

A L^AT_EX Package of utility macros *

Arthur Ogawa (<mailto:ogawa@teleport.com>), 1.0rc5b
Copyright (C) 1999 Arthur Ogawa

August 3, 2001

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

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

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 is 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	2
1.2	Bill of Materials	3
1.2.1	Primary Source	3
1.2.2	Generated by <code>latex ltxutil.dtx</code>	3
1.2.3	Generated by <code>tex ltxutil.ins</code>	3
1.2.4	Documentation	3
1.2.5	Auxiliary	3
2	Code common to all modules	3
3	The driver module <code>doc</code>	4
3.1	The Preamble	4
3.1.1	Docstrip and info directives	4
3.2	The installer file	4
3.3	The “Read Me” File	6
3.4	The Document Body	7

*This file has version number 1.0rc5b, last revised 2001/07/31. For version number and date, search on “1.0rc5b” in the `.dtx` file, or see the end of the `00readme.txt` file.

4	Using this package	7
4.1	Invoking the package	8
5	Compatability with L^AT_EX's Required Packages	8
5.1	array	8
5.2	longtable	8
6	Implementation of package	9
6.1	Beginning of the ltxutil DOCSTRIP module	9
6.2	Banner	9
6.3	Errors and warnings	9
6.4	New Tools	10
6.5	Boolean Control	11
6.6	Begin Document Structure	13
6.7	Type Tools	15
6.8	Display Math	15
6.9	Footnotes	16
6.10	Floats	19
6.10.1	Usage notes	19
6.10.2	Robustifying fragile commands	20
6.10.3	Preparing for the hyperref package	20
6.10.4	Footnotes within floats, unfloating floats, float font	21
6.10.5	Writing floats out to a file	23
6.11	Counters	25
6.12	Customization of Sections	26
6.13	Patch the tabular and array Environments	29
6.14	Repair other broken parts of L ^A T _E X	49
6.15	Syntax	49
6.16	Auto-indented Contents	49
6.17	Lists	52
6.18	End of the ltxutil DOCSTRIP module	53
	Index	54

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.ins`. The typeset documentation that you are now reading is generated from the same file by typesetting it with L^AT_EX or pdftex via `latex ltxutil.dtx` or `pdflatex ltxutil.dtx`.

1.1 Build Instructions

You may bootstrap this suite of files solely from `ltxutil.dtx`. Prepare by installing L^AT_EX 2_ε (and either `tex` or `pdftex`) on your computer, then carry out the following steps:

1. Within an otherwise empty directory, typeset `ltxutil.dtx` with L^AT_EX or `pdflatex`; you will obtain the typeset documentation you are now reading, along with the installer `ltxutil.ins`, and the file `00readme.txt`.

Note: you will have to run \LaTeX twice, then `makeindex`, then \LaTeX again in order to obtain a valid index and table of contents.

2. Now typeset `ltxutil.ins`, thereby generating the package file `ltxutil.sty`.
3. Install `ltxutil.sty` by moving it to a location in your filesystem where they will be found by \LaTeX .

1.2 Bill of Materials

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

1.2.1 Primary Source

One single file generates all.

```
%ltxutil.dtx
%
```

1.2.2 Generated by `latex ltxutil.dtx`

Typesetting the source file under \LaTeX generates the readme and the installer.

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

1.2.3 Generated by `tex ltxutil.ins`

Typesetting the installer generates the package files.

```
%ltxutil.sty
%
```

1.2.4 Documentation

The following are the online documentation:

```
%ltxutil.pdf
%
```

1.2.5 Auxiliary

The following are auxiliary files generated in the course of running \LaTeX :

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

2 Code common to all modules

The following may look a bit klutzy, but we want to require only one place in this file where the version number is stated, and we also want to ensure that the version number is embedded into every generated file.

Now we declare that these files can only be used with $\LaTeX 2\epsilon$. An appropriate message is displayed if a different \TeX format is used.

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

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

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

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

```
8 %<*doc|ltxutil>
9 [2001/07/31 1.0rc5b utilities package]% \fileversion
10 %</doc|ltxutil>
```

3 The driver module doc

This module, consisting of the present section, typesets the programmer's documentation, generating the `.ins` installer and `00readme.txt` as required.

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

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

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

3.1 The Preamble

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

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

3.1.1 Docstrip and info directives

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

```
20 \setcounter{StandardModuleDepth}{1}
```

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

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

3.2 The installer file

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

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

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

```
22 \begin{filecontents}{ltxutil.ins}
23 %% This file will generate documentation and runtime files
24 %% from ltxutil.dtx when run through LaTeX or TeX.
25 \input docstrip
26 \preamble
27
28 This is a generated file;
29 altering it directly is inadvisable;
30 instead, modify the original source file.
31 See the URL in the file 00readme.txt.
32
33 Copyright notice.
34
35   These files are distributed
36   WITHOUT ANY WARRANTY; without even the implied warranty of
37   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
38
39 \endpreamble
40 \keepsilent
41   \generate{%
42     \file{ltxutil.drv}{\from{ltxutil.dtx}{doc}}%
43     \file{ltxutil.sty}{%
44       \from{ltxutil.dtx}{ltxutil,ltxutil-krn}%
45     }%
46   }%
47 \iftoplevel{
48 \Msg{*****}
49 \Msg{*}
50 \Msg{* To finish the installation, please move}
51 \Msg{*   ltxutil.sty}
52 \Msg{* into a directory searched by TeX.}
53 \Msg{*}
```

```

54 \Msg{* To produce the documentation,
55       run ltxutil.dtx through LaTeX.}
56 \Msg{*}
57 \Msg{* Happy TeXing}
58 \Msg{*****}
59 }
60 \endbatchfile
61 \end{filecontents}

```

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

3.3 The “Read Me” File

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

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

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

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

```

62 \begin{filecontents*}{00readme.txt}
63 \title{%
64 A \LaTeX\ Package of utility macros%
65 \thanks{%
66 This file has version number \fileversion,
67 last revised \filedate.%
68 % For version number and date,
69 % search on "\fileversion" in the .dtx file,
70 % or see the end of the 00readme.txt file.
71 }%
72 }%
73
74 \author{%
75 Arthur Ogawa (\texttt{mailto:ogawa@teleport.com}),
76 \fileversion\Copyright (C) 1999 Arthur Ogawa
77 }%
78 \maketitle
79
80 This file embodies the \classname{ltxutil} package,
81 the implementation and its user documentation.
82
83 The distribution point for this work is
84 \url{ftp://ftp.teleport.com/users/ogawa/macros/latex/contrib/supported/ltxutil...}
85 which contains fully unpacked, prebuilt runtime files and documentation.
86
87 The \classname{ltxutil} package was commissioned by the American Physical Society
88 and is distributed under the terms of the \LaTeX\ Project Public License,
89 the same license under which all the portions of \LaTeX\ itself is distributed.

```

90 Please see `\url{http://ctan.tug.org/macros/latex/base/lppl.txt}` for details.
91
92 To use this document class, you must have a working
93 `\TeX` installation equipped with `\LaTeXe`
94 and possibly `pdftex` and Adobe Acrobat Reader or equivalent.
95
96 To install, retrieve the distribution,
97 unpack it into a directory on the target computer,
98 and move the file `\file{ltxutil.sty}`
99 into a location in your filesystem where it will be found by `\LaTeX`.
100
101 To use, read the user documentation `\file{ltxutil.pdf}`.
102
103 `\tableofcontents`
104
105 `\section{Processing Instructions}`
106
107 The package file `\file{ltxutil.sty}`
108 is generated from this file, `\file{ltxutil.dtx}`,
109 using the `{\sc docstrip}` facility of `\LaTeX`
110 via `|tex ltxutil.ins|`.
111 The typeset documentation that you are now reading is generated from
112 the same file by typesetting it with `\LaTeX` or `pdftex`
113 via `|latex ltxutil.dtx|` or `|pdflatex ltxutil.dtx|`.
114
115 `\subsection{Build Instructions}`
116
117 You may bootstrap this suite of files solely from `\file{ltxutil.dtx}`.
118 Prepare by installing `\LaTeXe` (and either `tex` or `pdftex`) on your computer,
119 then carry out the following steps:
120 `\begin{enumerate}`
121 `\item`
122 Within an otherwise empty directory,
123 typeset `\file{ltxutil.dtx}` with `\LaTeX` or `pdflatex`;
124 you will obtain the typeset documentation you are now reading,
125 along with
126 the installer `\file{ltxutil.ins}`,
127 and the file `\file{00readme.txt}`.
128
129 Note: you will have to run `\LaTeX` twice, then `\file{makeindex}`, then
130 `\LaTeX` again in order to obtain a valid index and table of contents.
131 `\item`
132 Now typeset `\file{ltxutil.ins}`,
133 thereby generating the package file `\file{ltxutil.sty}`.
134 `\item`
135 Install `\classname{ltxutil.sty}`
136 by moving it to a location
137 in your filesystem where they will be found by `\LaTeX`.
138 `\end{enumerate}`
139 `\end{filecontents*}`

3.4 The Document Body

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

```
140 \begin{document}%
141 \expandafter\DocInput\expandafter{\jobname.dtx}%
142 % ^^A\PrintChanges
143 \end{document}
144 %</doc>
```

4 Using this package

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

4.1 Invoking the package

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

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

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

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

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

5 Compatibility with \LaTeX 's Required Packages

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

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

Hereinafter follows some notes on specific \LaTeX packages.

5.1 array

This package alters the way tabular environments are done, therefore it could run afoul of the 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@superscript.com>) in his SuperScript line of commercial typesetting tools, and which are used here with his permission.

6.1 Beginning of the `ltxutil` DOCSTRIP module

```
145 %<*ltxutil>
146 \def\package@name{ltxutil}%
147 \expandafter\PackageInfo\expandafter{\package@name}{%
148 Utility macros for \protect\LaTeXe,
149 by A. Ogawa (ogawa@teleport.com)%
150 }%
151 %</ltxutil>
```

6.2 Banner

Credit where due.

```
152 %<*ltxutil-krn>
153 \typeout{%
154 ltxutil: portions licensed from W. E. Baxter (web@superscript.com)%
155 }%
```

6.3 Errors and warnings

```
\class@err A few shorthands for Class messages. Your document class should define \class@name.
\class@warn 156 \def\class@err#1{\ClassError{\class@name}{#1}\@eha}%
\class@info 157 \def\class@warn#1{\ClassWarningNoLine{\class@name}{#1}}%
158 \def\class@info#1{\ClassInfo{\class@name}{#1}}%
159 \def\obsolete@command#1{%
160 \class@warn@end{Command \string#1\space is obsolete.^^JPlease remove from your do
161 \global\let#1\@empty
162 #1%
163 }%
164 \def\replace@command#1#2{%
165 \class@warn@end{Command \string#1\space is obsolete;^^JUUse \string#2\space instea
166 \global\let#1#2%
```

```

167 #1%
168 }%
169 \def\replace@environment#1#2{%
170 \class@warn@end{Environment #1 is obsolete; ^JUse #2 instead}%
171 \glet@environment{#1}{#2}%
172 \@nameuse{#1}%
173 }%
174 \def\incompatible@package#1{%
175 \@ifpackageloaded{#1}{%
176 \def\@tempa{I cannot continue. You must remove the \string\usepackage\ statement
177 \ClassError{\class@name}{The #1 package cannot be used with \class@name}%
178 \@tempa\stop
179 }{%
180 \class@info{#1 was not loaded (OK!)}%
181 }%
182 }%
183 \def\class@warn@end#1{%
184 \gapdef\class@enddocumenthook{\class@warn{#1}}%
185 }%
186 \AtEndOfClass{%
187 \@ifxundefined\class@name{\def\class@name{Generic Class}}{}}%
188 }%

```

6.4 New Tools

```

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

\dimen@iii
190 \dimendef\dimen@iii\thr@@

\halign@
191 \def\halignt@{\halign\t@}%

\four Analogous to \@ne, \tw@, and \thr@@.
192 \chardef\four=4\relax
193 \chardef\cat@letter=11\relax
194 \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.
195 \def\let@environment#1#2{%
196 \expandafter\let
197 \csname#1\expandafter\endcsname\csname#2\endcsname
198 \expandafter\let
199 \csname end#1\expandafter\endcsname\csname end#2\endcsname
200 }%
201 \def\glet@environment#1#2{%
202 \global\expandafter\let
203 \csname#1\expandafter\endcsname\csname#2\endcsname
204 \global\expandafter\let
205 \csname end#1\expandafter\endcsname\csname end#2\endcsname
206 }%

```

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

```
207 \newcommand\tracingplain{%
208 \tracingonline\z@\tracingcommands\z@\tracingstats\z@
209 \tracingpages\z@\tracingoutput\z@\tracinglostchars\@ne
210 \tracingmacros\z@\tracingparagraphs\z@\tracingrestores\z@
211 \showboxbreadth5\showboxdepth3\relax %\errorstopmode
212 }%
213 \newcommand\traceoutput{%
214 \appdef\@resetactivechars{\showoutput}%
215 }%
```

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

```
216 \newcommand\say[1]{\typeout{<\noexpand#1=\meaning#1>}}%
217 \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.

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

`\count@i`
`\count@ii`

```
219 \countdef\count@i\@ne
220 \countdef\count@ii\tw@
```

6.5 Boolean Control

We introduce just enough of the Boolean calculus for \TeX . Alan Jeffrey was the pioneer here, with an article in TUGboat (Vol. 11, No. 2, page 237). This implementation owes a debt to William Baxter (web@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 `\prepdef` prepends to the given macro the tokens specified in its second argument. 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.

```
221 \long\def\prepdef#1#2{%
222 \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
223 \toks@ii{#2}%
224 \edef#1{\the\toks@ii\the\toks@}%
225 }%
226 \long\def\appdef#1#2{%
227 \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
228 \toks@ii{#2}%
229 \edef#1{\the\toks@\the\toks@ii}%
230 }%
231 \long\def\gappdef#1#2{%
232 \@ifxundefined#1{\toks@{}}{\toks@\expandafter{#1}}%
233 \toks@ii{#2}%
```

```

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

```

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

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

`\@argswap@val`

```

250 \long\def\@ifxundefined#1{\@ifx{\undefined#1}}%
251 \long\def\@ifnotrelax#1#2#3{\@ifx{\relax#1}{#3}{#2}}%
252 \long\def\@argswap#1#2{#2#1}%
253 \long\def\@argswap@val#1#2{#2{#1}}%
254 \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.

`\@boole@def`

```

255 \def\@boolean#1#2{%
256 \long\def#1{%
257 #2% \if<something>
258 \expandafter\true@sw
259 \else
260 \expandafter\false@sw
261 \fi
262 }%
263 }%
264 \def\@boole@def#1#\@boolean{#1}% Implicit #2

```

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

`\@booleanfalse`

```

265 \def\@booleantrue#1{\let#1\true@sw}%
266 \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: `\@ifcat`, `\@ifempty`, `\@xifempty`, and `\@ifnotempty`.

