

# The L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> Sources

Johannes Braams  
David Carlisle  
Alan Jeffrey  
Leslie Lamport  
Frank Mittelbach  
Chris Rowley  
Rainer Schöpf

2015/10/01

This file is maintained by the L<sup>A</sup>T<sub>E</sub>X Project team.  
Bug reports can be opened (category `latex`) at  
<http://latex-project.org/bugs.html>.

## Contents

<b>a</b>	<b>ltdirchk.dtx</b>	<b>1</b>
<b>1</b>	<b>L<sup>A</sup>T<sub>E</sub>X System Dependent Initialisations</b>	<b>1</b>
<b>2</b>	<b>Initialisation</b>	<b>2</b>
2.1	INITEX . . . . .	2
2.2	Some bits of 2e . . . . .	4
<b>3</b>	<b>texsys.cfg</b>	<b>5</b>
3.1	texsys.cfg . . . . .	5
3.2	UNIX (web2c) . . . . .	6
3.3	UNIX (other) . . . . .	7
3.4	MSDOS (emtex) . . . . .	7
3.5	MSDOS (other) . . . . .	7
3.6	VMS (DECUS T <sub>E</sub> X, PD VMS 3.6) . . . . .	7
3.7	VMS (???) . . . . .	7
3.8	MACINTOSH (OzT <sub>E</sub> X 1.6) . . . . .	8
3.9	MACINTOSH (other) . . . . .	8
3.10	FAKE EXAMPLE . . . . .	8
<b>4</b>	<b>Setting \@currdir</b>	<b>9</b>
<b>5</b>	<b>Setting \input@path</b>	<b>10</b>

<b>6</b>	<b>Filename Parsing</b>	<b>11</b>
<b>7</b>	<b>T<sub>E</sub>X Versions</b>	<b>13</b>
<b>8</b>	<b>ltxcheck.tex</b>	<b>13</b>
<b>b</b>	<b>lplain.dtx</b>	<b>14</b>
<b>9</b>	<b>Plain T<sub>E</sub>X</b>	<b>14</b>
<b>c</b>	<b>ltvers.dtx</b>	<b>32</b>
<b>10</b>	<b>Version Identification</b>	<b>32</b>
<b>d</b>	<b>ltdefns.dtx</b>	<b>34</b>
<b>11</b>	<b>Definitions</b>	<b>34</b>
	11.1 Initex initialisations . . . . .	34
	11.2 Saved versions of T <sub>E</sub> X primitives . . . . .	34
	11.3 Command definitions . . . . .	35
	11.4 Robust commands and protect . . . . .	43
	11.5 Internal defining commands . . . . .	46
<b>e</b>	<b>lalloc.dtx</b>	<b>49</b>
<b>12</b>	<b>Counters</b>	<b>49</b>
<b>f</b>	<b>lcntrl.dtx</b>	<b>51</b>
<b>13</b>	<b>Program control structure</b>	<b>51</b>
<b>g</b>	<b>lterror.dtx</b>	<b>55</b>
<b>14</b>	<b>Error handling</b>	<b>55</b>
	14.1 General commands . . . . .	55
	14.2 Specific errors . . . . .	60
<b>h</b>	<b>ltpar.dtx</b>	<b>64</b>
<b>15</b>	<b>Paragraphs</b>	<b>64</b>
	15.1 Implementation . . . . .	64
<b>i</b>	<b>ltspace.dtx</b>	<b>66</b>

<b>16 Spacing</b>	<b>66</b>
16.1 User Commands . . . . .	66
16.2 Chris' comments . . . . .	66
16.3 Some immediate actions . . . . .	68
16.4 The code . . . . .	69
16.5 Vertical spacing . . . . .	74
16.6 Horizontal space (and breaks) . . . . .	77
 <b>j ltlogos.dtx</b>	 <b>80</b>
<b>17 Logos</b>	<b>80</b>
 <b>k ltfiles.dtx</b>	 <b>81</b>
<b>18 File Handling</b>	<b>81</b>
18.1 Safe Input Macros . . . . .	87
18.2 Listing files . . . . .	89
 <b>l ltoutenc.dtx</b>	 <b>91</b>
<b>19 Font encodings</b>	<b>91</b>
19.1 Removing encoding-specific commands . . . . .	93
19.2 The order of declarations . . . . .	94
19.3 Docstrip modules . . . . .	94
19.4 Definitions for the kernel . . . . .	94
19.4.1 Declaration commands . . . . .	94
19.4.2 Hyphenation . . . . .	101
19.4.3 Miscellania . . . . .	102
19.4.4 Default encodings . . . . .	102
19.4.5 Math material . . . . .	104
19.5 Definitions for the OT1 encoding . . . . .	105
19.6 Definitions for the T1 encoding . . . . .	107
19.7 Definitions for the OMS encoding . . . . .	111
19.8 Definitions for the OML encoding . . . . .	112
19.9 Definitions for the OT4 encoding . . . . .	112
19.10Definitions for the TS1 encoding . . . . .	114
 <b>20 Package files</b>	 <b>118</b>
20.1 The fontenc package . . . . .	118
20.2 The textcomp package . . . . .	120
20.2.1 Supporting oldstyle digits . . . . .	129
20.2.2 Subset encoding defaults . . . . .	129
 <b>m ltcounts.dtx</b>	 <b>132</b>
<b>21 Counters and Lengths</b>	<b>132</b>
21.1 Environment Counter Macros . . . . .	132

<b>n</b> <code>ltnlength.dtx</code>	<b>138</b>
<b>22 Lengths</b>	<b>138</b>
<b>o</b> <code>ltfssbas.dtx</code>	<b>139</b>
<b>23 Preliminary macros</b>	<b>139</b>
<b>24 Macros for setting up the tables</b>	<b>140</b>
<b>25 Selecting a new font</b>	<b>145</b>
25.1   Macros for the user . . . . .	145
25.2   Macros for loading fonts . . . . .	149
<b>26 Assigning math fonts to <i>versions</i></b>	<b>154</b>
<b>p</b> <code>ltfssstrc.dtx</code>	<b>159</b>
<b>27 Introduction</b>	<b>159</b>
<b>28 A driver for this document</b>	<b>159</b>
<b>29 The Implementation</b>	<b>160</b>
<b>30 Handling Options</b>	<b>160</b>
<b>31 Macros common to <code>fam.tex</code> and <code>tracefmt.sty</code></b>	<b>162</b>
31.1   General font loading . . . . .	162
31.2   Math fonts setup . . . . .	166
31.2.1   Outline of algorithm for math font sizes . . . . .	166
31.2.2   Code for math font size setting . . . . .	168
31.2.3   Other code for math . . . . .	169
<b>32 Scaled font extraction</b>	<b>171</b>
32.1   Sizefunctions . . . . .	178
<b>q</b> <code>ltfsscmp.dtx</code>	<b>181</b>
<b>r</b> <code>ltfssdcl.dtx</code>	<b>185</b>
<b>33 Interface Commands</b>	<b>185</b>
<b>s</b> <code>ltfssini.dtx</code>	<b>208</b>
<b>34 NFSS Initialisation</b>	<b>208</b>
34.1   Providing math <i>versions</i> . . . . .	208
34.2   Miscellaneous . . . . .	210

<b>t fontdef.dtx</b>	<b>214</b>
<b>35 Introduction</b>	<b>214</b>
<b>36 Customization</b>	<b>214</b>
<b>37 The docstrip modules</b>	<b>215</b>
<b>38 A driver for this document</b>	<b>215</b>
<b>39 The fonttext.ltx file</b>	<b>216</b>
39.1 Encodings . . . . .	216
39.2 Defaults . . . . .	217
<b>40 The fontmath.ltx file</b>	<b>218</b>
40.1 The font encodings used . . . . .	218
40.1.1 Symbolfont and Alphabet declarations . . . . .	219
40.2 Math font sizes . . . . .	219
40.3 The math symbol assignments . . . . .	220
40.3.1 The letters . . . . .	220
40.3.2 The digits . . . . .	221
40.3.3 Punctuation, brace, etc. keys . . . . .	221
40.3.4 Delimitercodes for characters . . . . .	222
40.4 Symbols accessed via control sequences . . . . .	222
40.4.1 Greek letters . . . . .	222
40.4.2 Ordinary symbols . . . . .	223
40.4.3 Large Operators . . . . .	224
40.4.4 Binary symbols . . . . .	224
40.4.5 Relations . . . . .	225
40.4.6 Arrows . . . . .	226
40.4.7 Punctuation symbols . . . . .	227
40.4.8 Math accents . . . . .	227
40.4.9 Radicals . . . . .	227
40.4.10 Over and under something, etc . . . . .	228
40.4.11 Delimiters . . . . .	228
40.5 Math versions of text commands . . . . .	229
40.6 Other special functions and parameters . . . . .	230
40.6.1 Biggggg . . . . .	230
40.6.2 The log-like functions . . . . .	230
40.6.3 Parameters . . . . .	230
<b>41 Default cfg files</b>	<b>230</b>
<b>u preload.dtx</b>	<b>232</b>
<b>42 Overview</b>	<b>232</b>
<b>43 Customization</b>	<b>232</b>
<b>44 Module switches for the DOCSTRIP program</b>	<b>233</b>

<b>45 A driver for this document</b>	<b>233</b>
<b>46 The code</b>	<b>233</b>
<b>v ltfntcmd.dtx</b>	<b>236</b>
<b>47 Introduction</b>	<b>236</b>
<b>48 The implementation</b>	<b>238</b>
<b>49 Initialization</b>	<b>243</b>
<b>w ltpageno.dtx</b>	<b>244</b>
<b>50 Page Numbering</b>	<b>244</b>
<b>x ltxref.dtx</b>	<b>245</b>
<b>51 Cross Referencing</b>	<b>245</b>
51.1 Cross Referencing . . . . .	245
51.2 An extension of counter referencing . . . . .	247
<b>y ltmiscen.dtx</b>	<b>249</b>
<b>52 Miscellaneous Environments</b>	<b>249</b>
52.1 Environments . . . . .	249
52.2 Center, Flushright, Flushleft . . . . .	253
52.3 Verbatim . . . . .	255
<b>z ltmath.dtx</b>	<b>258</b>
<b>53 Math setup</b>	<b>258</b>
53.1 Math commands based on plain $\TeX$ . . . . .	258
53.1.1 The log-like functions . . . . .	258
53.1.2 Biggggg . . . . .	259
53.1.3 The UNSORTED Rest . . . . .	259
53.2 Math Environments . . . . .	262
53.3 External options to the standard document classes . . . . .	266
53.3.1 Left equation numbering . . . . .	266
53.3.2 Flush left equations . . . . .	266
<b>A ltlists.dtx</b>	<b>269</b>

<b>54 List, and related environments</b>	<b>269</b>
54.1 List and Trivlist	270
54.2 Vertical Spacing (skips)	271
54.3 Penalties	271
54.4 Horizontal Spacing (dimens)	271
54.5 Default Values	272
54.6 Itemize and Enumerate	282
 <b>B ltboxes.dtx</b>	 <b>284</b>
<b>55 L<sup>A</sup>T<sub>E</sub>X Box commands</b>	<b>284</b>
55.1 Some low-level constructs	295
 <b>C lttab.dtx</b>	 <b>296</b>
<b>56 Tabbing, Tabular and Array Environments</b>	<b>296</b>
56.1 tabbing	296
56.2 array and tabular environments	304
 <b>D ltpictur.dtx</b>	 <b>318</b>
<b>57 Picture Mode</b>	<b>318</b>
57.1 Curves	336
 <b>E ltthm.dtx</b>	 <b>339</b>
<b>58 Theorem Environments</b>	<b>339</b>
 <b>F ltsect.dtx</b>	 <b>343</b>
<b>59 Sectioning Commands</b>	<b>343</b>
59.1 The Title	343
59.2 Sectioning	344
59.2.1 Initializations	350
59.3 Table of Contents etc.	350
59.3.1 Convention	350
59.3.2 Commands	350
 <b>G ltfloat.dtx</b>	 <b>353</b>
<b>60 Floats</b>	<b>353</b>
60.1 Floating Environments	353
60.2 Footnotes	366

<b>H</b>	<b>ltxglo.dtx</b>	<b>373</b>
<b>61</b>	<b>Index and Glossary Generation</b>	<b>373</b>
<b>I</b>	<b>ltxbibl.dtx</b>	<b>375</b>
<b>62</b>	<b>Bibliography Generation</b>	<b>375</b>
62.1	Default definitions . . . . .	378
<b>J</b>	<b>ltxpage.dtx</b>	<b>379</b>
<b>63</b>	<b>Page styles and related commands</b>	<b>379</b>
63.1	Page Style Commands . . . . .	379
63.2	How a page style makes running heads and feet . . . . .	379
63.3	marking conventions . . . . .	379
<b>K</b>	<b>ltxoutput.dtx</b>	<b>382</b>
<b>64</b>	<b>Output Routine</b>	<b>382</b>
64.1	Floats . . . . .	382
64.1.1	Kludgeins . . . . .	435
64.1.2	Float control . . . . .	436
64.1.3	Float placement parameters . . . . .	448
<b>L</b>	<b>ltxclass.dtx</b>	<b>451</b>
<b>65</b>	<b>Introduction</b>	<b>451</b>
<b>66</b>	<b>User interface</b>	<b>451</b>
66.1	Option processing . . . . .	452
<b>67</b>	<b>Class and Package interface</b>	<b>452</b>
67.1	Class name and version . . . . .	452
67.2	Package name and version . . . . .	453
67.3	Requiring other packages . . . . .	453
67.4	Declaring new options . . . . .	454
67.5	Safe Input Macros . . . . .	454
<b>68</b>	<b>Implementation</b>	<b>455</b>
68.1	Hooks . . . . .	465
68.2	Providing shipment . . . . .	466
<b>69</b>	<b>After Preamble</b>	<b>468</b>
<b>M</b>	<b>ltxhyphen.dtx</b>	<b>470</b>

<b>N</b>	<b>ltnatex.dtx</b>	<b>472</b>
<b>70</b>	<b>Overview</b>	<b>472</b>
<b>71</b>	<b>Core T<sub>E</sub>X functionality</b>	<b>472</b>
<b>72</b>	<b>Plain T<sub>E</sub>X interface</b>	<b>473</b>
<b>73</b>	<b>Lua functionality</b>	<b>473</b>
73.1	Allocators in Lua . . . . .	473
73.2	Lua access to T <sub>E</sub> X register numbers . . . . .	473
73.3	Module utilities . . . . .	475
73.4	Callback management . . . . .	475
<b>74</b>	<b>Implementation</b>	<b>476</b>
74.1	Minimum LuaT <sub>E</sub> X version . . . . .	476
74.2	Older L <sup>A</sup> T <sub>E</sub> X/Plain T <sub>E</sub> X setup . . . . .	476
74.2.1	Fixes to <code>etex.src/etex.sty</code> . . . . .	476
74.2.2	luatex specific settings . . . . .	477
74.3	Attributes . . . . .	478
74.4	Category code tables . . . . .	478
74.5	Named Lua functions . . . . .	479
74.6	Custom whatsits . . . . .	480
74.7	Lua bytecode registers . . . . .	480
74.8	Lua chunk registers . . . . .	480
74.9	Lua loader . . . . .	480
74.10	Lua module preliminaries . . . . .	481
74.11	Lua module utilities . . . . .	482
74.11.1	Module tracking . . . . .	482
74.11.2	Module messages . . . . .	482
74.12	Accessing register numbers from Lua . . . . .	483
74.13	Attribute allocation . . . . .	484
74.14	Custom whatsit allocation . . . . .	485
74.15	Bytecode register allocation . . . . .	485
74.16	Lua chunk name allocation . . . . .	485
74.17	Lua callback management . . . . .	486
74.17.1	Housekeeping . . . . .	486
74.17.2	Handlers . . . . .	488
74.17.3	Public functions for callback management . . . . .	489
<b>O</b>	<b>ltfinal.dtx</b>	<b>494</b>
<b>75</b>	<b>Final settings</b>	<b>494</b>
75.1	Debugging . . . . .	494
75.2	Typesetting parameters . . . . .	494
75.3	Lccodes for hyphenation . . . . .	495
75.4	Hyphenation . . . . .	497
75.5	Font loading . . . . .	498
75.6	Input encoding . . . . .	498
75.7	Lccodes and uccodes . . . . .	499

75.8 Applying Patch files . . . . .	501
75.9 Freeing Memory . . . . .	502
75.10Initialise file list . . . . .	502
75.11Dumping the format . . . . .	503

# File a

## ltdirchk.dtx

### 1 L<sup>A</sup>T<sub>E</sub>X System Dependent Initialisations

This file implements the semi-automatic determination of various system dependent parts of the initialisation. The actual definitions may be placed in a file `texsys.cfg`. Thus for operating systems for which the tests here do not result in acceptable settings, a ‘hand written’ `texsys.cfg` may be produced.

The macros that must be defined are:

`\@currdir`      `\@currdir{filename}<space>` should expand to a form of the filename that uniquely refers to the ‘current directory’ if this is possible. (The expansion should also end with a space.) on UNIX, this is `\def\@currdir{./}`. For more exotic operating systems you may want to make `\@currdir` a macro with arguments delimited by `.` and/or `<space>`. If the operating system has no concept of directory structure, this macro should be defined to be empty.

`\input@path`      If the primitive `\openin` searches the same directories as the primitive `\input`, then it is possible to tell (using `\ifeof`) whether a file exists before trying to input it. For systems like this, `\input@path` should be left undefined.

If `\openin` does not ‘follow’ `\input` then `\input@path` must be defined to be a list of directories to search for input files. The format for each directory is as for `\@currdir`, normally just a prefix is required, but it may be a macro with space-delimited argument. That is, if `<dir>` is an entry in the input path, T<sub>E</sub>X will try to load the expansion of `<dir><filename><space>`

So either `<dir>` should be defined as a macro with argument delimited by space, or it should just expand to a directory name, including the final directory separator, so that it may be concatenated with the `<filename>`. This means that for UNIX-like syntax, each `<dir>` should end with a slash, `/`.

`\input@path` should expand to a list of such directories, each in a `{}` group.

`\filename@parse`      After a call of the form: `\filename@parse{<filename>}`, the three macros `\filename@area`, `\filename@base`, `\filename@ext` should be defined to be the ‘area’ (or directory), basename and extension respectively. If there was no extension specified in `<filename>`, `\filename@ext` should be `\let` to `\relax` (so this case may be tested with `\ifundefined{filename@ext}` and, perhaps a default extension substituted).

Normally one would not need to define this macro in `texsys.cfg` as the automatic tests can supply parsers that work with UNIX and VMS and Macintosh syntax, as well as a basic parser that will cover many other cases. However some operating systems may need a ‘hand produced’ parser in which case it should be defined in this file.

The UNIX parser also works for most MSDOS T<sub>E</sub>X versions. Currently if the UNIX, VMS or Macintosh parser is not used, `\filename@parse` is defined to always return an empty area, and to split the argument into basename and extension at the first ‘.’ that occurs in the name. Parsers for other formats may be defined in `texsys.cfg`, in which case they will be used in preference to the default definitions.

`\@TeXversion`      `\@TeXversion` is now set automatically by the initialisation tests in this file. You should not need to set it in `texsys.cfg`, however the following documentation

is left for information. L<sup>A</sup>T<sub>E</sub>X does not set this variable exactly, the automatic tests set it to:

2 for any version,  $v$ ,  $v < 3.0$   
 3 for any version,  $v$ ,  $3.0 \leq v \leq 3.14$   
 $\langle undefined \rangle$  otherwise.

However these values are accurate enough for L<sup>A</sup>T<sub>E</sub>X to take appropriate action for these old T<sub>E</sub>Xs.

If your T<sub>E</sub>X is older than version 3.141, then you should define `\@TeXversion` (using `\def`) to be the version number. If you do not do this<sup>1</sup>, L<sup>A</sup>T<sub>E</sub>X will not work around a bug in old T<sub>E</sub>X versions, and so error messages will appear in a very strange format, with `^^J` appearing instead of line breaks:

```
! LaTeX Error: \rubbish undefined.^^J^^JSee the LaTeX manual or LaTeX Companion
for explanation.^^JType H <return> for immediate help.
...
```

```
1.3 \renewcommand{\rubbish}
      {}
?
```

However if you put `\def\@TeXversion{3.14}` in `texsys.cfg` the following format will be used:

```
! LaTeX Error: \rubbish undefined.
```

```
See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
! .
...
```

```
1.3 \renewcommand{\rubbish}
      {}
?
```

Note that this has an extra line `! .` which does not appear in error messages that use the default settings with a current version of T<sub>E</sub>X, but this should not cause any confusion we hope.

## 2 Initialisation

As this file is read at a very early stage, some definitions that are normally considered to be part of the format must be made here.

### 2.1 INITEX

```
1 \<dircheck>
2 \<initex>
3 \<initex>\ifnum\catcode'\{=1
4 \<initex> \errmessage
5 \<initex> {LaTeX must be made using an initex with no format preloaded}
```

---

<sup>1</sup>Actually if your T<sub>E</sub>X is really old, version 2, L<sup>A</sup>T<sub>E</sub>X can detect this, and sets `\@TeXversion` to 2 if it is not set in the `cfg` file.

```

6 <initex>\fi
7 \catcode'\{=1
8 \catcode'\}=2

```

If LuaTeX is in use the extensions and other new primitives have to be activated: this is done as early as possible. Older versions of LuaTeX do not hide the primitives: a version check is not needed as the version itself will be missing in the case where action is needed!

```

9 \ifx\directlua\undefined
10 \else
11 \ifx\luatexversion\undefined

```

Enable e-TeX/pdfTeX/Umath primitives with their natural names

```

12 \directlua{tex.enableprimitives("",%
13 tex.extraprimitives('etex', 'pdftex', 'umath'))}

```

In current formats enable primitives with unprefix names. the latexrelease guards allow the primitives to be defined with a \luatex prefix if older formats are specified.

```

14 </initex>
15 </dircheck>
16 <*initex, latexrelease>
17 <latexrelease>\ifx\directlua\undefined\else
18 <latexrelease>\IncludeInRelease{2015/10/01}{\luatexluafunction}
19 <latexrelease>{LuaTeX (prefixed names)}%
20 \directlua{tex.enableprimitives("",%
21 tex.extraprimitives("omega", "aleph", "luatex"))}
22 <latexrelease>\EndIncludeInRelease
23 <latexrelease>\IncludeInRelease{0000/00/00}{\luatexluafunction}
24 <latexrelease>{LuaTeX (prefixed names)}%
25 <latexrelease>\directlua{
26 <latexrelease> tex.enableprimitives(
27 <latexrelease> "luatex",
28 <latexrelease> tex.extraprimitives("core","omega", "aleph", "luatex")
29 <latexrelease> )
30 <latexrelease> local i
31 <latexrelease> local t = { }
32 <latexrelease> for _,i in pairs(tex.extraprimitives("luatex")) do
33 <latexrelease> if not string.match(i,"^U") then
34 <latexrelease> if not string.match(i, "^luatex") then
35 <latexrelease> table.insert(t,i)
36 <latexrelease> end
37 <latexrelease> else
38 <latexrelease> if string.match(i,"^Uchar$") then
39 <latexrelease> table.insert(t,i)
40 <latexrelease> end
41 <latexrelease> end
42 <latexrelease> end
43 <latexrelease> for _,i in pairs(t) do
44 <latexrelease> tex.print(
45 <latexrelease> "\noexpand\\let\noexpand\\" .. i
46 <latexrelease> .. "\noexpand\\undefined"
47 <latexrelease> )
48 <latexrelease> end
49 <latexrelease>}
50 <latexrelease>\EndIncludeInRelease

```

```

51 \latexrelease\fi
52 \end{initex, latexrelease}
53 \end{dircheck}
54 \end{initex}

55 \fi
56 \fi

```

That distraction over, back to the basics of a format.

```

57 \catcode'\#=6
58 \catcode'\^=7
59 \chardef\active=13
60 \catcode'\@=11
61 \countdef\count@=255
62 \let\bgroup={ \let\egroup=}
63 \ifx@@input\undefined\let@@input\input\fi
64 \ifx@@end\undefined\let@@end\end\fi
65 \chardef\@inputcheck0
66 \chardef\sixt@n=16
67 \newlinechar'\^^J
68 \def\typeout{\immediate\write17}
69 \def\dospecials{\do\ \do\\\do\{\do\}\do\$\do\&%
70 \do\#\do\^\do\_ \do\% \do\~}
71 \def\makeoother#1{\catcode'#1=12\relax}
72 \def\space{ }
73 \def\@tempswafalse{\let\if@tempswa\iffalse}
74 \def\@tempswatrue{\let\if@tempswa\iftrue}
75 \let\if@tempswa\iffalse
76 \def\loop#1\repeat{\def\iterate{#1\relax\expandafter\iterate\fi}%
77 \iterate \let\iterate\relax}
78 \let\repeat\fi
79 \end{initex}

```

## 2.2 Some bits of 2e

```

80 \end{2kernel}
81 \def\two@digits#1{\ifnum#1<10 0\fi\number#1}
82 \long\def\firstoftwo#1#2{#1}
83 \long\def\secondoftwo#1#2{#2}

```

This is a special version of \ProvidesFile for initex use.

```

84 \def\ProvidesFile#1{%
85 \begingroup
86 \catcode'\ 10 %
87 \ifnum \endlinechar<256 %
88 \ifnum \endlinechar>\m@ne
89 \catcode\endlinechar 10 %
90 \fi
91 \fi
92 \makeoother\%
93 \ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]
94 \def\@providesfile#1[#2]{%
95 \wlog{File: #1 #2}%
96 \@addtofilelist{ #2}%
97 \endgroup}
98 \long\def\@addtofilelist#1{}

```

```

99 \def\@empty{}
100 \catcode'\%=12
101 \def\@percentchar{%}
102 \catcode'\%=14
103 \let\@currdir\@undefined
104 \let\input@path\@undefined
105 \let\filename@parse\@undefined

\strip@prefix
106 \def\strip@prefix#1>{}
107 </2ekernel>

```

### 3 texsys.cfg

As mentioned above, any site specific definitions required to describe the filename handling must be entered into a file `texsys.cfg`. If `texsys.cfg` can not be located by `\openin`, we write a default version out. The default version only contains comments, so we do not actually input the file in that case. The automatic tests later will, hopefully, correctly define the required macros.

The tricky code below checks to see if `texsys.cfg` exists. If it does not, all the text in this file between `START` and `END` is copied verbatim to a new file `texsys.cfg`. If `texsys.cfg` is found, then it is simply input. This is only done when this file is being used unstripped.

```

108 <*docstrip>
109 \openin15=texsys.cfg
110 \ifeof15
111 \typeout{** Writing a default texsys.cfg}
112 \immediate\openout15=texsys.cfg
113 \begingroup
114 \catcode'\^M\active%
115 \let^M\par%
116 \def\reserved@a#1^M{%
117   \def\reserved@b{#1}%
118   \ifx\reserved@b\reserved@c\endgroup\else%
119     \immediate\write15{#1}%
120     \expandafter\reserved@a{fi}%
121   \def\reserved@d#1START^M{\let\do\makeother\dospecials\reserved@a}%
122   \catcode'\%=12
123   \def\reserved@c{END}
124   \reserved@d

```

START

#### 3.1 texsys.cfg

This file contains the site specific definitions of the four macros `\@currdir`, `\input@path`, `\filename@parse` and `\TeXversion`.

As distributed it only contains comments, however this ‘empty’ file will work on many systems because of the automatic tests built into `ltdirchk.dtx`. You *are* allowed to edit this file to add definitions of these macros appropriate to your system.

The macros that must be defined are:

`\@currdir`      `\@currdir{filename}⟨space⟩` should expand to a form of the filename that uniquely refers to the ‘current directory’ if this is possible. (The expansion should also end with a space.) on UNIX, this is `\def\@currdir{./}`. For more exotic operating systems you may want to make `\@currdir` a macro with arguments delimited by `.` and/or `⟨space⟩`. If the operating system has no concept of directory structure, this macro should be defined to be empty.

`\input@path`      If the primitive `\openin` searches the same directories as the primitive `\input`, then it is possible to tell (using `\ifeof`) whether a file exists before trying to input it. For systems like this, `\input@path` should be left undefined.

                  If `\openin` does not ‘follow’ `\input` then `\input@path` must be defined to be a list of directories to search for input files. The format for each directory is as for `\@currdir`, normally just a prefix is required, but it may be a macro with space-delimited argument. That is, if `⟨dir⟩` is an entry in the input path, `TeX` will try to load the expansion of

`⟨dir⟩⟨filename⟩⟨space⟩`

                  So either `⟨dir⟩` should be defined as a macro with argument delimited by space, or it should just expand to a directory name, including the final directory separator, so that it may be concatenated with the `⟨filename⟩`. This means that for UNIX-like syntax, each `⟨dir⟩` should end with a slash, `/`. One exception to this rule is that the input path should *always* contain the empty directory `{}` as this will allow ‘full pathnames’ to be used, and the ‘current directory’ to be searched.

`\filename@parse`      `\input@path` should expand to a list of such directories, each in a `{}` group.

                  After a call of the form: `\filename@parse{⟨filename⟩}`, the three macros `\filename@area`, `\filename@base`, `\filename@ext` should be defined to be the ‘area’ (or directory), basename and extension respectively. If there was no extension specified in `⟨filename⟩`, `\filename@ext` should be `\let` to `\relax` (so this case may be tested with `\@ifundefined{filename@ext}` and, perhaps a default extension substituted).

