19 Hypertext capabilities

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

```

\URL@prefix 2056 \providecommand\href[0]{\begingroup\@sanitize\@href}%
2057 \def\@href#1{\@startlink{#1}\@@href}%
2058 \def\@@href#1{\endgroup#1\@@endlink}%
2059 \providecommand \url [0]{\begingroup\@sanitize \@url }%
2060 \def \@url #1{\endgroup\@href {#1}{\URL@prefix#1}}%
2061 \providecommand \URL@prefix [0]{URL }%

```

`\@startlink` How we define `\@startlink` and `\@endlink` will depend on whether we are
`\@endlink` running under PDFLATEX. If so, and if PDF output is requested, then we
`\pdfstartlink@attr` 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.

```

2062 \@ifundefined \pdfoutput {\true@sw}{\@ifnum{z@=\pdfoutput}{\true@sw}{\false@sw}}%
2063 {%
2064 \def\@startlink#1{\leavevmode\special{html:<a href="#1">}}%
2065 \def\@endlink{\special{html:</a>}}%
2066 }%
2067 \def\@startlink#1{%
2068 \leavevmode
2069 \pdfstartlink\pdfstartlink@attr
2070 user{/Subtype/Link/A<</Type/Action/S/URI/URI(#1)>>}%
2071 \relax
2072 }%
2073 \def\@endlink{\pdfendlink}%
2074 \def\pdfstartlink@attr{attr{/Border[0 0 1 ]/H/I/C[0 1 1]}}%
2075 }%
2076 \def\typeout@org#1{%
2077 \begingroup
2078 \set@display@protect
2079 \immediate\write\@unused{#1}%
2080 \endgroup

```

```
2081 }%
2082 \long\def\typeout@ltx#1{%
2083 \begingroup
2084 \set@display@protect
2085 \immediate\write\@unused{#1}%
2086 \endgroup
2087 }%
2088 \@ifx{\typeout\typeout@org}{%
2089 \class@info{Making \string\typeout\space \string\long}%
2090 \let\typeout\typeout@ltx
2091 }{}}%
```

6.20 End of the kernel DOCSTRIP module

Here ends the module.

```
2092 %</kernel>
```

Index

Symbols	
<code>\$TEXMF/</code>	3
<code>\%</code>	840
<code>\(</code>	605
<code>\)</code>	606
<code>.aux</code>	38
<code>.cls</code>	18
<code>.dtx</code>	5
<code>.rtx</code>	17
<code>\@</code>	409, 417
<code>\@array</code>	1169, 1235, 1238
<code>\@end</code>	336, 397
<code>\@endlink</code>	66
<code>\@endlink</code>	2058, 2062
<code>\@endpbox</code>	1336, 1359, 1465, 1474
<code>\@href</code>	2057, 2058
<code>\@par</code>	735, 911, 951, 1342, 1366, 1492
<code>\@startlink</code>	66
<code>\@startlink</code>	2057, 2062
<code>\@startpbox</code>	1336, 1358, 1465, 1474
<code>\@Esphack</code>	828, 830
<code>\@M</code>	45
<code>\@MM</code>	25
<code>\@acol</code>	55
<code>\@acol</code>	1197, 1206, 1223, 1292, 1303, 1319, 1593, 1594, 1672, 1674, 1772, 1818, 1885, 1888, 1896
<code>\@acolampacol</code> 1770, 1772, 1780, 1788, 1808, 1816, 1832, 1840, 1884, 1886, 1894, 1898
<code>\@acoll</code> 1204, 1221, 1301, 1317, 1790, 1842, 1901
<code>\@acolr</code>	55
<code>\@acolr</code> 1205, 1222, 1302, 1318, 1623, 1631, 1721, 1724
<code>\@addamp</code> 1004, 1094, 1745 , 1771, 1786, 1814, 1838, 1885, 1889, 1896, 1903
<code>\@addamp@LaTeX</code>	1004, 1745
<code>\@addamp@ltx</code>	1094, 1748
<code>\@addtopreamble</code> 1580, 1676, 1728, 1749, 1755, 1758, 1764, 1767, 1793, 1797, 1801, 1845, 1849, 1853, 1859, 1862, 1865, 1868, 1871, 1875
<code>\@afterindentfalse</code>	872
<code>\@afterindenttrue</code>	870
<code>\@ampacol</code>	1770, 1782, 1810, 1834
<code>\@argarraycr</code>	1264, 1268
<code>\@argswap</code>	13
<code>\@argswap</code>	275
<code>\@argswap@val</code>	13
<code>\@argswap@val</code>	275 , 505
<code>\@argtabularcr</code>	1246, 1247
<code>\@array</code>	47
<code>\@array</code> 996, 1086, 1111, 1168, 1169, 1229, 1232, 1325
<code>\@array@LaTeX</code>	996, 1325
<code>\@array@align@</code>	1481
<code>\@array@align@b</code>	1482
<code>\@array@align@c</code>	51
<code>\@array@align@c</code>	1483, 1498
<code>\@array@align@t</code>	1481
<code>\@array@align@v</code>	51
<code>\@array@align@v</code>	1484, 1498
<code>\@array@array</code>	1111, 1378
<code>\@array@array@new</code>	1168, 1400
<code>\@array@ltx</code>	1086, 1345
<code>\@array@sw</code>	43
<code>\@array@sw</code> 995, 1110, 1192
<code>\@array@sw@LaTeX</code>	1110, 1192
<code>\@array@sw@array</code>	995, 1193
<code>\@arrayacol</code> 1005, 1095, 1177, 1292, 1301–1303, 1317–1319, 1751
<code>\@arrayacol@LaTeX</code>	1005, 1751
<code>\@arrayacol@ltx</code>	1095, 1177, 1754
<code>\@arrayclassiv</code>	1009, 1099, 1294, 1305, 1864
<code>\@arrayclassiv@LaTeX</code>	1009, 1864
<code>\@arrayclassiv@ltx</code>	1099, 1867