```

\@ifdim
\@ifeof
\@ifhbox
\@ifhmode
\@ifinner
\@ifmmode
\@ifnum
\@ifodd
\@ifvbox
\@ifvmode
\@ifvoid

```

267 \@boole@def\@ifx#1{\ifx#1}%
268 \@boole@def\@ifx@empty#1{\ifx\@empty#1}%

```

269 \@boole@def\@if@empty#1{\if!#1!}%
270 %\@boole@def\@if@sw#1{\csname if#1\endcsname}%
271 \def\@if@sw#1#2{#1\expandafter\true@sw\else\expandafter>false@sw#2}%
272 \@boole@def\@ifdim#1{\ifdim#1}%
273 \@boole@def\@ifeof#1{\ifeof#1}%
274 \@boole@def\@ifhbox#1{\ifhbox#1}%
275 \@boole@def\@ifhmode{\ifhmode}%
276 \@boole@def\@ifinner{\ifinner}%
277 \@boole@def\@ifmmode{\ifmmode}%
278 \@boole@def\@ifnum#1{\ifnum#1}%
279 \@boole@def\@ifodd#1{\ifodd#1}%
280 \@boole@def\@ifvbox#1{\ifvbox#1}%
281 \@boole@def\@ifvmode{\ifvmode}%
282 \@boole@def\@ifvoid#1{\ifvoid#1}%

```

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

```

283 \long\def\true@sw#1#2{#1}%
284 \long\def>false@sw#1#2{#2}%

```

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

```

285 \long\def\loopuntil#1{#1}{\loopuntil{#1}}%
286 \long\def\loopwhile#1{#1{\loopwhile{#1}}}%

```

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

```

287 \def\@provide#1{%
288 \@ifx{\undefined#1}{\true@sw}{\@ifx{\relax#1}{\true@sw}{\false@sw}}%
289 {\def#1}{\def\junk}%
290 }%

```

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`.

The price we pay for this facility is to depend 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`.

```

291 \prepdef\document{%
292 \endgroup

```

```

293 \init@documenthook
294 \set@typesize@hook
295 \normalsize
296 \set@pica@hook
297 \true@sw{ }%
298 }%

```

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

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

```

299 \def\init@documenthook{ }%
300 \AtBeginDocument{ %
301   \class@documenthook
302 }%
303 \AtEndDocument{ %
304   \class@enddocumenthook
305 }%
306 \def\class@documenthook{ }%
307 \def\class@enddocumenthook{ }%

```

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

```

308 \def\set@typesize@hook{ }%
309 \def\set@pica@hook{ }%

```

`\enddocument` The standard L^AT_EX `\end{document}` processing is a potential problem, particularly
`\check@aux` when the output routine has been changed by `ltxgrid`. We separate out the procedure
`\do@check@aux` 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.

```

310 \def\enddocument{ %
311   \@enddocumenthook
312   \@checkend{document}%
313   \clear@document
314   \check@aux
315   \deadcycles\z@
316   \@@end
317 }%
318 \def\clear@document{\clearpage}%
319 \def\check@aux{\do@check@aux}%
320 \def\do@check@aux{ %
321   \@if@sw\if@filesw\fi{ %
322     \immediate\closeout\@mainaux
323     \let\@setckpt@gobbletwo
324     \let\@newl@bel\@testdef
325     \@tempswafalse
326     \makeatletter
327     \input\jobname.aux\relax

```

```

328 }}}%
329 \@dofilelist
330 \@ifdim{\font@submax >\fontsubfuzz\relax}{%
331   \@font@warning{%
332     Size substitutions with differences\MessageBreak
333     up to \font@submax\space have occured.\@gobbletwo
334   }%
335 }}}%
336 \@defaultsubs
337 \@refundefined
338 \@if@sw@if@filesw\fi{%
339   \@ifx{\@multiplelabels\relax}{%
340     \@if@sw@if@tempswa\fi{%
341       \@latex@warning@no@line{%
342         Label(s) may have changed.
343         Rerun to get cross-references right
344       }%
345     }}}%
346   }{%
347     \@multiplelabels
348   }%
349 }}}%
350 }%

```

6.7 Type Tools

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

```

351 \def\flushing{%
352   \let\\\@normalcr
353   \leftskip\z@skip
354   \rightskip\z@skip
355   \@rightskip\z@skip
356   \parfillskip\@flushglue
357 }%

```

6.8 Display Math

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

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

```

358 \def\eqnarray@LaTeX{%
359   \stepcounter{equation}%
360   \def\@currentlabel{\p@equation\theequation}%
361   \global\@eqnswtrue
362   \m@th
363   \global\@eqcnt\z@
364   \tabskip\@centering
365   \let\\\@eqnocr
366   $$\everycr{\halign\t@displaywidth\bgroup
367     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1

```

```

368 &\global\@eqcnt\@ne\hskip \tw@\arraycolsep \hfil$\{##}\$ \hfil
369 &\global\@eqcnt\tw@ \hskip \tw@\arraycolsep
370 $\displaystyle{\##}\$ \hfil\tabskip\@centering
371 &\global\@eqcnt\thr@@ \hbxt@\z@\bgroup\hss##\egroup
372 \tabskip\z@skip
373 \cr
374 }
375 \long\def\eqnarray@fleqn@fixed{%
376 \stepcounter{equation}\def\@currentlabel{\p@equation\theequation}%
377 \global\@eqnswtrue\m@th\global\@eqcnt\z@
378 \tabskip\mathindent
379 \let\@=\@eqncr
380 \setlength\abovedisplayskip{\topsep}%
381 \ifvmode\addtolength\abovedisplayskip{\partopsep}\fi
382 \addtolength\abovedisplayskip{\parskip}%
383 \setlength\belowdisplayskip{\abovedisplayskip}%
384 \setlength\belowdisplayskip{\abovedisplayskip}%
385 \setlength\abovedisplayskip{\abovedisplayskip}%
386 $$%
387 \everycr{}%
388 \halign\@linewidth\bgroup
389 \hskip\@centering$\displaystyle\tabskip\z@skip{\##}\$ \@eqnsele
390 &\global\@eqcnt\@ne
391 \hskip\tw@\eqncolsep
392 \hfil$\{\}\{##\}\$ \hfil
393 &\global\@eqcnt\tw@
394 \hskip\tw@\eqncolsep
395 $\displaystyle{\##}\$ \hfil\tabskip\@centering
396 &\global\@eqcnt\thr@@\hbxt@\z@\bgroup\hss##\egroup
397 \tabskip\z@skip
398 \cr
399 }%
400 \@ifx{\eqnarray\eqnarray@LaTeX}{%
401 \class@info{Repairing broken LaTeX eqnarray}%
402 \let\eqnarray\eqnarray@fleqn@fixed
403 \newlength\eqncolsep
404 \setlength\eqncolsep\z@
405 \let\eqnarray@LaTeX\relax
406 \let\eqnarray@fleqn@fixed\relax
407 }{}%
408 \def\mathindent{\@centering}%
409 \def\set@eqnarray@skips{}%

```

6.9 Footnotes

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

While we are at it, we rewrite both the `\footnote` and `\footnotemark` pro-

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

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

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

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

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

```
410 \def\footnote{%
411   \@ifnextchar[\@xfootnote{\@yfootnote\@footnotetext}%
412 }%
413 \def\footnotemark{%
414   \@ifnextchar[\@xfootnotemark{\@yfootnote}%
415 }%
416 \def\@xfootnote[#1]{%
417   \@xfootnotemark[#1]%
418   \@footnotetext
419 }%
420 \def\@xfootnotemark@ltx[#1]{%
421   \begingroup
422     \csname c@\@mpfn\endcsname #1\relax
423     \unrestored@protected@xdef\@thefnmark{\thempfn}%
424   \endgroup
425   \H@@footnotemark
426 }%
427 \def\@yfootnote{%
428   \stepcounter\@mpfn
429   \protected@xdef\@thefnmark{\thempfn}%
430   \H@@footnotemark
431 }%
```

Note on `hyperref` compatibility: In its “Automated \LaTeX hypertext cross-references”, the `hyperref` package alters footnote processing, thereby imperiling these fixes and necessitating defensive measures.

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

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

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

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

```
432 \appdef\class@documenthook{%
433 \@ifxundefined\H@@footnotemark{%
434 \let\H@@footnotemark\@footnotemark
435 }{}}%
436 \let\@xfootnotemark\@xfootnotemark@ltx
437 }%
```

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

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

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

```
\@footnotetext
\@mpfootnotetext
\@tpfootnotetext
\make@footnotetext
\set@footnotewidth
```

The two procedures `\@footnotetext` and `\@mpfootnotetext` share code. We make that explicit here.

Note that the procedure calling `\make@footnotetext` will open a group with `\bgroup` which is then closed by `\minipagefootnote@drop`.

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

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

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

```
438 \long\def\@footnotetext{%
439 \insert\footins\bgroup
440 \make@footnotetext
441 }%
442 \long\def\@mpfootnotetext{%
443 \minipagefootnote@pick
444 \make@footnotetext
445 }%
446 \def\make@footnotetext#1{%
447 \reset@font\footnotesize
448 \interlinepenalty\interfootnotelinepenalty
449 \splittopskip\footnotesep
450 \splitmaxdepth\dp\strutbox
451 % \floatingpenalty\@MM
452 \set@footnotewidth
453 \@parboxrestore
```

```

454 \protected@edef\@currentlabel{%
455   \csname p@footnote\endcsname\@thefnmark
456 }%
457 \color@begingroup
458   \@makefnmark
459   \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox
460 }%
461 \color@endgroup
462 \minipagefootnote@drop
463 }%
464 \def\set@footnotewidth{%
465   \hsize\columnwidth
466   \linewidth\hsize
467 }%
468 \def\set@footnotewidth@ii{%
469   \hsize\textwidth
470   \advance\hsize\columnsep
471   \divide\hsize\tw@
472   \advance\hsize-\columnsep
473   \linewidth\hsize
474 }%

```

6.10 Floats

6.10.1 Usage notes

We extend the \LaTeX kernel for three purposes:

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

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

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

```

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

```

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

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

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

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

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

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

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

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

6.10.2 Robustifying fragile commands

Certain of \LaTeX '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
475 \def\addtocontents#1#2{%
476 \protected@write\@auxout{%
477 \let \label \@gobble \let \index \@gobble \let \glossary \@gobble
478 \def\(\{\string\}%
479 \def\)\{\string\}%
480 \def\{\{\string\}\}%
481 }\{\string \@writefile {#1}{#2}}%
482 }%
```

6.10.3 Preparing for the hyperref package

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

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

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

```
483 \def\addcontentsline#1#2#3{%
484 \addtocontents{#1}{%
```

```

485 \protect\contentsline{#2}{#3}{\thepage}{}%
486 }%
487 }%
488 \def\contentsline#1#2#3#4{%
489 \csname l@#1\endcsname{#2}{#3}%
490 }%
491 \def\label#1{%
492 \@bsphack
493 \protected@write\@auxout{}{%
494 \string\newlabel{#1}{\@currentlabel}{\thepage}{}{}{}}%
495 }%
496 \@esphack
497 }%

```

6.10.4 Footnotes within floats, unfloating floats, float font

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

```

498 \appdef\class@documenthook{%
499 \prepdef\caption{\minipagefootnote@here}%
500 }%

```

Note on `hyperref` compatability: this change to the `\caption` command is compatible with the “Automated \LaTeX 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. These procedures encapsulate all uses of the `\@mpfootins` box.

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

```

\minipagefootnote@foot
\minipagefootnote@pick
\minipagefootnote@drop
501 \def\minipagefootnote@init{%
502 \setbox\@mpfootins\box\voidb@x
503 }%
504 \def\minipagefootnote@pick{%
505 \global\setbox\@mpfootins\vbox\bgroup
506 \unvbox\@mpfootins

```

```

507 }%
508 \def\minipagefootnote@drop{%
509 \egroup
510 }%
511 \def\minipagefootnote@here{%
512   \par
513   \@ifvoid\@mpfootins{}{%
514     \vskip\skip\@mpfootins
515     \fullinterlineskip
516     \@ifinner{%
517       \vtop{\unvcopy\@mpfootins}%
518       {\setbox\z@\lastbox}%
519     }{}%
520     \unvbox\@mpfootins
521   }%
522 }%
523 \def\minipagefootnote@foot{%
524   \@ifvoid\@mpfootins{}{%
525     \insert\footins\bgroup\unvbox\@mpfootins\egroup
526   }%
527 }%
528 \def\endminipage{%
529   \par
530   \unskip
531   \minipagefootnote@here
532   \@minipagefalse %% added 24 May 89
533   \color@endgroup
534   \egroup
535   \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}%
536 }%

```

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

```

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

```

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

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

```

537 \let\@xfloat@LaTeX\@xfloat
538 \def\@xfloat#1[#2]{%
539   \@xfloat@prep
540   \@nameuse{fp@proc@#2}%
541   \@ifxundefined\floats@sw{\global\@booleantrue\floats@sw}{}%
542   \floats@sw{\@xfloat@LaTeX{#1}[#2]}\@xfloat@anchored{#1}[]%
543 }%
544 \def\@xfloat@prep{%

```

```

545 \let\footnote\footnote@latex
546 \def\@mpfn{mpfootnote}%
547 \def\thempfn{\thempfootnote}%
548 % \def\thefootnote{\thempfootnote}%
549 \c@mpfootnote\z@
550 \let\@footnotetext\@mpfootnotetext
551 \let\H@@footnotetext\@mpfootnotetext
552 \let\@makefnmark\@mpmakefnmark
553 % \samepage
554 }%
555 \appdef\class@documenthook{%
556 \let\footnote@latex\footnote
557 }%
558 %\def\fp@proc@h{\@booleanfalse\floats@sw}%
559 %\def\fp@proc@H{\@booleanfalse\floats@sw}%
560 \def\@xfloat@anchored#1[#2]{%
561 \def\@capttype{#1}%
562 \begin@float@pagebreak
563 %\vbox\bgroup
564 \let\end@float\end@float@anchored
565 \let\end@dblfloat\end@float@anchored
566 % do as \@xfloat does:
567 \hsize\columnwidth
568 \@parboxrestore
569 \@floatboxreset
570 \minipagefootnote@init
571 % \pagegrid@col\@ne % Klotch to avoid processing as a float
572 }%
573 \def\end@float@anchored{%
574 \minipagefootnote@here
575 \par\vskip\z@skip %% \par\vskip\z@ added 15 Dec 87
576 %\egroup
577 \par
578 \end@float@pagebreak
579 }%
580 \def\begin@float@pagebreak{\par\addvspace\intextsep}%
581 \def\end@float@pagebreak{\par\addvspace\intextsep}%
582 \def\@mpmakefnmark#1{%
583 \parindent=1em
584 \noindent
585 \hb@xt@1em{\hss\@makefnmark}%
586 #1%
587 }%

```

6.10.5 Writing floats out to a file

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

```

588 \def\do@if@floats#1#2{%
589 \@ifxundefined\floats@sw{\global\@booleantrue\floats@sw}{}%
590 \floats@sw}{}%

```

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

```

591 \expandafter\newwrite
592         \csname#1write\endcsname
593 \expandafter\def
594         \csname#1@stream\endcsname{\jobname#2}%
595 \expandafter\immediate
596 \expandafter\openout
597         \csname#1write\endcsname
598         \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`.