                  Normally one would not need to define this macro in `texsys.cfg` as the automatic tests can supply parsers that work with UNIX and VMS syntax, as well as a basic parser that will cover many other cases. However some operating systems may need a ‘hand produced’ parser in which case it should be defined in this file.

                  The UNIX parser also works for most MSDOS `TeX` versions. Currently if the UNIX or VMS parser is not used, `\filename@parse` is defined to always return an empty area, and to split the argument into basename and extension at the first ‘.’ that occurs in the name. Parsers for other formats may be defined in `texsys.cfg`, in which case they will be used in preference to the default definitions.

`\@TeXversion`      You should not need to set this macro in `texsys.cfg`. `LaTeX` tests to set this automatically. See the comments in the opening section of `ltdirchk.dtx`.

                  The following sections give examples of definitions which might work on various systems. These are currently mainly untested as I only have access to a few systems, all of which do not need this file as the automatic tests work. All the code is commented out.

### 3.2 UNIX (web2c)

This implementation does make `\openin` and `\input` look in the same places. Acceptable settings are made by `ltdirchk.dtx`, and so this file may be empty. The definitions below are therefore just for information.

```
125 %\def\@currdir{./}
```

```
126 % \let\input@path\@undefined
```

### 3.3 UNIX (other)

Apparently some commercial UNIX implementations have different paths for `\openin` and `\input`. For these one could use definitions like the following (with whatever directories are used at your site): note that the directory names should end with `/`.

```
127 % \def\currdir{./}
128 % \def\input@path{%
129 %   {/usr/local/lib/tex/inputs/distrib/}%
130 %   {/usr/local/lib/tex/inputs/contrib/}%
131 %   {/usr/local/lib/tex/inputs/local/}%
132 % }
```

### 3.4 MSDOS (emtex)

This implementation does make `\openin` and `\input` look in the same places. Acceptable settings are made by `ltdirchk.dtx`, and so this file may be empty. The definitions below are therefore just for information.

```
133 % \def\currdir{./}
134 % \let\input@path\@undefined
```

### 3.5 MSDOS (other)

Some PC implementations have different paths for `\openin` and `\input`. For these one could use definitions like the following (with whatever directories are used at your site): note that the directory names should end with `/`. This assumes the implementation uses UNIX style `/` as the directory separator.

```
135 % \def\currdir{./}
136 % \def\input@path{%
137 %   {c:/tex/inputs/distrib/}%
138 %   {c:/tex/inputs/contrib/}%
139 %   {c:/tex/inputs/local/}%
140 % }
```

### 3.6 VMS (DECUS T<sub>E</sub>X, PD VMS 3.6)

This implementation does make `\openin` and `\input` look in the same places. Acceptable settings are made by `ltdirchk.dtx`, and so this file may be empty. The definitions below are therefore just for information.

```
141 % \def\currdir{[] }
142 % \let\input@path\@undefined
```

### 3.7 VMS (???)

Some VMS implementations have different paths for `\openin` and `\input`. For these one could use definitions like the following:

```
143 % \def\currdir{[] }
144 % \def\input@path{%
145 %   {tex_inputs:}%
146 % }
```

```

146 %   {SOMEDISK:[SOME.TEX.DIRECTORY]]}%
147 % }

```

### 3.8 MACINTOSH (OzTeX 1.6)

This implementation does make `\openin` and `\input` look in the same places. Acceptable settings are made by `ltdirchk.dtx`, and so this file may be empty. The definitions below are therefore just for information.

```

148 % \def\@currdir{:}
149 % \let\input@path\@undefined

```

### 3.9 MACINTOSH (other)

Some Macintosh implementations have different paths for `\openin` and `\input`. For these one could use definitions like the following (with whatever folders are used on your machine): note that the directory names should end with `:`, and they should contain *no* spaces.

```

150 % \def\@currdir{:}
151 % \def\input@path{%
152 %   {Hard-Disk:Applications:TeX:TeX-inputs:}%
153 %   {Hard-Disk:Applications:TeX:My-inputs:}%
154 % }

```

### 3.10 FAKE EXAMPLE

This example is for an operating system that has filenames of the form `<area>name`. For maximum compatibility with macro sets, you want `name.ext` to be mapped to `<ext>name`. and `<area>name.ext` to be mapped to `<area.ext>name`. `\input` does this mapping automatically, but `\openin` does not, and does not look in the same places as `\input`. `<>name` is the desired ‘current directory’ syntax.

the following code would possibly work:

```

155 % \def\@dir#1#2 {%
156 %   \@d@r{#1}#2..\@nil}
157 % \def\@d@r#1#2.#3.#4\@nil{%
158 %   <\ifx\@dir#1\@dir\else#1\ifx\@dir#3\@dir\else.\fi\fi#3>#2 }
159 %
160 % \def\@currdir{\@dir{}}
161 % \def\input@path{%
162 %   {\@dir{area.one}}}%
163 %   {\@dir{area.two}}}%
164 % }

```

END

```

165 \immediate\closeout15

```

If `texsys.cfg` did exist, then input it.

```

166 \else
167 \typeout{** Using the existing texsys.cfg}
168 \closein15
169 \input texsys.cfg
170 \fi
171 </docstrip>

```

If the stripped version of this file is being used (in latex2e.ltx) then texsys.cfg should be there, so just input it.

172 <dircheck>\input texsys.cfg

## 4 Setting \@currdir

**\@currdir** This is a local definition of \IfFileExists. It tries to relocate texsys.aux. If  
**\IfFileExists** it succeeds, then the \@currdir syntax has been determined. If all the tests fail then \@currdir will be set to \@empty, and ltxcheck will warn of this when it checks the format.

```
173 \begingroup
174 \count@ \time
175 \divide \count@ 60
176 \count2 = - \count@
177 \multiply \count2 60
178 \advance \count2 \time
```

**\today** The current date and time stamp.

```
179 \edef \today {%
180   \the \year / \two@digits {\the \month} / \two@digits {\the \day} : %
181   \two@digits {\the \count@} : \two@digits {\the \count2}}
```

Create a file texsys.aux (hopefully in the current directory), then try to locate it again.

```
182 \immediate \openout15 = texsys.aux
183 \immediate \write15 {\today ^^J}
184 \immediate \closeout15 %
```

#1 is the file to try, #2 is what to do on success, #3 on failure.

```
185 \def \IfFileExists #1#2#3 {%
186   \openin \@inputcheck #1 %
187   \ifeof \@inputcheck
188     #3 \relax
189   \else
190     \read \@inputcheck to \reserved@a
191     \ifx \reserved@a \today
192       \typeout {#1 found} #2 \relax
193     \else
194       \typeout {BAD: old file \reserved@a (should be \today)} %
195       #3 \relax
196     \fi
197   \fi
198   \closein \@inputcheck}
199 \endlinechar = -1
```

If \@currdir has not been pre-defined in texsys.cfg then test for UNIX, VMS and Oz-T<sub>E</sub>X-Mac. syntax.

```
200 \ifx \@currdir \@undefined
201   \IfFileExists {./texsys.aux} {\gdef \@currdir {./}} %
202   {\IfFileExists {[]texsys.aux} {\gdef \@currdir {[]}} %
203   {\IfFileExists {:texsys.aux} {\gdef \@currdir {:}} {}}}
```

If it is still undefined at this point, all the above tests failed. Earlier versions interactively prompted for a definition at this point, but it seems impossible to reliably obtain information from users at this point in the installation. This version of the file produces a format with no user-interaction. Later if the format is not suitable for the system, `texsys.cfg` may be edited and the format re-made.

```

204 \ifx\@currdir\undefined
205 \global\let\@currdir\empty
206 \typeout{^^J^^J%
207     !! No syntax for the current directory could be found^^J%
208 }%
209 \fi

```

Otherwise `\@currdir` was defined in `texsys.cfg`. In this case check that the syntax specified works on this system. (In case a complete L<sup>A</sup>T<sub>E</sub>X system has been copied from one system to another.) If the test fails, give up. The installer should remove or correct the offending `texsys.cfg` and try again.

```

210 \else
211 \IfFileExists{\@currdir texsys.aux}{\}{%
212 \edef\reserved@a{\errhelp{%
213     texsys.cfg specifies the current directory syntax to be^^J%
214     \meaning\@currdir^^J%
215     but this does not work on this system.^^J%
216     Remove texsys.cfg and restart.}}\reserved@a
217 \errmessage{Bad texsys.cfg file: \noexpand\@currdir}\@@end}

```

The version of `\@currdir` in `texsys.cfg` looks OK.

```

218 \fi

219 \immediate\closeout15 %
220 \endgroup

221 \typeout{^^J^^J%
222     \noexpand\@currdir set to:
223     \expandafter\strip@prefix\meaning\@currdir.^^J%
224 }

```

Stop here if the file is being used unstripped.

```

225 <*\docstrip>
226 \relax\endinput
227 </docstrip>

```

## 5 Setting `\input@path`

Earlier versions of this file attempted to automatically test whether `\input@path` was required, and interactively prompt for a path if necessary. This was not found to be very reliable. The first-time installer of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> can not be expected to have enough information to supply the correct information to the prompts. Now the interaction is omitted. After the format is made the installer can attempt to run the test document `ltxcheck.tex` through L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. This will check, amongst other things, whether `texsys.cfg` will need to be edited and the format remade.

`\input@path` Now set up the `\input@path`.

`\input@path` should either be undefined, or a list of directories as described in the introduction.

```

228 \typeout{^^J%
229     Assuming \noexpand\openin and \noexpand\input^^J%
230     \ifx\input@path\@undefined
\input@path has not been pre-defined.
231     have the same search path.^^J%
232     \else
\input@path has been defined in texsys.cfg.
233     have different search paths.^^J%
234     LaTeX will use the path specified by \noexpand\input@path:^^J%
235     \fi
236 }
```

## 6 Filename Parsing

`\filename@parse` Split a filename into its components.

```

237 \ifx\filename@parse\@undefined
238 \def\reserved@a{.}\ifx\currdir\reserved@a
\filename@parse was not specified in texsys.cfg, but \@currdir looks like
UNIX...
```

```

239 \typeout{^^JDefining UNIX/DOS style filename parser.^^J}
240 \def\filename@parse#1{%
241     \let\filename@area\@empty
242     \expandafter\filename@path#1/\}
```

Search for the last /.

```

243 \def\filename@path#1/#2\{%
244     \ifx\#2\%
245         \def\reserved@a{\filename@simple#1.\}%
246     \else
247         \edef\filename@area{\filename@area#1/}%
248         \def\reserved@a{\filename@path#2\}%
249     \fi
250     \reserved@a}
```

```

251 \else\def\reserved@a{[]}\ifx\currdir\reserved@a
```

`\filename@parse` was not specified in texsys.cfg, but \@currdir looks like VMS...

```

252 \typeout{^^JDefining VMS style filename parser.^^J}
253 \def\filename@parse#1{%
254     \let\filename@area\@empty
255     \expandafter\filename@path#1[]\}
```

Search for the last ].

```

256 \def\filename@path#1[]#2\{%
257     \ifx\#2\%
258         \def\reserved@a{\filename@simple#1.\}%
259     \else
260         \edef\filename@area{\filename@area#1[]}%
261         \def\reserved@a{\filename@path#2\}%

```

```

262     \fi
263     \reserved@a}

264 \else\def\reserved@a{:}\ifx\@currdir\reserved@a

\filename@parse was not specified in texsys.cfg, but \@currdir looks like Mac-
intosh...

265     \typeout{^^JDefining Mac style filename parser.^^J}
266     \def\filename@parse#1{%
267         \let\filename@area\@empty
268         \expandafter\filename@path#1:\}

    Search for the last :.

269     \def\filename@path#1:#2\{%
270         \ifx\\#2\%
271             \def\reserved@a{\filename@simple#1.\}%
272         \else
273             \edef\filename@area{\filename@area#1:}%
274             \def\reserved@a{\filename@path#2\}%
275         \fi
276         \reserved@a}

277 \else

\filename@parse was not specified in texsys.cfg. So just make a simple parser
that always sets \filename@area to empty.

278     \typeout{^^JDefining generic filename parser.^^J}
279     \def\filename@parse#1{%
280         \let\filename@area\@empty
281         \expandafter\filename@simple#1.\}
282     \fi\fi\fi

    \filename@simple is used by all three versions. Finally we can split off the
    extension.

283     \def\filename@simple#1.#2\{%
284         \ifx\\#2\%
285             \let\filename@ext\relax
286         \else
287             \edef\filename@ext{\filename@dot#2\}%
288         \fi
289         \edef\filename@base{#1}

    Remove a final dot, added earlier.

290     \def\filename@dot#1.\{#1}

291 \else

Otherwise, \filename@parse was specified in texsys.cfg.

292     \typeout{^^J^^J}
293     \noexpand\filename@parse was defined in texsys.cfg:^^J%
294     \expandafter\strip@prefix\meaning\filename@parse.^^J%
295     }
296 \fi

```

## 7 T<sub>E</sub>X Versions

`\@TeXversion` T<sub>E</sub>X versions older than 3.141 require `\@TeXversion` to be set. This can be determined automatically due to a trick suggested by Bernd Raichle. (Actually this will not always get the correct version number, eg T<sub>E</sub>X3.14 would be detected as T<sub>E</sub>X3, but L<sup>A</sup>T<sub>E</sub>X only needs to take account of T<sub>E</sub>X's older than 3, or between 3 and 3.14.

```
297 \ifx\@TeXversion\@undefined
298   \ifx\@undefined\inputlineno
299     \def\@TeXversion{2}
300   \else
301     {\catcode'\^^J=\active
302      \def\reserved@a#1#2\@@{\if#1\string^3\fi}
303      \edef\reserved@a{\expandafter\reserved@a\string^^J\@@}
304      \ifx\reserved@a\empty\else\gdef\@TeXversion{3}\fi}
305   \fi
306 \fi
307 </dircheck>
```

## 8 ltxcheck.tex

After the format has been made, and `article.cls` moved with the other files to the 'standard input directory' as specified in `install.txt`, the format may be checked by running the file `ltxcheck.tex`.

## File b

# lplain.dtx

## 9 Plain T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X includes almost all of the functionality of Knuth's original 'Basic Macros'. That is, the plain T<sub>E</sub>X format described in Appendix B of the T<sub>E</sub>XBook. However, some of the user commands are not much use so, in order to save memory, we may remove them from the kernel into a package. Here is a list of the commands that may be removed (PROBABLY NOT COMPLETE).

```
\magstep      \magstephalf
\mathhexbox
\vglue        \vgl@
\hglue        \hgl@
```

This file is by now very small as most of it has been moved to more appropriate kernel files: it may disappear completely one day.

L<sup>A</sup>T<sub>E</sub>X font definitions are done using NFSS2 so none of PLAIN's font definitions are in L<sup>A</sup>T<sub>E</sub>X.

L<sup>A</sup>T<sub>E</sub>X has its own tabbing environment, so PLAIN's is disabled.

L<sup>A</sup>T<sub>E</sub>X uses its own output routine, so most of the plain one was removed.

```
1 (*2kernel)
2 \catcode'\{=1 % left brace is begin-group character
3 \catcode'\}=2 % right brace is end-group character
4 \catcode'\$=3 % dollar sign is math shift
5 \catcode'\&=4 % ampersand is alignment tab
6 \catcode'\#=6 % hash mark is macro parameter character
7 \catcode'\^=7 % circumflex and uparrow are for superscripts
8 \catcode'\_ =8 % underline and downarrow are for subscripts
9 \catcode'\^I=10 % ascii tab is a blank space
10 \chardef\active=13 \catcode'\^=\active % tilde is active
11 \catcode'\^L=\active \outer\def^L{\par}% ascii form-feed is \outer\par
12 \message{catcodes,}
```

We had to define the `\catcodes` right away, before the message line, since `\message` uses the `{` and `}` characters. When INITEX (the T<sub>E</sub>X initializer) starts up, it has defined the following `\catcode` values:

```
\catcode'\^@=9 %  ascii null is ignored
\catcode'\^M=5 %  ascii return is end-line
\catcode'\ =0 %    backslash is TeX escape character
\catcode'\%=14 %   percent sign is comment character
\catcode'\ =10 %   ascii space is blank space
\catcode'\^?=15 %  ascii delete is invalid
\catcode'\A=11 ... \catcode'\Z=11 % uppercase letters
\catcode'\a=11 ... \catcode'\z=11 % lowercase letters
all others are type 12 (other)
```

Here is a list of the characters that have been specially catcoded:

```
13 \def\dospecials{\do\ \do\\\do\{\do\}\do\$ \do\&%
14 \do\# \do\^ \do\_ \do\% \do\^}
```

(not counting ascii null, tab, linefeed, formfeed, return, delete) Each symbol in the list is preceded by , which can be defined if you want to do something to every item in the list.

We make @ signs act like letters, temporarily, to avoid conflict between user names and internal control sequences of plain format.

```
15 \catcode'@=11
```

To make the plain macros more efficient in time and space, several constant values are declared here as control sequences. If they were changed, anything could happen; so they are private symbols.

```
\@ne Small constants are defined using \chardef.
\tw@ 16 \chardef\@ne=1
\thr@@ 17 \chardef\tw@=2
\sixt@@n 18 \chardef\thr@@=3
\@cclv 19 \chardef\sixt@@n=16
        20 \chardef\@cclv=255

\@cclvi Constants above 255 defined using \mathchardef.
\@m 21 \mathchardef\@cclvi=256
\@M 22 \mathchardef\@m=1000
\@MM 23 \mathchardef\@M=10000
      24 \mathchardef\@MM=20000
```

Allocation of registers

Here are macros for the automatic allocation of \count, \box, \dimen, \skip, \muskip, and \toks registers, as well as \read and \write stream numbers, \fam codes, \language codes, and \insert numbers.

```
25 \message{registers,}
```

When a register is used only temporarily, it need not be allocated; grouping can be used, making the value previously in the register return after the close of the group. The main use of these macros is for registers that are defined by one macro and used by others, possibly at different nesting levels. All such registers should be defined through these macros; otherwise conflicts may occur, especially when two or more macro packages are being used at the same time.

The following counters are reserved:

- 0 to 9 page numbering
- 10 count allocation
- 11 dimen allocation
- 12 skip allocation
- 13 muskip allocation
- 14 box allocation
- 15 toks allocation
- 16 read file allocation
- 17 write file allocation
- 18 math family allocation
- 19 language allocation
- 20 insert allocation
- 21 the most recently allocated number
- 22 constant -1

New counters are allocated starting with 23, 24, etc. Other registers are allocated starting with 10. This leaves 0 through 9 for the user to play with safely, except that counts 0 to 9 are considered to be the page and subpage numbers (since they are displayed during output). In this scheme, `\count 10` always contains the number of the highest-numbered counter that has been allocated, `\count 14` the highest-numbered box, etc. Inserts are given numbers 254, 253, etc., since they require a `\count`, `\dimen`, `\skip`, and `\box` all with the same number; `\count 20` contains the lowest-numbered insert that has been allocated. Of course, `\box255` is reserved for `\output`; `\count255`, `\dimen255`, and `\skip255` can be used freely.

It is recommended that macro designers always use `\global` assignments with respect to registers numbered

1, 3, 5, 7, 9,

and always non-`\global` assignments with respect to registers

0, 2, 4, 6, 8, 255.

This will prevent “save stack buildup” that might otherwise occur.

```
26 \count10=22 % allocates \count registers 23, 24, ...
27 \count11=9 % allocates \dimen registers 10, 11, ...
28 \count12=9 % allocates \skip registers 10, 11, ...
29 \count13=9 % allocates \muskip registers 10, 11, ...
30 \count14=9 % allocates \box registers 10, 11, ...
31 \count15=9 % allocates \toks registers 10, 11, ...
32 \count16=-1 % allocates input streams 0, 1, ...
33 \count17=-1 % allocates output streams 0, 1, ...
34 \count18=3 % allocates math families 4, 5, ...
35 \count19=0 % allocates \language codes 1, 2, ...
36 \count20=255 % allocates insertions 254, 253, ...
```

`\insc@unt` The insertion counter and most recent allocation.

```
\allocationnumber 37 \countdef\insc@unt=20
38 \countdef\allocationnumber=21
```

`\m@ne` The constant  $-1$ .

```
39 \countdef\m@ne=22 \m@ne=-1
```

`\wlog` Write on log file (only)

```
40 \def\wlog{\immediate\write\m@ne}
```

`\count@` Here are abbreviations for the names of scratch registers that don't need to be allocated.

```
\dimen@i 41 \countdef\count@=255
\dimen@ii 42 \dimendef\dimen@=0
\skip@ 43 \dimendef\dimen@i=1 % global only
\toks@ 44 \dimendef\dimen@ii=2
45 \skipdef\skip@=0
46 \toksdef\toks@=0
```

`\newcount` Now, we define `\newcount`, `\newbox`, etc. so that you can say `\newcount\foo` and `\foo` will be defined (with `\countdef`) to be the next counter.

`\newdimen` To find out which counter `\foo` is, you can look at `\allocationnumber`.

`\newskip` Since there's no `\boxdef` command, `\chardef` is used to define a `\newbox`, `\newmuskip` `\newinsert`, `\newfam`, and so on.

`\newread`

`\newwrite`

`\newlanguage` File b: ltplain.dtx Date: 2015/08/30 Version v2.1a

L<sup>A</sup>T<sub>E</sub>X change: remove `\outer` from `\newcount` and `\newdimen` (FMi) This is necessary to use `\newcount` inside `\if...` later on. Also remove from `\newskip`, `\newbox` `\newwrite` and `\newfam` (DPC) to save later redefinition.

```

47 /2kernel)
48 (*2kernel | latexrelease)
49 (latexrelease)\IncludeInRelease{2015/01/01}%
50 (latexrelease)          {\newcount}{Extended Allocation}%

51 \def\newcount {\e@alloc\count \countdef {\count10}\insc@unt\float@count}
52 \def\newdimen {\e@alloc\dimen \dimendef {\count11}\insc@unt\float@count}
53 \def\newskip  {\e@alloc\skip \skipdef {\count12}\insc@unt\float@count}
54 \def\newmuskip
55       {\e@alloc\muskip\muskipdef{\count13}\m@ne\e@alloc@top}

```

For compatibility use `\chardef` in the classical range.

```

56 \def\newbox    {\e@alloc\box
57               {\ifnum\allocationnumber<\@ccclvi
58               \expandafter\chardef
59               \else
60               \expandafter\e@alloc@chardef
61               \fi}
62               {\count14}\insc@unt\float@count}
63 \def\newtoks    {\e@alloc\toks \toksdef{\count15}\m@ne\e@alloc@top}
64 \def\newread    {\e@alloc\read \chardef{\count16}\m@ne\sixt@@n}
65 \def\newwrite   {\e@alloc\write \chardef{\count17}\m@ne\sixt@@n}
66 \def\new@mathgroup
67       {\e@alloc\mathgroup\chardef{\count18}\m@ne\e@mathgroup@top}
68 \def\newlanguage {\e@alloc\language \chardef{\count19}\m@ne\@ccclvi}
69 \let\newfam\new@mathgroup
70 /2kernel | latexrelease)

71 (latexrelease)\EndIncludeInRelease
72 (latexrelease)\IncludeInRelease{0000/00/00}%
73 (latexrelease)          {\newcount}{Extended Allocation}%
74 (latexrelease)\def\newcount{\alloc@0\count\countdef\insc@unt}
75 (latexrelease)\def\newdimen{\alloc@1\dimen\dimendef\insc@unt}
76 (latexrelease)\def\newskip{\alloc@2\skip\skipdef\insc@unt}
77 (latexrelease)\def\newmuskip{\alloc@3\muskip\muskipdef\@ccclvi}
78 (latexrelease)\def\newbox{\alloc@4\box\chardef\insc@unt}
79 (latexrelease)\def\newtoks{\alloc@5\toks\toksdef\@ccclvi}
80 (latexrelease)\def\newread{\alloc@6\read\chardef\sixt@@n}
81 (latexrelease)\def\newwrite{\alloc@7\write\chardef\sixt@@n}
82 (latexrelease)\def\new@mathgroup{\alloc@8\fam\chardef\sixt@@n}
83 (latexrelease)\def\newlanguage{\alloc@9\language\chardef\@ccclvi}
84 (latexrelease)\let\newfam\new@mathgroup
85 (latexrelease)\EndIncludeInRelease

```

`\e@alloc@chardef` The upper limit of extended registers, which leaves this number (eg `\dimen32767`)  
`\e@alloc@top` always unallocated by these macros. cf traditional `\dimen255`.

```

86 (*2kernel | latexrelease)
87 (latexrelease)\IncludeInRelease{2015/01/01}%
88 (latexrelease)          {\e@alloc@chardef}{Extended Allocation}%

89 \ifx\directlua\@undefined
90   \ifx\widowpenalties\@undefined

```

classic tex has  $2^8$  registers.

```
91 \mathchardef\@alloc@top=255
92 \let\@alloc@chardef\chardef
93 \else
```

etex and xetex have  $2^{15}$  registers.

```
94 \mathchardef\@alloc@top=32767
95 \let\@alloc@chardef\mathchardef
96 \fi
97 \else
```

luatex has  $2^{16}$  registers.

```
98 \chardef\@alloc@top=65535
99 \let\@alloc@chardef\chardef
100 \fi
101 </2ekernel | latexrelease>
102 <latexrelease>\EndIncludeInRelease
103 <latexrelease>\IncludeInRelease{0000/00/00}%
104 <latexrelease> \{\@alloc@chardef\}{Extended Allocation}%
105 <latexrelease>\let\@alloc@top\@undefined
106 <latexrelease>\let\@alloc@chardef\@undefined
107 <latexrelease>\EndIncludeInRelease
```

**\@mathgroup@top** The upper limit of extended math groups (`\fam`) 16 in classic T<sub>E</sub>X and e-T<sub>E</sub>X, but 256 in Unicode TeX variants.

```
108 <*2ekernel | latexrelease>
109 <latexrelease>\IncludeInRelease{2015/01/01}%
110 <latexrelease> \{\@mathgroup@top\}{Extended Allocation}%
111 \ifx\Umathcode\@undefined
```

classic and e tex have 16 fam (0–15).

```
112 \chardef\@mathgroup@top=16
113 \else
```

xetex and luatex have 256 fam (0–255).

```
114 \chardef\@mathgroup@top=256
115 \fi
116 </2ekernel | latexrelease>
117 <latexrelease>\EndIncludeInRelease
118 <latexrelease>\IncludeInRelease{0000/00/00}%
119 <latexrelease> \{\@mathgroup@top\}{Extended Allocation}%
120 <latexrelease>\let\@mathgroup@top\@undefined
121 <latexrelease>\EndIncludeInRelease
```

**\@alloc** A modified version of `\alloc@` that takes the count register rather than just the final digit of its number (assuming `\count1x`). It also has an extra argument to give the top of the extended range.

```

#1 #2 #3 #4 #5 #6
\@alloc type defcmd current top extended-top newname
```

Note that if just a single allocation range is required (not omitting a range up to 255 for inserts) then `−1` should be used for the first upper bound argument, `#4`.

```
122 <*2ekernel | latexrelease>
123 <latexrelease>\IncludeInRelease{2015/01/01}\{\@alloc\}{Extended Allocation}%
```

```

124 \def\@alloc#1#2#3#4#5#6{%
125   \global\advance#3\@ne
126   \@ch@ck{#3}{#4}{#5}#1%
127   \allocationnumber#3\relax
128   \global#2#6\allocationnumber
129   \wlog{\string#6=\string#1\the\allocationnumber}}%
130 \endkernel | latexrelease)
131 \end{latexrelease}\EndIncludeInRelease
132 \end{latexrelease}\IncludeInRelease{0000/00/00}{\@alloc}{Extended Allocation}%
133 \end{latexrelease}\let\@alloc\undefined
134 \end{latexrelease}\EndIncludeInRelease
135 \end{*2kernel}

```

`\@ch@ck` Extended check command. If the first range is exceeded, bump to 256 (or 266 for counts) and try again, testing the extended range.