<code>\@arrayclassz</code>	. 1007, 1097, 1293, 1304, 1769	<code>\@centering</code>	. 435, 438, 441, 460, 466, 479
<code>\@arrayclassz@LaTeX</code>	1007, 1769	<code>\@chclass</code>	1591, 1592, 1607, 1620, 1665, 1670, 1696, 1718
<code>\@arrayclassz@ltx</code>	.. 1097, 1778	<code>\@checkend</code> 332
<code>\@arraycr</code> 45	<code>\@chnum</code>	.. 1774, 1792, 1822, 1844
<code>\@arraycr</code>	1120, 1181, 1240 , 1295, 1306, 1398, 1424	<code>\@classi</code>	. 1591, 1610, 1665, 1699
<code>\@arraycr@array</code> 1120, 1252	<code>\@classii</code>	1591, 1612, 1665, 1701
<code>\@arraycr@new</code> 1181, 1257	<code>\@classiii</code> 1591, 1614
<code>\@arrayleft</code> 1391, 1421	<code>\@classiv</code>	1199, 1208, 1294, 1305, 1592, 1616
<code>\@arrayright</code> 1441, 1445	<code>\@classv</code>	. 1011, 1101, 1592, 1618, 1666, 1706, 1870
<code>\@arstrut</code> 52	<code>\@classv@LaTeX</code> 1011, 1870
<code>\@arstrut</code>	1335, 1351, 1388, 1414, 1466, 1475, 1548, 1550, 1567, 1578	<code>\@classv@ltx</code> 1101, 1874
<code>\@arstrut@hline</code> 52	<code>\@classvi</code> 1666, 1708
<code>\@arstrut@hline</code> 1542	<code>\@classvii</code> 1667, 1710
<code>\@arstrut@hline@clnc</code> 52	<code>\@classviii</code> 1667, 1712
<code>\@arstrut@hline@clnc</code>	1557, 1569	<code>\@classx</code>	. 1118, 1176, 1668, 1669, 1714, 1716, 1882
<code>\@arstrut@hook</code> 52	<code>\@classx@array</code> 1118, 1882
<code>\@arstrut@hook</code>	1528, 1529, 1542 , 1578, 1579	<code>\@classx@array@new</code>	. 1176, 1892
<code>\@arstrut@org</code> 1542	<code>\@classz</code>	. 1198, 1207, 1293, 1304, 1591, 1608, 1665, 1697
<code>\@arstrutbox</code> 47 , 52	<code>\@clsextension</code> 17 , 18
<code>\@arstrutbox</code>	<code>\@clsextension</code>	... 391, 408, 416
.. 1248, 1249, 1272, 1278, 1328, 1372, 1381, 1403		<code>\@ctrerr</code> 864
<code>\@arstrutbox@hline</code> 52	<code>\@currbox</code> 32 , 33
<code>\@arstrutbox@hline</code> 1542	<code>\@currbox</code>	780, 783, 785, 786, 792, 793, 802
<code>\@auxout</code> 610, 619, 1983	<code>\@currentHref</code> 546, 977
<code>\@boole@def</code> 13	<code>\@currentlabel</code>	431, 447, 573, 620
<code>\@boole@def</code> 280 , 292–295, 297–307	<code>\@currentlabelname</code> 37
<code>\@boolean</code> 13	<code>\@currentx</code> 17
<code>\@boolean</code> 280	<code>\@currentx</code>	373, 390, 391, 407, 415
<code>\@booleanfalse</code> 14	<code>\@currname</code> 17
<code>\@booleanfalse</code>	... 290 , 743, 788	<code>\@currname</code>	... 373, 389, 406, 414
<code>\@booleantrue</code> 14	<code>\@currnamestack@ltx</code>	. 401, 405, 410, 418, 420
<code>\@booleantrue</code>	... 290 , 673, 786	<code>\@dblarg</code> 880, 881
<code>\@bsphack</code> 618, 833	<code>\@dblfloat</code> 724, 726, 748
<code>\@caption</code> 28	<code>\@dblfloat@LaTeX</code> 724, 748
<code>\@capytype</code> 32	<code>\@defaultsubs</code> 356
<code>\@capytype</code> 689	<code>\@depth</code>	.. 1248, 1249, 1273, 1279, 1330, 1374, 1383, 1405, 1560
<code>\@centercr</code> 18 , 19	<code>\@dofilelist</code> 349
<code>\@centercr</code> 428		

<code>\@eha</code>	180	<code>\@ifdim</code> ..	292 , 350, 805, 904, 944, 1558, 2009, 2016
<code>\@enddocumenthook</code>	331	<code>\@ifempty</code>	14
<code>\@endpbox</code>	55	<code>\@ifeof</code>	292
<code>\@endpbox</code> 1336, 1359, 1465, 1474, 1586, 1602, 1656, 1687, 1872, 1879		<code>\@ifhbox</code>	292 , 801, 820
<code>\@eqcnt</code> ..	434, 439, 440, 442, 448, 461, 464, 467	<code>\@ifhmode</code>	292
<code>\@eqncr</code>	45	<code>\@ifinner</code>	292 , 652, 1489
<code>\@eqncr</code>	436, 450	<code>\@ifl@ter</code>	393
<code>\@eqnsel</code>	438, 460	<code>\@ifmmode</code> .	292 , 1300, 1316, 1483, 1485, 1544
<code>\@eqnswtrue</code>	432, 448	<code>\@ifnextchar</code>	488, 497, 512, 1229, 1232, 1235, 1238, 1246, 1247, 1263, 1267
<code>\@eqpen</code>	45	<code>\@ifnotempty</code>	14
<code>\@esphack</code>	622	<code>\@ifnotrelax</code>	13
<code>\@expast</code>	1587, 1603	<code>\@ifnotrelax</code> .	275 , 625, 800, 819
<code>\@extension</code>	17	<code>\@ifnum</code> ..	292 , 785, 886, 915, 929, 1914, 2062
<code>\@finalstrut</code>	578	<code>\@ifodd</code>	292 , 1916
<code>\@firstampfalse</code>		<code>\@ifpackageloaded</code>	199, 537, 985, 1109
..	1746, 1749, 1772, 1790, 1818, 1842, 1888, 1901	<code>\@ifstar</code> ..	879, 1244, 1245, 1255, 1260
<code>\@firstamptrue</code>	1582, 1597, 1655, 1684	<code>\@ifundefined</code>	895, 908, 913, 924, 927, 948, 953, 958, 961, 1973, 1979
<code>\@float</code>	26 , 31 , 33	<code>\@ifvbox</code>	292
<code>\@float@LaTeX</code>	722, 723, 747	<code>\@ifvmode</code>	292 , 482
<code>\@floatboxreset</code>	695	<code>\@ifvoid</code>	292 , 649, 660
<code>\@flushglue</code>	426, 1571	<code>\@ifx</code>	13 , 14
<code>\@font@warning</code>	351	<code>\@ifx</code>	275, 276, 279, 292 , 313, 359, 471, 550, 847, 996–1016, 1111–1123, 1192, 1193, 1919, 2088
<code>\@footnotemark</code>	24	<code>\@ifx@empty</code>	292
<code>\@footnotetext</code>	21 , 24	<code>\@ifxundefined</code>	13
<code>\@footnotetext</code> 518, 538, 539, 554		<code>\@ifxundefined</code> ...	247, 252, 257, 275 , 722, 2062
<code>\@gobble@opt@one</code>	61	<code>\@ifxundefined@cs</code>	279, 730, 741, 761
<code>\@gobble@opt@one</code>	1925	<code>\@iiparbox</code>	671
<code>\@halignto</code> 1296, 1307, 1312, 1322, 1334, 1368, 1387, 1432		<code>\@itemlabel</code>	2033
<code>\@hang@from</code>	908, 939	<code>\@lastchclass</code>	
<code>\@hang@froms</code>	948, 967	..	1582, 1592, 1593, 1598, 1620, 1622, 1655, 1670, 1671, 1683, 1718, 1720, 1770, 1779,
<code>\@hangfrom</code>	35–37		
<code>\@hangfrom</code>	939		
<code>\@hangfrom@section</code>	36		
<code>\@height</code> .	1248, 1249, 1329, 1373, 1382, 1404, 1506, 1559, 1566		
<code>\@href</code>	2056, 2057, 2060		
<code>\@ialph</code>	859, 860		
<code>\@if@empty</code>	292 , 2014		
<code>\@if@sw</code> ..	295, 296, 341, 358, 360		
<code>\@ifcat</code>	292		