```

599 \@ifundefined\@float@LaTeX{%
600   \let\@float@LaTeX\@float
601   \let\@dblfloat@LaTeX\@dblfloat
602   \let\@float\write@float
603   \let\@dblfloat\write@floats
604 }{%
605 \let@environment{#1@float}{#1}%
606 \let@environment{#1@floats}{#1*}%
607 \@ifundefined@cs{#1@write}{}{%
608   \let@environment{#1}{#1@write}%
609 }%
610 }%
611 }%

```

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

```

612 \def\triggerpar{\leavevmode@@par}%
613 \def\onepage{\def\begin@float@pagebreak{\newpage}\def\end@float@pagebreak{\newpage}}
614 \def\print@float#1#2{%
615   \@ifundefined@cs{#1write}{}{%
616     \begingroup
617       \@booleanfalse\floats@sw
618       #2%
619       \raggedbottom
620       \def\array@default{v}% floats must
621       \let\@float\@float@LaTeX
622       \let\@dblfloat\@dblfloat@LaTeX
623       \let\trigger@float@par\triggerpar
624       \let@environment{#1}{#1@float}%
625       \let@environment{#1*}{#1@floats}%
626       \expandafter\prepdef\csname#1\endcsname{\trigger@float@par}%
627       \expandafter\prepdef\csname#1*\endcsname{\trigger@float@par}%
628       \@namedef{fps@#1}{h!}%
629       \expandafter\immediate
630       \expandafter\closeout
631         \csname#1write\endcsname
632     \everypar{%
633       \global\let\trigger@float@par\relax
634       \global\everypar{}\setbox\z@\lastbox
635       \@ifundefined@cs{#1sname}{}{%
636         \begin@float@pagebreak

```

```

637     \expandafter\section
638     \expandafter*%
639     \expandafter{%
640         \csname#1sname\endcsname
641     }%
642 }%
643 }%
644 \input{\csname#1stream\endcsname}%
645 \endgroup
646 \global\expandafter\let\csname#1write\endcsname\relax
647 }%
648 }%

```

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

```

\write@floats
\write@@float
649 \def\write@float#1{\write@@float{#1}{#1}}%
650 \def\endwrite@float{\@Esphack}%
651 \def\write@floats#1{\write@@float{#1*}{#1}}%
652 \def\endwrite@floats{\@Esphack}%

```

`\write@@float`

```

653 \def\write@@float#1#2{%
654     \ifhmode
655         \@bsphack
656         \fi
657     \chardef\@tempc\csname#2write\endcsname
658     \toks@\{\begin{#1}\}%
659     \def\@tempb{#1}%
660     \expandafter\let\csname end#1\endcsname\endwrite@float
661     \catcode'\^^M\active
662     \@makeother\{\@makeother\}\@makeother\%
663     \write@floatline
664 }%

```

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

```

\@write@floatline
\float@end@tag
665 \begingroup
666 \catcode'\[\the\catcode'\{\catcode'\}\the\catcode'\}\@makeother\{\@makeother\}%
667 \gdef\float@end@tag#1\end{#2}#3@nul[%
668     \def\@tempa[#2]%
669     \@ifx[\@tempa\@tempb][\end{#2}][\write@floatline]%
670 ]%
671 \obeylines%
672 \gdef\write@floatline#1^^M[%
673     \begingroup%
674     \newlinechar'\^^M%
675     \toks@\expandafter[\the\toks@#1]\immediate\write\@tempc[\the\toks@]%
676     \endgroup%
677     \toks@[ ]%
678     \float@end@tag#1\end{\}\@nul%
679 ]%
680 \endgroup

```

6.11 Counters

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

```
681 \def\@alph#1{\ifcase#1\or a\or b\or c\or d\else\@ialph{#1}\fi}
682 \def\@ialph#1{\ifcase#1\or \or \or \or \or e\or f\or g\or h\or i\or j\or
683 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
684 y\or z\or aa\or bb\or cc\or dd\or ee\or ff\or gg\or hh\or ii\or jj\or
685 kk\or ll\or mm\or nn\or oo\or pp\or qq\or rr\or ss\or tt\or uu\or
686 vv\or ww\or xx\or yy\or zz\else\@ctrerr\fi}
```

6.12 Customization of Sections

Patch the standard L^AT_EX sectioning procedure to:

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

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

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

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

```
687 \def\@startsection#1#2#3#4#5#6{%
688 \@startsection@hook
689 \if@noskipsec \leavevmode \fi
690 \par
691 \@tempskipa #4\relax
692 \@afterindenttrue
693 \ifdim \@tempskipa <\z@
694 \@tempskipa -\@tempskipa \@afterindentfalse
695 \fi
696 \if@nobreak
697 \everypar{}%
698 \else
699 \addpenalty\@secpenalty\addvspace\@tempskipa
700 \fi
701 \@ifstar
702 {\@dblarg{\@ssect@ltx{#1}{#2}{#3}{#4}{#5}{#6}}}%
703 {\@dblarg{\@sect@ltx {#1}{#2}{#3}{#4}{#5}{#6}}}%
```

```
704 }%
705 \def\@startsection@hook{ }%
```

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

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

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

```
706 \class@info
707 {Repairing broken LaTeX \string\@sect}%
708 \def\@sect@ltx#1#2#3#4#5#6[#7]#8{%
709   \@ifnum{#2>\c@secnumdepth}{%
710     \def\H@svsec{\phantomsection}%
711     \let\@svsec\@empty
712   }{%
713     \H@refstepcounter{#1}%
714     \def\H@svsec{%
715       \phantomsection
716     }%
717     \protected@edef\@svsec{{#1}}%
718     \@ifundefined{@#1cntformat}{%
719       \prepdef\@svsec\@secntformat
720     }{%
721       \expandafter\prepdef
722       \expandafter\@svsec
723         \csname @#1cntformat\endcsname
724     }%
725   }%
726   \@tempskipa #5\relax
727   \@ifdim{\@tempskipa>z@}{%
728     \begingroup
729     \interlinepenalty \@M
730     #6{%
731       \@ifundefined{@hangfrom@#1}{\@hang@from}{\csname @hangfrom@#1\endcsname}%
732       {\hskip#3\relax\H@svsec}{\@svsec}{#8}%
733     }%
734     \@@par
735     \endgroup
736     \@ifundefined{#1mark}{\@gobble}{\csname #1mark\endcsname}{#7}%
737     \addcontentsline{toc}{#1}{%
738       \@ifnum{#2>\c@secnumdepth}{%
739         \protect\numberline{ }%
740       }{%
741         \protect\numberline{\csname the#1\endcsname}%
742       }%
743       #8}%
744   }{%
745     \def\@svsechd{%
```

```

746 #6{%
747 \ifundefined{@runin@to@#1}{\@runin@to}{\csname @runin@to@#1\endcsname}%
748 {\hskip#3\relax\H@svsec}{\@svsec}{#8}%
749 }%
750 \@ifundefined{#1mark}{\@gobble}{\csname #1mark\endcsname}{#7}%
751 \addcontentsline{toc}{#1}{%
752 \@ifnum{#2>\c@secnumdepth}{%
753 \protect\numberline}{%
754 }{%
755 \protect\numberline{\csname the#1\endcsname}%
756 }%
757 #8}%
758 }%
759 }%
760 \@xsect{#5}%
761 }%
762 \def\@hang@from#1#2#3{\@hangfrom{#1#2}#3}%
763 \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`.

```

764 \def\@ssect@ltx#1#2#3#4#5#6[#7]#8{%
765 % \def\@currentlabelname{#8}%
766 \def\H@svsec{\phantomsection}%
767 \@tempskipa #5\relax
768 \@ifdim{\@tempskipa>\z@}{%
769 \begingroup
770 \interlinepenalty \@M
771 #6{%
772 \ifundefined{@hangfroms@#1}{\@hang@froms}{\csname @hangfroms@#1\endcsname}%
773 % {\hskip#3\relax\H@svsec}{\Sectionformat{#8}{#1}}%
774 {\hskip#3\relax\H@svsec}{#8}%
775 }%
776 \@@par
777 \endgroup
778 \@ifundefined{#1smark}{\@gobble}{\csname #1smark\endcsname}{#7}%
779 \addcontentsline{toc}{#1}{\protect\numberline{}}#8}%
780 }{%
781 \def\@svsechd{%
782 #6{%
783 \ifundefined{@runin@tos@#1}{\@runin@tos}{\csname @runin@tos@#1\endcsname}%
784 % {\hskip#3\relax\H@svsec}{\Sectionformat{#8}{#1}}%
785 {\hskip#3\relax\H@svsec}{#8}%
786 }%
787 \@ifundefined{#1smark}{\@gobble}{\csname #1smark\endcsname}{#7}%
788 \addcontentsline{toc}{#1}{\protect\numberline{}}#8}%

```

```

789     }%
790 }%
791 \@xsect{#5}%
792 }%
793 \def\@hang@froms#1#2{#1#2}%
794 \def\@runin@tos #1#2{#1#2}%

```

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

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

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

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

```

795 \appdef\init@documenthook{%
796   \providecommand\phantomsection{}%
797   \@ifx{\Sectionformat\@undefined}{\let\Sectionformat\@firstoftwo}{}%
798   \providecommand\hyper@anchor[1]{}%
799   \providecommand\hyper@last{}%
800   \providecommand\Hy@raisedlink[1]{#1}%
801   \providecommand\hyper@anchorstart[1]{}%
802   \providecommand\hyper@anchorend{}%
803   \providecommand\hyper@linkstart[2]{}%
804   \providecommand\hyper@linkend{}%
805 }%
806 \let\H@refstepcounter\refstepcounter

```

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

The following setting gives the \LaTeX default.

```

807 \def\sec@upcase#1{\relax{#1}}%

```

6.13 Patch the `tabular` and `array` Environments

`\endtabular` We headpatch the begin processing and tailpatch the end processing of the `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 \LaTeX `array` and `tabular` environments. I know of none that do so.

```

808 \appdef\class@documenthook{%

```

```

809 \@ifpackageloaded{array}{\switch@array}{\switch@tabular}%
810 \prepdef\endtabular{\endtabular@hook}%
811 \@provide\endtabular@hook{%
812 \prepdef\endarray{\endarray@hook}%
813 \@provide\endarray@hook{%
814 \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.

```

815 \prepdef\@tabular{\tabular@hook}%
816 \@provide\@tabular@hook{%
817 }%

```

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

```

818 \def\switch@tabular{%
819 \let\@array@sw\@array@sw@array
820 \@ifx{\@array\@array@LaTeX}{%
821 \@ifx{\multicolumn\multicolumn@LaTeX}{%
822 \@ifx{\@tabular\@tabular@LaTeX}{%
823 \@ifx{\@tabarray\@tabarray@LaTeX}{%
824 \@ifx{\array\array@LaTeX}{%
825 \@ifx{\endarray\endarray@LaTeX}{%
826 \@ifx{\endtabular\endtabular@LaTeX}{%
827 \@ifx{\@mkpream\@mkpream@LaTeX}{%
828 \@ifx{\@addamp\@addamp@LaTeX}{%
829 \@ifx{\@arrayacol\@arrayacol@LaTeX}{%
830 \@ifx{\@tabacol\@tabacol@LaTeX}{%
831 \@ifx{\@arrayclassz\@arrayclassz@LaTeX}{%
832 \@ifx{\@tabclassiv\@tabclassiv@LaTeX}{%
833 \@ifx{\@arrayclassiv\@arrayclassiv@LaTeX}{%
834 \@ifx{\@tabclassz\@tabclassz@LaTeX}{%
835 \@ifx{\@classv\@classv@LaTeX}{%
836 \@ifx{\hline\hline@LaTeX}{%
837 \@ifx{\@tabularcr\@tabularcr@LaTeX}{%
838 \@ifx{\@xtabularcr\@xtabularcr@LaTeX}{%
839 \@ifx{\@xargarraycr\@xargarraycr@LaTeX}{%
840 \@ifx{\@yargarraycr\@yargarraycr@LaTeX}{%
841 \true@sw
842 }{%
843 \false@sw
844 }%
845 }{%
846 \false@sw
847 }%
848 }{%
849 \false@sw
850 }%
851 }{%
852 \false@sw
853 }%
854 }{%
855 \false@sw

```

```

856         }%
857     }{%
858         \false@sw
859     }%
860 }{%
861     \false@sw
862 }%
863 }{%
864     \false@sw
865 }%
866 }{%
867     \false@sw
868 }%
869 }{%
870     \false@sw
871 }%
872 }{%
873     \false@sw
874 }%
875 }{%
876     \false@sw
877 }%
878 }{%
879     \false@sw
880 }%
881 }{%
882     \false@sw
883 }%
884 }{%
885     \false@sw
886 }%
887 }{%
888     \false@sw
889 }%
890 }{%
891     \false@sw
892 }%
893 }{%
894     \false@sw
895 }%
896 }{%
897     \false@sw
898 }%
899 }{%
900     \false@sw
901 }%
902 }{%
903     \false@sw
904 }%
905 {%
906     \class@info{Patching LaTeX tabular.}%
907 }{%
908     \class@info{Unrecognized LaTeX tabular. Please update this document class! (Proc
909 }%

```