`\extrafloats` Allocate matching registers from the top of the extended range and add to `\@freelist`.

```

136 \endkernel)
137 \end{*2kernel | latexrelease)
138 \end{latexrelease}\IncludeInRelease{2015/10/01}
139 \end{latexrelease}          {\@ch@ck}{Extended Allocation (checking)}%
140 \gdef\@ch@ck#1#2#3#4{%
141   \ifnum#1<#2\else

```

If we've reached the classical top limit, bump to 256 or 266 for counts (count 256-265 are reserved by the allocation system).

```

142     \ifnum#1=#2\relax
143       #1\@cclvi
144       \ifx\count#4\advance#1 10 \fi
145     \fi

```

Check we are below the extended limit.

```

146     \ifnum#1<#3\relax
147     \else
148       \errmessage{No room for a new \string#4}%
149     \fi
150   \fi}%
151 \end{latexrelease}\EndIncludeInRelease
152 \end{latexrelease}\IncludeInRelease{2015/01/01}%
153 \end{latexrelease}          {\@ch@ck}{Extended Allocation (checking)}%
154 \end{latexrelease}\gdef\@ch@ck#1#2#3#4{%
155 \end{latexrelease}   \ifnum#1<#2\else
156 \end{latexrelease}   \ifnum#1=#2\relax
157 \end{latexrelease}   #1\@cclvi
158 \end{latexrelease}   \ifx\count#4\advance#1 10 \fi
159 \end{latexrelease}   \fi
160 \end{latexrelease}   \ifnum#1<#3\relax
161 \end{latexrelease}   \else
162 \end{latexrelease}   \errmessage{No room for a new #4}%
163 \end{latexrelease}   \fi
164 \end{latexrelease} \fi}%
165 \end{latexrelease}\EndIncludeInRelease
166 \end{latexrelease}\IncludeInRelease{0000/00/00}%

```

```

167 \latexrelease\let\ech@ck\@undefined
168 \latexrelease\EndIncludeInRelease
169 \latexrelease\IncludeInRelease{2015/01/01}%
170 \latexrelease\let\float@count\ech@alloc@top
171 \latexrelease\let\float@count\ech@alloc@top

\extrafloats

173 \ifx\numexpr\@undefined
In classic TeX use \newinsert to allocate float boxes.
174 \def\extrafloats#1{%
175 \count@#1\relax
176 \ifnum\count@>\z@
177 \newinsert\reserved@a
178 \expandafter\chardef
179 \csname bx@\the\allocationnumber\endcsname\allocationnumber
180 \@cons\@freelist{\csname bx@\the\allocationnumber\endcsname}%
181 \advance\count@\m@ne
182 \expandafter\extrafloats
183 \expandafter\count@
184 \fi
185 }%
186 \else
In e-tex take float boxes from the top of the extended range.
187 \def\extrafloats#1{%
188 \ifnum#1>\z@
189 \count@\numexpr\float@count-1\relax
190 \ch@ck0\count@\count
191 \ch@ck1\count@\dimen
192 \ch@ck2\count@\skip
193 \ch@ck4\count@\box
194 \ech@alloc@chardef\float@count\count@
195 \expandafter\ech@alloc@chardef
196 \csname bx@\the\float@count\endcsname\float@count
197 \@cons\@freelist{\csname bx@\the\float@count\endcsname}%
198 \expandafter
199 \extrafloats\expandafter{\numexpr#1-1\relax}%
200 \fi}%
201 \fi
202 \ifx\@kernel\@undefined
203 \latexrelease\EndIncludeInRelease
204 \latexrelease\IncludeInRelease{0000/00/00}%
205 \latexrelease\let\float@count\@undefined
206 \latexrelease\let\extrafloats\@undefined
207 \latexrelease\EndIncludeInRelease
208 \ifx\@kernel\@undefined
209 \ifx\@kernel\@undefined

\alloc@

210 \def\alloc@#1#2#3#4#5{\global\advance\count1#1\@ne

```

```

211 \ch@ck#1#4#2% make sure there's still room
212 \allocationnumber\count1#1%
213 \global#3#5\allocationnumber
214 \wlog{\string#5=\string#2\the\allocationnumber}}

```

`\newinsert`

```

215 </2ekernel>
216 <*2ekernel | latexrelease>
217 <latexrelease> \IncludeInRelease{2015/10/01}
218 <latexrelease> { \newinsert } { Extended \newinsert } %
219 \ifx\numexpr\@undefined

```

If e-TeX is not available use the original plain TeX definition of `\newinsert`.

```

220 \def\newinsert#1{\global\advance\insc@unt \m@ne
221 \ch@ck0\insc@unt\count
222 \ch@ck1\insc@unt\dimen
223 \ch@ck2\insc@unt\skip
224 \ch@ck4\insc@unt\box
225 \allocationnumber\insc@unt
226 \global\chardef#1\allocationnumber
227 \wlog{\string#1=\string\insert\the\allocationnumber}}
228 \else

```

The highest register allowed with `\insert`.

```

229 \ifx\directlua\@undefined
230 \chardef\@insert@top255
231 \else
232 \chardef\@insert@top\@alloc@top
233 \fi

```

If the classic registers are exhausted, take an insert from the free float list and use `\extrafloats` to add a new float to that list.

```

234 \def\newinsert#1{%
235 \@tempswafalse
236 \ifnum\count10<\insc@unt
237 \ifnum\count11<\insc@unt
238 \ifnum\count12<\insc@unt
239 \ifnum\count14<\insc@unt
240 \@tempswatruetrue
241 \fi\fi\fi\fi
242 \if@tempswa
243 \global\advance\insc@unt\m@ne
244 \allocationnumber\insc@unt
245 \else
246 \extrafloats\@one
247 \@next\@currbox\@freelist
248 {\ifnum\@currbox<\@insert@top
249 \allocationnumber\@currbox
250 \else
251 \ch@ck0\m@ne\insert
252 \fi}%
253 {\ch@ck0\m@ne\insert}}%
254 \fi
255 \global\chardef#1\allocationnumber

```

```

256 \wlog{\string#1=\string\insert\the\allocationnumber}%
257 }

258 \fi
259 /2ekernel | latexrelease)

260 (latexrelease)\EndIncludeInRelease
261 (latexrelease)\IncludeInRelease{0000/00/00}%
262 (latexrelease)          {\newinsert}{Extended \newinsert}%
263 (latexrelease)\let\@insert@top\@undefined
264 (latexrelease)\def\newinsert#1{\global\advance\insc@unt \m@ne
265 (latexrelease) \ch@ck0\insc@unt\count
266 (latexrelease) \ch@ck1\insc@unt\dimen
267 (latexrelease) \ch@ck2\insc@unt\skip
268 (latexrelease) \ch@ck4\insc@unt\box
269 (latexrelease) \allocationnumber\insc@unt
270 (latexrelease) \global\chardef#1\allocationnumber
271 (latexrelease) \wlog{\string#1=\string\insert\the\allocationnumber}}
272 (latexrelease)\EndIncludeInRelease
273 (*2ekernel)

\ch@ck

274 \gdef\ch@ck#1#2#3{%
275   \ifnum\count1#1<#2\else
276     \errmessage{No room for a new #3}%
277   \fi}

\newhelp

278 \def\newhelp#1#2{\newtoks#1#1\expandafter{\csname#2\endcsname}}

\maxdimen Here are some examples of allocation.
\hideskip 279 \newdimen\maxdimen \maxdimen=16383.99999pt % the largest legal <dimen>
280 \newskip\hideskip \hideskip=-1000pt plus 1fill % negative but can grow

\p@
\z@ 281 \newdimen\p@ \p@=1pt % this saves macro space and time
\z@skip 282 \newdimen\z@ \z@=0pt % can be used both for 0pt and 0
\voidb@x 283 \newskip\z@skip \z@skip=0pt plus0pt minus0pt
284 \newbox\voidb@x % permanently void box register

285 \message{compatibility for TeX 2, }

If this file is used in an old TEX we define the new features of TEX 3.0 as simple
macros or counters so that files that uses these features can be processed in such
an environment (They will however produce some other results).

286 \ifx\@undefined\inputlineno
287   \newcount\inputlineno

This could be used to detect that an old TEX is in force
288   \inputlineno-1

```

Extra test for MLTeX 2, RmS 91/11/07.

```
289 \ifx\@undefined\language
290 \newcount\language
291 \fi
292 \newcount\lefthyphenmin
293 \newcount\riquiryphenmin
294 \newcount\errorcontextlines
295 \newcount\holdinginserts
296 \newdimen\emergencystretch
297 \newcount\badness
298 \let\noboundary\relax
299 \newcount\setlanguage
300 \fi
```

Assign initial values to TeX's parameters

```
301 \message{parameters,}
```

All of TeX's numeric parameters are listed here, but the code is commented out if no special value needs to be set. INITEX makes all parameters zero except where noted.

```
302 \pretolerance=100
303 \tolerance=200 % INITEX sets this to 10000
304 \hbadness=1000
305 \vbadness=1000
306 \linepenalty=10
307 \hyphenpenalty=50
308 \exhyphenpenalty=50
309 \binoppenalty=700
310 \relpenalty=500
311 \clubpenalty=150
312 \widowpenalty=150
313 \displaywidowpenalty=50
314 \brokenpenalty=100
315 \predisplaypenalty=10000
    \postdisplaypenalty=0
    \interlinepenalty=0
    \floatingpenalty=0, set during \insert
    \outputpenalty=0, set before TeX enters \output
316 \doublehyphendemerits=10000
317 \finalhyphendemerits=5000
318 \adjdemerits=10000
    \looseness=0, cleared by TeX after each paragraph
    \pausing=0
    \holdinginserts=0
    \tracingonline=0
    \tracingmacros=0
    \tracingstats=0
    \tracingparagraphs=0
    \tracingpages=0
    \tracingoutput=0
319 \tracinglostchars=1
```

```

\tracingcommands=0
\tracingrestores=0
\language=0
320 \uchyph=1

\lefthyphenmin=2 \righthyphenmin=3 set below
\globaldefs=0
\maxdeadcycles=25 % INITEX does this
\hangafter=1 % INITEX does this, also TeX after each paragraph
\fam=0
\mag=1000 % INITEX does this
\escapechar='\ % INITEX does this
321 \defaultthyphenchar='\-
322 \defaultskewchar=-1

\endlinechar='\^M % INITEX does this
\newlinechar=-1 \LaTeX\ sets this in ltdefs.dtx.
323 \delimiterfactor=901

\time=now % TeX does this at beginning of job
\day=now % TeX does this at beginning of job
\month=now % TeX does this at beginning of job
\year=now % TeX does this at beginning of job

In LATEX we don't want box information in the transcript unless we do a full
tracing.
324 \showboxbreadth=-1
325 \showboxdepth=-1
326 \errorcontextlines=-1

327 \hfuzz=0.1pt
328 \vfuzz=0.1pt
329 \overfullrule=5pt
330 \maxdepth=4pt
331 \splitmaxdepth=\maxdimen
332 \boxmaxdepth=\maxdimen

\lineskiplimit=0pt, changed by \normalbaselines
333 \delimitershortfall=5pt
334 \nulldelimiterspace=1.2pt
335 \scriptspace=0.5pt

\mathsurround=0pt
\predisplaysize=0pt, set before TeX enters $$
\displaywidth=0pt, set before TeX enters $$
\displayindent=0pt, set before TeX enters $$
336 \parindent=20pt

\hangindent=0pt, zeroed by TeX after each paragraph
\hoffset=0pt
\voffset=0pt

\baselineskip=0pt, changed by \normalbaselines

```

```

\lineskip=0pt, changed by \normalbaselines
337 \parskip=0pt plus 1pt
338 \abovedisplayskip=12pt plus 3pt minus 9pt
339 \abovedisplayshortskip=0pt plus 3pt
340 \belowdisplayskip=12pt plus 3pt minus 9pt
341 \belowdisplayshortskip=7pt plus 3pt minus 4pt

\leftskip=0pt
\rightskip=0pt

342 \topskip=10pt
343 \splittopskip=10pt

\tabskip=0pt
\spaceskip=0pt
\xspaceskip=0pt
344 \parfillskip=0pt plus 1fil

\normalbaselineskip We also define special registers that function like parameters:
\normallineskip 345 \newskip\normalbaselineskip \normalbaselineskip=12pt
\normallineskiplimit 346 \newskip\normallineskip \normallineskip=1pt
347 \newdimen\normallineskiplimit \normallineskiplimit=0pt

\interfootlinepenalty

348 \newcount\interfootnotelinepenalty \interfootnotelinepenalty=100

Definitions for preloaded fonts

\magstephalf
\magstep 349 \def\magstephalf{1095 }
350 \def\magstep#1{\ifcase#1 \@m\or 1200\or 1440\or 1728\or
351 2074\or 2488\fi\relax}

Macros for setting ordinary text

\frenchspacing
\nonfrenchspacing 352 \def\frenchspacing{\sfcode'\.\@m \sfcode'\?\@m \sfcode'\!\@m
353 \sfcode'\:\@m \sfcode'\;\@m \sfcode'\,\@m}
354 \def\nonfrenchspacing{\sfcode'\.3000\sfcode'\?3000\sfcode'\!3000%
355 \sfcode'\:2000\sfcode'\;1500\sfcode'\,1250 }

\normalbaselines
356 \def\normalbaselines{\lineskip\normallineskip
357 \baselineskip\normalbaselineskip \lineskiplimit\normallineskiplimit}

\M Save a bit of space by using \let here.
\I 358 \def\^^M{\ } % control <return> = control <space>
359 \let\^^I\^^M % same for <tab>

\lq
\rq 360 \def\lq{' }
361 \def\rq{' }

```

```

\lbrack
\rbrack 362 \def\lbrack{[}
          363 \def\rbrack{]}

\aa      These are not from plain.tex but they are similar to other commands found here
\AA      and nowhere else, being alternate input forms for characters.
          364 \def \aa {\r a}
          365 \def \AA {\r A}

\endgraf
\endline 366 \let\endgraf=\par
          367 \let\endline=\cr

\space
          368 \def\space{ }

\empty   This probably ought to go altogether, but let it to the LATEX version to save space.
          369 \let\empty\@empty

\null
          370 \def\null{\hbox{}}

\bgroup
\egroup  371 \let\bgroup={
          372 \let\egroup=}

\obeylines In \obeylines, we say \let^^M=\par instead of \def^^M{\par} since this allows,
\obeyspaces for example, \let\par=\cr \obeylines \halign{...
          373 {\catcode'\^^M=\active % these lines must end with %
          374   \gdef\obeylines{\catcode'\^^M=\active \let^^M\par}%
          375   \global\let^^M\par} % this is in case ^^M appears in a \write
          376 \def\obeyspaces{\catcode'\ \active}
          377 {\obeyspaces\global\let \space=}

\loop    We use Kabelschacht's method of doing loops, see TUB 8#2 (1987). (unless that
\iterate breaks something :-). It turned out to need an extra \relax: see pr/642 (\loop
\repeat  could do one iteration too much in certain cases).
          378 \long\def \loop #1\repeat{%
          379   \def\iterate{#1\relax % Extra \relax
          380     \expandafter\iterate\fi
          381     }%
          382   \iterate
          383   \let\iterate\relax
          384 }

This setting of \repeat is needed to make \loop...\if...\repeat skippable
within another \if....
          385 \let\repeat=\fi

LATEX defines \smallskip, etc. in ltspac.dtx.

```

```

\nointerlineskip
\offinterlineskip 386 \def\nointerlineskip{\prevdepth-\@m\p@}
387 \def\offinterlineskip{\baselineskip-\@m\p@
388 \lineskip\z@ \lineskiplimit\maxdimen}

\vg glue
\hg glue 389 \def\vg glue{\afterassignment\vg l@ \skip@=}
390 \def\vg l@{\par \dimen@ \prevdepth \hrule \@height\z@
391 \nobreak\vskip\skip@ \prevdepth\dimen@}
392 \def\hg glue{\afterassignment\hg l@ \skip@=}
393 \def\hg l@{\leavevmode \count@ \spacefactor \vrule \@width\z@
394 \nobreak\hskip\skip@ \spacefactor\count@}

LATEX defines ~ in ltdefns.dtx.

\slash
395 \def\slash{/\penalty\exhyphenpenalty} % a '/' that acts like a '-'

\break
\nobreak 396 \def\break{\penalty-\@M}
\allowbreak 397 \def\nobreak{\penalty \@M}
398 \def\allowbreak{\penalty \z@}

\filbreak
\goodbreak 399 \def\filbreak{\par\vfil\penalty-200\vfilneg}
400 \def\goodbreak{\par\penalty-500 }

\ej ect Define \ej ect as in plain TEX but define \supereject only in the compatibility
file.
401 \def\ej ect{\par\break}

\removelastskip
402 \def\removelastskip{\ifdim\lastskip=\z@\else\vskip-\lastskip\fi}

\smallbreak
\medbreak 403 \def\smallbreak{\par\ifdim\lastskip<\smallskipamount
\bigbreak 404 \removelastskip\penalty-50\smallskip\fi}
405 \def\medbreak{\par\ifdim\lastskip<\medskipamount
406 \removelastskip\penalty-100\medskip\fi}
407 \def\bigbreak{\par\ifdim\lastskip<\bigskipamount
408 \removelastskip\penalty-200\bigskip\fi}

\m@th
409 \def\m@th{\mathsurround\z@}

\underbar Due to LATEX's redefinition of \underline plain TEX's \underbar can be done in
a simpler fashion (but do we need it at all?).
410 \def\underbar#1{\underline{\sbox\tw@{#1}\dp\tw@\z@\box\tw@}}

\strutbox LATEX sets \strutbox in \set@fontsize.
\strut 411 \newbox\strutbox
412 \def\strut{\relax\ifmmode\copy\strutbox\else\unhcopy\strutbox\fi}

```

`\hidewidth` For alignment entries that can stick out.

```
413 \def\hidewidth{\hskip\hideskip}
```

`\narrower`

```
414 \def\narrower{%
415   \advance\leftskip\parindent
416   \advance\rightskip\parindent}
```

L<sup>A</sup>T<sub>E</sub>X defines `\ae` and similar commands elsewhere.

```
417 \chardef\%='\'
418 \chardef\&='&
419 \chardef\#='#
```

Most text commands are actually encoding specific and therefore defined later, so commented out or removed from this file.

`\leavevmode` begins a paragraph, if necessary

```
420 \def\leavevmode{\unhbox\voidb@x}
```

`\mathhexbox`

```
421 \def\mathhexbox#1#2#3{\mbox{$\m@th \mathchar"#1#2#3$}}
```

`\ialign`

```
422 \def\ialign{\everycr{}}\tabskip\z@skip\halign} % initialized \halign
```

`\oalign`

```
\o@lign 423 \def\oalign#1{\leavevmode\vtop{\baselineskip\z@skip \lineskip.25ex%
\oalign 424   \ialign{##\crrc#1\crrc}}}%
425 \def\o@lign{\lineskiplimit\z@ \oalign}
426 \def\oalign{\lineskiplimit-\maxdimen \oalign}
```

`\sh@ft` The definition of this macro in `plain.tex` was improved in about 1997; but as a result its usage was changed and its new definition is not appropriate for L<sup>A</sup>T<sub>E</sub>X.

Since the version given here has been in use by L<sup>A</sup>T<sub>E</sub>X for many years it does not seem prudent to remove it now. As far as we can tell it has only been used to define `\b` and `\d` but this cannot be certain.

```
427 \def\sh@ft#1{\dimen@.00#1ex\multiply\dimen@\fontdimen1\font
428   \kern-.0156\dimen@} % compensate for slant in lowered accents
```

`\ltx@sh@ft` This is the L<sup>A</sup>T<sub>E</sub>X version of the second incarnation of the plain macro `\sh@ft`, which takes a dimension as its argument. It shifts a pseudo-accent horizontally by an amount proportional to the product of its argument and the slant-per-point (`fontdimen 1`).

```
429 \def\ltx@sh@ft #1{%
430   \dimen@ #1%
431   \kern \strip@pt
432   \fontdimen1\font \dimen@
433 } % kern by #1 times the current slant
```

L<sup>A</sup>T<sub>E</sub>X change: the text commands such as `\d`, `\b`, `\c`, `\copyright`, `\TeX` are now defined elsewhere.

L<sup>A</sup>T<sub>E</sub>X change: Make `\t` work in a moving argument. Now defined elsewhere.

```

\hrulefill  LATEX change: \kern\z@ added to end of \hrulefill and \dotfill to make them
\dotfill    work in ‘tabular’ and ‘array’ environments. (Change made 24 July 1987). LATEX
            change: \leavevmode added at beginning of \dotfill and \hrulefill so that
            they work as expected in vertical mode.
434 \def\hrulefill{\leavevmode\leaders\hrule\hfill\kern\z@}

The box in \dotfill originally contained (in plain.tex):
\mkern 1.5mu .\mkern 1.5mu;
the width of .44em differs from this by .04pt which is probably an acceptable
difference within leaders.

435 \def\dotfill{%
436   \leavevmode
437   \cleaders \hb@xt@ .44em{\hss.\hss}\hfill
438   \kern\z@}

INITEX sets \sfcode x=1000 for all x, except that \sfcode‘X’=999 for upper-
case letters. The following changes are needed:
439 \sfcode‘\’=0 \sfcode‘\’=0 \sfcode‘\’=0

The \nonfrenchspacing macro will make further changes to \sfcode values.
Definitions related to output
\magnification doesn’t work in LATEX.

\def\magnification{\afterassignment\m@g\count@}
\def\m@g{\mag\count@
\hspace6.5truein\vspace8.9truein\dimen\footins8truein}

\showoverfull The following commands are used in debugging:
440 \def\showoverfull{\tracingonline\@ne}

\showoutput
\loggingoutput 441 \gdef\loggingoutput{\tracingoutput\@ne
442   \showboxbreadth\maxdimen\showboxdepth\maxdimen\errorstopmode}
443 \gdef\showoutput{\loggingoutput\showoverfull}
444 /2ekernel)

\tracingall
\loggingall 445 (latexrelease)\IncludeInRelease{2015/01/20}{\loggingall}{etex tracing}%
446 (*2ekernel| latexrelease)
447 \ifx\tracingscantokens\@undefined
448 \gdef\loggingall{%
449   \tracingstats\tw@
450   \tracingpages\@ne
451   \tracinglostchars\@ne
452   \tracingparagraphs\@ne
453   \errorcontextlines\maxdimen
454   \loggingoutput
455   \tracingmacros\tw@
456   \tracingcommands\tw@
457   \tracingrestores\@ne
458   }%
459 \else
460 \gdef\loggingall{%
461   \tracingstats\tw@

```

```

462 \tracingpages\@ne
463 \tracinglostchars\tw@
464 \tracingparagraphs\@ne
465 \tracinggroups\@ne
466 \tracingifs\@ne
467 \tracingscantokens\@ne
468 \tracingnesting\@ne
469 \errorcontextlines\maxdimen
470 \loggingoutput
471 \tracingmacros\tw@
472 \tracingcommands\thr@@
473 \tracingrestores\@ne
474 \tracingassigns\@ne
475 }%
476 \fi
477 \gdef\tracingall{\showoverfull\loggingall}
478 </2ekernel | latexrelease>
479 (latexrelease)\EndIncludeInRelease
480 (latexrelease)\IncludeInRelease{0000/00/00}{\loggingall}{etex tracing}%
481 (latexrelease)\gdef\loggingall{\tracingcommands\tw@\tracingstats\tw@
482 (latexrelease) \tracingpages\@ne\tracinglostchars\@ne
483 (latexrelease) \tracingmacros\tw@\tracingparagraphs\@ne\tracingrestores\@ne
484 (latexrelease) \errorcontextlines\maxdimen\loggingoutput}
485 (latexrelease) \gdef\tracingall{\loggingall\showoverfull}
486 (latexrelease)\EndIncludeInRelease

\tracingnone
\hideoutput 487 (latexrelease)\IncludeInRelease{2015/01/20}{\tracingnone}%
488 (latexrelease) {turn off etex tracing}%
489 (*2ekernel | latexrelease>
490 \ifx\tracingscantokens\@undefined
491 \def\tracingnone{%
492 \tracingonline\z@
493 \tracingcommands\z@
494 \showboxdepth\m@ne
495 \showboxbreadth\m@ne
496 \tracingoutput\z@
497 \errorcontextlines\m@ne
498 \tracingrestores\z@
499 \tracingparagraphs\z@
500 \tracingmacros\z@
501 \tracinglostchars\@ne
502 \tracingpages\z@
503 \tracingstats\z@
504 }%
505 \else
506 \def\tracingnone{%
507 \tracingassigns\z@
508 \tracingrestores\z@
509 \tracingonline\z@
510 \tracingcommands\z@
511 \showboxdepth\m@ne
512 \showboxbreadth\m@ne
513 \tracingoutput\z@

```

```

514 \errorcontextlines\m@ne
515 \tracingnesting\z@
516 \tracingscantokens\z@
517 \tracingifs\z@
518 \tracinggroups\z@
519 \tracingparagraphs\z@
520 \tracingmacros\z@
521 \tracinglostchars\@ne
522 \tracingpages\z@
523 \tracingstats\z@
524 }%
525 \fi

526 \def\hideoutput{%
527 \tracingoutput\z@
528 \showboxbreadth\m@ne
529 \showboxdepth\m@ne
530 \tracingonline\m@ne
531 }%

532 </2ekernel | latexrelease>
533 <latexrelease>\EndIncludeInRelease
534 <latexrelease>\IncludeInRelease{0000/00/00}{\tracingnone}%
535 <latexrelease>                                {turn off etex tracing}%
536 <latexrelease>\let\tracingnone\@undefined
537 <latexrelease>\let\hideoutput\@undefined
538 <latexrelease>\EndIncludeInRelease

    LATEX change: \showhyphens Defined later.
    Punctuation affects the spacing.

539 <*2ekernel>
540 \nonfrenchspacing
541 </2ekernel>

```

File c  
ltvers.dtx

## 10 Version Identification

First we identify the date and version number of this release of L<sup>A</sup>T<sub>E</sub>X, and set `\everyjob` so that it is printed at the start of every L<sup>A</sup>T<sub>E</sub>X run.

```

1 \fntname
2 \fntversion 1 (*2ekernel)
3 \patcj@level 2 \def\fntname{LaTeX2e}
4 \edef\fntversion 3 \edef\fntversion
5 4 (/2ekernel)
6 5 (latexrelease)\edef\latexreleaseversion
7 6 (*2ekernel | latexrelease)
8 7 {2015/10/01}
9 8 (/2ekernel | latexrelease)
10 9 (*2ekernel)
11 10 \def\patch@level{1}

```

Check that the format being made is not too old. The error message complains about ‘more than 5 years’ but in fact the error is not triggered until 65 months.

This code is currently not activated as we don't know if we already got to the last official 2e version (due to staff shortage or due to a successor (think positive:)).

```

11 \iffalse
12 \def\reserved@a#1/#2/#3\@nil{%
13   \count@ \year
14   \advance \count@ -#1 \relax
15   \multiply \count@ by 12 \relax
16   \advance \count@ \month
17   \advance \count@ -#2 \relax}
18 \expandafter \reserved@a \fmtversion \@nil

```

\count@ is now the age of this file in months. Take a generous definition of ‘year’ so this message is not generated too often.

```

19 \ifnum\count@>65
20   \typeout{^^J%
21   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
22   ! You are attempting to make a LaTeX format from a source file^^J%
23   ! That is more than five years old.^^J%
24   !^^J%
25   ! If you enter <return> to scroll past this message then the format^^J%
26   ! will be built, but please consider obtaining newer source files^^J%
27   ! before continuing to build LaTeX.^^J%
28   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
29 }
30   \errhelp{To avoid this error message, obtain new LaTeX sources.}
31   \errmessage{LaTeX source files more than 5 years old!}
32 \fi
33 \let\reserved@a\relax
34 \fi
```

```

35 \ifnum\patch@level=0
36 \everyjob\expandafter{\the\everyjob
37 \typeout{\fmtname \space<\fmtversion>}}
38 \immediate
39 \write16{\fmtname \space<\fmtversion>}
40 \else\ifnum\patch@level>0
41 \everyjob\expandafter{\the\everyjob
42 \typeout{\fmtname \space<\fmtversion> patch level \patch@level}}
43 \immediate
44 \write16{\fmtname \space<\fmtversion> patch level \patch@level}
45 \else
46 \everyjob\expandafter{\the\everyjob
47 \typeout{\fmtname \space<\fmtversion> pre-release\patch@level}}
48 \immediate
49 \write16{\fmtname \space<\fmtversion> pre-release\patch@level}
50 \fi
51 \fi
52 </2ekernel>

```

\IncludeInRelease

```

53 <*2ekernel | latexrelease>
54 \def\IncludeInRelease#1{\kernel@ifnextchar[%
55 {\@IncludeInRelease{#1}}
56 {\@IncludeInRelease{#1}[#1]}}

    If a specific date has not been specified in latexrelease use ‘#1’.
57 \def\@IncludeInRelease#1[#2]{\@IncludeInRelease{#2}}

58 \def\@IncludeInRelease#1#2#3{%
59 \toks@{[#1] #3}%
60 \expandafter\ifx\csname\string#2+\@currname+IIR\endcsname\relax
61 \ifnum\expandafter\@parse@version#1//00\@nil
62 >\expandafter\@parse@version\fmtversion//00\@nil
63 \GenericInfo{}{Skipping: \the\toks@}%
64 \expandafter\expandafter\expandafter\@gobble@IncludeInRelease
65 \else
66 \GenericInfo{}{Applying: \the\toks@}%
67 \expandafter\let\csname\string#2+\@currname+IIR\endcsname\empty
68 \fi
69 \else
70 \GenericInfo{}{Already applied: \the\toks@}%
71 \expandafter\@gobble@IncludeInRelease
72 \fi
73 }

74 \long\def\@gobble@IncludeInRelease#1\EndIncludeInRelease{}
75 \let\EndIncludeInRelease\relax
76 </2ekernel | latexrelease>

```

# File d

## ltdfn.s.dtx

### 11 Definitions

This section contains commands used in defining other macros.

```
1 (*2kernel)
```

#### 11.1 Initex initialisations

`\two@digits` Prefix a number less than 10 with ‘0’.

```
2 \def\two@digits#1{\ifnum#1<10 0\fi\number#1}
```

`\typeout` Display something on the terminal.

```
3 \def\typeout#1{\begingroup\set@display@protect
4   \immediate\write\@unused{#1}\endgroup}
```

`\newlinechar` A char to be used as new-line in output to files.

```
5 \newlinechar'\^^J
```

#### 11.2 Saved versions of T<sub>E</sub>X primitives

The TeX primitive `\foo` is saved as `\@@foo`. The following primitives are handled in this way:

`\@@par`

```
6 \let\@@par=\par
7 %\let\@@input=\input      %%% moved earlier
8 %\let\@@end=\end          %%%
```

`\@@hyph` The following comment was added when these commands were first set up, 19 April 1986: the `\-` command is redefined to allow it to work in the `\ttfamily` type style, where automatic hyphenation is suppressed by setting `\hyphenchar` to `-1`. The original primitive T<sub>E</sub>X definition is saved as `\@@hyph` just in case anyone needs it.