1807, 1831, 1883, 1893	\@onefilewithoptions	17
\@latex@warning@no@line . . .	\@outputdblcol	9
\@listdepth	\@p@pfilename	404
\@mainaux	\@p@pfilename@ltx	401, 413
\@mainaux	\@parboxrestore	572, 694
\@makecol	\@pass@ptions	379
\@makefnmark	\@preamble 1333, 1335, 1343, 1349,	
\@makefnmark	1353, 1368, 1387, 1388, 1398,	
\@makefntext	1410, 1412, 1416, 1432, 1441,	
\@makeoother	1446, 1466, 1476, 1580, 1583,	
\@minipagefalse	1599, 1655, 1682, 1746, 1752,	
\@mklab	1761, 1773, 1820, 1821	
\@mkpream	\@preamerr 1342, 1366, 1594, 1626,	
\@mkpream 1003, 1093, 1117, 1175,	1628, 1675, 1676, 1678, 1726,	
1332, 1348, 1386, 1409, 1463,	1728, 1732	
1471, <u>1581</u> , <u>1654</u>	\@provide	312, 987, 989, 992
\@mkpream@LaTeX	\@pushfilename	404
\@mkpream@array	\@pushfilename@ltx	384, 404
\@mkpream@array@new 1175, 1681	\@refundefined	357
\@mkpream@ltx	\@reset@ptions	18
\@mkpream@relax	\@reset@ptions	387, 402
\@mkpream@relax <u>1646</u> , 1686, <u>1736</u>	\@resetactivechars	239
\@mpargs	\@rightskip	425
\@mpfn	\@runin@to	36
\@mpfn	\@runin@to	924, 940
491, 495, 504, 517, 523,	\@runin@tos	958, 968
527, 530, 533, 682	\@sanitize	2056, 2059
\@mpfootins	\@secntformat	35, 36
\@mpfootins	\@secntformat	896
638, 641, 642, 649,	\@secpenalty	877
650, 653, 656, 660, 661	\@sect	35
\@mpfootnotetext	\@sect	884
\@mpfootnotetext	\@sect@ltx	881, 885
\@mpfootnotetext	\@setckpt	343
\@mpmakefntext	\@sharp	1339, 1362, 1394, 1427,
\@multiplelabels	1464, 1472, 1585, 1601, 1636,	
\@namedef	1642, 1656, 1676, 1685, 1728,	
\@namedef	1775, 1776, 1794, 1798, 1802,	
\@nameuse	1823, 1825, 1827, 1846, 1850,	
196, 677, 1346, 1422,	1854, 1872, 1879	
1974, 2000	\@ssect	35, 37
\@ne	\@ssect	941
\@newl@bel	\@ssect@ltx	880, 941
\@nextchar 1589, 1590, 1604, 1606,	\@startcolumn	9
1662, 1693, 1859, 1862, 1865,	\@startpbox	55
1868, 1871, 1878		
\@nmbriestfalse		
\@nobreakfalse		
\@normalcr		
\@normalcr		
\@nul		

<code>\@startpbox</code>	<code>\@tabularcr@LaTeX</code> .. 1013, 1244
.. 1336, 1358, 1465, 1474,	<code>\@tabularcr@ltx</code> 1103, 1245
1586, 1602, 1652, 1656, 1687,	<code>\@tbpen</code>
1871, 1877	45
<code>\@startsection</code>	<code>\@tbpen</code>
35, 37	1240
<code>\@startsection</code>	<code>\@temp@sw</code>
865	786, 788, 792
<code>\@startsection@hook</code> .. 866, 883	<code>\@tempa</code>
<code>\@starttoc</code>	200, 202, 798, 800,
63	801, 806, 807, 809, 810, 817,
<code>\@starttoc</code>	819–822, 846, 847
1926	<code>\@tempb</code>
<code>\@svsec</code>	837, 847
36	<code>\@tempboxa</code>
<code>\@svsec</code> .. 888, 894, 896, 899, 909,	64
925	<code>\@tempc</code>
<code>\@svsechd</code>	835, 853
922, 956	<code>\@tempcnta</code>
<code>\@tabacol</code> 1006, 1096, 1178, 1197,	33
1206, 1223, 1757	<code>\@tempcntb</code> 822, 1906, 1909, 1910,
<code>\@tabacol@LaTeX</code> 1006, 1760	1913, 1915, 1916
<code>\@tabacol@ltx</code> .. 1096, 1178, 1763	<code>\@tempdima</code>
<code>\@tabacoll</code> 1204, 1221, 1757	65
<code>\@tabacolr</code> 1205, 1222, 1766	<code>\@tempskipa</code> . 869, 871, 872, 877,
<code>\@tabarray</code>	903, 904, 943, 944
44	<code>\@temptokena</code> . 1657, 1663, 1688,
<code>\@tabarray</code> 999, 1089, 1113, 1171,	1693
1201, 1210, 1218, 1226, 1228,	<code>\@testdef</code>
1297, 1308, 1313, 1323	344
<code>\@tabarray@LaTeX</code> 999, 1228	<code>\@testpach</code> 1590, 1606, 1664, 1695
<code>\@tabarray@array</code> ... 1113, 1234	<code>\@testtrue</code>
<code>\@tabarray@array@new</code> 1171, 1237	1910, 1916
<code>\@tabarray@ltx</code>	<code>\@textsuperscript</code>
1089, 1231	556
<code>\@tabclassiv</code> . 1008, 1098, 1199,	<code>\@tfor</code> ... 1588, 1604, 1662, 1693
1208, 1858	<code>\@thefnmark</code> .. 524, 531, 557, 574
<code>\@tabclassiv@LaTeX</code> . 1008, 1858	<code>\@toodeep</code>
<code>\@tabclassiv@ltx</code> ... 1098, 1861	2025
<code>\@tabclassz</code> .. 1010, 1100, 1198,	<code>\@totalleftmargin</code> . 2044, 2047,
1207, 1806	2051, 2053
<code>\@tabclassz@LaTeX</code> .. 1010, 1806	<code>\@tpfootnotetext</code>
<code>\@tabclassz@ltx</code>	554
1100, 1830	<code>\@trivlist</code>
<code>\@tabular</code>	2037
43	<code>\@twopowerfourteen</code>
<code>\@tabular</code> . 991, 998, 1088, 1112,	32
1170, 1194	<code>\@twopowerfourteen</code> ... 776, 804
<code>\@tabular@LaTeX</code>	<code>\@twopowertwo</code>
998, 1194	32
<code>\@tabular@array</code> 1112, 1212	<code>\@twopowertwo</code> 777, 821, 822
<code>\@tabular@array@new</code> 1170, 1220	<code>\@undefined</code>
<code>\@tabular@ltx</code>	43
1088, 1203	<code>\@undefined</code>
<code>\@tabularcr</code>	16, 400
45	<code>\@unexpandable@protect</code> .. 1584,
<code>\@tabularcr</code> .. 1013, 1103, 1192,	1600
1200, 1209, 1240	<code>\@unused</code>
<code>\@tabularcr@LaTeX</code>	2079, 2085
45	<code>\@unusedoptionlist</code> ... 379, 380
	<code>\@url</code>
	2059, 2060
	<code>\@whilesw</code>
	1658, 1689
	<code>\@width</code> .. 1248, 1249, 1273, 1279,
	1331, 1375, 1384, 1406, 1561
	<code>\@write@floatline</code>
	34
	<code>\@write@floatline</code>
	843