```

910 \let\@array\@array@ltx
911 \let\multicolumn\multicolumn@ltx
912 \let\@tabular\@tabular@ltx
913 \let\@tabarray\@tabarray@ltx
914 \let\array\array@ltx
915 \let\endarray\endarray@ltx
916 \let\endtabular\endtabular@ltx
917 \let\@mkpream\@mkpream@ltx
918 \let\@addamp\@addamp@ltx
919 \let\@arrayacol\@arrayacol@ltx
920 \let\@tabacol\@tabacol@ltx
921 \let\@arrayclassz\@arrayclassz@ltx
922 \let\@tabclassiv\@tabclassiv@ltx
923 \let\@arrayclassiv\@arrayclassiv@ltx
924 \let\@tabclassz\@tabclassz@ltx
925 \let\@classv\@classv@ltx
926 \let\hline\hline@ltx
927 \let\@tabularcr\@tabularcr@ltx
928 \let\@xtabularcr\@xtabularcr@ltx
929 \let\@xargarraycr\@xargarraycr@ltx
930 \let\@yargarraycr\@yargarraycr@ltx
931 }%
932 \def\switch@array{%
933 \let\@array@sw\@array@sw@LaTeX
934 \@ifx{\@array\@array@array}{%
935 \@ifx{\@tabular\@tabular@array}{%
936 \@ifx{\@tabarray\@tabarray@array}{%
937 \@ifx{\array\array@array}{%
938 \@ifx{\endarray\endarray@array}{%
939 \@ifx{\endtabular\endtabular@array}{%
940 \@ifx{\@mkpream\@mkpream@array}{%
941 \@ifx{\@classx\@classx@array}{%
942 \@ifx{\insert@column\insert@column@array}{%
943 \@ifx{\@arraycr\@arraycr@array}{%
944 \@ifx{\@xarraycr\@xarraycr@array}{%
945 \@ifx{\@xargarraycr\@xargarraycr@array}{%
946 \@ifx{\@yargarraycr\@yargarraycr@array}{%
947 \true@sw
948 }{%
949 \false@sw
950 }%
951 }{%
952 \false@sw
953 }%
954 }{%
955 \false@sw
956 }%
957 }{%
958 \false@sw
959 }%
960 }{%
961 \false@sw
962 }%
963 }{%

```

```

964         \false@sw
965     }%
966     }{%
967         \false@sw
968     }%
969     }{%
970         \false@sw
971     }%
972     }{%
973         \false@sw
974     }%
975     }{%
976         \false@sw
977     }%
978     }{%
979         \false@sw
980     }%
981     }{%
982         \false@sw
983     }%
984     }{%
985         \false@sw
986     }{%
987     \class@info{Patching array package.}%
988     }{%
989     \class@info{Unrecognized array package. Please update this document class! (Proc
990     }%
991     \let\@array      \@array@array@new
992     \let\@@array     \@array % Così fan tutti
993     \let\@tabular    \@tabular@array@new
994     \let\@tabarray   \@tabarray@array@new
995     \let\array       \array@array@new
996     \let\endarray    \endarray@array@new
997     \let\endtabular  \endtabular@array@new
998     \let\@mkpream    \@mkpream@array@new
999     \let\@classx     \@classx@array@new
1000    \let\@arrayacol  \@arrayacol@ltx
1001    \let\@tabacol    \@tabacol@ltx
1002    \let\insert@column\insert@column@array@new
1003    \expandafter\let\csname endtabular*\endcsname\endtabular % Così fan tutti
1004    \let\@arraycr    \@arraycr@new
1005    \let\@xarraycr   \@xarraycr@new
1006    \let\@xargarraycr\@xargarraycr@new
1007    \let\@yargarraycr\@yargarraycr@new
1008 }%

```

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

```

1009 \def\@array@sw@LaTeX{\@ifx{\@tabularcr}}%
1010 \def\@array@sw@array{\@ifx{\dollarbegin\beginngroup}}%

```

`\@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).

```

1011 \def\@tabular@LaTeX{%
1012 \leavevmode
1013 \hbox\bgroup$%
1014 \let\@acol\@tabacol
1015 \let\@classz\@tabclassz
1016 \let\@classiv\@tabclassiv
1017 \let\\\@tabularcr
1018 \@tabarray
1019 }%
1020 \def\@tabular@ltx{%
1021 \let\@acoll\@tabacoll
1022 \let\@acolr\@tabacolr
1023 \let\@acol\@tabacol
1024 \let\@classz\@tabclassz
1025 \let\@classiv\@tabclassiv
1026 \let\\\@tabularcr
1027 \@tabarray
1028 }%
1029 \def\@tabular@array{%
1030 \leavevmode
1031 \hbox\bgroup$%
1032 \col@sep\tabcolsep
1033 \let\dollarbegin\begingroup
1034 \let\dollarend\endgroup
1035 \@tabarray
1036 }%
1037 \def\@tabular@array@new{%
1038 \let\@acoll\@tabacoll
1039 \let\@acolr\@tabacolr
1040 \let\@acol\@tabacol
1041 \let\col@sep\@undefined
1042 \let\dollarbegin\begingroup
1043 \let\dollarend\endgroup
1044 \@tabarray
1045 }%

```

\@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.

```

1046 \def\@tabarray@LaTeX{%
1047 \m@th\@ifnextchar[\@array{\@array[c]}%
1048 }%
1049 \def\@tabarray@ltx{%
1050 \m@th\@ifnextchar[\@array{\expandafter\@array\expandafter[\@array@default]}%
1051 }%
1052 \def\@tabarray@array{%
1053 \@ifnextchar[{\@@array}{\@@array[c]}%
1054 }%
1055 \def\@tabarray@array@new{%
1056 \@ifnextchar[{\@@array}{\expandafter\@@array\expandafter[\@array@default]}%
1057 }%

```

\@tabularcr We provide for the \\ command within tabular to provide control over page breaking,

```

\@tbpen
\@tabularcr
\@xtabularcr
\@xargarraycr
\@yargarraycr
\@arraycr
\@xarraycr

```

just the same as that of `eqnarray`.

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.

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.

```

1058 \newcount\intertabularlinepenalty
1059 \intertabularlinepenalty=100
1060 \newcount\@tbpen
1061 \appdef\samepage{\intertabularlinepenalty\@M}%
1062 \def\@tabularcr@LaTeX{\ifnum 0='}\fi \ifstar \@xtabularcr \@xtabularcr}%
1063 \def\@tabularcr@ltx{\ifnum 0='}\fi \ifstar {\global \@tbpen \@M \@xtabularcr }{\
1064 \def\@xtabularcr@LaTeX{\ifnextchar [\@argtabularcr {\ifnum 0='{ \fi } \cr }}%
1065 \def\@xtabularcr@ltx{\ifnextchar [\@argtabularcr {\ifnum 0='{ \fi } \cr \noalign {\
1066 \def\@xargarraycr@LaTeX#1{\@tempdima #1\advance \@tempdima \dp \@arstrutbox \vrule \
1067 \def\@xargarraycr@ltx#1{\@tempdima #1\advance \@tempdima \dp \@arstrutbox \vrule \
1068 \def\@yargarraycr@LaTeX#1{\cr \noalign {\vskip #1}}%
1069 \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.

1070 \def\@arraycr@array{%
1071 \relax
1072 \iffalse{\fi\ifnum 0='}\fi
1073 \ifstar \@xarraycr \@xarraycr
1074 }%
1075 \def\@arraycr@new{%
1076 \relax
1077 \iffalse{\fi\ifnum 0='}\fi
1078 \ifstar {\global \@tbpen \@M \@xarraycr }{\global \@tbpen \intertabularlinepenal
1079 }%
1080 \def\@xarraycr@array{%
1081 \ifnextchar [%]
1082 \@argarraycr {\ifnum 0='{ \fi } \cr}%
1083 }%
1084 \def\@xarraycr@new{%
1085 \ifnextchar [%]
1086 \@argarraycr {\ifnum 0='{ \fi } \cr \noalign {\penalty \@tbpen }}%
1087 }%
1088 \def\@xargarraycr@array#1{%
1089 \unskip
1090 \@tempdima #1\advance \@tempdima \dp \@arstrutbox
1091 \vrule \@depth \@tempdima \@width \z@
1092 \cr

```

```

1093 }%
1094 \def\xargarraycr@new#1{%
1095 \unskip
1096 \@tempdima #1\advance\@tempdima \dp\@arstrutbox
1097 \vrule \@depth\@tempdima \@width\z@
1098 \cr
1099 \noalign {\penalty \@tbpen }%
1100 }%
1101 \def\yargarraycr@array#1{%
1102 \cr
1103 \noalign{\vskip #1}%
1104 }%
1105 \def\yargarraycr@new#1{%
1106 \cr
1107 \noalign{\penalty \@tbpen \vskip #1}%
1108 }%

```

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

```

1109 \def\array@LaTeX{%
1110 \let\@acol\@arrayacol
1111 \let\@classz\@arrayclassz
1112 \let\@classiv\@arrayclassiv
1113 \let\\\@arraycr
1114 \let\@halignto\@empty
1115 \@tabarray
1116 }%
1117 \def\array@ltx{%
1118 \@ifmmode{}\@badmath$}%
1119 \let\@acoll\@arrayacol
1120 \let\@acolr\@arrayacol
1121 \let\@acol\@arrayacol
1122 \let\@classz\@arrayclassz
1123 \let\@classiv\@arrayclassiv
1124 \let\\\@arraycr
1125 \let\@halignto\@empty
1126 \@tabarray
1127 }%
1128 \def\array@array{%
1129 \col@sep\arraycolsep
1130 \def\d@llarbegin{$}\let\d@llarend\d@llarbegin\gdef\@halignto{}%
1131 \@tabarray
1132 }
1133 \def\array@array@new{%
1134 \@ifmmode{}\@badmath$}%
1135 \let\@acoll\@arrayacol
1136 \let\@acolr\@arrayacol
1137 \let\@acol\@arrayacol
1138 \let\col@sep\@undefined
1139 \def\d@llarbegin{$}%
1140 \let\d@llarend\d@llarbegin
1141 \gdef\@halignto{}%

```

```

1142 \@tabarray
1143 }%

```

`\@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.

```

1144 \def\@array@LaTeX[#1]#2{%
1145   \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi\fi
1146   \bgroup
1147   \setbox\@arstrutbox\hbox{%
1148     \vrule \@height\arraystretch\ht\strutbox
1149           \@depth\arraystretch \dp\strutbox
1150           \@width\z@}%
1151   \@mkpream{#2}%
1152   \edef\@preamble{%
1153     \ialign \noexpand\@halignto
1154           \bgroup \@arstrut \@preamble \tabskip\z@skip \cr}%
1155   \let\@startpbox\@startpbox \let\@endpbox\@endpbox
1156   \let\@tabularnewline\%
1157   \let\@par\@empty
1158   \let\@sharp##%
1159   \set@typeset@protect
1160   \lineskip\z@skip\baselineskip\z@skip
1161   \ifhmode \@preamerr\z@ \@@par\fi
1162   \@preamble
1163 }%
1164 \def\@array@ltx[#1]#2{%
1165   \@nameuse{array@align@#1}%
1166   \set@arstrutbox
1167   \@mkpream{#2}%
1168   \prepdef\@preamble{%
1169     \tabskip\tabmid@skip
1170     \@arstrut
1171   }%
1172   \appdef\@preamble{%
1173     \tabskip\tabright@skip
1174     \cr
1175     \array@row@pre
1176   }%
1177 % \let\@startpbox\@startpbox
1178 % \let\@endpbox\@endpbox
1179 \let\@tabularnewline\%
1180 \let\@par\@empty
1181 \let\@sharp##%
1182 \set@typeset@protect
1183 \lineskip\z@skip\baselineskip\z@skip
1184 \tabskip\tableft@skip\relax
1185 \ifhmode \@preamerr\z@ \@@par\fi

```

```

1186 \everycr{ }%
1187 \expandafter\halign\expandafter\@halignto\expandafter\bgroup\@preamble
1188 }%
1189 %
1190 \def\set@arstrutbox{%
1191   \setbox\@arstrutbox\hbox{%
1192     \vrule \@height\arraystretch\ht\strutbox
1193           \@depth\arraystretch \dp\strutbox
1194           \@width\z@
1195   }%
1196 }%

```

\@array@array

```

1197 \def\@array@array[#1]#2{%
1198   \@tempdima \ht \strutbox
1199   \advance \@tempdima by\extrarowheight
1200   \setbox \@arstrutbox \hbox{\vrule
1201     \@height \arraystretch \@tempdima
1202     \@depth \arraystretch \dp \strutbox
1203     \@width \z@}%
1204   \begingroup
1205   \@mkpream{#2}%
1206   \xdef\@preamble{\noexpand \ialign \@halignto
1207     \bgroup \@arstrut \@preamble
1208     \tabskip \z@ \cr}%
1209   \endgroup
1210   \@arrayleft
1211   \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi \fi
1212   \bgroup
1213   \let \@sharp ##\let \protect \relax
1214   \lineskip \z@
1215   \baselineskip \z@
1216   \m@th
1217   \let\\\@arraycr \let\tabularnewline\\\let\par\@empty \@preamble
1218 }%
1219 \def\@array@array@new[#1]#2{%
1220   \@tempdima\ht\strutbox
1221   \advance\@tempdima by\extrarowheight
1222   \setbox\@arstrutbox\hbox{%
1223     \vrule \@height\arraystretch\@tempdima
1224           \@depth \arraystretch\dp\strutbox
1225           \@width \z@
1226   }%
1227   \begingroup
1228   \@mkpream{#2}%
1229   \xdef\@preamble{\@preamble}%
1230   \endgroup
1231   \prepdef\@preamble{%
1232     \tabskip\tabmid@skip
1233     \@arstrut
1234   }%
1235   \appdef\@preamble{%
1236     \tabskip\tabright@skip
1237   \cr

```

```

1238 \array@row@pre
1239 }%
1240 \@arrayleft
1241 \@nameuse{@array@align@#1}%
1242 \m@th
1243 \let\\\@arraycr
1244 \let\tabularnewline\\%
1245 \let\par\@empty
1246 \let\@sharp##%
1247 \set@typeset@protect
1248 \lineskip\z@\baselineskip\z@
1249 \tabskip\tableft@skip
1250 \everycr{}%
1251 \expandafter\halign\expandafter\@halignto\expandafter\bgroup\@preamble
1252 }%

```

`\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`.

```

1253 \def\endarray@LaTeX{%
1254 \crrc\egroup\egroup
1255 }%
1256 \def\endarray@ltx{%
1257 \crrc\array@row@pst\egroup\egroup
1258 }%
1259 \def\endarray@array{%
1260 \crrc \egroup \egroup \@arrayright \gdef\@preamble{}%
1261 }%
1262 \def\endarray@array@new{%
1263 \crrc\array@row@pst\egroup\egroup % Same as \endarray@ltx
1264 \@arrayright
1265 \global\let\@preamble\@empty
1266 }%

```

`\endtabular`

```

1267 \def\endtabular@LaTeX{%
1268 \crrc\egroup\egroup $\egroup
1269 }%
1270 \def\endtabular@ltx{%
1271 \endarray
1272 }%
1273 \def\endtabular@array{%
1274 \endarray $\egroup
1275 }%
1276 \def\endtabular@array@new{%
1277 \endarray
1278 }%

```

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

```

1279 \@namedef{endtabular*}{\endtabular}%

```

`\multicolumn`

```
1280 \long\def\multicolumn@LaTeX#1#2#3{%
1281 \multispan{#1}\begingroup
1282 \@mkpream{#2}%
1283 \def\@sharp{#3}\set@typeset@protect
1284 \let\@startpbox\@startpbox\let\@endpbox\@endpbox
1285 \@arstrut \@preamble\hbox{ }\endgroup\ignorespaces
1286 }%
1287 \long\def\multicolumn@ltx#1#2#3{%
1288 \multispan{#1}%
1289 \begingroup
1290 \@mkpream{#2}%
1291 \def\@sharp{#3}%
1292 \set@typeset@protect
1293 %\let\@startpbox\@startpbox\let\@endpbox\@endpbox
1294 \@arstrut
1295 \@preamble
1296 \hbox{ }%
1297 \endgroup
1298 \ignorespaces
1299 }%
```

`\array@align@` Here are the various procedures for the vertical alignment options. The change 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`.

`\array@default`

We establish here the default vertical alignment.