There is a need for a robust command for a discretionary hyphen since its exact representation depends on the glyphs available in the current font. For example, with suitable fonts and the T1 font encoding it is possible to use hanging hyphens.

A suitable robust definition that allows for many possible types of font and encoding may be as follows:

```
\DeclareRobustCommand {\-}{%
\discretionary {%
\char \ifnum\hyphenchar\font<\z@
\defaultthyphenchar
\else
\hyphenchar\font
\fi
}{ }{}%
}
```

The redefinition (via `\let`) of `\-` within tabbing also makes the use of a robust command advisable since then any redefinition of `\-` via `\DeclareRobustCommand` will not cause a conflict.

Therefore, macro writers should be hereby warned that these internals will probably change! It is likely that a future release of L<sup>A</sup>T<sub>E</sub>X will make `\-` effectively an encoding specific text command.

```

9 \let\@chyph=\-          % Save original primitive definition
10 \def\-\{\discretionary{-}{-}{-}}

\@dischyph
11 \let\@dischyph=\-

\@italiccorr Save the original italic correction.
12 \let\@italiccorr=\/

\@height The following definitions save token space. E.g., using \@height instead of height
\@depth saves 5 tokens at the cost in time of one macro expansion.
\@width 13 \def\@height{height} \def\@depth{depth} \def\@width{width}
\@minus 14 \def\@minus{minus}
\@plus 15 \def\@plus{plus}

\hbxt@ The next one is another 100 tokens worth.
16 \def\hbxt@{\hbox to}

17 \message{hacks,}
```

### 11.3 Command definitions

This section defines the following commands:

<code>\@namedef</code>	<code>{\NAME}</code>	Expands to <code>\def\{\NAME}</code> , except name can contain any characters.
<code>\@nameuse</code>	<code>{\NAME}</code>	Expands to <code>\{\NAME}</code> .
<code>\@ifnextchar</code>	<code>X{\YES}\{\NO}</code>	Expands to <code>\YES</code> if next character is an ‘X’, and to <code>\NO</code> otherwise. (Uses <code>\reserved@a</code> – <code>\reserved@c</code> .) NOTE: GOBBLES ANY SPACE FOLLOWING IT.
<code>\@ifstar</code>	<code>{\YES}\{\NO}</code>	Gobbles following spaces and then tests if next the character is a ‘*’. If it is, then it gobbles the ‘*’ and expands to <code>\YES</code> , otherwise it expands to <code>\NO</code> .
<code>\@dblarg</code>	<code>{\CMD}\{\ARG}</code>	Expands to <code>\{\CMD}\[\ARG]\{\ARG}</code> . Use <code>\@dblarg\CS</code> when <code>\CS</code> takes arguments <code>[ARG1]{ARG2}</code> , where default is <code>ARG1 = ARG2</code> .
<code>\@ifundefined</code>	<code>{\NAME}\{\YES}\{\NO}</code>	: If <code>\NAME</code> is undefined then it executes <code>\YES</code> , otherwise it executes <code>\NO</code> . More precisely, true if <code>\NAME</code> either undefined or = <code>\relax</code> .
<code>\@ifdefinable</code>	<code>\NAME{\YES}</code>	Executes <code>\YES</code> if the user is allowed to define <code>\NAME</code> , otherwise it gives an error. The user can define <code>\NAME</code> if <code>\@ifundefined{\NAME}</code> is true, ‘NAME’ ≠ ‘relax’ and the first three letters of ‘NAME’ are not ‘end’, and if <code>\endNAME</code> is not defined.
<code>\newcommand</code>	<code>*{\FOO}\{\i}\{\TEXT}</code>	

User command to define `\F00` to be a macro with  $i$  arguments ( $i = 0$  if missing) having the definition  $\langle TEXT \rangle$ . Produces an error if `\F00` already defined.

Normally the command is defined to be `\long` (ie it may take multiple paragraphs in its argument). In the star-form, the command is not defined as `\long` and a blank line in any argument to the command would generate an error.

`\renewcommand`       $\star{\langle F00 \rangle}[\langle i \rangle]{\langle TEXT \rangle}$

Same as `\newcommand`, except it checks if `\F00` already defined.

`\newenvironment`       $\star{\langle F00 \rangle}[\langle i \rangle]{\langle DEF1 \rangle}{\langle DEF2 \rangle}$

equivalent to:

`\newcommand{\F00}[i]{DEF1} \def{\endF00}{DEF2}`

(or the appropriate star forms).

`\renewenvironment`

Obvious companion to `\newenvironment`.

`\@cons`                : See description of `\output` routine.

`\@car`                `\@car T1 T2 ... Tn\@nil == T1` (unexpanded)

`\@cdr`                `\@cdr T1 T2 ... Tn\@nil == T2 ... Tn` (unexpanded)

`\typeout`              $\langle message \rangle$

Produces a warning message on the terminal.

`\typein`               $\langle message \rangle$

Types message, asks the user to type in a command, then executes it

`\typein`               $[\langle CS \rangle]{\langle MSG \rangle}$

Same as above, except defines `\CS` to be the input instead of executing it.

`\typein`

```

18 \def\typein{%
19   \let\@typein\relax
20   \@testopt\@xtypein\@typein}

21 \ifx\directlua\@undefined

22 \def\@xtypein[#1]#2{%
23   \typeout{#2}%
24   \advance\endlinechar\@M
25   \read\@inputcheck to#1%
26   \advance\endlinechar-\@M
27   \@typein}%

28 \else

29 \def\@xtypein[#1]#2{%
30   \typeout{#2}%
31   \begingroup \endlinechar\m@ne
32   \read\@inputcheck to#1%
33   \expandafter\endgroup
34   \expandafter\def\expandafter#1\expandafter{#1}%
35   \@typein}%

36 \fi

```

`\@namedef`

```

37 \def\@namedef#1{\expandafter\def\csname #1\endcsname}

```

`\@nameuse`

```

38 \def\@nameuse#1{\csname #1\endcsname}

```

```

\@cons
39 \def\@cons#1#2{\begingroup\let\@elt\relax\xdef#1{#1\@elt #2}\endgroup}

\@car
\@cdr
40 \def\@car#1#2\@nil{#1}
41 \def\@cdr#1#2\@nil{#2}

\@carcube \@carcube T1 ... Tn\@nil = T1 T2 T3 ,  $n > 3$ 
42 \def\@carcube#1#2#3#4\@nil{#1#2#3}

\@onlypreamble This macro adds its argument to the list of commands stored in \@preamblecmds
\@preamblecmds to be disabled after \begin{document}. These commands are redefined to gener-
ate \@notprerr at this point.
43 \def\@preamblecmds{}
44 \def\@onlypreamble#1{%
45   \expandafter\gdef\expandafter\@preamblecmds\expandafter{%
46     \@preamblecmds\do#1}}
47 \@onlypreamble\@onlypreamble
48 \@onlypreamble\@preamblecmds

\@star@or@long Look ahead for a *. If present reset \l@ngrel@x so that the next definition, #1,
will be non-long.
49 \def\@star@or@long#1{%
50   \ifstar
51     {\let\l@ngrel@x\relax#1}%
52     {\let\l@ngrel@x\long#1}}

\l@ngrel@x This is either \relax or \long depending on whether the *-form of a definition
command is being executed.
53 \let\l@ngrel@x\relax

\newcommand User level \newcommand.
54 \def\newcommand{\@star@or@long\new@command}

\new@command
55 \def\new@command#1{%
56   \@testopt{\@newcommand#1}0}

\@newcommand Handling arguments for \newcommand.
\@argdef 57 \def\@newcommand#1[#2]{%
\@xargdef 58   \kernel@ifnextchar [{\@xargdef#1[#2]}%
59     {\@argdef#1[#2]}}
Define #1 if it is definable.
Both here and in \@xargdef the replacement text is absorbed as an argument
because if we are not allowed to make the definition we have to get rid of it
completely.
60 \long\def\@argdef#1[#2]#3{%
61   \@ifdefinable #1{\@yargdef#1\@ne{#2}{#3}}
Handle the second optional argument.
62 \long\def\@xargdef#1[#2] [#3]#4{%
63   \@ifdefinable#1{%

```

Define the actual command to be:

```
\def\foo{\@protected@testopt\foo\\foo{default}}
```

where `\\foo` is a csname generated from applying `\csname` and `\string` to `\foo`, ie the actual name contains a backslash and therefore can't clash easily with existing command names. "Default" is the contents of the second optional argument of `(re)newcommand`.

```
64 \expandafter\def\expandafter#1\expandafter{%
65     \expandafter
66     \@protected@testopt
67     \expandafter
68     #1%
69     \csname\string#1\endcsname
70     {#3}}%
```

Now we define the internal macro ie `\\foo` which is supposed to pick up all arguments (optional and mandatory).

```
71 \expandafter\@yargdef
72 \csname\string#1\endcsname
73 \tw@
74 {#2}%
75 {#4}}}
```

`\@testopt` This macro encapsulates the most common call to `\@ifnextchar`, saving several tokens each time it is used in the definition of a command with an optional argument. `#1` The code to execute in the case that there is a `[` need not be a single token but can be any sequence of commands that 'expects' to be followed by `[`. If this command were only used in `\newcommand` definitions then `#1` would be a single token and the braces could be omitted from `{#1}` in the definition below, saving a bit of memory.

```
76 \long\def\@testopt#1#2{%
77 \kernel@ifnextchar[{#1}]{#1[{#2}]}}
```

`\@protected@testopt` Robust version of `\@testopt`. The extra argument (`#1`) must be a single token. If protection is needed the call expands to `\protect` applied to this token, and the 2nd and 3rd arguments are discarded (by `\@xprotect`). Otherwise `\@testopt` is called on the 2nd and 3rd arguments.

This method of making commands robust avoids the need for using up two csnames per command, the price is the extra expansion time for the `\ifx` test.

```
78 \def\@protected@testopt#1{%%
79 \ifx\protect\@typeset@protect
80 \expandafter\@testopt
81 \else
82 \@xprotect#1%
83 \fi}
```

`\@yargdef` These generate a primitive argument specification, from a L<sup>A</sup>T<sub>E</sub>X [*<digit>*] form; in fact *<digit>* can be anything such that `\number <digit>` is single digit.

`\@yargdef` Reorganised slightly so that `\renewcommand{\reserved@a}[1]{foo}` works. I am not sure this is worth it, as a following `\newcommand` would over-write the definition of `\reserved@a`.

Recall that L<sup>A</sup>T<sub>E</sub>X2.09 goes into an infinite loop with  
`\renewcommand[1]{\@tempa}{foo}`  
(DPC 6 October 93).

Reorganised again (DPC 1999). Rather than make a loop to construct the argument spec by counting, just extract the required argument spec by using a delimited argument (delimited by the digit). This is faster and uses less tokens. The coding is slightly odd to preserve the old interface (using `#2 = \tw@` as the flag to surround the first argument with `[]`). But the new method did not allow for the number of arguments `#3` not being given as an explicit digit; hence (further expansion of this argument and use of) `\number` was added later in 1999.

It is not clear why these are still `\long`.

```

84 \long \def \@yargdef #1#2#3{%
85   \ifx#2\tw@
86     \def\reserved@b##11{####1}}%
87   \else
88     \let\reserved@b\@gobble
89   \fi
90   \expandafter
91     \@yargd@f \expandafter{\number #3}#1%
92 }

93 \long \def \@yargd@f#1#2{%
94   \def \reserved@a ##1#1##2##{%
95     \expandafter\def\expandafter#2\reserved@b ##1#1%
96   }%
97   \l@ngrel@x \reserved@a 0##1##2##3##4##5##6##7##8##9##1%
98 }

```

`\@reargdef`

```

99 \long\def\@reargdef#1[#2]{%
100   \@yargdef#1\@ne{#2}}

```

`\renewcommand` Check the command name is already used. If not give an error message. Then temporarily disable `\@ifdefinable` then call `\newcommand`. (Previous version `\let#1=\relax` but this does not work too well if `#1` is `\@tempa-e`.)

```

101 \def\renewcommand{\@star@or@long\renew@command}

```

`\renew@command`

```

102 \def\renew@command#1{%
103   \begingroup \escapechar\m@ne\xdef\@gtempa{\string#1}\endgroup
104   \expandafter\@ifundefined\@gtempa
105     {\@latex@error{noexpand#1undefined}\@ehc}%
106     \relax
107   \let\@ifdefinable\@rc@ifdefinable
108   \new@command#1}

```

`\@ifdefinable` Test is user is allowed to define a command.

```

\@ifdefinable 109 \long\def\@ifdefinable #1#2{%
\@rc@ifdefinable 110   \edef\reserved@a{\expandafter\@gobble\string #1}%
111   \@ifundefined\reserved@a
112     {\edef\reserved@b{\expandafter\@carcube \reserved@a xxx\@nil}%
113     \ifx \reserved@b\@qend \@notdefinable\else

```

```

114         \ifx \reserved@a\@qrelax \@notdefinable\else
115             #2%
116         \fi
117     \fi}%
118     \@notdefinable}

```

Saved definition of \@ifdefinable.

```
119 \let\@@ifdefinable\@ifdefinable
```

Version of \@ifdefinable for use with \renewcommand. Does not do the check this time, but restores the normal definition.

```

120 \long\def\@rc@ifdefinable#1#2{%
121     \let\@ifdefinable\@@ifdefinable
122     #2}

```

**\newenvironment** Define a new user environment. #1 is the environment name. #2# Grabs all the tokens up to the first {. These will be any optional arguments. They are not parsed at this point, but are just passed to \@newenv which will eventually call \newcommand. Any optional arguments will then be parsed by \newcommand as it defines the command that executes the ‘begin code’ of the environment.

This #2# trick removed with version 1.2i as it fails if a { occurs in the optional argument. Now use \@ifnextchar directly.

```
123 \def\newenvironment{\@star@or@long\new@environment}
```

**\new@environment**

```

124 \def\new@environment#1{%
125     \@testopt{\@newenva#1}0}

```

**\@newenva**

```

126 \def\@newenva#1[#2]{%
127     \kernel@ifnextchar [{\@newenvb#1[#2]}{\@newenv{#1}{[#2]}}}]

```

**\@newenvb**

```
128 \def\@newenvb#1[#2][#3]{\@newenv{#1}{[#2][#3]}}
```

**\renewenvironment** Redefine an environment. For \renewenvironment disable \@ifdefinable and then call \newenvironment. It is OK to \let the argument to \relax here as there should not be a @temp... environment.

```
129 \def\renewenvironment{\@star@or@long\renew@environment}
```

**\renew@environment**

```

130 \def\renew@environment#1{%
131     \ifundefined{#1}%
132     {\@latex@error{Environment #1 undefined}\@ehc
133     }\relax
134     \expandafter\let\csname#1\endcsname\relax
135     \expandafter\let\csname end#1\endcsname\relax
136     \new@environment{#1}}

```

**\@newenv** The internal version of \newenvironment.

Call \newcommand to define the *begin-code* for the environment. \def is used for the *end-code* as it does not take arguments. (but may contain \pars)

Make sure that an attempt to define a ‘graf’ or ‘group’ environment fails.

```

137 \long\def\@newenv#1#2#3#4{%
138   \@ifundefined{#1}%
139     {\expandafter\let\csname#1\expandafter\endcsname
140      \csname end#1\endcsname}%
141   \relax
142   \expandafter\new@command
143     \csname #1\endcsname#2{#3}%
144     \l@ngrel@x\expandafter\def\csname end#1\endcsname{#4}}

```

`\newif` And here's a different sort of allocation: For example, `\newif\iffoo` creates `\footrue`, `\foofalse` to go with `\iffoo`.

```

145 \def\newif#1{%
146   \count@\escapechar \escapechar\m@ne
147   \let#1\iffalse
148   \@if#1\iftrue
149     \@if#1\iffalse
150   \escapechar\count@}

```

`\@if`

```

151 \def\@if#1#2{%
152   \expandafter\def\csname\expandafter\@gobbletwo\string#1%
153     \expandafter\@gobbletwo\string#2\endcsname
154     {\let#1#2}}

```

`\providecommand` `\providecommand` takes the same arguments as `\newcommand`, but discards them if #1 is already defined. Otherwise it just acts like `\newcommand`. This implementation currently leaves any discarded definition in `\reserved@a` (and possibly `\reserved@a`) this wastes a bit of space, but it will be reclaimed as soon as these scratch macros are redefined.

```

155 \def\providecommand{\@star@or@long\provide@command}

```

`\provide@command`

```

156 \def\provide@command#1{%
157   \begingroup
158   \escapechar\m@ne\xdef\@gtempa{\string#1}%
159   \endgroup
160   \expandafter\@ifundefined\@gtempa
161     {\def\reserved@a{\new@command#1}}%
162     {\def\reserved@a{\renew@command\reserved@a}}%
163   \reserved@a}%

```

`\CheckCommand` `\CheckCommand` takes the same arguments as `\newcommand`. If the command already exists, with the same definition, then nothing happens, otherwise a warning is issued. Useful for checking the current state before a macro package starts redefining things. Currently two macros are considered to have the same definition if they are the same except for different default arguments. That is, if the old definition was: `\newcommand\xxx[2][a]{(#1)(#2)}` then `\CheckCommand\xxx[2][b]{(#1)(#2)}` would *not* generate a warning, but, for instance `\CheckCommand\xxx[2]{(#1)(#2)}` would.

```

164 \def\CheckCommand{\@star@or@long\check@command}

```

`\CheckCommand` is only available in the preamble part of the document.

```

165 \@onlypreamble\CheckCommand

```

```

\check@command
166 \def\check@command#1#2#{\@check@c#1{#2}}
167 \@onlypreamble\check@command

\@check@c \CheckCommand itself just grabs all the arguments we need, without actually look-
ing for [ optional argument forms. Now define \reserved@a. If \reserved@a is
then defined, compare it with the “\#1’ otherwise compare \reserved@a with #1.
168 \long\def\@check@c#1#2#3{%
169   \expandafter\let\csname\string\reserved@a\endcsname\relax
170   \renew@command\reserved@a#2{#3}%
171   \@ifundefined{\string\reserved@a}%
172   {\@check@eq#1\reserved@a}%
173   {\expandafter\@check@eq
174     \csname\string#1\expandafter\endcsname
175     \csname\string\reserved@a\endcsname}}
176 \@onlypreamble\@check@c

\@check@eq Complain if #1 and #2 are not \ifx equal.
177 \def\@check@eq#1#2{%
178   \ifx#1#2\else
179     \@latex@warning@no@line
180       {Command \noexpand#1 has
181        changed.\MessageBreak
182        Check if current package is valid}%
183   \fi}
184 \@onlypreamble\@check@eq

\@gobble The \@gobble macro is used to get rid of its argument.
\@gobbletwo 185 \long\def \@gobble #1{}
\@gobblefour 186 \long\def \@gobbletwo #1#2{}
187 \long\def \@gobblefour #1#2#3#4{}

\@firstofone Some argument-grabbers.
\@firstoftwo 188 \long\def \@firstofone#1{#1}
\@secondoftwo 189 \long\def \@firstoftwo#1#2{#1}
190 \long\def \@secondoftwo#1#2{#2}

\@iden \@iden is another name for \@firstofone for compatibility reasons.
191 \let\@iden\@firstofone

\@thirdofthree Another grabber now used in the encoding specific section.
192 \long\def \@thirdofthree#1#2#3{#3}

\@expandtwoargs A macro to totally expand two arguments to another macro
193 \def\@expandtwoargs#1#2#3{%
194 \edef\reserved@a{\noexpand#1{#2}{#3}}\reserved@a}

\@backslashchar A category code 12 backslash.
195 \edef\@backslashchar{\expandafter\@gobble\string\}

```

## 11.4 Robust commands and protect

Fragile and robust commands are one of the thornier issues in L<sup>A</sup>T<sub>E</sub>X's commands. Whilst typesetting documents, L<sup>A</sup>T<sub>E</sub>X makes use of many of T<sub>E</sub>X's features, such as arithmetic, defining macros, and setting variables. However, there are (at least) three different occasions when these commands are not safe. These are called 'moving arguments' by L<sup>A</sup>T<sub>E</sub>X, and consist of:

- writing information to a file, such as indexes or tables of contents.
- writing information to the screen.
- inside an `\edef`, `\message`, `\mark`, or other command which evaluates its argument fully.

The method L<sup>A</sup>T<sub>E</sub>X uses for making fragile commands robust is to precede them with `\protect`. This can have one of five possible values:

- `\relax`, for normal typesetting. So `\protect\foo` will execute `\foo`.
- `\string`, for writing to the screen. So `\protect\foo` will write `\foo`.
- `\noexpand`, for writing to a file. So `\protect\foo` will write `\foo` followed by a space.
- `\@unexpandable@protect`, for writing a moving argument to a file. So `\protect\foo` will write `\protect\foo` followed by a space. This value is also used inside `\edefs`, `\marks` and other commands which evaluate their arguments fully.
- `\@unexpandable@noexpand`, for performing a deferred write inside an `\edef`. So `\protect\foo` will write `\foo` followed by a space. If you want `\protect\foo` to be written, you should use `\@unexpandable@protect`. (Removed as never used).

```
\@unexpandable@protect  These commands are used for setting \protect inside \edefs.
\@unexpandable@noexpand 196 \def\@unexpandable@protect{\noexpand\protect\noexpand}
                        197 %\def\@unexpandable@noexpand{\noexpand\noexpand\noexpand}
```

```
\DeclareRobustCommand  This is a package-writers command, which has the same syntax as \newcommand,
\declare@robustcommand but which declares a protected command. It does this by having
                        \DeclareRobustCommand\foo
                        define \foo to be \protect\foo<space>,
                        and then use \newcommand\foo<space>.
                        Since the internal command is \foo<space>, when it is written to an auxiliary
                        file, it will appear as \foo.
```

We have to be a bit cleverer if we're defining a short command, such as `\_`, in order to make sure that the auxiliary file does not include a space after the command, since `\_ a` and `\_a` aren't the same. In this case we define `\_` to be:

```
\x@protect\_ \protect\_<space>
```

which expands to:

```

\ifx\protect\@typeset@protect\else
  \x@protect\
\fi
\protect\_<space>

```

Then if `\protect` is `\@typeset@protect` (normally `\relax`) then we just perform `\_<space>`, and otherwise `\x@protect@` gobbles everything up and expands to `\protect\_.`

*Note:* setting `\protect` to any value other than `\relax` whilst in ‘typesetting’ mode will cause commands to go into an infinite loop! In particular, setting `\relax` to `\@empty` will cause `\_` to loop forever. It will also break lots of other things, such as protected `\ifmmodes` inside `\haligns`. If you really really have to do such a thing, then please set `\@typeset@protect` to be `\@empty` as well. (This is what the code for `\patterns` does, for example.)

More fun with `\expandafter` and `\csname`.

```

198 \def\DeclareRobustCommand{\@star@or@long\declare@robustcommand}
199 \def\declare@robustcommand#1{%
200   \ifx#1\@undefined\else\ifx#1\relax\else
201     \@latex@info{Redefining \string#1}%
202   \fi\fi
203   \edef\reserved@a{\string#1}%
204   \def\reserved@b{#1}%
205   \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
206   \edef#1{%
207     \ifx\reserved@a\reserved@b
208       \noexpand\x@protect
209       \noexpand#1%
210     \fi
211     \noexpand\protect
212     \expandafter\newcommand\csname
213       \expandafter\@gobble\string#1 \endcsname
214   }%
215   \let\@ifdefinable\rc@ifdefinable
216   \expandafter\newcommand\csname
217     \expandafter\@gobble\string#1 \endcsname
218 }

```

```

\x@protect
\x@protect 219 \def\x@protect#1{%
220   \ifx\protect\@typeset@protect\else
221     \x@protect#1%
222   \fi
223 }
224 \def\x@protect#1\fi#2#3{%
225   \fi\protect#1%
226 }

```

```

\@typeset@protect
227 \let\@typeset@protect\relax

```

```

\set@display@protect These macros set \protect appropriately for typesetting or displaying.
\set@typeset@protect 228 \def\set@display@protect{\let\protect\string}
229 \def\set@typeset@protect{\let\protect\@typeset@protect}

\protected@edef The commands \protected@edef and \protected@xdef perform ‘safe’ \edefs
\protected@xdef and \xdefs, saving and restoring \protect appropriately. For cases where restoring
\unrestored@protected@xdef \protect doesn’t matter, there’s an ‘unsafe’ \unrestored@protected@xdef,
\restore@protect useful if you know what you’re doing!
230 \def\protected@edef{%
231   \let\@protect\protect
232   \let\protect\@unexpandable@protect
233   \afterassignment\restore@protect
234   \edef
235 }
236 \def\protected@xdef{%
237   \let\@protect\protect
238   \let\protect\@unexpandable@protect
239   \afterassignment\restore@protect
240   \xdef
241 }
242 \def\unrestored@protected@xdef{%
243   \let\protect\@unexpandable@protect
244   \xdef
245 }
246 \def\restore@protect{\let\protect\@protect}

\protect The normal meaning of \protect
247 \set@typeset@protect

\MakeRobust The macro firstly checks if the controls sequence in question exists at all.
248 </2ekernel>
249 <latexrelease>\IncludeInRelease{2015/01/01}{\MakeRobust}{\MakeRobust}%
250 <*2ekernel|latexrelease>
251 \def\MakeRobust#1{%
252   \ifundefined{\expandafter\@gobble\string#1}{%
253     \latex@error{The control sequence ‘\string#1’ is undefined!%
254     \MessageBreak There is nothing here to make robust}%
255     \@eha
256   }%

Then we check if the macro is already robust. We do this by testing if the internal
name for a robust macro is defined, namely \foo_␣. If it is already defined do
nothing, otherwise set \foo_␣ equal to \foo and redefine \foo so that it acts like
a macro defined with \DeclareRobustCommand.

257   {%
258     \ifundefined{\expandafter\@gobble\string#1\space}%
259     {%
260       \expandafter\let\csname
261       \expandafter\@gobble\string#1\space\endcsname=#1%
262       \edef\reserved@a{\string#1}%
263       \def\reserved@b{#1}%
264       \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
265       \edef#1{%

```

```

266      \ifx\reserved@a\reserved@b
267      \noexpand\x@protect\noexpand#1%
268      \fi
269      \noexpand\protect\expandafter\noexpand
270      \csname\expandafter\@gobble\string#1\space\endcsname}%
271  }%
272  {\@latex@info{The control sequence ‘\string#1’ is already robust}}%
273  }%
274 }%
275 </2ekernel | latexrelease>
276 <latexrelease>\EndIncludeInRelease
277 <latexrelease>\IncludeInRelease{0000/00/00}{\MakeRobust}{\MakeRobust}%
278 <latexrelease>\let\MakeRobust\@undefined
279 <latexrelease>\EndIncludeInRelease
280 (*2ekernel)

```

## 11.5 Internal defining commands

These commands are used internally to define other L<sup>A</sup>T<sub>E</sub>X commands.

**\@ifundefined** Check if first arg is undefined or **\relax** and execute second or third arg depending,

```

281 \def\@ifundefined#1{%
282   \expandafter\ifx\csname#1\endcsname\relax
283   \expandafter\@firstoftwo
284   \else
285   \expandafter\@secondoftwo
286   \fi}

```

**\@qend** The following define **\@qend** and **\@qrelax** to be the strings ‘end’ and ‘relax’  
**\@qrelax** with the characters **\catcoded 12**.

```

287 \edef\@qend{\expandafter\@cdr\string\end\@nil}
288 \edef\@qrelax{\expandafter\@cdr\string\relax\@nil}

```

**\@ifnextchar** **\@ifnextchar** peeks at the following character and compares it with its first argument. If both are the same it executes its second argument, otherwise its third.

```

289 \long\def\@ifnextchar#1#2#3{%
290   \let\reserved@d=#1%
291   \def\reserved@a{#2}%
292   \def\reserved@b{#3}%
293   \futurelet\@let@token\@ifnch}

```

**\kernel@ifnextchar** This macro is the kernel version of **\@ifnextchar** which is used in a couple of places to prevent the AMS variant from being used since in some places this produced chaos (for example if an **fd** file is loaded in a random place then the optional argument to **\ProvidesFile** could get printed there instead of being written only in the log file. This happened when there was a space or a newline between the mandatory and optional arguments! It should really be fixed in the **amsmath** package one day, but...