<code>\@writefile</code>	610	<code>\^</code>	839, 852
<code>\@x@sf</code>	502, 509	<code>_</code>	36
<code>\@xargarraycr</code>	45	<code>00readme</code>	3, 5
<code>\@xargarraycr</code>	1015, 1105, 1122, 1183, 1240	<code>_</code>	25, 31, 51, 54, 55, 59, 76, 78, 84, 89, 96, 200
<code>\@xargarraycr@LaTeX</code>	1015, 1248	A	
<code>\@xargarraycr@array</code>	1122, 1270	<code>\abovedisplayshortskip</code>	... 456
<code>\@xargarraycr@ltx</code>	1105, 1249	<code>\abovedisplayskip</code>	... 451–456
<code>\@xargarraycr@new</code>	1183, 1276	<code>\active</code>	839
<code>\@xarraycr</code>	45	<code>\addcontentsline</code>	37
<code>\@xarraycr</code>	1121, 1182, 1240	<code>\addcontentsline</code>	612, 914, 928, 954, 962
<code>\@xarraycr@array</code>	1121, 1262	<code>\addtocontents</code>	600, 613
<code>\@xarraycr@new</code>	1182, 1266	<code>\aftergroup</code>	51
<code>\@xbitor</code>	1906	<code>\aftergroup</code>	1483, 1490, 2004
<code>\@xbitor@LaTeX</code>	1906, 1919	<code>\appdef</code>	13, 15, 27
<code>\@xbitor@ltx</code>	1912, 1924	<code>\appdef</code>	239, 246, 322, 535, 630, 634, 980, 984, 1243, 1353, 1416, 1554, 1578, 1580, 1736
<code>\@xfloat</code>	674	<code>\appdef@e</code>	264
<code>\@xfloat@LaTeX</code>	674, 678	<code>\appdef@eval</code>	269
<code>\@xfloat@anchored</code>	678, 688	<code>\appdef@val</code>	261, 270
<code>\@xfloat@prep</code>	676, 680	<code>aps.rtx</code>	17
<code>\@xfootnotemark</code>	21	argument	
<code>\@xfootnotenext</code>	24	class	17
<code>\@xhline</code>	1507, 1513, 1514	float	34
<code>\@xhline@unneeded</code>	1516	text	22, 39
<code>\@xifempty</code>	14	<code>your_tokens_here</code>	15
<code>\@xssect</code>	937, 965	argument, optional	
<code>\@xtabularcr</code>	45	number	22
<code>\@xtabularcr</code>	1014, 1104, 1240	<code>\array</code>	46
<code>\@xtabularcr@LaTeX</code>	1014, 1246	<code>\array</code>	1000, 1090, 1114, 1172, 1291
<code>\@xtabularcr@ltx</code>	1104, 1247	array document class	10, 39, 43, 45, 46, 55, 57
<code>\@xvi</code>	775, 785, 786	array environment	2, 39, 44
<code>\@xxxii</code>	781, 782, 784	<code>\array@array</code>	1114, 1310
<code>\@yargarraycr</code>	45	<code>\array@array@new</code>	1172, 1315
<code>\@yargarraycr</code>	1016, 1106, 1123, 1184, 1240	<code>\array@default</code>	746, 1232, 1238, 1481
<code>\@yargarraycr@LaTeX</code>	1016, 1250	<code>\array@hook</code>	990
<code>\@yargarraycr@array</code>	1123, 1283	<code>\array@LaTeX</code>	1000, 1291
<code>\@yargarraycr@ltx</code>	1106, 1251		
<code>\@yargarraycr@new</code>	1184, 1287		
<code>\[</code>	21		
<code>\[</code>	844		
<code>\]</code>	18, 19, 45		
<code>\{</code>	117, 840, 844		
<code>\}</code>	117, 840, 844		
<code>\]</code>	21		
<code>\]</code>	844		

<code>\array@ltx</code>	1090, 1299	<code>\catcode</code>	409, 417, 839, 844
<code>\array@message</code>	1109, 1186	<code>\cell@fil</code>	1533, 1576, 1850, 1854
<code>\array@row@pre</code>	1356, 1419, 1497	<code>\cell@font</code>	1532, 1539, 1641, 1651, 1846, 1850, 1854, 1879
<code>\array@row@pst</code>	1438, 1444, 1497	<code>\centering</code>	18
<code>\array@row@rst</code>	51	<code>\changes</code>	121–161
<code>\array@row@rst</code> 1497 , 1641, 1650, 1794, 1798, 1802, 1846, 1850, 1854, 1879	<code>\chardef</code>	217–219, 775, 835
<code>\arraycolsep</code>	439, 440, 1311, 1752, 1755	<code>\check@aux</code>	330
<code>\arrayrulewidth</code>	1506, 1520, 1566	<code>\langle class \rangle</code> placeholder	17
<code>\arraystretch</code> 1329, 1330, 1373, 1374, 1382, 1383, 1404, 1405, 1558, 1560	<code>class</code> , argument	17
<code>\AtBeginDocument</code>	15 , 27 , 44	<code>\class@documenthook</code>	15 , 31
<code>\AtBeginDocument</code>	323	<code>\class@documenthook</code>	322 , 535, 634
<code>\AtEndDocument</code>	15	<code>\class@enddocumenthook</code>	15 , 27
<code>\AtEndDocument</code>	325	<code>\class@enddocumenthook</code>	208, 322
<code>\AtEndOfClass</code>	17	<code>\class@err</code>	180
<code>\author</code>	34	<code>\class@ext@hook</code>	371
B		<code>\class@extension</code>	371
<code>\baselineskip</code>	12 , 21	<code>\class@extensionfile</code>	17
<code>\baselineskip</code>	1341, 1364, 1396, 1429	<code>\class@extensionfile</code>	371
<code>\baselineskip</code>	12	<code>\class@info</code>	180 , 394, 396, 472, 551, 791, 823, 884, 1082, 1084, 1164, 1187, 1190, 1920, 1922, 2089
<code>\begin</code>	15	<code>\class@name</code>	10 , 11
<code>\begin@float@pagebreak</code>	690, 704, 736, 762	<code>\class@name</code>	180–182, 201, 210, 211
<code>\belowdisplayskip</code>	455	<code>\class@warn</code>	180 , 212
<code>\belowdisplayskip</code>	454	<code>\class@warn@end</code>	184, 189, 194, 207
<code>\bgroup</code>	24 , 51	<code>\langle class customization commands \rangle</code> place- holder	9
<code>\body@font</code>	1503, 1538	<code>\ClassError</code>	180, 201
<code>\botrule</code>	51	<code>classes.dtx</code>	26
<code>\botrule</code>	1502	<code>classes.dtx</code> document class	31
<code>\box</code>	64	<code>\ClassInfo</code>	182
C		<code>\classname</code>	45, 53, 105, 107, 109, 124, 148, 149, 161
<code>\c@mpfootnote</code>	548, 549, 684	<code>\classoption</code>	151
<code>\c@secnumdepth</code>	886, 915, 929	<code>\ClassWarningNoLine</code>	181
<code>\caption</code>	28	<code>\clear@document</code>	333, 338
<code>\caption</code>	634	<code>\clearpage</code>	338
<code>\cat@letter</code>	218	<code>\closeout</code>	342, 756
<code>\catcode</code>	17	<code>\cmd</code>	117
		<code>\col@sep</code>	43
		<code>\col@sep</code>	1215, 1311