```
1300 \def\@array@align@t{\leavevmode\vtop\bgroup}%
1301 \def\@array@align@b{\leavevmode\vbox\bgroup}%
1302 \def\@array@align@c{\leavevmode\@ifmmode{\vcenter\bgroup}{\vcenter\bgroup\aftergroup}
1303 \def\@array@align@v{%
1304 \ifmmode{%
1305 \badmath
1306 \vcenter\bgroup
1307 }{%
1308 \@ifinner{%
1309 $\vcenter\bgroup\aftergroup$
1310 }{%
1311 \@@par\bgroup
1312 }%
1313 }%
1314 }%
1315 \def\array@default{c}%
```

`\array@row@pre`

`\array@row@pst`

`\array@row@rst`

The procedure `\array@row@rst` reestablishes a default context for an alignment, so that they can be nested. Any environment or procedure that alters the way 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.

```
1316 \def\array@row@rst{%
1317 \let\@array@align@v\@array@align@c
1318 }%
```

```

1319 \def\array@row@pre{ }%
1320 \def\array@row@pst{ }%

\toprule   Default definitions for \toprule, \colrule, \botrule
\colrule  1321 \newcommand\toprule{\tab@rule{\column@font}{\column@fil}{\frstrut}}%
\botrule  1322 \newcommand\colrule{\unskip\lrstrut\\\tab@rule{\body@font}{\frstrut}}%
          1323 \newcommand\botrule{\unskip\lrstrut\\\noalign{\hline@rule}}%

\hline

1324 \def\hline@LaTeX{%
1325   \noalign{\ifnum0='}\fi\hrule \@height \arrayrulewidth \futurelet
1326     \reserved@a\@xhline
1327 }%
1328 \def\hline@ltx{%
1329   \noalign{%
1330     \ifnum0='}\fi
1331     \hline@rule
1332     \futurelet\reserved@a\@xhline
1333   }% \noalign ended in \@xhline
1334 }%
1335 \def\@xhline@unneeded{%
1336   \say\reserved@a
1337   \ifx\reserved@a\hline
1338     \vskip\doublerulesep
1339     \vskip-\arrayrulewidth
1340   \fi
1341   \ifnum0='}\fi}%
1342 }%
1343 \def\tab@rule#1#2#3{%
1344   \crrc
1345   \noalign{%
1346     \hline@rule
1347     \gdef\@arstrut@hook{%
1348       \global\let\@arstrut@hook\@empty
1349     }%
1350   }%
1351   \gdef\cell@font{#1}%
1352   \gdef\cell@fil{#2}%
1353 }%
1354 }%
1355 \def\column@font{ }%
1356 \def\column@fil{ }%
1357 \def\body@font{ }%
1358 \def\cell@font{ }%
1359 \def\frstrut{ }%
1360 \def\lrstrut{ }%

```

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

```

1361 \def\@arstrut@hline{%
1362 \relax
1363 \@ifmmode{\copy}{\unhcopy}\@arstrutbox@hline
1364 \@arstrut@hook
1365 }%
1366 %
1367 \let\@arstrut@org\@arstrut
1368 \def\@arstrut@hook{%
1369 \global\let\@arstrut\@arstrut@org
1370 }%
1371 %
1372 \newbox\@arstrutbox@hline
1373 \appdef\set@arstrutbox{%
1374 \setbox\@arstrutbox@hline\hbox{%
1375 \setbox\z@\hbox{$0^{0}_{-}$}%
1376 \dimen@ht\z@\advance\dimen@\@arstrut@hline@clnc
1377 \@ifdim{\dimen@<\arraystretch\ht\strutbox}{\dimen@=\arraystretch\ht\strutbox}{
1378 \vrule \@height\dimen@
1379 \quad \@depth\arraystretch \dp\strutbox
1380 \quad \@width\z@
1381 }%
1382 }%
1383 %
1384 \def\hline@rule{%
1385 \hrule \@height \arrayrulewidth
1386 \global\let\@arstrut\@arstrut@hline
1387 }%
1388 \def\@arstrut@hline@clnc{2\p@}% % Klootch: magic number

```

`\tableleft@skip`

```

1389 \def\tableleft@skip{\z@skip}%
1390 \def\tabmid@skip{\z@skip}%\@flushglue
1391 \def\tabright@skip{\z@skip}%
1392 \def\tableleftsep{\tabcolsep}%
1393 \def\tabmidsep{\tabcolsep}%
1394 \def\tabrightsep{\tabcolsep}%
1395 \def\cell@fil{}%
1396 \def\pbox@hook{}%

```

`\@arstrut`

```

1397 \appdef\@arstrut{\@arstrut@hook}%
1398 \let\@arstrut@hook\@empty
1399 \def\@addtopreamble{\appdef\@preamble}%

```

`\@mkpream`

```

1400 \def\@mkpream@LaTeX#1{%
1401 \quad \@firststamptrue\@lastchclass6
1402 \quad \let\@preamble\@empty
1403 \quad \let\protect\@unexpandable@protect
1404 \quad \let\@sharp\relax

```

```

1405 \let\@startpbox\relax\let\@endpbox\relax
1406 \@expast{#1}%
1407 \expandafter\@tfor \expandafter
1408   \@nextchar \expandafter:\expandafter=\reserved@a\do
1409   {\@testpach\@nextchar
1410    \ifcase \@chclass \@classz \or \@classi \or \@classii \or \@classiii
1411     \or \@classiv \or \@classv \fi\@lastchclass\@chclass}%
1412 \ifcase \@lastchclass \@acol
1413   \or \or \@preamerr \@ne\or \@preamerr \tw@\or \or \@acol \fi
1414 }%
1415 \def\@mkpream@ltx#1{%
1416   \@firstamptrue
1417   \@lastchclass6
1418   \let\@preamble\@empty
1419   \let\protect\@unexpandable@protect
1420   \let\@sharp\relax
1421   %\let\@startpbox\relax\let\@endpbox\relax
1422   \@expast{#1}%
1423   \expandafter\@tfor \expandafter\@nextchar \expandafter:\expandafter=\reserved@a
1424   \do{%
1425     \expandafter\@testpach \expandafter{\@nextchar}%
1426     \ifcase \@chclass
1427       \@classz
1428       \or
1429       \@classi
1430       \or
1431       \@classii
1432       \or
1433       \@classiii
1434       \or
1435       \@classiv
1436       \or
1437       \@classv
1438       \fi
1439     \@lastchclass\@chclass
1440   }%
1441   \ifcase \@lastchclass
1442     \@acolr % right-hand column
1443   \or
1444   \or
1445     \@preamerr\@ne
1446   \or
1447     \@preamerr\tw@
1448   \or
1449   \or
1450     \@acolr % right-hand column
1451   \fi
1452 }%

```

\insert@column

```

1453 \def\insert@column@array{%
1454   \the@toks \the \@tempcnta
1455   \ignorespaces \@sharp \unskip
1456   \the@toks \the \count@ \relax

```

```

1457 }%
1458 \def\insert@column@array@new{%
1459 \the@toks\the\@tempcnta
1460 \array@row@rst\cell@font
1461 \ignorespaces\@sharp\unskip
1462 \the@toks\the\count@
1463 \relax
1464 }%

```

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

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

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

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

```

1465 \def\@mkpream@relax{%
1466 \let\tableleftsep\relax
1467 \let\tabmidsep\relax
1468 \let\tabrightsep\relax
1469 \let\array@row@rst\relax
1470 \let\cell@font\relax
1471 \let\@startpbox\relax
1472 }%

```

`\@mkpream`

```

1473 \def\@mkpream@array#1{%
1474 \gdef\@preamble{ }\@lastchclass 4 \@firststamptrue
1475 \let\@sharp\relax \let\@startpbox\relax \let\@endpbox\relax
1476 \@temptokena{#1}\@tempswatrue
1477 \@whilesw\if@tempswa\fi{\@tempswafalse\the\NC@list}%
1478 \count@\m@ne
1479 \let\the@toks\relax
1480 \prepnext@tok
1481 \expandafter \@tfor \expandafter \@nextchar
1482 \expandafter : \expandafter = \the \@temptokena \do
1483 { \@testpach
1484 \ifcase \@chclass \@classz \or \@classi \or \@classii
1485 \or \save@decl \or \or \@classv \or \@classvi
1486 \or \@classvii \or \@classviii
1487 \or \@classx
1488 \or \@classx \fi
1489 \@lastchclass\@chclass}%
1490 \ifcase \@lastchclass
1491 \@acol \or
1492 \or
1493 \@acol \or
1494 \@preamerr \thr@@ \or
1495 \@preamerr \tw@ \@addtopreamble\@sharp \or
1496 \or

```

```

1497 \else \@preamerr \@ne \fi
1498 \def\the@toks{\the\toks}%
1499 }%
1500 \def\@mkpream@array@new#1{%
1501 \gdef\@preamble{}%
1502 \@lastchclass\@f@ur
1503 \@firstamptrue
1504 \let\@sharp\relax
1505 \@mkpream@relax
1506 %\let\@startpbox\relax\let\@endpbox\relax
1507 \@temptokena{#1}\@tempswattrue
1508 \@whilesw\if@tempswa\fi{\@tempswafalse\the\NC@list}%
1509 \count@\m@ne
1510 \let\the@toks\relax
1511 \prepnext@tok
1512 \expandafter\@tfor\expandafter\@nextchar\expandafter:\expandafter=\the\@temptokena
1513 \do{%
1514 \@testpach
1515 \ifcase\@chclass
1516 \@classz
1517 \or
1518 \@classi
1519 \or
1520 \@classii
1521 \or
1522 \save@decl
1523 \or
1524 \or
1525 \@classv
1526 \or
1527 \@classvi
1528 \or
1529 \@classvii
1530 \or
1531 \@classviii
1532 \or
1533 \@classx
1534 \or
1535 \@classx
1536 \fi
1537 \@lastchclass\@chclass
1538 }%
1539 \ifcase\@lastchclass
1540 \@acolr % right-hand column
1541 \or
1542 \or
1543 \@acolr % right-hand column
1544 \or
1545 \@preamerr\thr@@
1546 \or
1547 \@preamerr\tw@\@addtopreamble\@sharp
1548 \or
1549 \or
1550 \else

```

```

1551 \@preamerr\@ne
1552 \fi
1553 \def\the@toks{\the\toks}%
1554 }%

\@addamp
1555 \def\@addamp@LaTeX{%
1556 \iffirstamp\@firstampfalse\else\edef\@preamble{\@preamble &}\fi
1557 }%
1558 \def\@addamp@ltx{%
1559 \iffirstamp\@firstampfalse\else\@addtopreamble{&}\fi
1560 }%

\@arrayacol
1561 \def\@arrayacol@LaTeX{%
1562 \edef\@preamble{\@preamble \hskip \arraycolsep}%
1563 }%
1564 \def\@arrayacol@ltx{%
1565 \@addtopreamble{\hskip\arraycolsep}%
1566 }%

\@tabacol
1567 \def\@tabacol1{%
1568 \@addtopreamble{\hskip\tableftsep\relax}%
1569 }%
1570 \def\@tabacol@LaTeX{%
1571 \edef\@preamble{\@preamble \hskip \tabcolsep}%
1572 }%
1573 \def\@tabacol@ltx{%
1574 \@addtopreamble{\hskip\tabmidsep\relax}%
1575 }%
1576 \def\@tabacolr{%
1577 \@addtopreamble{\hskip\tabrightsep\relax}%
1578 }%

\@arrayclassz
1579 \def\@arrayclassz@LaTeX{%
1580 \ifcase \@lastchclass \@acolampacol \or \@ampacol \or
1581 \or \or \@addamp \or
1582 \@acolampacol \or \@firstampfalse \@acol \fi
1583 \edef\@preamble{\@preamble
1584 \ifcase \@chnum
1585 \hfil$\relax\@sharp$\hfil \or $\relax\@sharp$\hfil
1586 \or \hfil$\relax\@sharp$\fi}%
1587 }%
1588 \def\@arrayclassz@ltx{%
1589 \ifcase\@lastchclass
1590 \@acolampacol
1591 \or
1592 \@ampacol
1593 \or
1594 \or
1595 \or
1596 \@addamp

```

```

1597 \or
1598 \@acolampacol
1599 \or
1600 \@firstampfalse\@acoll
1601 \fi
1602 \ifcase\@chnum
1603 \@addtopreamble{%
1604   \hfil\array@row@rst$\relax\@sharp$\hfil
1605 }%
1606 \or
1607 \@addtopreamble{%
1608   \array@row@rst$\relax\@sharp$\hfil
1609 }%
1610 \or
1611 \@addtopreamble{%
1612   \hfil\array@row@rst$\relax\@sharp$%
1613 }%
1614 \fi
1615 }%

```

\@tabclassz

```

1616 \def\@tabclassz@LaTeX{%
1617   \ifcase\@lastchclass
1618     \@acolampacol
1619   \or
1620     \@ampacol
1621   \or
1622   \or
1623   \or
1624     \@addamp
1625   \or
1626     \@acolampacol
1627   \or
1628     \@firstampfalse\@acol
1629   \fi
1630   \edef\@preamble{%
1631     \@preamble{%
1632       \ifcase\@chnum
1633         \hfil\ignorespaces\@sharp\unskip\hfil
1634       \or
1635         \hskip1sp\ignorespaces\@sharp\unskip\hfil
1636       \or
1637         \hfil\hskip1sp\ignorespaces\@sharp\unskip
1638       \fi}}%
1639 }%
1640 \def\@tabclassz@ltx{%
1641   \ifcase\@lastchclass
1642     \@acolampacol
1643   \or
1644     \@ampacol
1645   \or
1646   \or
1647   \or
1648     \@addamp

```

```

1649 \or
1650 \@acolampacol
1651 \or
1652 \@firstampfalse\@acoll
1653 \fi
1654 \ifcase\@chnum
1655 \@addtopreamble{%
1656   {\hfil\array@row@rst\cell@font\ignorespaces\@sharp\unskip\hfil}%
1657 }%
1658 \or
1659 \@addtopreamble{%
1660   {\cell@fil\hskip1sp\array@row@rst\cell@font\ignorespaces\@sharp\unskip\hfil}%
1661 }%
1662 \or
1663 \@addtopreamble{%
1664   {\hfil\hskip1sp\array@row@rst\cell@font\ignorespaces\@sharp\unskip\cell@fil}%
1665 }%
1666 \fi
1667 }%