Note that there may be other places in the kernel where this version should be used rather than the original, but variable, version.

```

294 \let\kernel@ifnextchar\@ifnextchar

```

`\@ifnch` `\@ifnch` is a tricky macro to skip any space tokens that may appear before the character in question. If it encounters a space token, it calls `xifnch`.

```
295 \def\@ifnch{%
296   \ifx\@let@token\@sptoken
297     \let\reserved@c\@xifnch
298   \else
299     \ifx\@let@token\reserved@d
300       \let\reserved@c\reserved@a
301     \else
302       \let\reserved@c\reserved@b
303     \fi
304   \fi
305   \reserved@c}
```

`\@sptoken` The following code makes `\@sptoken` a space token. It is important here that the control sequence `\:` consists of a non-letter only, so that the following whitespace is significant. Together with the fact that the equal sign in a `\let` may be followed by only one optional space the desired effect is achieved. NOTE: the following hacking must precede the definition of `\:` as math medium space.

```
306 \def\:\{\let\@sptoken= } \: % this makes \@sptoken a space token
```

`\@xifnch` In the following definition of `\@xifnch`, `\:` is again used to get a space token as delimiter into the definition.

```
307 \def\:\{\@xifnch} \expandafter\def\:\: {\futurelet\@let@token\@ifnch}
```

`\makeatletter` Make internal control sequences accessible or inaccessible.

```
\makeatother 308 \def\makeatletter{\catcode'\@11\relax}
309 \def\makeatother{\catcode'\@12\relax}
```

`\@ifstar` The new implementation below avoids passing the *<true code>* Through one more `\def` than the *<false code>*, which previously meant that `#` had to be written as `####` in one argument, but `##` in the other. The `*` is gobbled by `\@firstoftwo`.

```
310 \def\@ifstar#1{\@ifnextchar *{\@firstoftwo{#1}}}
```

`\@dblarg`

```
\@xdblarg 311 \long\def\@dblarg#1{\kernel@ifnextchar[{\#1}{\@xdblarg{#1}}}
312 \long\def\@xdblarg#1#2#1[{\#2}]{\#2}}
```

`\@sanitize` The command `\@sanitize` changes the catcode of all special characters except for braces to ‘other’. It can be used for commands like `\index` that want to write their arguments verbatim. Needless to say, this command should only be executed within a group, or chaos will ensue.

```
313 \def\@sanitize{\@makeother\ \@makeother\\\@makeother$\@makeother\&%
314 \@makeother\#\@makeother^\@makeother_\@makeother%\@makeother\~}
```

`\@onelevel@sanitize` This makes the whole “meaning” of `#1` (its one-level expansion) into catcode 12 tokens: it could be used in `\DeclareRobustCommand`.

If it is to be used on default float specifiers, this should be done when they are defined.

```
315 \def \@onelevel@sanitize #1{%
316   \edef #1{\expandafter\strip@prefix
317             \meaning #1}%
318 }
```

319  $\langle /2\text{ekernel} \rangle$

# File e

## lalloc.dtx

### 12 Counters

This section deals with counter and other variable allocation.

1 (\*2ekernel)

The following are from plain T<sub>E</sub>X:

\z@ A zero dimen or number. It's more efficient to write \parindent\z@ than  
\parindent 0pt.

\@ne The number 1.

\m@ne The number −1.

\tw@ The number 2.

\sixt@@n The number 16.

\@m The number 1000.

\@MM The number 20000.

\@xxxii The constant 32.

2 \chardef\@xxxii=32

\@Mi Constants 1001–1004.

\@Mii 3 \mathchardef\@Mi=10001

\@Miii 4 \mathchardef\@Mii=10002

\@Miv 5 \mathchardef\@Miii=10003

6 \mathchardef\@Miv=10004

\@tempcnta Scratch count registers used by L<sup>A</sup>T<sub>E</sub>X kernel commands.

\@tempcntb 7 \newcount\@tempcnta

8 \newcount\@tempcntb

\if@tempswa General boolean switch used by L<sup>A</sup>T<sub>E</sub>X kernel commands.

9 \newif\if@tempswa

\@tempdima Scratch dimen registers used by L<sup>A</sup>T<sub>E</sub>X kernel commands.

\@tempdimb 10 \newdimen\@tempdima

\@tempdimc 11 \newdimen\@tempdimb

12 \newdimen\@tempdimc

\@tempboxa Scratch box register used by L<sup>A</sup>T<sub>E</sub>X kernel commands.

13 \newbox\@tempboxa

\@tempskipa Scratch skip registers used by L<sup>A</sup>T<sub>E</sub>X kernel commands.

\@tempskipb 14 \newskip\@tempskipa

15 \newskip\@tempskipb

```

\@temptokena Scratch token register used by LATEX kernel commands.
16 \newtoks\@temptokena

\@flushglue Glue used for \right- & \leftskip = 0pt plus 1fil
17 \newskip\@flushglue \@flushglue = 0pt plus 1fil
18 \>/2ekernel)

```

# File f

## ltnctrl.dtx

### 13 Program control structure

This section defines a number of control structure macros, such as while-loops and for-loops.

```

1 \<2kernel>
2 \message{control,}

\@whilenum TEST \do {BODY}
\@whiledim TEST \do {BODY} : These implement the loop
    while TEST do BODY od
    where TEST is a TeX \ifnum or \ifdim test, respectively.
    They are optimized for the normal case of TEST initially false.

\@whilesw SWITCH \fi {BODY} : Implements the loop
    while SWITCH do BODY od
    Optimized for normal case of SWITCH initially false.

\@for NAME := LIST \do {BODY} : Assumes that LIST expands to
A1,A2,
    ... ,An .
    Executes BODY n times, with NAME = Ai on the i-th
iteration.
    Optimized for the normal case of n = 1. Works for n=0.

\@tfor NAME := LIST \do {BODY}
    if, before expansion, LIST = T1 ... Tn where each Ti is a
    token or {...}, then executes BODY n times, with NAME = Ti
    on the i-th iteration. Works for n=0.

NOTES: 1. These macros use no \@temp sequences.
        2. These macros do not work if the body contains anything that
        looks syntactically to TeX like an improperly balanced \if
        \else \fi.

\@whilenum TEST \do {BODY} ==
BEGIN
    if TEST
    then BODY
        \@iwhilenum{TEST \relax BODY}
END

\@iwhilenum {TEST BODY} ==
BEGIN
    if TEST
    then BODY

```

```

        \@nextwhile = def(\@iwhilenum)
      else \@nextwhile = def(\@whilenoop)
    fi
    \@nextwhile {TEST BODY}
  END

\@whilesw SWITCH \fi {BODY} ==
BEGIN
  if SWITCH
    then BODY
      \@iwhilesw {SWITCH BODY}\fi
    fi
  END

\@iwhilesw {SWITCH BODY} \fi ==
BEGIN
  if SWITCH
    then BODY
      \@nextwhile = def(\@iwhilesw)
    else \@nextwhile = def(\@whileswnoop)
    fi
    \@nextwhile {SWITCH BODY} \fi
  END

\@whilenoop
\@whilenum 3 \long\def\@whilenum#1\do #2{\ifnum #1\relax #2\relax\@iwhilenum{#1\relax
\@iwhilenum 4 #2\relax}\fi}
5 \long\def\@iwhilenum#1{\ifnum #1\expandafter\@iwhilenum
6 \else\expandafter\@gobble\fi{#1}}

\@whiledim
\@iwhiledim 7 \long\def\@whiledim#1\do #2{\ifdim #1\relax#2\@iwhiledim{#1\relax#2}\fi}
8 \long\def\@iwhiledim#1{\ifdim #1\expandafter\@iwhiledim
9 \else\expandafter\@gobble\fi{#1}}

\@whileswnoop
\@whilesw 10 \long\def\@whilesw#1\fi#2{#1#2\@iwhilesw{#1#2}\fi\fi}
\@iwhilesw 11 \long\def\@iwhilesw#1\fi{#1\expandafter\@iwhilesw
12 \else\@gobbletwo\fi{#1}\fi}

\@for NAME := LIST \do {BODY} ==
BEGIN \@forloop expand(LIST),\@nil,\@nil \@@ NAME {BODY}
END

\@forloop CAR, CARCDR, CDRCDR \@@ NAME {BODY} ==
BEGIN
  NAME = CAR
  if def(NAME) = def(\@nnil)
    else BODY;

```

```

NAME = CARCDR
if def(NAME) = def(\@nnil)
  else BODY
    \@iforloop CDRCDR \@@ NAME \do {BODY}
  fi
fi
END

\@iforloop CAR, CDR \@@ NAME {BODY} =
  NAME = CAR
  if def(NAME) = def(\@nnil)
    then \@nextwhile = def(\@fornoop)
    else BODY ;
    \@nextwhile = def(\@iforloop)
  fi
  \@nextwhile name cdr {body}

\@tfor NAME := LIST \do {BODY}
  = \@tforloop LIST \@nil \@@ NAME {BODY}

\@tforloop car cdr \@@ name {body} =
  name = car
  if def(name) = def(\@nnil)
    then \@nextwhile == \@fornoop
    else body ;
    \@nextwhile == \@forloop
  fi
  \@nextwhile name cdr {body}

\@nnil
13 \def\@nnil{\@nil}

\@empty
14 \def\@empty{}

\@fornoop
15 \long\def\@fornoop#1\@#2#3{}

\@for
16 \long\def\@for#1:=#2\do#3{%
17   \expandafter\def\expandafter\@fortmp\expandafter{#2}%
18   \ifx\@fortmp\@empty \else
19     \expandafter\@forloop#2,\@nil,\@nil\@#1{#3}\fi}

\@forloop
20 \long\def\@forloop#1,#2,#3\@#4#5{\def#4{#1}\ifx #4\@nnil \else
21   #5\def#4{#2}\ifx #4\@nnil \else#5\@iforloop #3\@#4{#5}\fi\fi}

\@iforloop
22 \long\def\@iforloop#1,#2\@#3#4{\def#3{#1}\ifx #3\@nnil
23   \expandafter\@fornoop \else
24   #4\relax\expandafter\@iforloop\fi#2\@#3{#4}}

```

```

\@tfor
25 \def\@tfor#1:={\@tfor#1 }
26 \long\def\@tfor#1#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\space\else
27   \@tforloop#2\@nil\@nil\@#1{#3}\fi}
28 \long\def\@tforloop#1#2\@#3#4{\def#3{#1}\ifx #3\@nnil
29   \expandafter\@fornoop \else
30   #4\relax\expandafter\@tforloop\fi#2\@#3{#4}}

\@break@tfor Break out of a \@tfor loop. This should be called inside the scope of an \if. See
\iffilenamepath for an example.
31 \long\def\@break@tfor#1\@#2#3{\fi\fi}

\@removeelement Removes an element from a comma-separated list and puts it into a control se-
quence, called as \@removeelement{<element>}{<list>}{<cs>}. Due to the imple-
mentation method the <element> is not allowed to contain braces.
32 \def\@removeelement#1#2#3{%
33   \def\reserved@a##1,#1,##2\reserved@a{##1,##2\reserved@b}%
34   \def\reserved@b##1,\reserved@b##2\reserved@b{%
35     \ifx,##1\@empty\else##1\fi}%
36   \edef#3{%
37     \expandafter\reserved@b\reserved@a,#2,\reserved@b,#1,\reserved@a}}

38 </2ekernel>

```

# File g

## lterror.dtx

### 14 Error handling

This section defines L<sup>A</sup>T<sub>E</sub>X's error commands.

```
1 (*2ekernel)
```

The '2ekernel' code ensures that a `\usepackage{autoerr}` is essentially ignored if a 'full' format is being used that has the error messages already in the format.

These days we don't support autoloading approach any longer, but this part bit is kept in case it is used in old documents.

```
2 \expandafter\let\csname ver@autoerr.sty\endcsname\fmtversion
```

#### 14.1 General commands

**\MessageBreak** This command prints a new-line inside a message, followed by a continuation line begun with `\@msg@continuation`. Normally it is defined to be `\relax`, but inside messages, it is let to `\@message@break`.

```
3 \let\MessageBreak\relax
```

**\GenericInfo** This takes two arguments: a continuation and a message, and sends the result to the log file.

```
4 \DeclareRobustCommand{\GenericInfo}[2]{%
5   \begingroup
6     \def\MessageBreak{^^J#1}%
7     \set@display@protect
8     \immediate\write\m@ne{#2\on@line.}%
9   \endgroup
10 }
```

**\GenericWarning** This takes two arguments: a continuation and a message, and sends the result to the screen.

```
11 \DeclareRobustCommand{\GenericWarning}[2]{%
12   \begingroup
13     \def\MessageBreak{^^J#1}%
14     \set@display@protect
15     \immediate\write\@unused{^^J#2\on@line.^^J}%
16   \endgroup
17 }
```

**\GenericError** This macro takes four arguments: a continuation, an error message, where to go for further information, and the help information. It displays the error message, and sets the error help (the result of typing `h` to the prompt), and does a horrible hack to turn the last context line (which by default is the only context line) into just three dots. This could be made more efficient.

```
18 \bgroup
19 \lccode'\@=' \ %
```

```

20 \lccode'\~=' \ %
21 \lccode'\}=' \ %
22 \lccode'\{=' \ %
23 \lccode'\T=' \T%
24 \lccode'\H=' \H%
25 \catcode'\ =11\relax%
26 \lowercase{%
27 \egroup%

```

Unfortunately T<sub>E</sub>X versions older than 3.141 have a bug which means that `^^J` does not force a linebreak in `\message` and `\errmessage` commands. So for these old T<sub>E</sub>X's we use `\typeout` to produce the message, and then have an empty `\errmessage` command. This causes an extra line of the form

! .

To appear on the terminal, but if you do not like it, you can always upgrade your T<sub>E</sub>X! In order for your format to use this version, you must define the macro `\@TeXversion` to be the version number, e.g., 3.14 of the underlying T<sub>E</sub>X. See the comments in `ltdircheck.dtx`.

```

28 \dimen@ \ifx\@TeXversion\undefined4\else\@TeXversion\fi\p@%
29 \ifdim\dimen@>3.14\p@%

```

First the 'standard case'.

```

30 \DeclareRobustCommand{\GenericError}[4]{%
31 \begingroup%
32 \immediate\write\@unused{}%
33 \def\MessageBreak{^^J}%
34 \set@display@protect%
35 \edef%
36 %    %<-----do not delete this space!----->%
37 \@err@
38 {{#4}}%
39 \errhelp
40 %    %<-----do not delete this space!----->%
41 \@err@
42 \let
43 %    %<-----do not delete this space!----->%
44 \@err@
45 \@empty
46 \def\MessageBreak{^^J#1}%
47 \def~{\errmessage{%
48 #2.^^J^^J%
49 #3^^J%
50 Type H <return> for immediate help%
51 %    %<-----do not delete this space!----->%
52 \@err@
53 }}%
54 ~%
55 \endgroup}%
56 \else%

```

Secondly the version for old T<sub>E</sub>X's.

```

57 \DeclareRobustCommand{\GenericError}[4]{%
58 \begingroup%

```

```

59 \immediate\write\@unused{}%
60 \def\MessageBreak{^^J}%
61 \set@display@protect%
62 \edef%
63 %    %<-----do not delete this space!----->%
64 \@err@ %
65 {{#4}}%
66 \errhelp
67 %    %<-----do not delete this space!----->%
68 \@err@ %
69 \let
70 %    %<-----do not delete this space!----->%
71 \@err@ %
72 \errmessage
73 \def\MessageBreak{^^J#1}%
74 \def~{\typeout{! %
75 #2.^^J^^J%
76 #3^^J%
77 Type H <return> for immediate help.}%
78 %    %<-----do not delete this space!----->%
79 \@err@ %
80 {}}%
81 ~%
82 \endgroup}%
83 \fi}%

```

`\PackageError` These commands are intended for use by package and class writers, to give information to authors. The syntax is:

<code>\PackageWarning</code>	<code>\PackageError{&lt;package&gt;}{&lt;error&gt;}{&lt;help&gt;}</code>
<code>\PackageWarningNoLine</code>	<code>\PackageWarning{&lt;package&gt;}{&lt;warning&gt;}</code>
<code>\PackageInfo</code>	<code>\PackageWarningNoLine{&lt;package&gt;}{&lt;warning&gt;}</code>
<code>\ClassError</code>	<code>\PackageInfo{&lt;package&gt;}{&lt;info&gt;}</code>
<code>\ClassWarning</code>	
<code>\ClassWarningNoLine</code>	
<code>\ClassInfo</code>	

and similarly for classes. The `Error` commands print the `<error>` message, and present the interactive prompt; if the author types `h`, then the `<help>` information is displayed. The `Warning` commands produce a warning but do not present the interactive prompt. The `WarningNoLine` commands do the same, but don't print the input line number. The `Info` commands write the message to the `log` file. Within the messages, the command `\MessageBreak` can be used to break a line, `\protect` can be used to protect command names, and `\space` is a space, for example:

```

\newcommand{\foo}{F00}
\PackageWarning{ethel}{%
  Your hovercraft is full of eels,\MessageBreak
  and \protect\foo\space is \foo}

```

produces:

```

Package ethel warning: Your hovercraft is full of eels,
(ethel)                and \foo is F00 on input line 54.

```

```

84 \gdef\PackageError#1#2#3{%
85   \GenericError{%
86     (#1)\@spaces\@spaces\@spaces\@spaces
87   }{%
88     Package #1 Error: #2%
89   }{%
90     See the #1 package documentation for explanation.%
91   }{#3}%
92 }

93 \def\PackageWarning#1#2{%
94   \GenericWarning{%
95     (#1)\@spaces\@spaces\@spaces\@spaces
96   }{%
97     Package #1 Warning: #2%
98   }%
99 }

100 \def\PackageWarningNoLine#1#2{%
101   \PackageWarning{#1}{#2@gobble}%
102 }

103 \def\PackageInfo#1#2{%
104   \GenericInfo{%
105     (#1) \@spaces\@spaces\@spaces
106   }{%
107     Package #1 Info: #2%
108   }%
109 }

110 \gdef\ClassError#1#2#3{%
111   \GenericError{%
112     (#1) \space\@spaces\@spaces\@spaces
113   }{%
114     Class #1 Error: #2%
115   }{%
116     See the #1 class documentation for explanation.%
117   }{#3}%
118 }

119 \def\ClassWarning#1#2{%
120   \GenericWarning{%
121     (#1) \space\@spaces\@spaces\@spaces
122   }{%
123     Class #1 Warning: #2%
124   }%
125 }

126 \def\ClassWarningNoLine#1#2{%
127   \ClassWarning{#1}{#2@gobble}%
128 }

129 \def\ClassInfo#1#2{%
130   \GenericInfo{%
131     (#1) \space\space\@spaces\@spaces
132   }{%
133     Class #1 Info: #2%
134   }%
135 }

```

```

\@latex@error Errors and other info, for use in the LATEX core.
\@latex@warning 136 \gdef\@latex@error#1#2{%
\@latex@warning@no@line 137 \GenericError{%
\@latex@info 138 \space\space\space\@spaces\@spaces\@spaces
\@latex@info@no@line 139 }{%
140 LaTeX Error: #1%
141 }{%
142 See the LaTeX manual or LaTeX Companion for explanation.%
143 }{#2}%
144 }

145 \def\@latex@warning#1{%
146 \GenericWarning{%
147 \space\space\space\@spaces\@spaces\@spaces
148 }{%
149 LaTeX Warning: #1%
150 }%
151 }

152 \def\@latex@warning@no@line#1{%
153 \@latex@warning{#1\@gobble}}

154 \def\@latex@info#1{%
155 \GenericInfo{%
156 \@spaces\@spaces\@spaces
157 }{%
158 LaTeX Info: #1%
159 }%
160 }

161 \def\@latex@info@no@line#1{%
162 \@latex@info{#1\@gobble}}

\@font@warning and \@font@info are defined later since they have to be
redefined by the tracefnt package.

\def\@font@warning#1{%
\GenericWarning{%
{(font)\@spaces\@spaces}%
{Font Warning: #1}%
}
\def\@font@info#1{%
\GenericInfo{%
(font)\space\@spaces
}%
Font Info: #1%
}%
}

\c@errorcontextlines \errorcontextlines as a LATEX counter, so that it may be manipulated with
\setcounter (once it is defined :-))
163 \let\c@errorcontextlines\errorcontextlines
164 \c@errorcontextlines=-1

\on@line The message ‘ on input line n’, if possible.
165 \ifnum\inputlineno=\m@ne

```

```

166 \let\on@line\@empty
167 \else
168 \def\on@line{ on input line \the\inputlineno}
169 \fi

\@warning Older LATEX messages. For the moment, these \let to the new message commands.
\@warning They may be changed later, once only obsolete packages and classes contain them.
\@latexerr

170 \let\@warning\@latex@warning
171 \let\@warning\@latex@warning\on@line
172 \global\let\@latexerr\@latex@error

```

\@spaces Four spaces.

```

173 \def\@spaces{\space\space\space\space}

```

## 14.2 Specific errors

\@eha The more common error help messages.

```

\@ehb 174 \gdef\@eha{%
\@ehc 175 Your command was ignored.\MessageBreak
\@ehd 176 Type \space I <command> <return> \space to replace it %
177 with another command,\MessageBreak
178 or \space <return> \space to continue without it.}
179 \gdef\@ehb{%
180 You've lost some text. \space \@ehc}
181 \gdef\@ehc{%
182 Try typing \space <return> %
183 \space to proceed.\MessageBreak
184 If that doesn't work, type \space X <return> \space to quit.}
185 \gdef\@ehd{%
186 You're in trouble here. \space \@ehc}

```

\@notdefinable Error message generated in \@ifdefinable from calls to one of the commands \newcommand, \newlength or \newtheorem specifying an already-defined command name or one that begins \end....

```

187 \gdef\@notdefinable{%
188 \@latex@error{%
189 Command \@backslashchar\reserved@a\space
190 already defined.\MessageBreak
191 Or name \@backslashchar\@qend... illegal,
192 see p.192 of the manual}\@eha}

```

\@nolnerr Generated by \newline and \\ when called in vertical mode.

```

193 \gdef\@nolnerr{%
194 \@latex@error{There's no line here to end}\@eha}

```

\@nocounterr Generated by \setcounter, \addtocounter or \newcounter if applied to an undefined counter <cnt>.

\@nocnterr Obsolete error message generated in L<sup>A</sup>T<sub>E</sub>X 2.09 by \setcounter, \addtocounter or \newcounter for undefined counter. DO NOT use for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> it MIGHT vanish! Use \@nocounterr{<cnt>} instead.

```

195 \gdef\@nocounterr#1{%
196   \latexerror{No counter '#1' defined}\@eha}
197 \gdef\@nocnterr{\@nocounterr?}

\@ctrerr Called when trying to print the value of a counter numbered by letters that's
greater than 26.
198 \gdef\@ctrerr{%
199   \latexerror{Counter too large}\@ehb}

\@nodocument Error produced if paragraphs are typeset in the preamble.
200 \gdef\@nodocument{%
201   \latexerror{Missing \protect\begin{document}}\@ehd}

\@badend Called by \end that doesn't match its \begin. RmS 1992/08/24: added code to
\@badend to display position of non-matching \begin. FMi 1993/01/14: missing
space added.
202 \gdef\@badend#1{%
203   \latexerror{\protect\begin{\@currenvir}\@currenvline
204               \space ended by \protect\end{#1}}\@eha}

\@badmath Called by \[, \], \( or \) when used in wrong mode.
205 \gdef\@badmath{%
206   \latexerror{Bad math environment delimiter}\@eha}

\@toodeep Called by a list environment nested more than six levels deep, or an enumerate or
itemize nested more than four levels.
207 \gdef\@toodeep{%
208   \latexerror{Too deeply nested}\@ehd}

\@badpoptabs Called by \endtabbing when not enough \poptabs have occurred, or by \poptabs
when too many have occurred.
209 \gdef\@badpoptabs{%
210   \latexerror{\protect\pushtabs\space and \protect\poptabs
211               \space don't match}\@ehd}

\@badtab Called by \>, \+ , \- or \< when stepping to an undefined tab.
212 \gdef\@badtab{%
213   \latexerror{Undefined tab position}\@ehd}

\@preamerr This error is special: it appears in places where we normally have to \protect
expansions. However, to prevent a protection of the error message itself (which
would result in the message getting printed not issued on the terminal) we need
to locally reset \protect to \relax.
214 \gdef\@preamerr#1{%
215   \begingroup
216     \let\protect\relax
217     \latexerror{\ifcase #1 Illegal character\or
218               Missing @-exp\or Missing p-arg\fi\space
219               in array arg}\@ehd
220   \endgroup}

```

`\@badlinearg` Occurs in `\line` and `\vector` command when a bad slope argument is encountered.

```

221 \gdef\@badlinearg{%
222   \@latex@error{%
223     Bad \protect\line\space or \protect\vector
224     \space argument}\@ehb}

```

`\@parmoderr` Occurs in a float environment or a `\marginpar` when encountered in inner vertical mode.

```

225 \gdef\@parmoderr{%
226   \@latex@error{Not in outer par mode}\@ehb}

```

`\@fltovf` Occurs in float environment or `\marginpar` when there are no more free boxes for storing floats.

```

227 \gdef\@fltovf{%
228   \@latex@error{Too many unprocessed floats}\@ehb}

```

`\@latexbug` Occurs in output routine. This is bad news.

```

229 \gdef\@latexbug{%
230   \@latex@error{This may be a LaTeX bug}{Call for help}}

```

`\@badcrerr` This error was removed and replaced by `\@nolnerr`.

```

231 %\def\@badcrerr {\@latex@error{Bad use of \protect\\}\@ehc}

```

`\@noitemerr` `\addvspace` or `\addpenalty` was called when not in vmode. Probably caused by a missing `\item`.

```

232 \gdef\@noitemerr{%
233   \@latex@error{Something's wrong--perhaps a missing %
234     \protect\item}\@ehc}

```

`\@notprerr` A command that can be used only in the preamble appears after the command `\begin{document}`.

```

235 \gdef\@notprerr{%
236   \@latex@error{Can be used only in preamble}\@eha}

```

`\@inmatherr` Issued by commands that don't work correctly within math (like `\item`). There is no real error recovery happening, e.g., the user might get additional errors afterwards.

```

237 \gdef\@inmatherr#1{%
238   \relax
239   \ifmmode
240     \@latex@error{Command \protect#1 invalid in math mode}\@ehc
241     \fi}

```

`\@invalidchar` An error for use with invalid characters. This is commented out, since we decided to use catcode 15 instead.

```

242 %\def\@invalidchar{\@latex@error{Invalid character in input}\@ehc}
243 \endkernel

```

As well as the above error commands some error messages are directly coded to save space. The Messages already present in  $\text{\LaTeX}2.09$  included:

`Environment --- undefined`  
 Issued by `\begin` for undefined environment.

`tab overflow`  
 Occurs in `\=` when maximum number of tabs exceeded.

`\< in mid line`  
 Occurs in `\<` when it appears in middle of line.

`Float(s) lost`  
 In output routine, caused by a float environment or `\marginpar` occurring in inner vertical mode.

# File h

## ltpar.dtx

### 15 Paragraphs

This section of the kernel declares the commands used to set `\par` and `\everypar` when ever their function needs to be changed for a long time.

#### 15.1 Implementation

There are two situations in which `\par` may be changed:

- Long-term changes, in which the new value is to remain in effect until the current environment is left. The environments that change `\par` in this way are the following:
  - All list environments (itemize, quote, etc.)
  - Environments that turn `\par` into a noop: tabbing, array and tabular.
- Temporary changes, in which `\par` is restored to its previous value the next time it is executed. The following are all such uses.
  - `\end` when preceded by `\@endparenv`, which is called by `\endtrivlist`
  - The mechanism for avoiding page breaks and getting the spacing right after section heads.

`\@setpar` To permit the proper interaction of these two situations, long-term changes are made by the `\@setpar{<VAL>}` command. It's function is:

To set `\par`. It `\def`'s `\par` and `\@par` to `<VAL>`.

`\@restorepar` Short-term changes are made by the usual `\def\par` commands. The original values are restored after a short-term change by the `\@restorepar` commands.

`\@@par` `\@@par` always is defined to be the original `TEX` `\par`.

`\everypar` `\everypar` is changed only for the short term. Whenever `\everypar` is set non-null, it should restore itself to null when executed.

The following commands change `\everypar` in this way:

- `\item`
- `\end` when preceded by `\@endparenv`, which is called by `\endtrivlist`
- `\minipage`

When dealing with `\par` and `\everypar` remember the following two warnings:

1. Commands that make short-term changes to `\par` and `\everypar` must take account of the possibility that the new commands and the ones that do the restoration may be executed inside a group. In particular, `\everypar` is executed inside a group whenever a new paragraph begins with a left brace. The `\everypar` command that restores its definition should be local to the current group (in case the command is inside a minipage used inside someplace

where `\everypar` has been redefined). Thus, if `\everypar` is redefined to do an `\everypar{}` it could take several executions of `\everypar` before the restoration “holds”. This usually causes no problem. However, to prevent the extra executions from doing harm, use a global switch to keep anything harmful in the new `\everypar` from being done twice.

2. Commands that change `\everypar` should remember that `\everypar` might be supposed to set the following switches false:

- `@nobreak`
- `@minipage`

they should do the setting if necessary.

```
1 \*2kernel)
2 \message{par,}
```

`\@setpar` Initiate a long-term change to `\par`.

```
\@par 3 \def\@setpar#1{\def\par{#1}\def\@par{#1}}
```

The default definition of `\@par` will ensure that if `\@restorepar` defines `\par` to execute `\@par` it will redefine itself to the primitive `\@@par` after one iteration.

```
4 \def\@par{\let\par\@@par\par}
5 \*2kernel)
```

`\@restorepar` Restore from a short-term change to `\par`.

```
6 \def\@restorepar{\def\par{\@par}}
```

# File i

## ltspace.dtx

### 16 Spacing

This section deals with spacing, and line- and page-breaking.