<code>\color</code>	1740	<code>\CT@end</code>	44
<code>\color@begingroup</code>	576	<code>\CT@row@color</code>	1742
<code>\color@endgroup</code>	580, 669	<code>\CT@setup</code>	1737
<code>colortbl</code> document class .	43, 44, 55, 57, 85, 86	<code>\CT@start</code>	44
<code>\colortbl@message</code> ..	1109, 1189	<code>\CurrentOption</code>	386
<code>\colrule</code>	51	D	
<code>\colrule</code>	1502	<code>\dollarbegin</code> .	1193, 1216, 1224, 1312, 1320, 1321
<code>\column@fil</code>	1502, 1537	<code>\dollarend</code>	1217, 1225, 1312, 1321
<code>\column@font</code>	1502, 1536	<code>\deadcycles</code>	335
<code>\columnsep</code>	595, 597	<code>\DeclareRobustCommand</code>	19
<code>\columnwidth</code>	25	<code>\DeclareRobustCommand</code>	428
<code>\columnwidth</code>	590, 693	<code>\dimen</code>	62
<code>\contentsline</code>	28, 61, 62	<code>\dimen@</code> ..	803, 804, 806, 807, 809, 810, 1557–1559, 1995, 1996
<code>\contentsline</code>	614, 631, 632	<code>\dimen@iii</code>	215
<code>\contentsline@latex</code> ..	628, 631	<code>\dimendef</code>	215
<code>\copy</code>	1544	<code>displaymath</code> environment	21
<code>\copyright</code>	31	<code>\displaystyle</code> .	438, 441, 460, 466
<code>\count</code>	32, 33	<code>\displaywidth</code>	437
<code>\count</code> ...	780, 783, 785, 786, 792, 1906, 1913	<code>\do</code>	1589, 1605, 1663, 1694
<code>\count@</code> ..	1637, 1643, 1659, 1690	<code>\do@check@aux</code>	330
<code>\count@i</code>	244	<code>\do@if@floats</code>	27, 31
<code>\count@ii</code>	244	<code>\do@if@floats</code>	712
<code>\countdef</code>	244, 245	<code>doc</code>	5
<code>\crrc</code>	1435, 1438, 1441, 1444, 1449, 1525	<code>doc/</code>	3
<code>\cs</code>	121, 122, 124, 135–137, 139, 140, 142–147, 150, 152, 159, 161	<code>\DocInput</code>	8
<code>\csname</code>	13, 47	<code>\document</code>	15
<code>\csname</code>	16, 17, 222, 224, 228, 230, 279, 295, 373, 375, 381, 491, 495, 523, 527, 533, 545, 546, 574, 625, 626, 715, 717, 720, 721, 752, 753, 757, 766, 770, 772, 799, 818, 835, 838, 900, 908, 913, 918, 924, 927, 932, 948, 953, 958, 961, 1180, 1932, 1933, 1972, 1975, 1978, 1980, 1984, 1985, 1993–1995, 2004, 2032	<code>\document</code>	316
<code>\CT@cell@color</code>	1743	<code>document class</code>	
<code>\CT@color</code>	1738	<code>array</code>	10, 39, 43, 45, 46, 55, 57
<code>\CT@column@color</code>	1741	<code>classes.dtx</code>	31
<code>\CT@do@color</code>	1739	<code>colortbl</code>	43, 44, 55, 57, 85, 86
		<code>ftnright</code>	9
		<code>hyperref</code>	19, 21, 23, 24, 28, 37, 38, 61, 62, 65, 66, 84–86
		<code>hyperref_␣package</code>	2, 28
		<code>longtable</code>	10, 55
		<code>ltxdoc</code>	5, 8
		<code>ltxgrid</code>	10, 16, 25
		<code>ltxutil</code>	1, 9
		<code>ltxutil.dtx</code>	3
		<code>ltxutil.pdf</code>	3
		<code>ltxutil.sty</code>	3
		<code>textcase</code>	19, 86

document class option
 lengthcheck 86
document environment 5, 16
\document@inithook 15
\document@inithook .. 318, 321,
 322, 630, 980, 984
\doublerulesep 1519
\dp 587, 793, 809, 810,
 822, 1248, 1249, 1272, 1278,
 1330, 1374, 1383, 1405, 1560

E

\edef 55
\edef 249, 254,
 259, 502, 1333, 1746, 1752,
 1761, 1773, 1820, 1975
\emph 76
\end 34
\end@float@anchored . 691, 692,
 698
\end@float@pagebreak 702, 705,
 736
\endarray 44, 49
\endarray . 984, 1001, 1091, 1115,
 1173, 1434, 1452, 1455, 1458
\endarray@array 1115, 1440
\endarray@array@new 1173, 1443
\endarray@hook 988, 989
\endarray@LaTeX 1001, 1434
\endarray@ltx 49
\endarray@ltx .. 1091, 1437, 1444
\endcsname 16, 17, 222,
 224, 228, 230, 279, 295, 373,
 375, 381, 491, 495, 523, 527,
 533, 545, 546, 574, 625, 626,
 715, 717, 720, 721, 752, 753,
 757, 766, 770, 772, 799, 818,
 835, 838, 900, 908, 913, 918,
 924, 927, 932, 948, 953, 958,
 961, 1180, 1932, 1933, 1972,
 1975, 1978, 1980, 1984, 1985,
 1993–1995, 2004, 2032
\enddocument 330
\endgroup 15
\endminipage 664

\endtabular 984, 1002, 1092, 1116,
 1174, 1180, 1448, 1460
endtabular environment 50
\endtabular* 1460
\endtabular@array .. 1116, 1454
\endtabular@array@new ... 1174,
 1457
\endtabular@hook 986, 987
\endtabular@LaTeX .. 1002, 1448
\endtabular@ltx 1092, 1451
\endwrite@float 828, 838
\endwrite@floats 830
environment
 array 2, 39, 44
 displaymath 21
 document 5, 16
 endtabular 50
 eqnarray 19, 21, 45
 equation 21
 figure 26
 figure@write 26, 31
 longtable 26, 34
 minipage 21, 26
 tabular 2, 39, 44, 45
\eqnarray 19
\eqnarray 471, 473, 486
eqnarray environment . 19, 21, 45
\eqnarray@fleqn@fixed 429
\eqnarray@LaTeX 429
\eqncolsep ... 462, 465, 474, 475
\equation 21
\equation 485
equation environment 21
\errorstopmode 236
\extrarowheight 1380, 1402

F

\f@ur 217, 1683
\false@sw ... 285, 291, 296, 308,
 313, 1019, 1022, 1025, 1028,
 1031, 1034, 1037, 1040, 1043,
 1046, 1049, 1052, 1055, 1058,
 1061, 1064, 1067, 1070, 1073,
 1076, 1079, 1126, 1129, 1132,
 1135, 1138, 1141, 1144, 1147,

1150, 1153, 1156, 1159, 1162, 2062	
figure environment	26
figure@write environment	26, 31
\figuresname	27
file	
\$TEXMF/	3
.aux	38
.cls	18
.dtx	5
.rtx	17
00readme	3, 5
aps.rtx	17
classes.dtx	26
doc	5
doc/	3
kernel	2, 10, 67
latex/	3
ltxfloat.dtx	21
ltxutil.dtx	2, 3
ltxutil.pdf	1
ltxutil.sty	1–3
makeindex	3
nameref.sty	37
package	2, 10
pdfmark.def	37
revtex/	3
source/	3
tex/	3
texmf-local/	3
\file	64, 67, 73, 74, 83, 89, 91, 94, 95, 98, 99, 105, 107, 109, 111, 114
float, argument	34
\float@end@tag	34
\float@end@tag	843
\floatingpenalty	25
\floats@sw	30, 31
\floats@sw	673, 678, 713, 743
\flushing	421
\font@submax	350, 353
\font@subfuzz	350
\footins	562, 661
\footmark	22
\footnote	21, 22, 26
\footnote	488, 604
\footnote@latex	536
\footnotemark	21–24
\footnotemark	488
\footnotesep	578, 586
\footnotesize	584
\footnotetest	488
\footnotetext	21–23
\footnotetext	512
\frustrut	1502, 1503, 1540
ftnright document class	9
\ftype@	32
\fullinterlineskip	243, 651
\futurelet	1506, 1513
G	
\gappdef	208, 246
\GetFileInfo	22
\glet@environment	195, 220
\glossary	603
H	
\H@@footnotetext	21
\H@@footnotetext	538, 539, 685
\H@mpfootnotetext	540, 541, 685
\H@refstepcounter	36
\H@refstepcounter	890, 979
\H@svsec	887, 891, 909, 925, 942, 949, 959
\halign	49, 55
\halign@	216, 459
\hangindent	65
\hb@xt@	442, 467, 709, 2002, 2012
\hbox	20, 21, 65
\hline	52
\hline	1012, 1102, 1505
\hline@LaTeX	1012, 1505
\hline@ltx	1102, 1509
\hline@rule	52
\hline@rule	1504, 1512, 1527, 1542
\href	66
\href	2056
\hrule	12
\hspace	15
\ht	793, 803, 806, 807, 821, 1329, 1373, 1379,