\@tabclassiv
1668 \def\@tabclassiv@LaTeX{%
1669 \@addtopreamble\@nextchar
1670 }%
1671 \def\@tabclassiv@ltx{%
1672 \expandafter\@addtopreamble\expandafter{\@nextchar}%
1673 }%

\@arrayclassiv
1674 \def\@arrayclassiv@LaTeX{%
1675 \@addtopreamble{\$@nextchar$}%
1676 }%
1677 \def\@arrayclassiv@ltx{%
1678 \expandafter\@addtopreamble\expandafter{\expandafter$@nextchar$}%
1679 }%

\@classv
1680 \def\@classv@LaTeX{%
1681 \@addtopreamble{\@startpbox{\@nextchar}\ignorespaces
1682 \@sharp\@endpbox}%
1683 }%
1684 \def\@classv@ltx{%
1685 \expandafter\@addtopreamble
1686 \expandafter{%
1687 \expandafter \@startpbox
1688 \expandafter {\@nextchar}%
1689 \pbox@hook\array@row@rst\cell@font\ignorespaces\@sharp\@endpbox
1690 }%
1691 }%

\@classx
1692 \def\@classx@array{%
1693 \ifcase \@lastchclass
1694 \@acolampacol \or

```

```

1695 \@addamp \@acol \or
1696 \@acolampacol \or
1697 \or
1698 \@acol \@firstampfalse \or
1699 \@addamp
1700 \fi
1701 }%
1702 \def\@classx@array@new{%
1703 \ifcase \@lastchclass
1704 \@acolampacol
1705 \or
1706 \@addamp \@acol
1707 \or
1708 \@acolampacol
1709 \or
1710 \or
1711 \@firstampfalse\@acoll
1712 \or
1713 \@addamp
1714 \fi
1715 }%

```

6.14 Repair other broken parts of L^AT_EX

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

```

1716 \def\@xbitor@LaTeX #1{\@tempcntb \count#1
1717 \ifnum \@tempcnta =\z@
1718 \else
1719 \divide\@tempcntb\@tempcnta
1720 \ifodd\@tempcntb \@testtrue\fi
1721 \fi}%
1722 \def\@xbitor@ltx#1{%
1723 \@tempcntb\count#1%
1724 \@ifnum{\@tempcnta=\z@}{\}%
1725 \divide\@tempcntb\@tempcnta
1726 \@ifodd\@tempcntb{\@testtrue}{\}%
1727 }%
1728 }%
1729 \@ifx{\@xbitor\@xbitor@LaTeX}{%
1730 \class@info{Repairing broken LaTeX \string\@xbitor}%
1731 }{\%
1732 \class@info{Unrecognized LaTeX \string\@xbitor. Please update this document clas
1733 }%
1734 \let\@xbitor\@xbitor@ltx

```

6.15 Syntax

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

```

1735 \newcommand*\@gobble@opt@one[2][\]{%

```

6.16 Auto-indented Contents

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

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

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

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

```
1736 \def\@starttoc#1{%
1737   \begingroup
1738     \toc@pre
1739     \makeatletter
1740     \@input{\jobname.#1}%
1741     \if@filesw
1742       \expandafter\newwrite\csname tf@#1\endcsname
1743       \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
1744     \fi
1745     \@nobreakfalse
1746     \toc@post
1747   \endgroup
1748 }%
1749 \def\toc@pre{}%
1750 \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.

```
1751 \def\toc@@font{}{\footnotesize\rmfamily}%
1752 \def\@dotsep{\z@}{5.5pt}%
```

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

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

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

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

```
%}%  
%
```

Glom some \dimen registers.

```
1753 \let\tocdim@section \leftmargini  
1754 \let\tocdim@subsection \leftmarginii  
1755 \let\tocdim@subsubsection \leftmarginiii  
1756 \let\tocdim@paragraph \leftmarginiv  
1757 \let\tocdim@appendix \leftmarginv  
1758 \let\tocdim@pagenum \leftmarginvi
```

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

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

To activate this override of the standard L^AT_EX processing, the substyle does:

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

```
1759 \def\toc@pre@auto{%  
1760 \toc@@font  
1761 \@tempdima\z@  
1762 \toc@setindent\@tempdima{section}%  
1763 \toc@setindent\@tempdima{subsection}%  
1764 \toc@setindent\@tempdima{subsubsection}%  
1765 \toc@setindent\@tempdima{paragraph}%  
1766 \toc@letdimen{appendix}%  
1767 \toc@letdimen{pagenum}%  
1768 }%  
1769 \def\toc@post@auto{%  
1770 \if@filesw  
1771 \begingroup  
1772 \toc@writedimen{section}%  
1773 \toc@writedimen{subsection}%  
1774 \toc@writedimen{subsubsection}%  
1775 \toc@writedimen{paragraph}%  
1776 \toc@writedimen{appendix}%  
1777 \toc@writedimen{pagenum}%  
1778 \endgroup  
1779 \fi  
1780 }%
```

```
\toc@setindent
```

```
1781 \def\toc@setindent#1#2{%  
1782 \csname tocdim@#2\endcsname\tocdim@min\relax  
1783 \@ifundefined{tocmax@#2}{\@namedef{tocmax@#2}{\z@}}{}%  
1784 \advance#1\@nameuse{tocmax@#2}\relax  
1785 \expandafter\edef\csname tocleft@#2\endcsname{\the#1}%  
1786 }%
```

```
\toc@letdimen
```

```
1787 \def\toc@letdimen#1{%  
1788 \csname tocdim@#1\endcsname\tocdim@min\relax  
1789 \@ifundefined{tocmax@#1}{\@namedef{tocmax@#1}{\z@}}{}%  
1790 \expandafter\let\csname tocleft@#1\endcsname\csname tocmax@#1\endcsname  
1791 }%
```

`\toc@writedimen`

```
1792 \def\toc@writedimen#1{%
1793 \immediate\write\@auxout{%
1794 \gdef\expandafter\string\csname tocmax@#1\endcsname{%
1795 \expandafter\the\csname tocdim@#1\endcsname
1796 }%
1797 }%
1798 }%
```

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

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

```
1799 \def\l@sections#1#2#3#4{%
1800 % #1 - superior section
1801 % #2 - this section
1802 % #3 - content, including possible \numberline
1803 % #4 - page number
1804 \begingroup
1805 \everypar{}%
1806 \set@tocdim@pagenum{#4}%
1807 \global\@tempdima\csname tocdim@#2\endcsname
1808 \leftskip\csname tocdim@#2\endcsname\relax
1809 \dimen@\csname tocdim@#1\endcsname\relax
1810 \parindent-\leftskip\advance\parindent\dimen@
1811 \rightskip\tocleft@pagenum plus 1fil\relax
1812 \skip@\parfillskip\parfillskip\z@
1813 \let\numberline\numberline@@sections
1814 \@nameuse{l@f@#2}%
1815 \ignorespaces#3\unskip\nobreak\hskip\skip@
1816 \hbext@\rightskip{\hfil\unhbox\@tempboxa}\hskip-\rightskip\hskip\z@skip
1817 \par
1818 \expandafter\aftergroup\csname tocdim@#2\endcsname\expandafter
1819 \endgroup\the\@tempdima\relax
1820 }%
1821 \def\set@tocdim@pagenum#1{%
1822 \setbox\@tempboxa\hbox{\ignorespaces#1}%
1823 \@ifdim{\tocdim@pagenum<\wd\z@}{\global\tocdim@pagenum\wd\z@}{}%
1824 }%
```

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

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

```
1825 \def\numberline@@sections#1{%
1826 \leavevmode\hbext@-\parindent{%
1827 \hfil
1828 \@ifempty{#1}{}{%
1829 \setbox\z@\hbox{#1.\kern\@dotsep}%
1830 \@ifdim{\@tempdima<\wd\z@}{\global\@tempdima\wd\z@}{}%
1831 \unhbox\z@
1832 }%
```

```

1833 }%
1834 \ignorespaces
1835 }%
1836 \def\tocdim@min{z@}%

```

6.17 Lists

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

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

```

1837 \def\list#1#2{%
1838   \ifnum \@listdepth >5\relax
1839     \@toodeep
1840   \else
1841     \global\advance\@listdepth\@ne
1842   \fi
1843   \rightmargin\z@
1844   \listparindent\z@
1845   \itemindent\z@
1846   \csname @list\romannumeral\the\@listdepth\endcsname
1847   \def\@itemlabel{#1}%
1848   \let\makelabel\@mklab
1849   \@nmbrrlistfalse
1850   #2\relax
1851   \@trivlist
1852   \parskip\parsep
1853   \set@listindent
1854   \ignorespaces
1855 }%
1856 \def\set@listindent@parshape{%
1857   \parindent\listparindent
1858   \advance\@totalleftmargin\leftmargin
1859   \advance\linewidth-\rightmargin
1860   \advance\linewidth-\leftmargin
1861   \parshape\@ne\@totalleftmargin\linewidth
1862 }%
1863 \def\set@listindent@{%
1864   \parindent\listparindent
1865   \advance\@totalleftmargin\leftmargin
1866   \advance\rightskip\rightmargin
1867   \advance\leftskip\@totalleftmargin
1868 }%
1869 \let\set@listindent\set@listindent@parshape

```

6.18 End of the `ltxutil` DOCSTRIP module

Here ends the module.

```
1870 %</ltxutil-krn>
```

Here ends the programmer’s documentation.

Index

Symbols	
<code>\%</code>	662
<code>\(</code>	478
<code>\)</code>	479
<code>.dtx</code>	4, 5
<code>.ins</code>	4
<code>\@array</code>	992, 1053, 1056
<code>\@end</code>	316
<code>\@endpbox</code>	1155, 1178, 1284, 1293
<code>\@par</code> . 612, 734, 776, 1161, 1185, 1311	
<code>\@startpbox</code> . 1155, 1177, 1284, 1293	
<code>\@Esphack</code>	650, 652
<code>\@M</code>	34
<code>\@MM</code>	451
<code>\@acol</code> .. 1014, 1023, 1040, 1110, 1121, 1137, 1412, 1413, 1491, 1493, 1582, 1628, 1695, 1698, 1706	
<code>\@acolampacol</code>	
... 1580, 1582, 1590, 1598, 1618, 1626, 1642, 1650, 1694, 1696, 1704, 1708	
<code>\@acoll</code> . 1021, 1038, 1119, 1135, 1600, 1652, 1711	
<code>\@acolr</code> . 1022, 1039, 1120, 1136, 1442, 1450, 1540, 1543	
<code>\@addamp</code>	828, 918, <u>1555</u> , 1581, 1596, 1624, 1648, 1695, 1699, 1706, 1713
<code>\@addamp@LaTeX</code>	828, 1555
<code>\@addamp@ltx</code>	918, 1558
<code>\@addtopreamble</code>	
... 1399, 1495, 1547, 1559, 1565, 1568, 1574, 1577, 1603, 1607, 1611, 1655, 1659, 1663, 1669, 1672, 1675, 1678, 1681, 1685	
<code>\@afterindentfalse</code>	694
<code>\@afterindenttrue</code>	692
<code>\@ampacol</code> 1580, 1592, 1620, 1644	
<code>\@argarraycr</code>	1082, 1086
<code>\@argswap</code>	<u>12</u>
<code>\@argswap</code>	<u>250</u>
<code>\@argswap@val</code>	<u>12</u>
<code>\@argswap@val</code>	<u>250</u>
<code>\@argtabularcr</code> ...	1064, 1065
<code>\@array</code>	<u>36</u>
<code>\@array</code> . 820, 910, 934, 991, 992, 1047, 1050, <u>1144</u>	
<code>\@array@LaTeX</code>	820, 1144
<code>\@array@align@</code>	<u>1300</u>
<code>\@array@align@b</code>	1301
<code>\@array@align@c</code>	<u>40</u>
<code>\@array@align@c</code> .. 1302, 1317	
<code>\@array@align@t</code>	1300
<code>\@array@align@v</code>	<u>40</u>
<code>\@array@align@v</code> .. 1303, 1317	
<code>\@array@array</code>	934, <u>1197</u>
<code>\@array@array@new</code> . 991, 1219	
<code>\@array@ltx</code>	910, 1164
<code>\@array@sw</code>	<u>33</u>
<code>\@array@sw</code>	819, 933, <u>1009</u>
<code>\@array@sw@LaTeX</code> .. 933, 1009	
<code>\@array@sw@array</code> .. 819, 1010	
<code>\@arrayacol</code> 829, 919, 1000, 1110, 1119–1121, 1135–1137, <u>1561</u>	
<code>\@arrayacol@LaTeX</code> . 829, 1561	
<code>\@arrayacol@ltx</code> 919, 1000, 1564	
<code>\@arrayclassiv</code> 833, 923, 1112, 1123, <u>1674</u>	
<code>\@arrayclassiv@LaTeX</code> .. 833, 1674	
<code>\@arrayclassiv@ltx</code> 923, 1677	
<code>\@arrayclassz</code> . 831, 921, 1111, 1122, <u>1579</u>	
<code>\@arrayclassz@LaTeX</code> 831, 1579	
<code>\@arrayclassz@ltx</code> . 921, 1588	
<code>\@arraycr</code>	<u>35</u>
<code>\@arraycr</code> 943, 1004, <u>1058</u> , 1113, 1124, 1217, 1243	
<code>\@arraycr@array</code> ... 943, 1070	
<code>\@arraycr@new</code> 1004, 1075	
<code>\@arrayleft</code>	1210, 1240
<code>\@arrayright</code>	1260, 1264
<code>\@arstrut</code>	<u>41</u>
<code>\@arstrut</code> 1154, 1170, 1207, 1233, 1285, 1294, 1367, 1369, 1386, <u>1397</u>	
<code>\@arstrut@hline</code>	<u>41</u>
<code>\@arstrut@hline</code>	<u>1361</u>
<code>\@arstrut@hline@clnc</code> ...	<u>41</u>