#### 16.1 User Commands

`\nopagebreak` [ $\langle i \rangle$ ] :  $\langle i \rangle = 0, \dots, 4$ .  
 Default argument = 4. Puts a penalty into the vertical list output as follows:  
 0 : penalty = 0  
 1 : penalty = `\@lowpenalty`  
 2 : penalty = `\@medpenalty`  
 3 : penalty = `\@highpenalty`  
 4 : penalty = 10000  
`\pagebreak` [ $\langle i \rangle$ ] : same as except negatives of its penalty  
`\linebreak` [ $\langle i \rangle$ ] : analog of the above  
`\nolinebreak` [ $\langle i \rangle$ ] : analog of the above  
`\samepage` : inhibits page breaking most places by setting the following penalties to 10000:  
`\interlinepenalty`  
`\postdisplaypenalty`  
`\interdisplaylinepenalty`  
`\@beginparpenalty`  
`\@endparpenalty`  
`\@itempenalty`  
`\@secpenalty`  
`\interfootnotelinepenalty`  
`\` : initially defined to be `\newline`  
`\` [ $\langle length \rangle$ ] : initially defined to be `\vspace{\langle length \rangle}\newline`  
 Note: `\`\* adds a `\adjust{\penalty 10000}`  
 OBSOLETE COMMANDS (which never made it into the manual):  
`\obeycr` : defines `\CRi` == `\`  
`\restorecr` : restores `\CRi` to its usual meaning.

#### 16.2 Chris' comments

There are several aspects of the handling of space in horizontal mode that are inconsistent or do not work well in some cases. These are largely concerned with ignoring the effect of space tokens that would otherwise typeset an inter-word space.

Negating the effect of such space tokens is achieved by two mechanisms:

- `\unskip` is used to remove the glue just added by a space that has already had its effect; it is sometimes invoked after an `\ifdim` test on `\lastskip` (see below);
- `\ignorespaces` is used to ignore space-tokens yet to come.

The test done on `\lastskip` is sometimes for equality with zero and sometimes for being positive. Recall also that the test is only on the natural length of the glue and that no glue cannot be distinguished from glue whose natural length is zero: to summarise, a pretty awful test. It is not clear why these tests are not all the same; I think that they should all be for equality. One place where `\unskip` is often used is just before a `\par` (which itself internally does an `\unskip`) and one bit of code (in `\@item`) even has two `\unskips` before a `\par`. These uses may be fossil code but if they are necessary, maybe `\@killglue` would be even safer.

Such removal of glue by `\unskip` may sometimes have the wrong result, removing not the glue from a space-token but other explicit glue; this is sometimes not what is intended.

A common way to prevent such removal is to add an `\hskip\z@` after the glue that should not be removed. This protects that glue against one `\unskip` with no test but not against more than one. It does work for ‘tested `\unskips`’. This is used by `\hspace*` but not by `\hspace`; this is inconsistent as the star is supposed to prevent removal only at the beginning of a line, not at the end, or in a tabular, etc.

If this reason for removing glue were the only consideration then a tested-`\unskip` and protection by `\hskip\z@` would suffice but would need to be consistently implemented.

However, the class of invisibles, commands and environments tries to be even cleverer: one of these tries to leave only one inter-word space whenever there is one before it and one after it; and it does this quite well.

But problems can arise when there is not a space-token on both sides of it; in particular, when an invisible appears at the beginning or end of a piece of text the method still leaves one space token whereas usually in these cases it should leave none.

Also, the current rules do not work well when more than one such command appears consecutively, separated by space-tokens; it leaves glue between every other invisible.

There is also a question about what these commands should do when they occur next to spaces that do not come from space tokens but, for example, from `\hspace`. Should they still produce ‘just one space’? If so, which one? It is good to note that the manual is sufficiently cautious about invisibles that we are not obliged to make anything work.

Another interesting side-road to explore is whether the space-tokens either side of an `\hspace{...}` should be ignored.

One alternative to the current algorithm that is often suggested is that all glue around the invisible should be consolidated into a space after it (usually without stating how much glue should be put there). The command `\nolinebreak` is implemented this way (and `\linebreak` should also be). This does not work correctly for the following common case:

```
... some text
\index{some-word}
some-word and more text.
```

This is optimal coding since it is normal to index a word that gets split across a page-break on its starting page. This would, on the other hand, fix another common (and documented) failure of the current system: when the invisible is

the last thing in a paragraph the space before it is not removed and, worse, it is also hidden from the paragraph-ending mechanism so that an ‘empty’ line can be created at the end of the paragraph.

Another deficiency (I think) of the current system is that the following is treated as having the `\index` command between the paragraphs, which is probably not what the author intended (since there is no empty line after it).

```
\index{beginnings}
Beginnings of paragraphs ...
```

I know of no algorithm that will handle satisfactorily even all the most common cases; note that it could be that the best algorithm may be different for different invisibles since, for example, the common uses and expected behaviour of `\index`, `\marginpar`, `\linebreak`, `\pagebreak` and `\vspace` are somewhat different. [For example, is `\vspace` ever used in the middle of a paragraph?]

One method that can (and is) used to make invisible commands produce no space when used at the beginning of text is to put in some glue that is nearly enough the same as no glue or glue of zero length in all respects except for the precise test for not being exactly equal to zero; examples of such glue are `\hskip 1sp` and, possibly better but more complex, `\hskip -1sp \hskip 1sp`. However, this only works when it is known that user-supplied text is about to start.

Some similar concerns apply to the handling of space and penalties in vertical mode; there is an extra hurdle here as `\unskip` does not work on the main vertical list. The complexity of the tests done by `\addvspace` have never been explained.

The implementation of space hacks etc for vertical mode is another major area that needs further attention; my earlier experiments did not produce much improvement over the current unsatisfactory situation.

One particular problem is what happens when the following very natural coding is used (part of the problem here is that this looks like an hmode problem, but it is not):

```
... end of text.

\begin{enumerate}
  \item \label{item:xxx} Item text.
\end{enumerate}
```

### 16.3 Some immediate actions

- Fix bug in `\linebreak`.
- Fix bug in `\\*`.
- Reimplement `\\`, etc, removing extra `\vadjusts` and getting better error trapping (this seems to involve a lot more tokens).
- Investigate whether `\\`, etc need to be errors in vmode; I think that they could be noops (maybe with a warning).
- Make all(?) `\unskips` include test for zero skip (rather than other tests or no test).

- Consider replacing `\hskip 1sp` by something better (here called an ‘infinitesimal’ skip).
- Look at all `\hskip\z@` (or similar) to see if they should be changed to an ‘infinitesimal’ skip.
- Resolve the inconsistency between `\hspace` and `\hspace*`.
- Remove unnecessary `\unskips`.
- Investigate and rationalise the ‘newline’ code.
- Find better algorithms for all sorts of things or, easier(?), fix T<sub>E</sub>X itself.

## 16.4 The code

```

1 (*2ekernel)
2 \message{spacing,}

\pagebreak
\nopagebreak 3 \def\pagebreak{\@testopt{\@no@pgbk-}4}
4 \def\nopagebreak{\@testopt{\@no@pgbk4}

\@no@pgbk
5 \def\@no@pgbk #1[#2]{%
6   \ifvmode
7     \penalty #1\@getpen{#2}%
8   \else
9     \@bsphack
10    \vadjust{\penalty #1\@getpen{#2}}%
11    \@esphack
12  \fi}

\linebreak
\nolinebreak 13 \def\linebreak{\@testopt{\@no@lnbk-}4}
14 \def\nolinebreak{\@testopt{\@no@lnbk4}

\@no@lnbk
15 \def\@no@lnbk #1[#2]{%
16   \ifvmode
17     \@nolnerr
18   \else
19     \@tempskipa\lastskip
20     \unskip
21     \penalty #1\@getpen{#2}%
22     \ifdim\@tempskipa>\z@
23       \hskip\@tempskipa
24       \ignorespaces
25     \fi
26   \fi}

\samepage
27 \def\samepage{\interlinepenalty\@M
28   \postdisplaypenalty\@M

```

```

29 \interdisplaylinepenalty\@M
30 \@beginparpenalty\@M
31 \@endparpenalty\@M
32 \@itempenalty\@M
33 \@secpenalty\@M
34 \interfootnotelinepenalty\@M}

```

`\` The purpose of the new code is to fix a few bugs; however, it also attempts to optimize the following, in order of priority:

1. efficient execution of plain `\`;
2. efficient execution of `\[...]`;
3. memory use;
4. name-space use.

The changes should make no difference to the typeset output. It appears to be safe to use `\reserved@e` and `\reserved@f` here (other reserved macros are somewhat disastrous).

These changes made `\newline` even less robust than it had been, so now it is explicitly robust, like `\`.

```

\@normalcr The internal definition of the ‘normal’ definition of \.
35 \DeclareRobustCommand\{\%
36   \let \reserved@e \relax
37   \let \reserved@f \relax
38   \@ifstar{\let \reserved@e \vadjust \let \reserved@f \nobreak
39             \@xnewline}%
40             \@xnewline}
41 \expandafter\let\expandafter\@normalcr
42   \csname\expandafter\@gobble\string\ \endcsname

```

```

\newline A simple form of the ‘normal’ definition of \.
43 \DeclareRobustCommand\newline{\@normalcr\relax}

```

```

\@xnewline
44 \def\@xnewline{\@ifnextchar[% ] bracket matching
45               \@newline
46               {\@gnewline\relax}}

```

```

\@newline
47 \def\@newline[#1]{\let \reserved@e \vadjust
48                  \@gnewline {\vskip #1}}

```

`\@gnewline` The `\nobreak` added to prevent null lines when `\` ends an overfull line. Change made 24 May 89 as suggested by Frank Mittelbach and Rainer Schöpf

```

49 \def\@gnewline #1{%
50   \ifvmode
51     \@nolnerr
52   \else
53     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break
54   \fi}

```

```

\@getpen
55 \def\@getpen#1{\ifcase #1 \z@ \or \@lowpenalty\or
56     \@medpenalty \or \@highpenalty
57     \else \@M \fi}

\if@nbreak Switch used to avoid page breaks caused by \label after a section heading, etc.
It should be GLOBALLY set true after the \nbreak and globally set false by
the next invocation of \everypar.
Commands that reset \everypar should globally set it false if appropriate.
58 \def\@nbreakfalse{\global\let\if@nbreak\iffalse}
59 \def\@nbreaktrue {\global\let\if@nbreak\iftrue}
60 \@nbreakfalse

\@savsk Registers used to save the space factor and last skip.
\@savsf 61 \newdimen\@savsk
62 \newcount\@savsf

\@bsphack \@bsphack and \@esphack used by macros such as \index and \begin{@float}
... \end{@float} that want to be invisible — i.e., not leave any extra space when
used in the middle of text. Such a macro should begin with \@bsphack and end
with \@esphack The macro in question should not create any text, nor change the
mode.
Before giving the current definition we give an extended definition that is
currently not used (because it doesn't work as advertised:-)
These are generalised hacks which attempt to do sensible things when 'invisible
commands' appear in vmode too.
They need to cope with space in both hmode (plus spacefactor) and vmode,
and also cope with breaks etc. In vmode this means ensuring that any following
\addvspace, etc sees the correct glue in \lastskip.
In fact, these improved versions should be used for other cases of 'whatsits,
thingies etc' which should be invisible. They are only for commands, not environ-
ments (see notes on \@Esphack).
BTW, anyone know why the standard hacks are surrounded by \ifmmode\else
rather than simply \ifhmode?
And are there any cases where saving the spacefactor is essential? I have some
extensions where it is, but it does not appear to be so in the standard uses.

\def \@bsphack{%
  \relax \ifvmode
    \@savsk \lastskip
    \ifdim \lastskip=\z@
    \else
      \vskip -\lastskip
    \fi
  \else
    \ifhmode
      \@savsk \lastskip
      \@savsf \spacefactor
    \fi
  \fi
}

```

I think that, in vmode, it is the safest to put in a `\nobreak` immediately after such things since writes, inserts etc followed by glue give valid breakpoints and, in general, it is possible to create breaks but impossible to destroy them.

```
\def \@esphack{%
  \relax \ifvmode
    \nobreak
    \ifdim \@savsk=\z@
    \else
      \vskip\@savsk
    \fi
  \else
    \ifhmode
      \spacefactor \@savsf
      \ifdim \@savsk>\z@
        \ignorespaces
      \fi
    \fi
  \fi
}
```

For the moment we are going to ignore the vertical versions until they are correct.

```
63 \def\@bsphack{%
64   \relax
65   \ifhmode
66     \@savsk\lastskip
67     \@savsf\spacefactor
68   \fi}
```

`\@esphack` Companion to `\@bsphack`.

```
69 </2ekernel>
70 <latexrelease>\IncludeInRelease{2015/01/01}%
71 <latexrelease>          {\@esphack}{hyphenation after space hack}%
72 <*/2ekernel | latexrelease>
73 \def\@esphack{%
74   \relax
75   \ifhmode
76     \spacefactor\@savsf
77     \ifdim\@savsk>\z@
78       \nobreak \hskip\z@skip
79       \ignorespaces
80     \fi
81   \fi}%
82 </2ekernel | latexrelease>
83 <latexrelease>\EndIncludeInRelease
84 <latexrelease>\IncludeInRelease{0000/00/00}%
85 <latexrelease>          {\@esphack}{hyphenation after space hack}%
86 <latexrelease>\def\@esphack{%
87 <latexrelease>  \relax
88 <latexrelease>  \ifhmode
89 <latexrelease>    \spacefactor\@savsf
90 <latexrelease>    \ifdim\@savsk>\z@
91 <latexrelease>    \ignorespaces
```

```

92 <latexrelease> \fi
93 <latexrelease> \fi}%
94 <latexrelease>\EndIncludeInRelease
95 <*2ekernel>

```

**\@Esphack** A variant of \@esphack that sets the @ignore switch to true (as \@esphack used to do previously). This is currently used only for floats and similar environments.

```

w
96 </2ekernel>
97 <latexrelease>\IncludeInRelease{2015/01/01}%
98 <latexrelease> \Eesphack{hyphenation after space hack}%
99 <*2ekernel | latexrelease>
100 \def \@Esphack{%
101 \relax
102 \ifhmode
103 \spacefactor \@savsf
104 \ifdim \@savsk > \z@
105 \nobreak \hskip \z@skip
106 \@ignoretrue
107 \ignorespaces
108 \fi
109 \fi}%
110 </2ekernel | latexrelease>
111 <latexrelease>\EndIncludeInRelease
112 <latexrelease>\IncludeInRelease{0000/00/00}%
113 <latexrelease> \Eesphack{hyphenation after space hack}%
114 <latexrelease>\def \@Esphack{%
115 <latexrelease> \relax
116 <latexrelease> \ifhmode
117 <latexrelease> \spacefactor \@savsf
118 <latexrelease> \ifdim \@savsk > \z@
119 <latexrelease> \@ignoretrue
120 <latexrelease> \ignorespaces
121 <latexrelease> \fi
122 <latexrelease> \fi}%
123 <latexrelease>\EndIncludeInRelease
124 <*2ekernel>

```

**\@vbsphack** Another variant which is useful for invisible things which should not live in vmode (this is how some people feel about marginals).

If it occurs in vmode then it enters hmode and ensures that \@savsk is nonzero so that the \ignorespaces is put in later. It is not used at present.

```

\def \@vbsphack{ %
\relax \ifvmode
\leavevmode
\@savsk 1sp
\@savsf \spacefactor
\else
\ifhmode
\@savsk \lastskip
\@savsf \spacefactor
\fi
\fi
}

```

## 16.5 Vertical spacing

L<sup>A</sup>T<sub>E</sub>X supports the plain T<sub>E</sub>X commands `\smallskip`, `\medskip` and `\bigskip`. However, it redefines them using `\vspace` instead of `\vskip`.

Extra vertical space is added by the command `\addvspace{⟨skip⟩}`, which adds a vertical skip of `⟨skip⟩` to the document. The sequence `\addvspace{⟨s1⟩} \addvspace{⟨s2⟩}` is equivalent to `\addvspace{⟨maximum of s1, s2⟩}`.

`\addvspace` should be used only in vertical mode, and gives an error if it's not. The `\addvspace` command does *not* add vertical space if `@minipage` is true. The minipage environment uses this to inhibit the addition of extra vertical space at the beginning.

Penalties are put into the vertical list with the `\addpenalty{⟨penalty⟩}` command. It works properly when `\addpenalty` and `\addvspace` commands are mixed.

The `@nobreak` switch is set true used when in vertical mode and no page break should occur. (Right now, it is used only by the section heading commands to inhibit page breaking after a heading.)

```
\addvspace{SKIP} ==
BEGIN
  if vmode
    then if @minipage
      else if \lastskip =0
        then \vskip SKIP
        else if \lastskip < SKIP
          then \vskip -\lastskip
          \vskip SKIP
          else if SKIP < 0 and \lastskip >= 0
            then \vskip -\lastskip
            \vskip \lastskip + SKIP
          fi
        fi
      fi
    else useful error message (CAR).
  fi
END
```

`\@xaddvskip` Internal macro for `\vspace` handling the case that space has previously been added.

```
125 \def\@xaddvskip{%
126   \ifdim\lastskip<\@tempskipb
127     \vskip-\lastskip
128     \vskip\@tempskipb
129   \else
130     \ifdim\@tempskipb<\z@
131       \ifdim\lastskip<\z@
132         \else
133           \advance\@tempskipb\lastskip
134           \vskip-\lastskip
135           \vskip \@tempskipb
136         \fi
137       \fi
138     \fi}
```

`\addvspace` Add vertical space taking into account space already added, as described above.

```

139 \def\addvspace#1{%
140   \ifvmode
141     \if@minipage\else
142       \ifdim \lastskip =\z@
143         \vskip #1\relax
144       \else
145         \@tempskipb#1\relax
146         \@xaddvskip
147       \fi
148     \fi
149   \else
150     \@noitemerr
151   \fi}

```

`\addpenalty`

```

152 </2ekernel>
153 <latexrelease>\IncludeInRelease{2015/01/01}%
154 <latexrelease>          {\addpenalty}{\addpenalty}%
155 <*2ekernel | latexrelease>

```

Fix provided by Donald (though the original fix was not good enough). In 2005 Plamen Tanovski discovered that this fix wasn't good enough either as the `\vskip` kept getting bigger if several `\addpenalty` commands followed each other. Donald kindly send a new fix.

```

156 \def\addpenalty#1{%
157   \ifvmode
158     \if@minipage
159     \else
160       \if@nobreak
161     \else
162       \ifdim\lastskip=\z@
163         \penalty#1\relax
164       \else
165         \@tempskipb\lastskip

```

We have to make sure the final `\vskip` seen by  $\text{\TeX}$  is the correct one, namely `\@tempskipb`. However we may have to adjust for `\prevdepth` when placing the penalty but that should not affect the skip we pass on to  $\text{\TeX}$ .

```

166     \begingroup
167       \@tempskipa\@tempskipb
168       \advance \@tempskipb
169       \ifdim\prevdepth>\maxdepth\maxdepth\else

```

If `\prevdepth` is -1000pt due to `\nointerlineskip` we better not add it!

```

170       \ifdim \prevdepth = -\@m\p@ \z@ \else \prevdepth \fi
171       \fi
172       \vskip -\@tempskipb
173       \penalty#1%
174       \ifdim\@tempskipa=\@tempskipb

```

Do nothing if the `\prevdepth` check made no adjustment.

```

175       \else

```

Combine the `\prevdepth` adjustment into a single skip.

```

176             \advance\@tempskipb -\@tempskipa
177             \vskip \@tempskipb
178         \fi
The final skip is always the specified length.
179         \vskip \@tempskipa
180     \endgroup
181 \fi
182 \fi
183 \fi
184 \else
185     \@noitemerr
186 \fi}%

187 \</2ekernel | latexrelease>
188 \<latexrelease>\EndIncludeInRelease
189 \<latexrelease>\IncludeInRelease{0000/00/00}%
190 \<latexrelease>          {\addpenalty}{\addpenalty}%
191 \<latexrelease>\def\addpenalty#1{%
192 \<latexrelease> \ifvmode
193 \<latexrelease>     \if@minipage
194 \<latexrelease>     \else
195 \<latexrelease>     \if@nobreak
196 \<latexrelease>     \else
197 \<latexrelease>     \ifdim\lastskip=\z@
198 \<latexrelease>     \penalty#1\relax
199 \<latexrelease>     \else
200 \<latexrelease>     \@tempskipb\lastskip
201 \<latexrelease>     \vskip -\lastskip
202 \<latexrelease>     \penalty#1%
203 \<latexrelease>     \vskip\@tempskipb
204 \<latexrelease>     \fi
205 \<latexrelease>     \fi
206 \<latexrelease>     \fi
207 \<latexrelease> \else
208 \<latexrelease>     \@noitemerr
209 \<latexrelease> \fi}%
210 \<latexrelease>\EndIncludeInRelease
211 \<*2ekernel>

```

\vspace  
\@vspace  
\@vspacer

The new code for these commands depends on the following facts:

- The value of prevdepth is changed only when a box or rule is created and added to a vertical list;
- The value of prevdepth is used only when a box is created and added to a vertical list;
- The value of prevdepth is always local to the building of one vertical list.

```

212 \DeclareRobustCommand\vspace{\@ifstar\@vspacer\@vspace}
213 \def\@vspace #1{%
214 \ifvmode
215     \vskip #1
216     \vskip\z@skip
217 \else

```

```

218     \@bsphack
219     \adjust{\@restorepar
220             \vskip #1
221             \vskip\z@skip
222             }%
223     \@esphack
224 \fi}

225 \def\@vspacer#1{%
226   \ifvmode
227     \dimen@\prevdepth
228     \hrule \@height\z@
229     \nobreak
230     \vskip #1
231     \vskip\z@skip
232     \prevdepth\dimen@
233   \else
234     \@bsphack
235     \adjust{\@restorepar
236             \hrule \@height\z@
237             \nobreak
238             \vskip #1
239             \vskip\z@skip}%
240     \@esphack
241 \fi}

\smallskip
\medskip 242 \def\smallskip{\vspace\smallskipamount}
\bigskip 243 \def\medskip{\vspace\medskipamount}
244 \def\bigskip{\vspace\bigskipamount}

\smallskipamount
\medskipamount 245 \newskip\smallskipamount \smallskipamount=3pt plus 1pt minus 1pt
\bigskipamount 246 \newskip\medskipamount \medskipamount =6pt plus 2pt minus 2pt
247 \newskip\bigskipamount \bigskipamount =12pt plus 4pt minus 4pt

```

## 16.6 Horizontal space (and breaks)

`\nobreakdashes` This idea is borrowed from the `amsmath` package but here we define a robust command.

This command is a low-level command designed for use only before hyphens or dashes (such as `-`, `--`, or `---`).

It could probably be better implemented: it may need its own private token register and temporary command.

Setting the hyphen in a box and then unboxing it means that the normal penalty will not be added after it—and if the penalty is not there a break will not be taken (unless an explicit penalty or glue follows, thus the final `\nobreak`).

Note that even if it is not followed by a ‘-’, it still leaves `vmode` and sets the `spacefactor`; so use it carefully!

```

248 \DeclareRobustCommand{\nobreakdashes}{%
249   \leavevmode
250   \toks@{}%
251   \def\reserved@a##1{\toks@\expandafter{the\toks@-}%

```

```

252 \futurelet\@let@token \reserved@b}%
253 \def\reserved@b {\ifx\@let@token -%
254 \expandafter\reserved@a
255 \else
256 \setbox\z@ \hbox{\the\toks@\nobreak}%
257 \unhbox\z@
258 \spacefactor\sfcode'\-
259 \fi}%
260 \futurelet\@let@token \reserved@b
261 }

```

**\nobreakspace** This is a robust command that produces a horizontal space at which, in paragraph-mode, a line-break is not possible. We then define an active ~ to expand to it since this is the documented behaviour of ~. One reason for introducing this is that some 8-bit input encodings have a slot for such a space and we do not want to use active characters as the L<sup>A</sup>T<sub>E</sub>X internal commands.

The braces in the definition of ~ are needed to ensure that a following space is preserved when reading to/from internal files.

We need to keep \@xobeysp as it is widely used; so here it is let to the non-robust command \nobreakspace .

```

262 \DeclareRobustCommand{\nobreakspace}{%
263 \leavevmode\nobreak\ }
264 \catcode '\~=13
265 \def~{\nobreakspace{}}
266 \expandafter\let\expandafter\@xobeysp\csname nobreakspace \endcsname

```

**\,** Used in paragraph mode produces a \thinspace. It has the ordinary definition in math mode. Useful for quotes inside quotes, as in ‘\, ‘Foo’, he said.’

```

267 \DeclareRobustCommand{\,}{%
268 \relax\ifmmode\mskip\thinmuskip\else\thinspace\fi
269 }

```

**\@** Placed before a ‘.’, makes it a sentence-ending period. Does the right thing for other punctuation marks as well. Does this by setting spacefactor to 1000.

```

270 </2ekernel>
271 <latexrelease>\IncludeInRelease{2015/01/01}%
272 <latexrelease> \{\@}{Space after \@}%
273 <*2ekernel | latexrelease>
274 \def\@{\spacefactor\@m{}}%
275 </2ekernel | latexrelease>
276 <latexrelease>\EndIncludeInRelease
277 <latexrelease>\IncludeInRelease{0000/00/00}%
278 <latexrelease> \{\@}{Space after \@}%
279 <latexrelease>\def\@{\spacefactor\@m}%
280 <latexrelease>\EndIncludeInRelease
281 <*2ekernel>

```

**\hspace**

```

282 \DeclareRobustCommand\hspace{\@ifstar\@hspacer\@hspace}

```

**\@hspace**

```

283 \def\@hspace#1{\hskip #1\relax}

```

```

\@hspacer extra \hskip 0pt added 1985/17/12 to guard against a following \unskip \relax
added 13 Oct 88 for usual TEX lossage replaced both changes by \hskip\z@skip
27 Nov 91
284 \def\@hspacer#1{\vrule \@width\z@\nobreak
285          \hskip #1\hskip \z@skip}

\fill
286 \newskip\fill
287 \fill = 0pt plus 1fill

\stretch
288 \def\stretch#1{\z@ \@plus #1fill\relax}

\thinspace
\negthinspace 289 \def\thinspace{\kern .16667em }
\enspace      290 \def\negthinspace{\kern-.16667em }
291 \def\enspace{\kern.5em }

\enskip
\quad         292 \def\enskip{\hskip.5em\relax}
\qquad       293 \def\quad{\hskip1em\relax}
294 \def\qquad{\hskip2em\relax}

\obeycr      The following definitions will probably get deleted or moved to compatibility mode
\restorecr   soon.

295 {\catcode'\^M=13 \gdef\obeycr{\catcode'\^M13 \def^^M{\\relax}%
296     \gobblecr}%
297 {\catcode'\^M=13 \gdef\gobblecr{\@ifnextchar
298 \gobble\ignorespaces}}
299 \gdef\restorecr{\catcode'\^M5 }}

300 </2kernel>

```

# File j

## ltlogos.dtx

### 17 Logos

Various logos are defined here.