1401, 1557, 1558
 \Hy@footnote@currentHref . 505,
 546
 \Hy@raisedlink 972
 \hyper@anchorend 38
 \hyper@anchorend 974
 \hyper@anchorstart 38
 \hyper@anchorstart 973
 \hyper@last 38
 \hyper@linkend 38
 \hyper@linkend 507, 976
 \hyper@linkstart 38
 \hyper@linkstart 505, 975
 \hyper@makecurrent 971
 \hyperanchor 38
 hyperref document class . 19, 21,
 23, 24, 28, 37, 38, 61, 62,
 65, 66, 84–86
 hyperref_Lpackage document class
 2, 28

I

\ialign 47
 \ialign 1334, 1387
 \if 47
 \if 282, 294, 1326, 1392
 \if@filesw . 341, 358, 1931, 1960
 \if@firstamp 1746, 1749
 \if@nobreak 874
 \if@noskipsec 867
 \ifeof 298
 \iffalse 38, 1254, 1259
 \IfFileExists 372
 \ifhbox 299
 \ifinner 301
 \ifmmode 302
 \ifodd 304, 1910
 \ifvbox 305
 \ifvoid 307
 \ignorespaces
 . . . 578, 1466, 1479, 1636,
 1642, 1823, 1825, 1827, 1846,
 1850, 1854, 1871, 1879, 2001,
 2008, 2020, 2040
 \immediate . . 342, 718, 755, 853,
 1933, 1983, 2079, 2085

\incompatible@package 198
 \index 602
 \init@hyperref 969
 \insert 562, 661
 \insert@column . 1119, 1179, 1634
 \insert@column@array 1119, 1634
 \insert@column@array@new 1179,
 1639
 \interdisplaylinepenalty . . 45
 \interfootnotelinepenalty 585
 \interlinepenalty 585, 906, 946
 \intertabularlinepenalty . . 45
 \intertabularlinepenalty 1240,
 1241, 1243, 1245, 1260
 \intextsep 704, 705
 \item . . 87, 97, 100, 104, 106, 108,
 113, 115
 \itemindent 2031
 \itshape 548, 557

J

\j@nk 314

K

kernel 2, 10, 67

L

\l@ 62
 \l@@sections 62
 \l@@sections 1989
 \l@section 1943
 \label 601, 612
 \lastbox 482, 654, 760
 \LaTeX . 25, 54, 55, 65, 75, 76, 78,
 89, 93, 96, 98
 latex/ 3
 \LaTeXe 59, 84, 172
 \leftmargin . . . 2044, 2046, 2051
 \leftmargini 1943
 \leftmarginii 1944
 \leftmarginiii 1945
 \leftmarginiv 1946
 \leftmarginv 1947
 \leftmarginvi 1948
 \leftskip 65
 \leftskip . 423, 1994, 1996, 2053

lengthcheck document class option	10, 16, 25
.....	86
\lengthcheck@sw	738
\let	43, 63
\let@environment	11
\let@environment	220, 728, 729, 731, 750, 751
\lineskip	1341, 1364, 1395, 1429
\linewidth	459, 591, 598, 2045–2047
\list	65
\list	2023
\listparindent	2030, 2043, 2050
\LoadClass	17
longtable document class	10, 55
longtable environment	26, 34
\loop	14
\loopuntil	310
\loopwhile	310
\lrstrut	1503, 1504, 1541
\LT@array	10
ltxfloat.dtx	21
\ltx@contentsline	612
\ltx@def@footproc	22, 23
\ltx@def@footproc	490, 498, 513, 521
\ltx@footmark	22
\ltx@footmark	488
\ltx@footnote@pop	681, 687
\ltx@foottext	22, 23
\ltx@foottext	488
\ltx@make@current@footnote	488
\ltx@mathindent	20
\ltx@mathindent	449, 479
\ltx@stp@footproc	22, 23
\ltx@stp@footproc	494, 499, 514, 529
\ltx@thempfootnote	549, 552
\ltx@xfootmark	497, 498
\ltx@xfootnote	22
\ltx@xfootnote	488
\ltx@xfoottext	512, 513
\ltx@yfootmark	497, 499
\ltx@yfootnote	488, 493
\ltx@yfoottext	512, 514
ltxdoc document class	5, 8
ltxgrid document class	10, 16, 25
\ltxu@dotsep	1942, 2015
ltxutil document class	1, 9
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	24
\make@footnotetext	554
\makeatletter	346, 385, 1929
makeindex	3
\makelabel	2034
\maketitle	24
\maketitle	43
\mark	27
\mathchardef	776, 777
\mathindent	20
\meaning	241
\MessageBreak	352
minipage environment	21, 26
\minipagefootnote@drop	24
\minipagefootnote@drop	581, 637
\minipagefootnote@foot	637
\minipagefootnote@here	29
\minipagefootnote@here	635, 637, 699
\minipagefootnote@init	637, 696
\minipagefootnote@pick	566, 637
\mpfootnote	551
\multicolumn	997, 1087, 1461
\multicolumn@LaTeX	997, 1461
\multicolumn@ltx	1087, 1468
\multiply	782
\multispan	1462, 1469
N	
nameref.sty	37
\NC@list	1658, 1689
\newbox	1553
\newcount	1240, 1242
\newenvironment	34
\newlabel	28

`\robustify@contents` 600
`\romannumeral` 2032
`\rule` 578

S

`\samepage` 45
`\samepage` 1243
`\save@decl` 1666, 1703
`\say` 12
`\say` 241, 1517
`\saythe` 12
`\saythe` 241
`\sc` 75
`\sec@upcase` 39
`\sec@upcase` 983
`\section` 71, 763
`\Sectionformat` 37
`\set@arstrutbox` 52
`\set@arstrutbox` 1347, 1371, 1542
`\set@display@protect` 2078, 2084
`\set@eqnarray@skips` 480
`\set@footnotefont` 25
`\set@footnotefont` 570, 583
`\set@footnotewidth` 25, 26
`\set@footnotewidth` 554
`\set@footnotewidth@ii` 25
`\set@footnotewidth@ii` 593
`\set@listindent` 2039, 2055
`\set@listindent@` 2049
`\set@listindent@parshape` 2042, 2055
`\set@tocdim@pagenum` 64
`\set@tocdim@pagenum` 1992, 2007
`\set@typeset@protect` . . . 1340, 1363, 1428, 1464, 1473
`\shipout` 12
`\show@box@size@sw` 790
`\showboxbreadth` 236
`\showboxdepth` 236
`\showoutput` 239
`\skip@` 1998, 2001
`source/` 3
`\spacefactor` 502, 509
`\special` 2064, 2065
`\splitmaxdepth` 587
`\splittopskip` 586

`\stepcounter` 430, 447, 530
`\stop` 202
`\StopEventually` 5
`\string` 184, 189, 200, 551, 605–607, 610, 620, 884, 1920, 1922, 1984, 2089
`\strutbox` 578, 587, 1329, 1330, 1373, 1374, 1379, 1383, 1401, 1405, 1558, 1560
`\subsection` 81, 120
`\substyle@ext` 17
`\switch@array` 39, 45
`\switch@array` 985, 994
`\switch@array@info` . 1109, 1166
`\switch@tabular` 39, 45
`\switch@tabular` 985, 994