<code>\@arstrut@hline@c1nc</code>	. 1376, 1388	<code>\@currentlabel</code>	. 360, 376, 454, 494
<code>\@arstrut@hook</code> 41	<code>\@currentlabelname</code> 28
<code>\@arstrut@hook</code>	... 1347, 1348, 1361, 1397, 1398	<code>\@currentlabelname</code> 765
<code>\@arstrut@org</code> 1361	<code>\@dblarg</code> 702, 703
<code>\@arstrutbox</code> 36, 41	<code>\@dblfloat</code> 601, 603, 622
<code>\@arstrutbox</code>	. 1066, 1067, 1090, 1096, 1147, 1191, 1200, 1222	<code>\@dblfloat@LaTeX</code>	... 601, 622
<code>\@arstrutbox@hline</code> 41	<code>\@defaultsubs</code> 336
<code>\@arstrutbox@hline</code>	... 1361	<code>\@depth</code>	. 1066, 1067, 1091, 1097, 1149, 1193, 1202, 1224, 1379
<code>\@auxout</code> 476, 493, 1793	<code>\@dofilelist</code> 329
<code>\@boole@def</code> 12	<code>\@eha</code> 156
<code>\@boole@def</code> 255, 267–270, 272–282	<code>\@enddocumenthook</code> 311
<code>\@boolean</code> 12	<code>\@endpbox</code>	1155, 1178, 1284, 1293, 1405, 1421, 1475, 1506, 1682, 1689
<code>\@boolean</code> 255	<code>\@eqcnt</code>	. 363, 368, 369, 371, 377, 390, 393, 396
<code>\@booleanfalse</code> 12	<code>\@eqnocr</code> 34
<code>\@booleanfalse</code>	. 265, 558, 559, 617	<code>\@eqnocr</code> 365, 379
<code>\@booleantrue</code> 12	<code>\@eqnset</code> 367, 389
<code>\@booleantrue</code>	... 265, 541, 589	<code>\@eqnswtrue</code> 361, 377
<code>\@bsphack</code> 492, 655	<code>\@eqpen</code> 34
<code>\@caption</code> 21	<code>\@esphack</code> 496
<code>\@captertype</code> 561	<code>\@expast</code> 1406, 1422
<code>\@centering</code>	. 364, 367, 370, 389, 395, 408	<code>\@finalstrut</code> 459
<code>\@chclass</code>	1410, 1411, 1426, 1439, 1484, 1489, 1515, 1537	<code>\@firststampfalse</code> 1556, 1559, 1582, 1600, 1628, 1652, 1698, 1711
<code>\@checkend</code> 312	<code>\@firststamptrue</code>	... 1401, 1416, 1474, 1503
<code>\@chnum</code>	.. 1584, 1602, 1632, 1654	<code>\@firstoftwo</code> 797
<code>\@classi</code>	. 1410, 1429, 1484, 1518	<code>\@float</code> 19, 23
<code>\@classii</code>	1410, 1431, 1484, 1520	<code>\@float@LaTeX</code>	... 599, 600, 621
<code>\@classiii</code> 1410, 1433	<code>\@floatboxreset</code> 569
<code>\@classiv</code>	1016, 1025, 1112, 1123, 1411, 1435	<code>\@flushglue</code> 356, 1390
<code>\@classv</code>	.. 835, 925, 1411, 1437, 1485, 1525, 1680	<code>\@font@warning</code> 331
<code>\@classv@LaTeX</code> 835, 1680	<code>\@footnotemark</code> 16, 17
<code>\@classv@ltx</code> 925, 1684	<code>\@footnotemark</code> 434
<code>\@classvi</code> 1485, 1527	<code>\@footnotetext</code> 16–18
<code>\@classvii</code> 1486, 1529	<code>\@footnotetext</code>	. 411, 418, 438, 550
<code>\@classviii</code> 1486, 1531	<code>\@gobble@opt@one</code> 49
<code>\@classx</code>	.. 941, 999, 1487, 1488, 1533, 1535, 1692	<code>\@gobble@opt@one</code> 1735
<code>\@classx@array</code> 941, 1692	<code>\@halignto</code>	... 1114, 1125, 1130, 1141, 1153, 1187, 1206, 1251
<code>\@classx@array@new</code>	999, 1702	<code>\@hang@from</code> 731, 762
<code>\@classz</code>	1015, 1024, 1111, 1122, 1410, 1427, 1484, 1516	<code>\@hang@froms</code> 772, 793
<code>\@ctrerr</code> 686	<code>\@hangfrom</code> 26, 28
		<code>\@hangfrom</code> 762

`\@preamerr` 1161, 1185, 1413, 1445, 1447, 1494, 1495, 1497, 1545, 1547, 1551
`\@provide` ... [287](#), 811, 813, 816
`\@refundefined` 337
`\@resetactivechars` 214
`\@rightskip` 355
`\@runin@to` [26](#)
`\@runin@to` 747, 763
`\@runin@tos` 783, 794
`\@secCNTformat` [26](#)
`\@secCNTformat` 719
`\@secpenalty` 699
`\@sect` [26](#)
`\@sect` [706](#)
`\@sect@ltx` 703, 708
`\@setckpt` 323
`\@sharp` . 1158, 1181, 1213, 1246, 1283, 1291, 1404, 1420, 1455, 1461, 1475, 1495, 1504, 1547, 1585, 1586, 1604, 1608, 1612, 1633, 1635, 1637, 1656, 1660, 1664, 1682, 1689
`\@sssect` [26](#), [28](#)
`\@sssect` [764](#)
`\@sssect@ltx` 702, 764
`\@startcolumn` [8](#)
`\@startpbox`
... 1155, 1177, 1284, 1293, 1405, 1421, 1471, 1475, 1506, 1681, 1687
`\@startsection` [26](#), [28](#)
`\@startsection` [687](#)
`\@startsection@hook` 688, 705
`\@starttoc` [50](#)
`\@starttoc` [1736](#)
`\@svsec` [26](#)
`\@svsec` . 711, 717, 719, 722, 732, 748
`\@svsechd` 745, 781
`\@tabacol` . 830, 920, 1001, 1014, 1023, 1040, [1567](#)
`\@tabacol@LaTeX` ... 830, 1570
`\@tabacol@ltx` . 920, 1001, 1573
`\@tabacoll` ... 1021, 1038, 1567
`\@tabacolr` ... 1022, 1039, 1576
`\@tabarray` [34](#)
`\@tabarray` 823, 913, 936, 994, 1018, 1027, 1035, 1044, [1046](#), 1115, 1126, 1131, 1142
`\@tabarray@LaTeX` .. 823, 1046
`\@tabarray@array` .. 936, 1052
`\@tabarray@array@new` .. 994, 1055
`\@tabarray@ltx` 913, 1049
`\@tabclassiv` .. 832, 922, 1016, 1025, [1668](#)
`\@tabclassiv@LaTeX` 832, 1668
`\@tabclassiv@ltx` .. 922, 1671
`\@tabclassz` 834, 924, 1015, 1024, [1616](#)
`\@tabclassz@LaTeX` . 834, 1616
`\@tabclassz@ltx` ... 924, 1640
`\@tabular` [33](#)
`\@tabular` 815, 822, 912, 935, 993, [1011](#)
`\@tabular@LaTeX` ... 822, 1011
`\@tabular@array` ... 935, 1029
`\@tabular@array@new` 993, 1037
`\@tabular@ltx` 912, 1020
`\@tabularcr` [34](#)
`\@tabularcr` 837, 927, 1009, 1017, 1026, [1058](#)
`\@tabularcr@LaTeX` [34](#)
`\@tabularcr@LaTeX` . 837, 1062
`\@tabularcr@ltx` ... 927, 1063
`\@tbpen` [34](#), [35](#)
`\@tbpen` [1058](#)
`\@tempa` 176, 178, 668, 669
`\@tempb` 659, 669
`\@tempc` 657, 675
`\@tempcntb` ... 1716, 1719, 1720, 1723, 1725, 1726
`\@tempdima` [52](#)
`\@tempskipa` . 691, 693, 694, 699, 726, 727, 767, 768
`\@temptokena` . 1476, 1482, 1507, 1512
`\@testdef` 324
`\@testpach` 1409, 1425, 1483, 1514
`\@testtrue` 1720, 1726
`\@tfor` ... 1407, 1423, 1481, 1512
`\@thefnmark` 423, 429, 455
`\@toodeep` 1839
`\@totalleftmargin` 1858, 1861, 1865, 1867
`\@tpfootnotetext` [438](#)
`\@trivlist` 1851
`\@undefined` 16, 797, 1041, 1138

<code>\@unexpandable@protect</code>	1403, 1419	<code>\}</code>	662, 666
<code>\@whilesw</code>	1477, 1508	<code>\]</code>	666
<code>\@width</code>	1066, 1067, 1091, 1097, 1150, 1194, 1203, 1225, 1380	<code>\^</code>	661, 674
<code>\@write@floatline</code>	25	<code>00readme.txt</code>	2, 4
<code>\@write@floatline</code>	665	<code>_</code>	64, 88, 89, 93, 112, 118, 123, 129, 130, 176
<code>\@writefile</code>	481	A	
<code>\@xargarraycr</code>	35	<code>\abovedisplayshortskip</code>	385
<code>\@xargarraycr</code>	839, 929, 945, 1006, 1058	<code>\abovedisplayskip</code>	380–385
<code>\@xargarraycr@LaTeX</code>	839, 1066	<code>\active</code>	661
<code>\@xargarraycr@array</code>	945, 1088	<code>\addcontentsline</code>	28
<code>\@xargarraycr@ltx</code>	929, 1067	<code>\addcontentsline</code>	483, 737, 751, 779, 788
<code>\@xargarraycr@new</code>	1006, 1094	<code>\addtocontents</code>	475, 484
<code>\@xarraycr</code>	35	<code>\aftergroup</code>	40
<code>\@xarraycr</code>	944, 1005, 1058	<code>\aftergroup</code>	1302, 1309, 1818
<code>\@xarraycr@array</code>	944, 1080	<code>\appdef</code>	11, 13, 14, 20
<code>\@xarraycr@new</code>	1005, 1084	<code>\appdef</code>	214, 221, 432, 498, 555, 795, 808, 1061, 1172, 1235, 1373, 1397, 1399
<code>\@xbitor</code>	1716	<code>\appdef@e</code>	239
<code>\@xbitor@LaTeX</code>	1716, 1729	<code>\appdef@eval</code>	244
<code>\@xbitor@ltx</code>	1722, 1734	<code>\appdef@val</code>	236, 245
<code>\@xfloat</code>	537	argument	
<code>\@xfloat@LaTeX</code>	537, 542	float	25
<code>\@xfloat@anchored</code>	542, 560	text	29
<code>\@xfloat@prep</code>	539, 544	<code>\array</code>	36
<code>\@xfootnote</code>	410	<code>\array</code>	824, 914, 937, 995, 1109
<code>\@xfootnotemark</code>	16, 17	array document class	8, 29, 30, 33, 35, 36, 43, 44
<code>\@xfootnotemark</code>	410	array environment	2, 29, 34
<code>\@xfootnotemark@ltx</code>	420, 436	<code>\array@array</code>	937, 1128
<code>\@xfootnotenext</code>	22	<code>\array@array@new</code>	995, 1133
<code>\@xhline</code>	1326, 1332, 1333	<code>\array@default</code>	620, 1050, 1056, 1300
<code>\@xhline@unneeded</code>	1335	<code>\array@hook</code>	814
<code>\@xifempty</code>	12	<code>\array@LaTeX</code>	824, 1109
<code>\@xssect</code>	760, 791	<code>\array@ltx</code>	914, 1117
<code>\@xtabularcr</code>	35	<code>\array@row@pre</code>	1175, 1238, 1316
<code>\@xtabularcr</code>	838, 928, 1058	<code>\array@row@pst</code>	1257, 1263, 1316
<code>\@xtabularcr@LaTeX</code>	838, 1064	<code>\array@row@rst</code>	40
<code>\@xtabularcr@ltx</code>	928, 1065	<code>\array@row@rst</code>	1316, 1460, 1469, 1604, 1608, 1612, 1656, 1660, 1664, 1689
<code>\@yargarraycr</code>	35	<code>\arraycolsep</code>	368, 369, 1129, 1562, 1565
<code>\@yargarraycr</code>	840, 930, 946, 1007, 1058		
<code>\@yargarraycr@LaTeX</code>	840, 1068		
<code>\@yargarraycr@array</code>	946, 1101		
<code>\@yargarraycr@ltx</code>	930, 1069		
<code>\@yargarraycr@new</code>	1007, 1105		
<code>\@yfootnote</code>	410		
<code>\[</code>	666		
<code>\]</code>	34		
<code>\{</code>	662, 666		

D

`\dollarbegin` . 1010, 1033, 1042,
1130, 1139, 1140
`\dollarend` 1034, 1043, 1130, 1140
`\deadcycles` 315
`\dimen` 50
`\dimen@` .. 1376–1378, 1809, 1810
`\dimen@iii` 190
`\dimendef` 190
`\displaystyle` 367, 370, 389, 395
`\displaywidth` 366
`\do` 1408, 1424, 1482, 1513
`\do@check@aux` 310
`\do@if@floats` 19, 23
`\do@if@floats` 588
`doc` 4, 5
`\DocInput` 7
`\document` 13
`\document` 291
document class
 array 8, 29, 30, 33, 35, 36, 43,
 44
 classes.dtx 23
 ftnright 8
 hyperref 15, 17, 18, 20, 21, 28,
 29, 49, 52
 hyperref package 20
 longtable 8, 43, 44
 ltxdoc 4, 7
 ltxgrid 8, 14, 18
 ltxutil 1, 8
 ltxutil.sty 3
document environment 4, 14
`\doublerulesep` 1338
`\dp` .. 450, 1066, 1067, 1090, 1096,
1149, 1193, 1202, 1224, 1379

E

`\edef` 44
`\edef` .. 224, 229, 234, 1152, 1556,
1562, 1571, 1583, 1630, 1785
`\end` 25
`\end@float@anchored` 564, 565,
573
`\end@float@pagebreak` .. 578,
581, 613
`\endarray` 39
`\endarray` 808, 825, 915, 938, 996,
1253, 1271, 1274, 1277
`\endarray@array` 938, 1259

`\endarray@array@new` 996, 1262
`\endarray@hook` 812, 813
`\endarray@LaTeX` 825, 1253
`\endarray@ltx` 39
`\endarray@ltx` . 915, 1256, 1263
`\endbatchfile` 60
`\endcsname` 16, 17, 197,
199, 203, 205, 254, 270, 422,
455, 489, 592, 594, 597, 598,
626, 627, 631, 640, 644, 646,
657, 660, 723, 731, 736, 741,
747, 750, 755, 772, 778, 783,
787, 1003, 1742, 1743, 1782,
1785, 1788, 1790, 1794, 1795,
1807–1809, 1818, 1846
`\enddocument` 310
`\endgroup` 13
`\endminipage` 528
`\endpreamble` 39
`\endtabular` . 808, 826, 916, 939,
997, 1003, 1267, 1279
endtabular environment 39
`\endtabular*` 1279
`\endtabular@array` . 939, 1273
`\endtabular@array@new` . 997,
1276
`\endtabular@hook` 810, 811
`\endtabular@LaTeX` . 826, 1267
`\endtabular@ltx` 916, 1270
`\endwrite@float` 650, 660
`\endwrite@floats` 652
environment
 array 2, 29, 34
 document 4, 14
 endtabular 39
 eqnarray 15, 34
 figure 19
 figure@write 19, 23
 longtable 19, 25
 minipage 16, 19
 tabular 2, 29, 34
`\eqnarray` 15
`\eqnarray` 400, 402
eqnarray environment 15, 34
`\eqnarray@fleqn@fixed` . 358
`\eqnarray@LaTeX` 358
`\eqncolsep` .. 391, 394, 403, 404
`\errorstopmode` 211
`\extrarowheight` .. 1199, 1221