`\TeX` The  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  logo, adjusted so that a full stop after the logo counts as ending a sentence.

```
1 \langle*2ekernel\rangle
2 \def\TeX{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX\@}
```

`\LaTeX` The  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  logo.

```
3 \DeclareRobustCommand{\LaTeX}{L\kern-.36em%
4     {\sbox\z@ T%
5       \vbox to\ht\z@{\hbox{\check@mathfonts
6                             \fontsize\sf@size\z@
7                             \math@fontsfalse\selectfont
8                             A}%
9                             \vss}%
10    }%
11    \kern-.15em%
12    \TeX}
```

`\LaTeXe` The  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}_{2\epsilon}$  logo as proposed by A-W designers.

```
13 \DeclareRobustCommand{\LaTeXe}{\mbox{\m@th
14   \if b\expandafter\@car\f@series\@nil\boldmath\fi
15   \LaTeX\kern.15em2$_{\textstyle\varepsilon}$}}
16 \langle/2ekernel\rangle
```

# File k

## ltfiles.dtx

### 18 File Handling

The following user commands are defined in this part:

<code>\document</code>	<code>(ie \begin{document})</code> Reads in the .AUX files and <code>\catcode</code> 's @ to 12.
<code>\nofiles</code>	Suppresses all file output by setting <code>\@files</code> false.
<code>\includeonly</code>	<code>{\NAME1, ... ,NAMEn}</code> Causes only parts NAME1, ... ,NAMEn to be read by their <code>\include</code> commands. Works by setting <code>partsw</code> true and setting <code>\@partlist</code> to NAME1, ... ,NAMEn.
<code>\include</code>	<code>{\NAME}</code> Does an <code>\input</code> NAME unless <code>\@partsw</code> is true and NAME is not in <code>\@partlist</code> . If <code>\@files</code> is true, then it directs .AUX output to NAME.AUX, including a checkpoint at the end.
<code>\input</code>	<code>{\NAME}</code> The same as TeX's <code>\input</code> , except it allows optional braces around the file name. In $\text{\LaTeX 2}_{\epsilon}$ , it also avoids the primitive 'missing file' error, if the file can not be found.
<code>\IfFileExists</code>	<code>{\NAME}{\&lt;then&gt;}{\&lt;else&gt;}</code> If the file exists on the system, execute <i>then</i> otherwise execute <i>else</i> .
<code>\InputIfFileExists</code>	<code>{\NAME}{\&lt;then&gt;}{\&lt;else&gt;}</code> If the file exists on the system, execute <i>then</i> and input NAME otherwise execute <i>else</i> .

```
1 (*2ekernel)
2 \message{files,}
```

VARIABLES, SWITCHES AND INTERNAL COMMANDS:

<code>\@mainaux</code>	: Output file number for main .AUX file.
<code>\@partaux</code>	: Output file number for current part's .AUX file.
<code>\@auxout</code>	: Either <code>\@mainout</code> or <code>\@partout</code> , depending on which .AUX file output goes to.
<code>\@input{foo}</code>	: If file foo exists, then <code>\input</code> 's it, otherwise types a warning message.
<code>@files</code>	: Switch – set false if no .AUX, .TOC, .IDX etc files are to be written
<code>@partsw</code>	: Set true by a <code>\includeonly</code> command.
<code>\@partlist</code>	: Set to the argument of the <code>\includeonly</code> command.
<code>\cp@FOO</code>	: The checkpoint for <code>\include</code> 'd file FOO.TEX, written by <code>\writckpt</code> at the end of file FOO.AUX

```
\includeonly{FILELIST} ==
BEGIN
```

```

\@partsw := T
\@partlist := FILELIST
END

\include{FILE} ==
BEGIN
\clearpage
if \@filesw = T
then \immediate\write\@mainaux{\string\@input{FILE.AUX}}
fi
if \@partsw = T
then \@tempswa := F
\reserved@a == FILE
for \reserved@a := \@partlist
do if eval(\reserved@a) = eval(\reserved@b)
then \@tempswa := T fi
od
fi

if \@tempswa = T
then \@auxout := \@partaux
if \@filesw = T
then \immediate\openout\@partaux{FILE.AUX}
\immediate\write\@partaux{\relax}
fi
\@input{FILE.TEX}
\clearpage
\@writeckpt{FILE}
if @filesw then \closeout \@partaux fi
\@auxout := \@mainaux
else \cp@FILE
fi
END

\@writeckpt{FILE} ==
BEGIN
if \@filesw = T
\immediate\write on file \@partaux:
\@setckpt{FILE}{
%% }
for \reserved@a := \cl@ckpt
do \immediate\write on file \@partaux:
\global\string\setcounter

{eval(\reserved@a)}{eval(\c@eval(\reserved@a))}
od
%% {
\immediate\write on file \@partaux: }
fi
END

\@setckpt{FILE}{LIST} ==

```

```

BEGIN
  G \cp@FILE := LIST
END

INITIALIZATION
  \@tempswa := T

\@inputcheck Allocate read stream for testing and output stream.
  \@unused 3 \newread\@inputcheck
            4 \newwrite\@unused

  \@mainaux
  \@partaux 5 \newwrite\@mainaux
            6 \newwrite\@partaux

  \if@filesw
  \if@partsw 7 \newif\if@filesw \@fileswtrue
            8 \newif\if@partsw \@partswfalse

\@clubpenalty This stores the current normal (non-infinite) value of \@clubpenalty; it should
               therefore be reset whenever the normal value is changed (as in the bibliography
               in the standard styles).
               9 \newcount\@clubpenalty
               10 \@clubpenalty \@clubpenalty

\document Cancel the \begingroup from \begin
           11 \def\document{\endgroup
           If some options on \documentclass haven't been used by any package we will now
           give a warning since this is most certainly a misspelling.
           12 \ifx\@unusedoptionlist\@empty\else
           13 \latex@warning@no@line{Unused global option(s):^^J%
           14 \spaces[\@unusedoptionlist]}%
           15 \fi
           16 \@colht\textheight
           17 \@colroom\textheight \vsize\textheight
           18 \@columnwidth\textwidth
           19 \@clubpenalty\clubpenalty
           20 \if@twocolumn
           21 \advance\columnwidth -\columnsep
           22 \divide\columnwidth\tw@ \hsize\columnwidth \@firstcolumntrue
           23 \fi
           24 \hsize\columnwidth \linewidth\hsize
           25 \begingroup\@floatplacement\@dblfloatplacement
           26 \makeatletter\let\@writefile\@gobbletwo

           27 \global \let \@multiplelabels \relax
           28 \input{\jobname.aux}%
           29 \endgroup
           30 \if@filesw
           31 \immediate\openout\@mainaux\jobname.aux
           32 \immediate\write\@mainaux{\relax}%
           33 \fi

```

Dateline 1991/03/26: FMi added `\process@table` to support NFSS; This will also work with old fonts if no other style defines `\process@table`. The following line forces the initialization of the math fonts.

```

34 \process@table
35 \let\glb@currsizel@empty %% Force math initialization.

36 \normalsize
37 \everypar{}%
```

So that punctuation in headings is not disturbed by verbatim or other local changes to the space factor codes, save the document default here. This will be locally reset by the output routine. For special cases a class may want to define `\normalsfcode`s directly, in case that definition will be used. (This is an old bug, problem existed in L<sup>A</sup>T<sub>E</sub>X2.0x and plain T<sub>E</sub>X.)

```

38 \ifx\normalsfcode\@empty
39 \ifnum\scode'\.=\@m
40 \let\normalsfcode\frenchspacing
41 \else
42 \let\normalsfcode\nonfrenchspacing
43 \fi
44 \fi
```

Way back in 1991 (08/26) FMi & RmS set the `\noskipsec` switch to true in the preamble and to false here. This was done to trap lists and related text in the preamble but it does not catch everything; hence Change 1.1g was introduced.

```

45 \noskipsecfalse

46 \let \@refundefined \relax
```

Just before disabling the preamble commands we execute the begin document hook which contains any code contributed by `\AtBeginDocument`. Also disable the gathering of the file list, if no `\listfiles` has been issued. `\AtBeginDocument` is redefined at this point so that and such commands that get into the hook do not chase their tail...

```

47 \let\AtBeginDocument\@firstofone
48 \@begindocumenthook
```

Most of the following assignments will be done globally in case the user adds something like `\begin{multicols}` to the document hook, i.e. starts are group in `\begin{document}`.

Since a value of exactly 0pt for `\topskip` causes `\twocolumn[]` to misbehave, we add this check, hoping that it will not cause any problems elsewhere.

```

49 \ifdim\topskip<1sp\global\topskip 1sp\relax\fi
50 \global\@maxdepth\maxdepth
51 \global\let\@begindocumenthook\@undefined
52 \ifx\@listfiles\@undefined
53 \global\let\@filelist\relax
54 \global\let\@addtofilelist\@gobble
55 \fi
```

At the very end we disable all preamble commands. This has to happen after the begin document hooks was executed so that this hook can still use such commands.

```

56 \gdef\do##1{\global\let ##1\@notprerr}%
57 \@preamblecmds
```

The next line saves tokens and also allows `\@nocument` to be used directly to trap preamble errors.

```
58 \global\let \@nocument \relax
```

The next line is a pure safety measure in case a do list is ever expanded at the wrong place. In addition it will save a few tokens to get rid of the above definition.

```
59 \global\let\do\noexpand
```

Use of `\AtBeginDocument` hook might mean that we are already in horizontal mode, so ignore the space after `\begin{document}`.

```
60 \ignorespaces
```

```
61 \@onlypreamble\document
```

`\normalsfcodes` The setting of `\@empty` is just a flag. This command may be defined in a class or package file. If it is still `\@empty` at `\begin{document}` it will be defined to be `\frenchspacing` or `\nonfrenchspacing`, depending on which of those appears to be in effect at that point.

```
62 \let\normalsfcodes\@empty
```

`\nofiles` Set `\@fileswf` which suppresses the places where L<sup>A</sup>T<sub>E</sub>X makes `\immediate` writes. The `\makeindex` and `\makeglossary` are disabled. `\protected@write` is redefined not to write to the file specified, but rather to write a blank line to the log file. This ensures that a *⟨whatsit⟩* node is still created, and so spacing is not affected by the `\nofiles` command; to ensure this more generally, the `\if@nobreak` test is needed.

```
63 \def\nofiles{%
64   \@fileswf
65   \typeout{No auxiliary output files.^^J}%
66   \long\def\protected@write##1##2##3%
67     {\write\m@ne{}\if@nobreak\ifvmode\nobreak\fi\fi}%
68   \let\makeindex\relax
69   \let\makeglossary\relax}
70 \@onlypreamble\nofiles
```

`\protected@write` This takes three arguments: an output stream, some initialization code, and some text to write. It then writes this, with appropriate handling of `\protect` and `\thepage`.

```
71 \long\def \protected@write#1#2#3{%
72   \begingroup
73   \let\thepage\relax
74   #2%
75   \let\protect\@unexpandable@protect
76   \edef\reserved@a{\write#1{#3}}%
77   \reserved@a
78   \endgroup
79   \if@nobreak\ifvmode\nobreak\fi\fi
80 }
```

```
81 \let\@auxout=\@mainaux
```

`\includeonly`

```
82 \def\includeonly#1{%
83   \@partswtrue
```

```

84 \edef\@partlist{\zap@space#1 \@empty}}
85 \@onlypreamble\includeonly

\include In the definition of \include, \def\reserved@b changed to \edef\reserved@b
to be consistent with the \edef in \includeonly. (Suggested by Rainer Schöpf
& Frank Mittelbach. Change made 20 Jul 88.)
    Changed definition of \include to allow space at end of file name — otherwise,
    typing \include{foo } would cause LATEX to overwrite foo.tex. Change made
    24 May 89, suggested by Rainer Schöpf and Frank Mittelbach
    Made \include check for being used inside an \include'd file, as this will not
    work and cause surprising results.

86 \def\include#1{\relax
87 \ifnum\@auxout=\@partaux
88 \latexerror{\string\include\space cannot be nested}\@eha
89 \else \@include#1 \fi}

\@include

90 \def\@include#1 {%
91 \clearpage
92 \if@filesw
93 \immediate\write\@mainaux{\string\@input{#1.aux}}%
94 \fi
95 \@tempswatrue
96 \if@partsw
97 \@tempswafalse
98 \edef\reserved@b{#1}%
99 \@for\reserved@a:=\@partlist\do
100 {\ifx\reserved@a\reserved@b\@tempswatrue\fi}%
101 \fi
102 \if@tempswa
103 \let\@auxout\@partaux
104 \if@filesw
105 \immediate\openout\@partaux #1.aux
106 \immediate\write\@partaux{\relax}%
107 \fi
108 \@input@{#1.tex}%
109 \clearpage
110 \@writeckpt{#1}%
111 \if@filesw
112 \immediate\closeout\@partaux
113 \fi
114 \else
    If the file is not included, reset \deadcycles, so that a long list of non-included
    files does not generate an ‘Output loop’ error.
115 \deadcycles\z@
116 \@nameuse{cp@#1}%
117 \fi
118 \let\@auxout\@mainaux}

\@writeckpt

119 \def\@writeckpt#1{%
120 \if@filesw

```

```

121 \immediate\write\@partaux{\string\@setckpt{#1}\@charlb}%
122 {\let\@elt\@wckptelt \cl@ckpt}%
123 \immediate\write\@partaux{\@charrb}%
124 \fi}

\@wckptelt
125 \def\@wckptelt#1{%
126 \immediate\write\@partaux{%
127 \string\setcounter{#1}{\the\@nameuse{c@#1}}}}

\@setckpt RmS 93/08/31: introduced \@setckpt
128 \def\@setckpt#1{\global\@namedef{cp@#1}}

\@charlb The following defines \@charlb and \@charrb to be { and }, respectively with
\@charrb \catcode 11.
129 {\catcode'\@charlb=2
130 \catcode'\@charrb=11
131 \gdef\@charlb[{
132 \gdef\@charrb[}]
133 ]% }brace matching

```

## 18.1 Safe Input Macros

```

\IfFileExists
134 \long\def \IfFileExists#1#2#3{%
135 \openin\@inputcheck#1 %
136 \ifeof\@inputcheck
137 \ifx\input@path\@undefined
138 \def\reserved@a{#3}%
139 \else
140 \def\reserved@a{\@iffileonpath{#1}{#2}{#3}}%
141 \fi
142 \else
143 \closein\@inputcheck
144 \edef\@filef@und{#1}%
145 \def\reserved@a{#2}%
146 \fi
147 \reserved@a}

\@iffileonpath If the file is not found by \openin, and \input@path is defined, look in all the
directories specified in \input@path.
148 \long\def\@iffileonpath#1{%
149 \let\reserved@a\@secondoftwo
150 \expandafter\@tfor\expandafter\reserved@b\expandafter
151 : \expandafter=\input@path\do{%
152 \openin\@inputcheck\reserved@b#1 %
153 \ifeof\@inputcheck\else
154 \edef\@filef@und{\reserved@b#1}%
155 \let\reserved@a\@firstoftwo%
156 \closein\@inputcheck
157 \@break@tfor
158 \fi}%
159 \reserved@a}

```

`\InputIfFileExists` Now define `\InputIfFileExists` to input #1 if it seems to exist. Immediately prior to the input, #2 is executed. If the file #1 does not exist, execute ‘#3’.

```

160 \long\def \InputIfFileExists#1#2{%
161   \IfFileExists{#1}%
162     {#2\@addtofilelist{#1}\@input \@filef@und}}

```

`\input` Input a file: if the argument is given in braces use safe input macros, otherwise use TeX’s primitive `\input` command (which is called `\@input` in L<sup>A</sup>T<sub>E</sub>X).

```

163 \def\input{\@ifnextchar\bgroup\iinput\@input}

```

`\iinput` Define `\iinput` (i.e., `\input`) in terms of `\InputIfFileExists`.

```

164 \def\iinput#1{%
165   \InputIfFileExists{#1}{}%
166   {\filename@parse{#1}%
167    \edef\reserved@a{\noexpand\@missingfileerror
168      {\filename@area\filename@base}%
169      {\ifx\filename@ext\relax tex\else\filename@ext\fi}}%
170    \reserved@a}}

```

`\@input` Define `\@input` in terms of `\IfFileExists`. So this is a ‘safe input’ command, but the files input are not listed by `\listfiles`.

We don’t want .aux, .toc files etc be listed by `\listfiles`. However, something like .bbl probably should be listed and thus should be implemented not by `\@input`.

```

171 \def\@input#1{%
172   \IfFileExists{#1}{\@input\@filef@und}{\typeout{No file #1.}}}

```

`\@input@` Version of `\@input` that does add the file to `\@filelist`.

```

173 \def\@input@#1{\InputIfFileExists{#1}{\typeout{No file #1.}}}

```

`\@missingfileerror` This ‘error’ command avoids TeX’s primitive missing file loop.

Missing file error. Prompt for a new filename, offering a default extension.

```

174 \gdef\@missingfileerror#1#2{%
175   \typeout{^^J! LaTeX Error: File ‘#1.#2’ not found.^^J^^J%
176   Type X to quit or <RETURN> to proceed,^^J%
177   or enter new name. (Default extension: #2)^^J}%
178   \message{Enter file name: }%
179   {\endlinechar\m@ne
180    \global\read\m@ne to\@gtempa}%
181   \ifx\@gtempa\@empty
182   \else
183     \def\reserved@a{x}\ifx\reserved@a\@gtempa\batchmode\@end\fi
184     \def\reserved@a{X}\ifx\reserved@a\@gtempa\batchmode\@end\fi
185     \filename@parse\@gtempa
186     \edef\filename@ext{%
187       \ifx\filename@ext\relax#2\else\filename@ext\fi}%
188     \edef\reserved@a{%
189       \noexpand\InputIfFileExists
190       {\filename@area\filename@base.\filename@ext}%
191       }%
192     {\noexpand\@missingfileerror
193      {\filename@area\filename@base}{\filename@ext}}}%
194   \reserved@a
195   \fi}

```

`\@obsoletefile` For compatibility with L<sup>A</sup>T<sub>E</sub>X 2.09 document styles, we distribute files called `article.sty`, `book.sty`, `report.sty`, `slides.sty` and `letter.sty`. These use the command `\@obsoletefile`, which produces a warning message.

```
196 \def\@obsoletefile#1#2{%
197   \@latex@warning@no@line{inputting ‘#1’ instead of obsolete ‘#2’}}
198 \@onlypreamble\@obsoletefile
```

## 18.2 Listing files

`\@filelist` A list of files input so far. The initial value of `\@gobble` eats the comma before the first file name.

```
199 \let\@filelist\@gobble
```

`\@addtofilelist` Add to the list of files input so far. This ‘real’ definition is only used for ‘cfg’ files during initex. An initial definition of `\@gobble` has already been set.

```
200 %\def\@addtofilelist#1{\xdef\@filelist{\@filelist,#1}}
```

`\listfiles` A preamble command to cause `\end{document}` to list files input from the main file.

```
201 \def\listfiles{%
202   \let\listfiles\relax
203   \def\@listfiles##1##2##3##4##5##6##7##8##9\@{%
204     \def\reserved@d{\}%
205     \@tfor\reserved@c:=##1##2##3##4##5##6##7##8\do{%
206       \ifx\reserved@c\reserved@d
207         \edef\filename@area{ \filename@area}%
208       \fi}}%
209   \def\@dofilelist{%
210     \typeout{^^J *File List*}%
211     \@for\@currname:=\@filelist\do{%
212       \filename@parse\@currname
213       \edef\reserved@a{%
214         \filename@base.%
215         \ifx\filename@ext\relax tex\else\filename@ext\fi}%
216       \expandafter\let\expandafter\reserved@b
217         \csname ver@\reserved@a\endcsname
218       \expandafter\expandafter\expandafter\@listfiles\expandafter
219         \filename@area\filename@base\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\@
220       \typeout{%
221         \filename@area\reserved@a
222         \ifx\reserved@b\relax\else\@spaces\reserved@b\fi}}%
223     \typeout{ *****^^J}}
```

The `\@filelist` will be de-activated if `\listfiles` does not appear in the preamble. `\begin{document}` contains code equivalent to the following:

```
\AtBeginDocument{%
  \ifx\@listfiles\@undefined
    \let\@filelist\relax
    \let\@addtofilelist\@gobble
  \fi}
```

```
224 \@onlypreamble\listfiles
```

\@dofilelist

225 \let\@dofilelist\relax

226 \endkernel

## File 1

# ltoutenc.dtx

## 19 Font encodings

This section of the kernel contains commands for declaring encoding-specific commands, such as accents. It also contains the code for some of the encoding files, including `omlenc.def`, `omsenc.def`, `t1enc.def` and `ot1enc.def` files, which define the OLM, OMS, T1 and OT1 encodings, and the `fontenc` package for selecting encodings.

The `fontenc` package has options for encodings, of which the last option is the default encoding. For example, to use the OT2, OT3 and T1 encodings, with T1 as the default, you say:

```
\usepackage[OT2,OT3,T1]{fontenc}
```

The standard kernel set-up loads font encoding files and selects an encoding as follows.

```
\input {omlenc.def}
\input {t1enc.def}
\input {ot1enc.def}
\input {omsenc.def}
\fontencoding{OT1}
```

Note that the files in the standard `inputenc` package depend on this behaviour of the kernel.

The syntax for declaring encoding-specific commands is:

```
\DeclareTextCommand{<command>}{<encoding>}
                        [<number>] [<default>] {<commands>}
```

This command is like `\newcommand`, except that it defines a command which is specific to one encoding. The resulting command is always robust, even if its definition is fragile. For example, the definition of `\l` in the OT1 encoding is:

```
\DeclareTextCommand{\l}{OT1}{\@xxxii l}
```

`\DeclareTextCommand` takes the same optional arguments as `\newcommand`.

```
\ProvideTextCommand{<command>}{<encoding>}
                        [<number>] [<default>] {<commands>}
```

This acts like `\DeclareTextCommand`, but does nothing if the command is already defined.

```
\DeclareTextSymbol{<command>}{<encoding>}{<slot>}
```

This command defines a text symbol, with a particular slot in that encoding. The commands:

```
\DeclareTextSymbol{\ss}{OT1}{25}
\DeclareTextCommand{\ss}{OT1}{\char25 }
```

have the same effect, but the `\DeclareTextSymbol` is faster.

```
\DeclareTextAccent{<command>}{<encoding>}{<slot>}
```

This command declares a text accent. The commands:

```
\DeclareTextAccent{"}{OT1}{127}
\DeclareTextCommand{"}{OT1}{\add@accent {127}}
```

have the same effect.

```
\DeclareTextComposite{<command>}{<encoding>}{<argument>}{<slot>}
```

This command declares a composite letter, for example in the T1 encoding `\'a` is slot 225, which is declared by:

```
\DeclareTextComposite{"'}{T1}{a}{225}
```

The *command* will normally have been declared with `\DeclareTextAccent`, or as a one-argument `\DeclareTextCommand`.

`\DeclareTextComposite` is the most common example of using the more general declaration `\DeclareTextCompositeCommand`, which can define a composite to be an arbitrary piece of text.

```
\DeclareTextCompositeCommand{<command>}{<encoding>}{<argument>}{<text>}
```

For example, in the OT1 encoding Å has a hand-crafted definition this is declared as follows

```
\DeclareTextCompositeCommand{\r}{OT1}{A}
{\leavevmode\setbox\z@\hbox{!}\dimen@ \ht\z@\advance\dimen@-1ex%
\rlap{\raise.67\dimen@\hbox{\char23}}A}
```

The *command* will normally have been declared with `\DeclareTextAccent`, or as a one-argument `\DeclareTextCommand`.

The commands defined using the above declarations can be used in two ways. Normally they are used by just calling the command in the appropriate encoding, for example `\ss`. However, sometimes you may wish to use a command in an encoding where it is not defined. If the command has no arguments, then you can use it in another encoding by calling `\UseTextSymbol`:

```
\UseTextSymbol{<encoding>}{<command>}
```

For example, `\UseTextSymbol{OT1}{\ss}` has the same effect as:

```
{\fontencoding{OT1}\selectfont\ss}
```

If the command has one argument then you can use it in another encoding by calling `\UseTextAccent`:

```
\UseTextAccent{<encoding>}{<command>}{<text>}
```

For example, if the current encoding is OT2 then `\UseTextAccent{OT1}{\'a}` has the same effect as:

```
{\fontencoding{OT1}\selectfont\'{\fontencoding{OT2}\selectfont a}}
```

You can also declare a default definition for a text command, which will be used if the current encoding has no appropriate definition. Such use will also set the definition for this command in the current encoding to equal this default definition; this makes subsequent uses of the command much faster.

```
\DeclareTextCommandDefault{<command>}{<definition>}
```

For example, the default definition of the command `\textonequarter` (which produces the fraction  $\frac{1}{4}$ ) could be built using math mode:

```
\DeclareTextCommandDefault{\textonequarter}{\ensuremath {\frac{1}{4}}}
```

There is a matching `\Provide` command which will not override an existing default definition:

```
\ProvideTextCommandDefault{<command>}{<definition>}
```

The most common use for these commands is to use symbols from other encodings, so there are some optimizations provided:

```
\DeclareTextSymbolDefault{<command>}{<encoding>}
\DeclareTextAccentDefault{<command>}{<encoding>}
```

are short for:

```
\DeclareTextCommandDefault{<command>}
{\UseTextSymbol{<encoding>}}{<command>}}
\DeclareTextCommandDefault[1]{<command>}
{\UseTextAccent{<encoding>}}{<command>}}{#1}}
```

For example, to make OT1 the default encoding for `\ss` and `\'` you say:

```
\DeclareTextSymbolDefault{\ss}{OT1}
\DeclareTextAccentDefault{\'}{OT1}
```

Note that you can use these commands on any zero- or one-argument commands declared with `\DeclareText*` or `\ProvideText*`, not just those defined using `\DeclareTextSymbol` or `\DeclareTextAccent`.

## 19.1 Removing encoding-specific commands

In some cases encoding definitions are given to provide some limited support since nothing better is available, for example, the definition for `\textdollar` in OT1 is a hack since \$ and £ actually share the same slot in this encoding. Thus if such a glyph becomes available in a different encoding (e.g., TS1) one would like to get rid of the flaky one and make the default definition point to the new encoding. In such a case defining

```
\DeclareTextSymbol{\textdollar}{TS1}{36}
\DeclareTextSymbolDefault{\textdollar}{TS1}
```

is not enough since if typesetting in OT1 L<sup>A</sup>T<sub>E</sub>X will still find the encoding specific-definition for OT1 and therefore ignore the new default. Therefore to ensure that in this case the TS1 version is used we have to remove the OT1 declaration:

```
\UndeclareTextCommand{\textdollar}{OT1}
```

Since the \$ sign is a proper glyph in the T1 encoding there is no point removing its definition and forcing L<sup>A</sup>T<sub>E</sub>X to pick up the TS1 version if typesetting in this encoding. However, assume you want to use the variant dollar sign, i.e., \$ for your dollars. In that case you have to get rid of the T1 declaration as well, e.g., the following would do that for you:

```
\UndeclareTextCommand{\textdollar}{OT1}
\UndeclareTextCommand{\textdollar}{T1}
\DeclareTextCommandDefault{\textdollar}
{\UseTextSymbol{TS1}\textdollaroldstyle}
```

## 19.2 The order of declarations

If an encoding-specific command is defined for more than one encoding, then it will execute fastest in the encoding in which it was defined last since its top-level definition will be set up to execute in that encoding without any overhead.

For this reason the file `fonttext.ltx` currently first loads the definitions for the T1 encoding and then those for the OT1 encoding so that typesetting in OT1 is optimized since that is (still) the default. However, when T1 is explicitly requested (via `\usepackage[T1]{fontenc}`) the top-level definitions are automatically changed to favour T1 since its declarations are reloaded in the process.

For the same reason default declarations should never come last since they are implemented as a special encoding themselves (with the name ?). Specifying them last would simply mean to make those encoding-specific commands equally inefficient in all encodings. Therefore the `textcomp` package, for example, first sets up all defaults to point to TS1 and then declares the commands in the TS1 encoding.

## 19.3 Docstrip modules

This `.dtx` file is be used to generate several related files containing font encoding definitions. The mutually exclusive docstrip options are listed here.

T1	generates <code>t1enc.def</code> for the Cork encoding.
TS1	generates <code>ts1enc.def</code> for the Text Companion encoding.
TS1sty	generates <code>textcomp.sty</code> , package that sets up use of the Text Companion encoding.
OT1	generates <code>ot1enc.def</code> for Knuth's CM encoding.
OMS	generates <code>omsenc.def</code> for Knuth's math symbol encoding.
OML	generates <code>omlenc.def</code> for Knuth's math letters encoding.
OT4	generates <code>ot4enc.def</code> for the Polish extension to the OT1 encoding, created by B. Jackowski and M. Ryćko for use with the Polish version of Computer Modern and Computer Concrete.
package	generates <code>fontenc.sty</code> for selecting encodings.
2ekernel	for the kernel commands.

## 19.4 Definitions for the kernel

### 19.4.1 Declaration commands

This section contains definitions for commands such as accents which depend on the current encoding. These commands will usually be kept in `.def` files, for

example `ot1enc.def` contains the definitions for the OT1 encoding.

```
1 (*2ekernel)
2 \message{font encodings,}

Far too many macros in one block here!
```

If you say:

```
\DeclareTextCommand
\ProvideTextCommand
\DeclareTextSymbol
  \@dec@text@cmd
\chardef@text@cmd
  \@changed@cmd
  \@changed@x
\TextSymbolUnavailable
  \@inmathwarn

\DeclareTextCommand{\foo}{T1}...

then \foo is defined to be \T1-cmd \foo \T1\foo, where \T1\foo is one control
sequence, not two! We then call \newcommand to define \T1\foo.

3 \def\DeclareTextCommand{%
4   \@dec@text@cmd\newcommand}

5 \def\ProvideTextCommand{%
6   \@dec@text@cmd\providecommand}

7 \def\@dec@text@cmd#1#2#3{%
8   \expandafter\def\expandafter#2%
9     \expandafter{%
10       \csname#3-cmd\expandafter\endcsname
11       \expandafter#2%
12       \csname#3\string#2\endcsname
13     }%
14   \let\@ifdefinable\@rc@ifdefinable
15   \expandafter#1\csname#3\string#2\endcsname}
```

This command was introduced to fix a major bug in `\@dec@text@cmd` without changing that command itself. This was thought to be necessary because it is defined in more than one package. (Perhaps the more serious bug is to put complex low-level commands like this in packages?)

The problem it solves is that whereas both `\newcommand` and `\providecommand` (used just above) both handle the resetting of `\@ifdefinable` (following its disabling in `\@dec@text@cmd`), the primitive `\chardef` neither needs the disabling, nor does the resetting.

```
16 \def\chardef@text@cmd{%
17   \let\@ifdefinable\@ifdefinable
18   \chardef
19 }
20 \def\DeclareTextSymbol#1#2#3{%
21   \@dec@text@cmd\chardef@text@cmd#1{#2}#3\relax
22 }
```

The declarations are only available before `\begin{document}`.

```
23 \onlypreamble\DeclareTextCommand
24 \onlypreamble\DeclareTextSymbol
```

The sneaky bit in all this is what `\T1-cmd \foo \T1\foo` does. There are five possibilities, depending on the current values of `\protect`, `\cf@encoding` and `\ifmmode`:

- If `\protect` is `\@typeset@protect` and `\cf@encoding` is T1, then we execute `\T1\foo`. This should be the normal behaviour, and is optimized for speed.
- If `\protect` is `\@typeset@protect`, `\cf@encoding` is (say) OT1, and `\OT1\foo` is defined, then we execute `\OT1\foo`.