T

`\t@` 214, 216
`\tab@rule` 52
`\tab@rule` 1502, 1503, 1524
`\tabcolsep` 1215, 1573–1575, 1761
`\tableleft@skip` . . 1365, 1430, 1570
`\tableleftsep` . . . 1573, 1647, 1758
`\tableofcontents` 69
`\tabmid@skip` . . . 1350, 1413, 1571
`\tabmidsep` 1574, 1648, 1764
`\tabright@skip` . . 1354, 1417, 1572
`\tabrightsep` . . . 1575, 1649, 1767
`\tabskip` 20, 47
`\tabskip` 435, 438, 441, 443, 449, 460, 466, 468, 1335, 1350, 1354, 1365, 1389, 1413, 1417, 1430
`\tabular` 39
`tabular environment` 2, 39, 44, 45
`\tabular@hook` 991, 992
`\tabularnewline` 1337, 1360, 1398, 1425
`\tally@float` 32
`\tally@float` 775
`\TeX` 51, 59, 98
`tex/` 3
`texmf-local/` 3
`text, argument` 22, 39

<code>textcase document class</code>	19, 86
<code>\texttt</code>	36, 94, 95, 102, 111, 114, 117, 121
<code>\textwidth</code>	594, 805
<code>\thanks</code>	26, 30, 36
<code>\the@toks</code>	1635, 1637, 1640, 1643, 1660, 1679, 1691, 1734
<code>\theequation</code>	431, 447
<code>\thefootnote</code>	22
<code>\thempfn</code>	22
<code>\thempfn</code>	524, 531, 683
<code>\thempfootnote</code>	24
<code>\thempfootnote</code>	550, 552, 683
<code>\thempfootnote@latex</code>	548, 550
<code>\thepage</code>	614, 620
<code>\thr@@</code>	11
<code>\thr@@</code>	215, 442, 467, 1675, 1726
<code>\title</code>	24
<code>\toc@font</code>	62
<code>\toc@font</code>	1941, 1950
<code>\toc@letdimen</code>	1956, 1957, 1977
<code>\toc@post</code>	63
<code>\toc@post</code>	1936, 1940
<code>\toc@post@auto</code>	63
<code>\toc@post@auto</code>	1949
<code>\toc@pre</code>	63
<code>\toc@pre</code>	1928, 1939
<code>\toc@pre@auto</code>	63
<code>\toc@pre@auto</code>	1949
<code>\toc@setindent</code>	1952–1955, 1971
<code>\toc@writedimen</code>	1962–1967, 1982
<code>\tocdim@appendix</code>	1947
<code>\tocdim@min</code>	1972, 1978, 2022
<code>\tocdim@pagenum</code>	1948, 2009
<code>\tocdim@paragraph</code>	1946
<code>\tocdim@section</code>	1943
<code>\tocdim@subsection</code>	1944
<code>\tocdim@subsubsection</code>	1945
<code>\tocleft@pagenum</code>	1997
<code>\tocleft@section</code>	64
<code>\tocmax@section</code>	64
<code>\toks</code>	1679, 1734
<code>\toks@</code>	247, 249, 252, 254, 257, 259, 836, 853, 855
<code>\toks@ii</code>	13
<code>\toks@ii</code>	248, 249, 253, 254, 258, 259, 274
<code>\toksdef</code>	274
<code>\toprule</code>	51
<code>\toprule</code>	1502
<code>\topsep</code>	451
<code>\total@float</code>	32
<code>\total@float</code>	739, 775
<code>\traceoutput</code>	12
<code>\traceoutput</code>	238
<code>\tracingall</code>	12
<code>\tracingcommands</code>	233
<code>\tracinglostchars</code>	234
<code>\tracingmacros</code>	235
<code>\tracingonline</code>	233
<code>\tracingoutput</code>	12
<code>\tracingoutput</code>	234
<code>\tracingpages</code>	234
<code>\tracingparagraphs</code>	235
<code>\tracingplain</code>	12
<code>\tracingplain</code>	232
<code>\tracingrestores</code>	235
<code>\tracingstats</code>	233
<code>\trigger@float@par</code>	749, 752, 753, 759
<code>\triggerpar</code>	735, 749
<code>\true@sw</code>	283, 290, 296, 308, 313, 319, 1017, 1124, 2062
<code>\tw@</code>	11
<code>\typeout@ltx</code>	2082, 2090
<code>\typeout@org</code>	2076, 2088
U	
<code>\unhbox</code>	2002, 2017
<code>\unhcopy</code>	1544
<code>\unrestored@protected@xdef</code>	524
<code>\unskip</code>	666, 1271, 1277, 1503, 1504, 1636, 1642, 1823, 1825, 1827, 1846, 1850, 1854, 2001
<code>\unvbox</code>	642, 656, 661, 671
<code>\unvcopy</code>	653, 802
<code>\uppercase</code>	39
<code>\url</code>	66
<code>\url</code>	14, 49, 56, 2056
<code>\URL@prefix</code>	2056
<code>\usepackage</code>	3

V

`\vbox` . . . 641, 802, 1326, 1392, 1482
`\vcenter` 51
`\vcenter` . . . 1326, 1392, 1483, 1487,
1490
`\voidb@x` 638
`\vrule` 1248, 1249, 1273, 1279, 1329,
1373, 1381, 1404, 1559
`\vsize` 15
`\vtop` 653, 1326, 1392, 1481

W

`\write` 853, 1983, 2079, 2085
`\write@float` 827, 831
`\write@float` 26, 31, 34
`\write@float` 725, 827
`\write@floatline` 34
`\write@floatline` 841, 843
`\write@floats` 726, 827

X

`\xdef` 389, 405, 1387, 1410

Y

`your_toks_here`, argument . . . 15
`<your document here>` placeholder 9

Z

`\z@skip` . . . 423–425, 438, 443, 460,
468, 700, 1335, 1341, 1364,
1570–1572, 2002