F

`\f@ur` [192](#), [1502](#)
`\false@sw` [260](#), [266](#), [271](#), [283](#), [288](#),
 [843](#), [846](#), [849](#), [852](#), [855](#), [858](#),
 [861](#), [864](#), [867](#), [870](#), [873](#), [876](#),
 [879](#), [882](#), [885](#), [888](#), [891](#), [894](#),
 [897](#), [900](#), [903](#), [949](#), [952](#), [955](#),
 [958](#), [961](#), [964](#), [967](#), [970](#), [973](#),
 [976](#), [979](#), [982](#), [985](#)
figure environment [19](#)
figure@write environment [19](#), [23](#)
`\figuresname` [19](#)
file
 .dtx [4](#), [5](#)
 .ins [4](#)
 00readme.txt [2](#), [4](#)
 classes.dtx [19](#)
 doc [4](#), [5](#)
 ltxfloat.dtx [16](#)
 ltxutil [2](#), [9](#), [53](#)
 ltxutil.drv [5](#)
 ltxutil.dtx [2](#)
 ltxutil.ins [2-4](#)
 ltxutil.pdf [1](#)
 ltxutil.sty [1-3](#), [5](#)
 makeindex [2](#)
 nameref.sty [28](#)
 pdfmark.def [28](#)
`\file` [42](#), [43](#), [98](#), [101](#), [107](#), [108](#), [117](#),
 [123](#), [126](#), [127](#), [129](#), [132](#), [133](#)
float, argument [25](#)
`\float@end@tag` [25](#)
`\float@end@tag` [665](#)
`\floatingpenalty` [18](#)
`\floatingpenalty` [451](#)
`\floats@sw` [22](#), [23](#)
`\floats@sw` .. [537](#), [541](#), [542](#), [558](#),
 [559](#), [589](#), [590](#), [617](#)
`\flushing` [351](#)
`\font@submax` [330](#), [333](#)
`\fontsubfuzz` [330](#)
`\footins` [439](#), [525](#)
`\footnote` [16](#), [19](#)
`\footnote` [410](#), [545](#), [556](#)
`\footnote@latex` [545](#), [556](#)
`\footnotemark` [16](#)
`\footnotemark` [410](#)
`\footnotesep` [449](#), [459](#)
`\footnotesize` [447](#), [1751](#)
`\fp@proc@H` [559](#)

`\fp@proc@h` [558](#)
`\from` [42](#), [44](#)
`\frstrut` [1321](#), [1322](#), [1359](#)
ftnright document class [8](#)
`\fullinterlineskip` . [218](#), [515](#)
`\futurelet` [1325](#), [1332](#)

G

`\gappdef` [184](#), [221](#)
`\generate` [41](#)
`\GetFileInfo` [21](#)
`\glet@environment` .. [171](#), [195](#)
`\glossary` [477](#)

H

`\H@@footnotemark` [425](#), [430](#), [433](#),
 [434](#)
`\H@@footnotetext` [551](#)
`\H@refstepcounter` [27](#)
`\H@refstepcounter` .. [713](#), [806](#)
`\H@svsec` [710](#), [714](#), [732](#), [748](#), [766](#),
 [773](#), [774](#), [784](#), [785](#)
`\halign` [39](#), [44](#)
`\halign@` [191](#), [388](#)
`\hangindent` [52](#)
`\hb@xt@` [371](#), [396](#), [585](#), [1816](#), [1826](#)
`\hbox` [52](#)
`\hline` [41](#)
`\hline` [836](#), [926](#), [1324](#)
`\hline@LaTeX` [836](#), [1324](#)
`\hline@ltx` [926](#), [1328](#)
`\hline@rule` [41](#)
`\hline@rule` .. [1323](#), [1331](#), [1346](#),
 [1361](#)
`\hrule` [11](#)
`\hsize` [13](#)
`\ht` . [1148](#), [1192](#), [1198](#), [1220](#), [1376](#),
 [1377](#)
`\Hy@raisedlink` [800](#)
`\hyper@anchor` [798](#)
`\hyper@anchorend` [29](#)
`\hyper@anchorend` [802](#)
`\hyper@anchorstart` [29](#)
`\hyper@anchorstart` [801](#)
`\hyper@last` [28](#)
`\hyper@last` [799](#)
`\hyper@linkend` [29](#)
`\hyper@linkend` [804](#)
`\hyper@linkstart` [29](#)
`\hyper@linkstart` [803](#)

<code>\hyperanchor</code>	28	<code>\intextsep</code>	580, 581
<code>hyperref</code> document class	15, 17, 18, 20, 21, 28, 29, 49, 52	<code>\item</code>	121, 131, 134
<code>hyperref</code> package document class 20	<code>\itemindent</code>	1845
I			
<code>\ialign</code>	36	J	
<code>\ialign</code>	1153, 1206	<code>\j@nk</code>	289
<code>\if</code>	36	K	
<code>\if</code>	257, 269, 1145, 1211	<code>\keepsilent</code>	40
<code>\if@filesw</code> .	321, 338, 1741, 1770	L	
<code>\if@firstamp</code>	1556, 1559	<code>\l@</code>	50
<code>\if@nobreak</code>	696	<code>\l@sections</code>	50
<code>\if@noskipsec</code>	689	<code>\l@sections</code>	1799
<code>\ifeof</code>	273	<code>\l@section</code>	1753
<code>\iffalse</code>	1072, 1077	<code>\label</code>	477, 491
<code>\ifhbox</code>	274	<code>\lastbox</code>	518, 634
<code>\ifinner</code>	276	<code>\LaTeX</code> 64, 88, 89, 99, 109, 112, 123, 129, 130, 137	
<code>\ifmmode</code>	277	<code>\LaTeXe</code>	93, 118, 148
<code>\ifodd</code>	279, 1720	<code>\leftmargin</code> ..	1858, 1860, 1865
<code>\ifToplevel</code>	47	<code>\leftmargini</code>	1753
<code>\ifvbox</code>	280	<code>\leftmarginii</code>	1754
<code>\ifvoid</code>	282	<code>\leftmarginiii</code>	1755
<code>\ignorespaces</code>		<code>\leftmarginiv</code>	1756
	459, 1285, 1298, 1455, 1461, 1633, 1635, 1637, 1656, 1660, 1664, 1681, 1689, 1815, 1822, 1834, 1854	<code>\leftmarginv</code>	1757
<code>\immediate</code> ..	322, 595, 629, 675, 1743, 1793	<code>\leftmarginvi</code>	1758
<code>\incompatible@package</code> .	174	<code>\leftskip</code>	52
<code>\index</code>	477	<code>\leftskip</code> .	353, 1808, 1810, 1867
<code>\init@documenthook</code> .	293, 299, 795	<code>\let</code>	51
<code>\insert</code>	439, 525	<code>\let@environment</code>	10
<code>\insert@column</code> .	942, 1002, 1453	<code>\let@environment</code> 195, 605, 606, 608, 624, 625	
<code>\insert@column@array</code> ..	942, 1453	<code>\lineskip</code> .	1160, 1183, 1214, 1248
<code>\insert@column@array@new</code> 1002, 1458	<code>\linewidth</code>	388, 466, 473, 1859–1861
<code>\interdisplaylinepenalty</code> .	34	<code>\list</code>	53
<code>\interfootnotelinepenalty</code>	448	<code>\list</code>	1837
<code>\interlinepenalty</code> .	448, 729, 770	<code>\listparindent</code> .	1844, 1857, 1864
<code>\intertabularlinepenalty</code> .	34	<code>longtable</code> document class	8, 43, 44
<code>\intertabularlinepenalty</code> 1058, 1059, 1061, 1063, 1078	<code>longtable</code> environment	... 19, 25
		<code>\loop</code>	13
		<code>\loopuntil</code>	285
		<code>\loopwhile</code>	285
		<code>\lrstrut</code>	1322, 1323, 1360
		<code>\LT@array</code>	8
		<code>ltxfloat.dtx</code>	16
		<code>ltxdoc</code> document class 4, 7
		<code>ltxgrid</code> document class	. 8, 14, 18
		<code>ltxutil</code>	2, 9, 53

ltxutil document class	1, 8
ltxutil.drv	5
ltxutil.dtx	2
ltxutil.ins	2–4
ltxutil.pdf	1
ltxutil.sty	1–3, 5
ltxutil.sty document class	..	3
M		
\make@footnotetext	18
\make@footnotetext	438
\makeatletter	326, 1739
makeindex	2
\makelabel	1848
\maketitle	17, 18
\maketitle	78
\mark	20
\mathindent	378, 408
\meaning	216
\MessageBreak	332
minipage environment	16, 19
\minipagefootnote@drop	. 18	
\minipagefootnote@drop	462,	
	<u>501</u>	
\minipagefootnote@foot	<u>501</u>	
\minipagefootnote@here	. 21	
\minipagefootnote@here	499,	
	<u>501, 574</u>	
\minipagefootnote@init	<u>501</u>	
\minipagefootnote@pick	443,	
	<u>501</u>	
\Msg	48–54, 56–58
\multicolumn	... 821, 911, <u>1280</u>	
\multicolumn@LaTeX	821, <u>1280</u>	
\multicolumn@ltx	.. 911, 1287	
\multispan	1281, 1288
N		
nameref.sty	28
\NC@list	1477, 1508
\newbox	1372
\newcount	1058, 1060
\newenvironment	25
\newlabel	20
\newlabel	494
\newlinechar	674
\newwrite	591, 1742
\noalign	1065, 1067–1069, 1086,	
	1099, 1103, 1107, 1323, 1325,	
	1329, 1333, 1345	
\noexpand	.. 216, 217, 1153, 1206	
\normalsize	295
\numberline	52
\numberline@@sections	1813,	
	<u>1825</u>	
O		
\obeylines	671
\obsolete@command	159
\oneapage	613
\openout	596, 1743
\other	194
P		
\p@equation	360, 376
\package@name	146, 147
\PackageInfo	147
\pagegrid@col	571
\parindent	52
\parindent	583, 1810, 1826, 1857,	
	<u>1864</u>	
\parsep	1852
\parshape	52
\parshape	1861
\parskip	382, 1852
\partopsep	381
\pbox@hook	1396, 1689
pdfmark.def	28
\phantomsection	28
\phantomsection	710, 715, 766,	
	<u>796</u>	
placeholder		
	<i>< class customization commands ></i>	
	8
	<i>< your document here ></i> 8
\preamble	26
\prepdef	11
\prepdef <u>221</u> , 291, 499, 626,	
	627, 719, 721, 810, 812, 815,	
	<u>1168</u> , 1231	
\prepnext@tok	1480, 1511
\prevdepth	11
\prevdepth	218
\print@float	24
\print@float	<u>612</u>
\printfigures	20
\protected@edef	454, 717
\protected@write	... 476, 493	
\protected@xdef	429

`\providecommand` 796, 798–804, 814
`\ProvidesFile` 4, 6

R

`\raggedleft` 15
`\raggedright` 15
`\realfootnote` 17
`\refstepcounter` 27
`\relax` 26, 29
`\repeat` 13
`\replace@command` 164
`\replace@environment` .. 169
`\RequirePackage` ... 13, 15, 17
`\reserved@a` .. 1326, 1332, 1336, 1337, 1408, 1423
`\reset@font` 447
`\romannumeral` 1846
`\rule` 459

S

`\samepage` 34
`\samepage` 553, 1061
`\save@decl` 1485, 1522
`\say` 11
`\say` 216, 1336
`\saythe` 11
`\saythe` 216
`\sc` 109
`\sec@upcase` 29
`\sec@upcase` 807
`\section` 105, 637
`\Sectionformat` 28
`\Sectionformat` .. 773, 784, 797
`\set@arstrutbox` 41
`\set@arstrutbox` .. 1166, 1190, 1361
`\set@eqnarray@skips` ... 409
`\set@footnotewidth` 18
`\set@footnotewidth` 438
`\set@footnotewidth@ii` .. 18
`\set@footnotewidth@ii` . 468
`\set@listindent` .. 1853, 1869
`\set@listindent@` 1863
`\set@listindent@parshape` .
..... 1856, 1869
`\set@pica@hook` 14
`\set@pica@hook` 296, 308
`\set@tocdim@pagenum` .. 1806, 1821

`\set@typeset@protect` . 1159, 1182, 1247, 1283, 1292
`\set@typesize@hook` 14
`\set@typesize@hook` . 294, 308
`\shipout` 10
`\showboxbreadth` 211
`\showboxdepth` 211
`\showoutput` 214
`\skip@` 1812, 1815
`\splitmaxdepth` 450
`\splittopskip` 449
`\stepcounter` 359, 376, 428
`\stop` 178
`\StopEventually` 6
`\string` . 160, 165, 176, 478–481, 494, 707, 1730, 1732, 1794
`\strutbox` 450, 459, 1148, 1149, 1192, 1193, 1198, 1202, 1220, 1224, 1377, 1379
`\subsection` 115
`\switch@array` 30, 35
`\switch@array` 809, 818
`\switch@tabular` 30, 35
`\switch@tabular` 809, 818

T

`\t@` 189, 191, 366
`\tab@rule` 41
`\tab@rule` 1321, 1322, 1343
`\tabcolsep` 1032, 1392–1394, 1571
`\tableleft@skip` 1184, 1249, 1389
`\tableleftsep` .. 1392, 1466, 1568
`\tableofcontents` 103
`\tabmid@skip` . 1169, 1232, 1390
`\tabmidsep` ... 1393, 1467, 1574
`\tabright@skip` 1173, 1236, 1391
`\tabrightsep` . 1394, 1468, 1577
`\tabskip` 36
`\tabskip` 364, 367, 370, 372, 378, 389, 395, 397, 1154, 1169, 1173, 1184, 1208, 1232, 1236, 1249
`\tabular` 29
`tabular environment` ... 2, 29, 34
`\tabular@hook` 815, 816
`\tabularnewline` .. 1156, 1179, 1217, 1244
`\TeX` 93
`text, argument` 29

`\write@floatline` ... 663, [665](#)
`\write@floats` 603, [649](#)

X

`\xdef` 1206, 1229

Y

`<your document here>` placeholder 8

Z

`\z@` [52](#)
`\z@skip` . 353–355, 367, 372, 389,
397, 575, 1154, 1160, 1183,
1389–1391, 1816