Change History

4.0b		4.0c	
	<code>\@mpmakefn</code> text: AO: Removed superfluous <code>\defs</code> , changed to using <code>\floats@sw</code> as the flag. Also stopped using DPC's <code>\if@twocolumn</code> flag: using <code>\floats@sw</code> instead. Also added <code>\par\vskip\z@skip</code> after the <code>\minipagefootnotes</code> so that the float box would have zero depth like the kernel one. 30		<code>\@mpmakefn</code> text: (AO, 110) Install hooks for endfloats processing 30
	General: AO: Fixed spurious CR and (return) characters in output file. Also, if the document did not have the <code>\endfigure</code> on a line of its own, the macro wouldn't work. Fixed. 3		<code>\@ssect</code> : (AO, 116) Hyperref compatibility 37
	AO: Removed superfluous <code>\defs</code> , changed to using <code>\floats@sw</code> as the flag. Also stopped using DPC's <code>\if@twocolumn</code> flag: using <code>\floats@sw</code> instead. Also added <code>\par\vskip\z@skip</code> after the <code>\minipagefootnotes</code> so that the float box would have zero depth like the kernel one. 3		General: (AO, 110) Install hooks for endfloats processing 3
	only execute if there really were floats of the given type 3		(AO, 116) Hyperref compatibility 3
	Support the hack with <code>\prepdef</code> , and delay until <code>\AtBeginDocument</code> time, since hyperref clobbers <code>\caption</code> 3		(AO, 130) Interference from array package 3
	<code>\caption</code> : Support the hack with <code>\prepdef</code> , and delay until <code>\AtBeginDocument</code> time, since hyperref clobbers <code>\caption</code> 28		*-form mandates pagebreak at each float; only print section head if there is something there. 3
	<code>\print@float</code> : only execute if there really were floats of the given type 31		<code>\endarray</code> : (AO, 130) Interference from array package 39
	<code>\write@float</code> : AO: Fixed spurious CR and (return) characters in output file. Also, if the document did not have the <code>\endfigure</code> on a line of its own, the macro wouldn't work. Fixed. 34		<code>\print@float</code> : *-form mandates pagebreak at each float; only print section head if there is something there. 31
		4.0d	
			<code>\@mpmakefn</code> text: (AO, 127) Floats placed [h] to allow page breaks 30
			(AO, 224) Hyperref compatibility. 30
			General: (AO, 127) Floats placed [h] to allow page breaks 3
			(AO, 174) kernel fix 3, 21
			(AO, 224) Hyperref compatibility. 3
			Allow things to break over pages by setting <code>array@default</code> 3
			<code>\print@float</code> : Allow things to break over pages by setting <code>array@default</code> 31
		4.0e	
			<code>\@mpmakefn</code> text: (AO, 221) Remove <code>samepage</code> command from <code>@xfloat@prep</code> : If the float can break over pages, we want better control. 30
			General: (AO, 221) Remove <code>samepage</code> command from <code>@xfloat@prep</code> : If the float can break over pages, we want better control. 3

4.0f	<code>\@ssect:</code> (AO, 404) Hyperref compatibility 37	<code>\normalsize</code> directive 3
	General: (AO, 404) Hyperref compatibility 3	<code>\class@enddocumenthook:</code> <code>\class@documenthook</code> is the last <code>\AtBeginDocument</code> token now 15
4.1a	<code>\@mpmakefntext:</code> <code>\@xfloat@prep</code> calls <code>\ltx@footnote@pop</code> to restore the original <code>\ltx@footmark</code> and <code>\ltx@foottext</code> procedures, in case footnote processing has switched. 30	<code>\document:</code> Get rid of <code>\set@typesize@hook</code> <code>\set@pica@hook</code> and the <code>\normalsize</code> directive 15
	<code>\@p@pfilename:</code> Class extension mechanism <code>\@pushfilename@ltx</code> and <code>\@p@pfilename@ltx</code> 18	<code>\eqnarray@fleqn@fixed:</code> (AO, 475) I had not properly reproduced the LaTeX macro <code>\eqnarray</code> 19
	General: (AO) Make <code>\addtocontents</code> a <code>\long \def;</code> gobble up <code>\footnote</code> 3	<code>\footnote:</code> (AO) Remove code that avoided changes to <code>\xfootnotemark</code> 24
	(AO) Remove code that avoided changes to <code>\xfootnotemark</code> . . 3	<code>\ltx@make@current@footnote:</code> (AO, 438) Complete rewrite of footnote macros. 22
	(AO, 438) Complete rewrite of footnote macros. 3	<code>\numberline@@sections:</code> (AO, 461) Change the csname from <code>\@dotsep</code> to <code>\ltxu@dotsep</code> . The former is understood in mu. (What we wanted was a dimension.) 65
	(AO, 459) do not assume <code>\class@name</code> is defined . . . 3, 11	<code>\robustify@contents:</code> (AO) Make <code>\addtocontents</code> a <code>\long \def;</code> gobble up <code>\footnote</code> 27
	(AO, 461) Change the csname from <code>\@dotsep</code> to <code>\ltxu@dotsep</code> . The former is understood in mu. (What we wanted was a dimension.) . . . 3	<code>\toc@@font:</code> (AO, 461) Change the csname from <code>\@dotsep</code> to <code>\ltxu@dotsep</code> . The former is understood in mu. (What we wanted was a dimension.) . . . 62
	(AO, 475) I had not properly reproduced the LaTeX macro <code>\eqnarray</code> 3	
	<code>\@xfloat@prep</code> calls <code>\ltx@footnote@pop</code> to restore the original <code>\ltx@footmark</code> and <code>\ltx@foottext</code> procedures, in case footnote processing has switched. 3	4.1b
	<code>\class@documenthook</code> is the last <code>\AtBeginDocument</code> token now . 3	<code>\@mkpream:</code> (AO, 505) try to accommodate <code>colortbl</code> 55
	Class extension mechanism <code>\@pushfilename@ltx</code> and <code>\@p@pfilename@ltx</code> 3	<code>\@mkpream@relax:</code> (AO, 505) try to accommodate <code>colortbl</code> 57
	Class extension mechanism <code>\class@extension</code> , <code>\class@extensionfile</code> , and <code>\class@ext@hook</code> 3, 17	<code>\@mpmakefntext:</code> No need to protect against undefined <code>\float@sw</code> 30
	Get rid of <code>\set@typesize@hook</code> <code>\set@pica@hook</code> and the	<code>\@tabarray:</code> (AO, 505) try to accommodate <code>colortbl</code> 44
		<code>\@tabular:</code> (AO, 505) try to accommodate <code>colortbl</code> 43
		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> <code>savoire</code> 3

Default assignment of <code>\float@sw</code> now, not at <code>\AtBeginDocument</code> time.	3	<code>\URL@prefix:</code> (AO, 487) Support for video figures and the <code>\setfloatlink</code> command . . .	66
If class option <code>lengthcheck</code> is in effect, log the height of this float class.	3	4.1d	
No need to protect against undefined <code>\float@sw</code>	3	General: (AO, 511) Compatability with <code>lineno.sty</code> 's erroneous way of detecting <code>fleqn.clo</code>	3
Patch the array package even later: after all package patches go in.	3	<code>\eqnarray@fleqn@fixed:</code> (AO, 511) Compatability with <code>lineno.sty</code> 's erroneous way of detecting <code>fleqn.clo</code>	20
Refine toc processing: provide default.	3	4.1f	
Tally and log the height of a float class	3	General: (AO, 515) Hook for setting the font of a footnote	3
<code>\array:</code> (AO, 505) try to accommodate <code>colortbl</code>	47	(AO, 518) Tally register overflow when locument is long	3
<code>\do@if@floats:</code> No need to protect against undefined <code>\float@sw</code> .	31	<code>\set@footnotewidth:</code> (AO, 515) Hook for setting the font of a footnote	25
<code>\endarray:</code> Patch the array package even later: after all package patches go in.	39	<code>\total@float:</code> (AO, 518) Tally register overflow when locument is long	32
<code>\floats@sw:</code> Default assignment of <code>\float@sw</code> now, not at <code>\AtBeginDocument</code> time.	30	4.1g	
<code>\init@hyperref:</code> Acquire <code>hyperref</code> <code>savoire</code>	38	General: (AO, 525) Remove phantom paragraph above display math that is given in vertical mode	3, 20
<code>\ltx@contentsline:</code> Refine toc processing: provide default. . .	28	(AO, 532) Both arguments of <code>\href</code> get sanitized	3
<code>\print@float:</code> If class option <code>lengthcheck</code> is in effect, log the height of this float class. .	31	(AO, 539) Use of double-backslash in argument of <code>\section</code> gives error. The <code>textcase</code> package is involved.	3, 18
<code>\switch@array:</code> (AO, 505) Try to accommodate <code>colortbl</code>	42	<code>\URL@prefix:</code> (AO, 532) Both arguments of <code>\href</code> get sanitized .	66
<code>\total@float:</code> Tally and log the height of a float class	32		