- If `\protect` is `\@typeset@protect`, `\cf@encoding` is (say) `OT1`, we're in text mode, and `\OT1\foo` is undefined, then we define `\OT1\foo` to be the default value of `\foo`, and execute `\OT1\foo`.
- If `\protect` is `\@typeset@protect`, `\cf@encoding` is (say) `OT1`, we're in math mode, and `\OT1\foo` is undefined, then we execute the default value of `\foo`. (This is necessary so that things like `$X_\copyright$` work properly.)
- If `\protect` is not `\@typeset@protect` then we execute `\noexpand\foo`. For example, if we are writing to a file, then this results in `\foo` being written. If we are in a `\mark`, then `\foo` will be put in the mark—since `\foo` is robust, it will then survive all the things which may happen to it whilst it's a `\mark`.

So after all that, we will either execute the appropriate definition of `\foo` for the current encoding, or we will execute `\noexpand\foo`.

The default value of `\foo` is `\?\foo` if it is defined, and an error message otherwise.

When the encoding is changed from `T1` to `OT1`, `\T1-cmd` is defined to be `\@changed@cmd` and `\OT1-cmd` is defined to be `\@current@cmd`. This means that the test for what the current encoding is can be performed quickly.

```

25 \def\@current@cmd#1{%
26   \ifx\protect\@typeset@protect
27     \inmathwarn#1%
28   \else
29     \noexpand#1\expandafter\@gobble
30   \fi}

31 \def\@changed@cmd#1#2{%
32   \ifx\protect\@typeset@protect
33     \inmathwarn#1%
34     \expandafter\ifx\csname\cf@encoding\string#1\endcsname\relax
35       \expandafter\ifx\csname ?\string#1\endcsname\relax
36         \expandafter\def\csname ?\string#1\endcsname{%
37           \TextSymbolUnavailable#1%
38         }%
39       \fi
40       \global\expandafter\let
41         \csname\cf@encoding\string#1\expandafter\endcsname
42         \csname ?\string#1\endcsname
43     \fi
44     \csname\cf@encoding\string#1%
45       \expandafter\endcsname
46   \else
47     \noexpand#1%
48   \fi}

49 \gdef\TextSymbolUnavailable#1{%
50   \@latex@error{%
51     Command \protect#1 unavailable in encoding \cf@encoding%
52   }{\@eha}

```

The command `\inmathwarn` produces a warning message if we are currently in math mode. Note that since this command is used inside text commands, it can't

call `\relax` before the `\ifmmode`. This means that it is possible for the warning to fail to be issued at the beginning of a row of an `halign` whose template enters math mode. This is probably a bad feature, but there's not much that can be done about it, since adding a `\relax` would break ligatures and kerning between text symbols.

A more efficient solution would be to make `\@inmathwarn` and `\@inmatherr` equal to `\@empty` and `\relax` by default, and to have `\everymath` reset them to their usual definitions. This is left for future investigation (for example it may break some third party code).

```
53 \def\@inmathwarn#1{%
54   \ifmmode
55     \latex@warning{Command \protect#1 invalid in math mode}%
56   \fi}
```

`\DeclareTextCommandDefault` These define commands with encoding ?.

`\ProvideTextCommandDefault` Note that `\DeclareTextCommandDefault` can only be used in the preamble, but that the `\Provide` version is allowed in inputenc .def files, so is allowed anywhere.

```
57 \def\DeclareTextCommandDefault#1{%
58   \DeclareTextCommand#1?}

59 \def\ProvideTextCommandDefault#1{%
60   \ProvideTextCommand#1?}

61 \@onlypreamble\DeclareTextCommandDefault
62 %\@onlypreamble\ProvideTextCommandDefault
```

They require `\? -cmd` to be initialized as `\@changed@cmd`.

```
63 \expandafter\let\csname?-cmd\endcsname\@changed@cmd
```

`\DeclareTextAccent` This is just a disguise for defining a TeX `\accent` command.

```
64 \def\DeclareTextAccent#1#2#3{%
65   \DeclareTextCommand#1{#2}{\add@accent{#3}}
66 \@onlypreamble\DeclareTextAccent
```

`\add@accent` To save space this code is shared between all text accents that are set using the `\accent` primitive. The argument is pre-set in a box so that any font loading that is needed is already done within the box. This is needed because font-loading involves grouping and that would prevent the accent mechanism from working so that the accent would not be positioned over the argument. Declarations that change the font should be allowed (only low-level ones are at present) inside the argument of an accent command, but not size changes, as they involve `\setbox` operations which also inhibit the mechanism of the `\accent` primitive.

Note that the whole process is within a group. For a detailed discussion of this reimplementation and its deficiencies, see pr/3160.

```
67 \def\add@accent#1#2{\hmode@bgroup
```

Turn off the group in `\UseTextSymbol` in case this is used inside the argument of `\add@accent`.

```
68   \let\hmode@start@before@group\@firstofone
69   \setbox\@tempboxa\hbox{#2%
```

When presetting the argument in a box we record its `\spacefactor` for later use after the accent got typeset. This way something like `\‘A` gets the spacefactor of A (i.e., 999) rather than the default value of 1000.

```
70      \global\mathchardef\accent@spacefactor\spacefactor}%
71      \accent#1 #2\egroup\spacefactor\accent@spacefactor}
```

Default definition for `\accent@spacefactor` prevents a horrible death of the above macro inside an unprotected `\edef`.

```
72 \let\accent@spacefactor\relax
```

```
\hmode\bgroup
```

```
73 \def\hmode\bgroup{\leavevmode\bgroup}
```

`\DeclareTextCompositeCommand` Another amusing game to play with `\expandafter`, `\cename`, and `\string`. When you say `\DeclareTextCompositeCommand{\foo}{T1}{a}{bar}`, we look to see if the expansion of `\T1\foo` begins with `\@text@composite`, and if it doesn't, we redefine `\T1\foo` to be:

```
\@text@composite@x
\@strip@args
#1 -> \@text@composite \T1\foo #1\@empty \@text@composite {...}
```

where `...` is the previous definition of `\T1\foo`. Finally, we define `\\T1\foo-a` to expand to `bar`.

```
74 \def\DeclareTextCompositeCommand#1#2#3#4{%
75   \expandafter\let\expandafter\reserved@a\cename#2\string#1\endcename
76   \expandafter\expandafter\expandafter\ifx
77   \expandafter\@car\reserved@a\relax\relax\@nil \@text@composite \else
78     \edef\reserved@b##1{%
79       \def\expandafter\noexpand
80       \cename#2\string#1\endcename###1{%
81       \noexpand\@text@composite
82       \expandafter\noexpand\cename#2\string#1\endcename
83       ###1\noexpand\@empty\noexpand\@text@composite
84       {##1}}}%
85   \expandafter\reserved@b\expandafter{\reserved@a{##1}}%
86   \fi
87   \expandafter\def\cename\expandafter\string\cename
88   #2\endcename\string#1-\string#3\endcename{#4}}
89 \@onlypreamble\DeclareTextCompositeCommand
```

This all works because:

```
\@text@composite \T1\foo A\@empty \@text@composite {...}
```

expands to `\\T1\foo-A` if `\\T1\foo-A` has been defined, and `{...}` otherwise.

Note that `\@text@composite` grabs the first token of the argument and puts just that in the cname. This is so that `\‘{\textit{e}}` will work—it checks whether `\\T1\’-{\textit{e}}` is defined (which presumably it isn't) and so expands to `{\accent 1 \textit{e}}`.

This trick won't always work, for example `\‘{\itshape e}` will expand to (with spaces added for clarity):

```
\cename \string \T1\’ - \string {\itshape e} \@empty \endcename
```

which will die pretty horribly. Unfortunately there's not much can be done about this if we're going to use `\csname` lookups as a fast way of accessing composites.

This has an unfortunate 'misfeature' though, which is that in the T1 encoding, `\'{aa}` produces á. This is not the expected behaviour, and should perhaps be fixed if the fix doesn't affect performance too badly.

Finally, it's worth noting that the `\@empty` is used in `\@text@composite` so that accents will work even when the argument is empty. If you say `\'{}` then this looks up `\\T1\'-\@empty`, which ought to be `\relax`, and so all is well. If we didn't include the `\@empty`, then `\'{}` would expand to:

```
\csname \string \T1\' - \string \endcsname
```

so the `\endcsname` would be `\string`'ed and the whole of the rest of the document would be put inside the `\csname`. This would not be good.

```
90 \def\@text@composite#1#2#3\@text@composite{%
91   \expandafter\@text@composite@x
92     \csname\string#1-\string#2\endcsname}
```

Originally the `\@text@composite@x` macro had two arguments and if #1 was not `\relax` it was executed, otherwise #2 was executed. All this happened within the `\ifx` code so that neither #1 nor #2 could have picked up any additional arguments from the input stream. This has now been changed using the typical `\@firstoftwo / \@secondoftwo` coding. This way the final expansion will happen without any `\else` or `\fi` intervening in the case that we need to get a further token from the input stream.

```
93 \def\@text@composite@x#1{%
94   \ifx#1\relax
95     \expandafter\@secondoftwo
96   \else
97     \expandafter\@firstoftwo
98   \fi
99   #1}
```

The command `\DeclareTextComposite` uses `\DeclareTextCompositeCommand` to declare a command which expands out to a single glyph.

```
100 \catcode\z@=11\relax
101 \def\DeclareTextComposite#1#2#3#4{%
102   \def\reserved@a{\DeclareTextCompositeCommand#1{#2}{#3}}%
103   \bgroup
104     \lccode\z@#4%
105     \lowercase{%
106   \egroup
107     \reserved@a ^^@}}
108 \catcode\z@=15\relax
109 \@onlypreamble\DeclareTextComposite
```

<code>\UseTextAccent</code> <code>\UseTextSymbol</code> <code>\@use@text@encoding</code>	<p>These fragile commands access glyphs from different encodings. They use grotty low-level calls to the font selection scheme for speed, and in order to make sure that <code>\UseTextSymbol</code> doesn't do anything which you're not allowed to do between an <code>\accent</code> and its glyph.</p>
--	--

For a detailed discussion of this reimplementaion and its deficiencies, see pr/3160.

```

110 \def\UseTextAccent#1#2#3{%
111   \hmode@start@before@group
112   {%

```

Turn off the group in \UseTextSymbol in case this is used inside the arguments of \UseTextAccent.

```

113     \let\hmode@start@before@group\@firstofone
114     \let\@curr@enc\cf@encoding
115     \@use@text@encoding{#1}%
116     #2{\@use@text@encoding\@curr@enc#3}%
117   }}

```

```

118 \def\UseTextSymbol#1#2{%
119   \hmode@start@before@group
120   {%
121     \def\@wrong@font@char{\MessageBreak
122       for \noexpand\symbol'\string#2'}%
123     \@use@text@encoding{#1}%
124     #2%
125   }%
126 }

```

```

127 \def\@use@text@encoding#1{%
128   \edef\f@encoding{#1}%
129   \xdef\font@name{%
130     \csname\curr@fontshape/\f@size\endcsname}%
131   \pickup@font
132   \font@name
133   \@@enc@update}

```

`\hmode@start@before@group` The `\hmode@start@before@group` starts hmode and should be immediately followed by an explicit `{...}`. Its purpose is to ensure that hmode is started before this group is opened. Inside `\add@accent` and `\UseTextAccent` it is redefined to remove this group so that it doesn't conflict with the `\accent` primitive.

For a detailed discussion see pr/3160.

```

134 \let\hmode@start@before@group\leavevmode

```

`\DeclareTextSymbolDefault` Some syntactic sugar. Again, these should probably be optimized for speed.

```

\DeclareTextAccentDefault
135 \def\DeclareTextSymbolDefault#1#2{%
136   \DeclareTextCommandDefault#1{\UseTextSymbol{#2}#1}}
137 \def\DeclareTextAccentDefault#1#2{%
138   \DeclareTextCommandDefault#1{\UseTextAccent{#2}#1}}
139 \@onlypreamble\DeclareTextSymbolDefault
140 \@onlypreamble\DeclareTextAccentDefault

```

`\UndeclareTextCommand` This command safely removes an encoding specific declaration for a given encoding. It is helpful if one intends to use the default definition always and therefore wants to get rid of a declaration for some specific encoding.

```

141 \def\UndeclareTextCommand#1#2{%

```

If there is no declaration for the current encoding do nothing. (This makes a hash table entry but without eTeX we can't do anything about that).

```

142   \expandafter\ifx\csname#2\string#1\endcsname\relax
143   \else

```

Else: throw away that declaration.

```
144 \global\expandafter\let\csname#2\string#1\endcsname
145 \undefined
```

But this is unfortunately not enough, we have to take a look at the top-level definition of the encoding specific command which for a command `\foo` would look similar to `\T1-cmd \foo \T1\foo` (three tokens).

Of course, instead of `T1` one could see a different encoding name; which one depends the encoding for which `\foo` was declared last.

Now assume we have just removed the declaration for `\foo` in `T1` and the top-level of `\foo` expands to the above. Then we better change that pretty fast otherwise we do get an “undefined csname error” when we try to typeset `\foo` within `T1` instead of getting the default definition for `\foo`. And what is the best way to change that top-level definition? Well, the only “encoding” we know for sure will still be around is the default encoding denoted by `?`.

Thus in case the last token of the top-level expansion is now undefined we change the declaration to look like `\?-cmd \foo \?\foo` which is done by the following (readable?) code:

```
146 \expandafter\expandafter\expandafter
147 \ifx\expandafter\@thirdofthree#1\@undefined
148 \expandafter\gdef\expandafter#1\expandafter
149 {\csname ?-cmd\expandafter\endcsname\expandafter
150 #1\csname?\string#1\endcsname}%
151 \fi
152 \fi
153 }
154 \onlypreamble\UndeclareTextCommand
```

#### 19.4.2 Hyphenation

```
\patterns We redefine \patterns and \hyphenation to allow the use of commands declared
\@patterns with \DeclareText* to be used inside them.
\hyphenation
\@hyphenation 155 %\let\@patterns\patterns
156 %\let\@hyphenation\hyphenation
157 %\def\patterns{%
158 % \bgroup
159 % \let\protect\@empty
160 % \let\@typeset@protect\@empty
161 % \let\@changed@x\@changed@x@mouth
162 % \afterassignment\egroup
163 % \@patterns
164 %}
165 %\def\hyphenation{%
166 % \bgroup
167 % \let\protect\@empty
168 % \let\@typeset@protect\@empty
169 % \let\@changed@x\@changed@x@mouth
170 % \afterassignment\egroup
171 % \@hyphenation
172 %}
```

### 19.4.3 Miscellanea

`\a` The `\a` command is used to access the accent commands even when they have been redefined (for example by the `tabbing` environment). Its internal name is `\@tabacckludge`.

The `\string` within the `\csname` guards against something like `'` being active at the point of use.

```
173 \def\@tabacckludge#1{\expandafter\@changed@cmd
174                               \csname\string#1\endcsname\relax}
175 \let\a=\@tabacckludge
```

### 19.4.4 Default encodings

We define the default encodings for most commands to be either OT1, OML or OMS. These defaults are in the kernel and therefore fonts with these encodings must be available unless these defaults are redefined elsewhere. Recall that the standard kernel loads the encoding files for these encodings, and also that for the T1 encoding.

The naming conventions in the kernel are not what we would use if we were starting from scratch... Those defined by DEK (like `\ae` and `\ss`) or by the T<sub>E</sub>X Users Group Technical Working Group on multi-lingual typesetting (like `\th` and `\ng`) have short names. Those which were added to the kernel in 1993 and early 1994 are named after their Adobe glyph names (like `\guillemotleft` and `\quotedblbase`). Unfortunately, this naming scheme won't work for all glyphs, since some names (like `\space`) are already used, and some (like `\endash`) are very likely to be defined by users. So we're now using the naming scheme of `\text` followed by the Adobe name, (like `\textendash` and `\textsterling`). Except that some glyphs don't have Adobe names, so we're using the names used by fontinst for those (like `\textcompwordmark`). Sigh.

Some accents from OT1:

```
176 \DeclareTextAccentDefault{"}{OT1}
177 \DeclareTextAccentDefault{'}{OT1}
178 \DeclareTextAccentDefault{.}{OT1}
179 \DeclareTextAccentDefault{=}{OT1}
180 \DeclareTextAccentDefault{H}{OT1}
181 \DeclareTextAccentDefault{^}{OT1}
182 \DeclareTextAccentDefault{'}{OT1}
183 \DeclareTextAccentDefault{b}{OT1}
184 \DeclareTextAccentDefault{c}{OT1}
185 \DeclareTextAccentDefault{d}{OT1}
186 \DeclareTextAccentDefault{r}{OT1}
187 \DeclareTextAccentDefault{u}{OT1}
188 \DeclareTextAccentDefault{v}{OT1}
189 \DeclareTextAccentDefault{~}{OT1}
```

Some symbols from OT1:

```
190 %\DeclareTextSymbolDefault{AA}{OT1}
191 \DeclareTextSymbolDefault{AE}{OT1}
192 \DeclareTextSymbolDefault{L}{OT1}
193 \DeclareTextSymbolDefault{OE}{OT1}
194 \DeclareTextSymbolDefault{O}{OT1}
195 %\DeclareTextSymbolDefault{aa}{OT1}
```

```

196 \DeclareTextSymbolDefault{\ae}{OT1}
197 \DeclareTextSymbolDefault{\i}{OT1}
198 \DeclareTextSymbolDefault{\j}{OT1}

199 \DeclareTextSymbolDefault{\ij}{OT1}
200 \DeclareTextSymbolDefault{\IJ}{OT1}

201 \DeclareTextSymbolDefault{\l}{OT1}
202 \DeclareTextSymbolDefault{\oe}{OT1}
203 \DeclareTextSymbolDefault{\o}{OT1}
204 \DeclareTextSymbolDefault{\ss}{OT1}
205 \DeclareTextSymbolDefault{\textdollar}{OT1}
206 \DeclareTextSymbolDefault{\textemdash}{OT1}
207 \DeclareTextSymbolDefault{\textendash}{OT1}
208 \DeclareTextSymbolDefault{\textexclamdown}{OT1}
209 %\DeclareTextSymbolDefault{\texthyphenchar}{OT1}
210 %\DeclareTextSymbolDefault{\texthyphen}{OT1}
211 \DeclareTextSymbolDefault{\textquestiondown}{OT1}
212 \DeclareTextSymbolDefault{\textquotedblleft}{OT1}
213 \DeclareTextSymbolDefault{\textquotedblright}{OT1}
214 \DeclareTextSymbolDefault{\textquoteleft}{OT1}
215 \DeclareTextSymbolDefault{\textquoteright}{OT1}
216 \DeclareTextSymbolDefault{\textsterling}{OT1}

```

Some symbols from OMS:

```

217 \DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
218 \DeclareTextSymbolDefault{\textbackslash}{OMS}
219 \DeclareTextSymbolDefault{\textbar}{OMS}
220 \DeclareTextSymbolDefault{\textbardbl}{OMS}
221 \DeclareTextSymbolDefault{\textbraceleft}{OMS}
222 \DeclareTextSymbolDefault{\textbraceright}{OMS}
223 \DeclareTextSymbolDefault{\textbullet}{OMS}
224 \DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
225 \DeclareTextSymbolDefault{\textdagger}{OMS}
226 \DeclareTextSymbolDefault{\textparagraph}{OMS}
227 \DeclareTextSymbolDefault{\textperiodcentered}{OMS}
228 \DeclareTextSymbolDefault{\textsection}{OMS}
229 \DeclareTextAccentDefault{\textcircled}{OMS}

```

Some symbols from OML:

```

230 \DeclareTextSymbolDefault{\textless}{OML}
231 \DeclareTextSymbolDefault{\textgreater}{OML}
232 \DeclareTextAccentDefault{\t}{OML}

```

Some defaults we can fake.

The interface for defining `\copyright` changed, it used to use `\expandafter` to add braces at the appropriate points.

```

233 \DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
234 % \expandafter\def\expandafter
235 % \copyright\expandafter{\expandafter\copyright}

236 \DeclareTextCommandDefault{\textasciicircum}{\^{} }
237 \DeclareTextCommandDefault{\textasciitilde}{\~{} }
238 \DeclareTextCommandDefault{\textcompwordmark}{\leavevmode\kern\z@}
239 \DeclareTextCommandDefault{\textunderscore}{%
240 \leavevmode \kern.06em\vbox{\hrule\@width.3em}}

```

```

241 \DeclareTextCommandDefault{\textvisiblespace}{%
242   \mbox{\kern.06em\vrule \@height.3ex}%
243   \vbox{\hrule \@width.3em}%
244   \hbox{\vrule \@height.3ex}}

Using \fontdimen3 in the next definition is some sort of a kludge (since it
is the interword stretch) but it makes the ellipsis come out right in mono-spaced
fonts too (since there it is zero).

245 \DeclareTextCommandDefault{\textellipsis}{%
246   .\kern\fontdimen3\font
247   .\kern\fontdimen3\font
248   .\kern\fontdimen3\font}

249 %\DeclareTextCommandDefault{\textregistered}{\textcircled{\scshape r}}
250 \DeclareTextCommandDefault{\textregistered}{\textcircled{%
251   \check@mathfonts\fontsize\sf@size\z@\math@fontsfalse\selectfont R}}
252 \DeclareTextCommandDefault{\texttrademark}{\textsuperscript{TM}}
253 \DeclareTextCommandDefault{\SS}{SS}

254 \DeclareTextCommandDefault{\textordfeminine}{\textsuperscript{a}}
255 \DeclareTextCommandDefault{\textordmasculine}{\textsuperscript{o}}

```

### 19.4.5 Math material

Some commands can be used in both text and math mode:

```

256 \DeclareRobustCommand{\$}{\ifmmode\mathdollar\else\textdollar\fi}
257 \DeclareRobustCommand{\l}{\ifmmode\lbrace\else\textbraceleft\fi}
258 \DeclareRobustCommand{\r}{\ifmmode\rbrace\else\textbraceright\fi}
259 \DeclareRobustCommand{\P}{\ifmmode\mathparagraph\else\textparagraph\fi}
260 \DeclareRobustCommand{\S}{\ifmmode\mathsection\else\textsection\fi}
261 \DeclareRobustCommand{\dag}{\ifmmode{\dagger}\else\textdagger\fi}
262 \DeclareRobustCommand{\ddag}{\ifmmode{\ddagger}\else\textdaggerdbl\fi}

```

For historical reasons \copyright needs {} around the definition in maths.

```

263 \DeclareRobustCommand{\_}{%
264   \ifmmode\nfss@text{\textunderscore}\else\textunderscore\fi}
265 \DeclareRobustCommand{\copyright}{%
266   \ifmmode{\nfss@text{\textcopyright}}\else\textcopyright\fi}
267 \DeclareRobustCommand{\pounds}{%
268   \ifmmode\mathsterling\else\textsterling\fi}

269 \DeclareRobustCommand{\dots}{%
270   \ifmmode\mathellipsis\else\textellipsis\fi}

271 \let\ldots\dots

```

Default definition of comma below.

```

272 </2ekernel>
273 <latexrelease>\IncludeInRelease{2015/10/01}{\textcommabelow}{comma accent}%
274 <*2ekernel | latexrelease>
275 \DeclareTextCommandDefault\textcommabelow[1]
276   {\hmode\bgroup\ooalign{\null#1\cr\hidewidth\raise-.31ex
277   \hbox{\check@mathfonts\fontsize\ssf@size\z@
278   \math@fontsfalse\selectfont,}\hidewidth}\egroup}
279 <latexrelease>\EndIncludeInRelease
280 </2ekernel | latexrelease>

```

```

281 \latexrelease\IncludeInRelease{0000/00/00}{\textcommabelow}{comma accent}%
282 \latexrelease\let\textcommabelow\undefined
283 \latexrelease\EndIncludeInRelease

```

## 19.5 Definitions for the OT1 encoding

The definitions for the ‘T<sub>E</sub>X text’ (OT1) encoding.

Declare the encoding.

```

284 \*OT1
285 \DeclareFontEncoding{OT1}{}{}

```

Declare the accents.

```

286 \DeclareTextAccent{"}{OT1}{127}
287 \DeclareTextAccent{'}{OT1}{19}
288 \DeclareTextAccent{.}{OT1}{95}
289 \DeclareTextAccent{=}{OT1}{22}
290 \DeclareTextAccent{~}{OT1}{94}
291 \DeclareTextAccent{\'}{OT1}{18}
292 \DeclareTextAccent{\~}{OT1}{126}
293 \DeclareTextAccent{\H}{OT1}{125}
294 \DeclareTextAccent{\u}{OT1}{21}
295 \DeclareTextAccent{\v}{OT1}{20}
296 \DeclareTextAccent{\r}{OT1}{23}

```

Some accents have to be built by hand: Note that `\oalign` and `\o@lign` must be inside a group. In these definitions we no longer use the helper function `\sh@ft` from `plain.tex` since that now has two incompatible definitions.

```

297 \DeclareTextCommand{\b}{OT1}[1]
298   {\hmode\bgroup\o@lign{\relax#1\cr\hidewidth\ltx@sh@ft{-3ex}%
299     \vbox to.2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
300 \DeclareTextCommand{\c}{OT1}[1]
301   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent24 #1%
302     \else{\oalign{\unhbox\z@\cr\hidewidth\char24\hidewidth}}\fi}
303 \DeclareTextCommand{\d}{OT1}[1]
304   {\hmode\bgroup
305     \o@lign{\relax#1\cr\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}

```

Declare the text symbols.

```

306 \DeclareTextSymbol{\AE}{OT1}{29}
307 \DeclareTextSymbol{\OE}{OT1}{30}
308 \DeclareTextSymbol{\O}{OT1}{31}
309 \DeclareTextSymbol{\ae}{OT1}{26}
310 \DeclareTextSymbol{\i}{OT1}{16}
311 \DeclareTextSymbol{\j}{OT1}{17}
312 \DeclareTextSymbol{\oe}{OT1}{27}
313 \DeclareTextSymbol{\o}{OT1}{28}
314 \DeclareTextSymbol{\ss}{OT1}{25}
315 \DeclareTextSymbol{\textendash}{OT1}{124}
316 \DeclareTextSymbol{\textendash}{OT1}{123}

```

Using the ligatures helps with OT1 fonts that have `\textexclamdown` and `\textquestiondown` in unusual positions.

```

317 %\DeclareTextSymbol{\textexclamdown}{OT1}{60}
318 %\DeclareTextSymbol{\textquestiondown}{OT1}{62}
319 \DeclareTextCommand{\textexclamdown}{OT1}{!'}

```

```

320 \DeclareTextCommand{\textquestiondown}{OT1}{?'}
321 %\DeclareTextSymbol{\texthyphenchar}{OT1}{'\-}
322 %\DeclareTextSymbol{\texthyphen}{OT1}{'\-}
323 \DeclareTextSymbol{\textquotedblleft}{OT1}{'92}
324 \DeclareTextSymbol{\textquotedblright}{OT1}{'\"}
325 \DeclareTextSymbol{\textquoteleft}{OT1}{'\'}
326 \DeclareTextSymbol{\textquoteright}{OT1}{'\'}

```

Some symbols which are faked from others:

```

327 % \DeclareTextCommand{\aa}{OT1}
328 %     {\accent23a}
329 \DeclareTextCommand{\L}{OT1}
330     {\leavevmode\setbox\z@\hbox{L}\hb@xt@\wd\z@{\hss\@xxxii L}}
331 \DeclareTextCommand{\l}{OT1}
332     {\hmode\bgroup\@xxxii l\egroup}
333 % \DeclareTextCommand{\AA}{OT1}
334 %     {\leavevmode\setbox\z@\hbox{h}\dimen@ht\z@\advance\dimen@-1ex%
335 %     \rlap{\raise.67\dimen@\hbox{\char23}}A}

```

In the OT1 encoding Å has a hand-crafted definition, so we have here the first recorded explicit use of `\DeclareTextCompositeCommand`.

```

336 \DeclareTextCompositeCommand{\r}{OT1}{A}
337     {\leavevmode\setbox\z@\hbox{!}\dimen@ht\z@\advance\dimen@-1ex%
338     \rlap{\raise.67\dimen@\hbox{\char23}}A}

```

The dutch language uses the letter ‘ij’. It is available in T1 encoded fonts, but not in the OT1 encoded fonts. Therefor we fake it for the OT1 encoding.

```

339 \DeclareTextCommand{\ij}{OT1}{%
340     \nobreak\hskip\z@skip i\kern-0.02em j\nobreak\hskip\z@skip}
341 \DeclareTextCommand{\IJ}{OT1}{%
342     \nobreak\hskip\z@skip I\kern-0.02em J\nobreak\hskip\z@skip}

```

In the OT1 encoding, £ and \$ share a slot.

```

343 \DeclareTextCommand{\textdollar}{OT1}{\hmode\bgroup
344     \ifdim \fontdimen\@ne\font >\z@
345         \slshape
346     \else
347         \upshape
348     \fi
349     \char'\$\egroup}
350 \DeclareTextCommand{\textsterling}{OT1}{\hmode\bgroup
351     \ifdim \fontdimen\@ne\font >\z@
352         \itshape
353     \else
354         \fontshape{ui}\selectfont
355     \fi
356     \char'\$\egroup}

```

Here we are adding some more composite commands to the OT1 encoding. This makes the use of certain accents with `i` compatible with their use with the T1 encoding; this enables them to become true L<sup>A</sup>T<sub>E</sub>X internal representations. However, it will make these accents work a little less fast since a check will always be made for the existence of a composite.

```

357 \DeclareTextComposite{\.}{OT1}{i}{'\i}
358 \DeclareTextComposite{\.}{OT1}{i}{'\i}

```

```

359 \DeclareTextCompositeCommand{\'}{OT1}{i}{\@tabacckludge'\i}
360 \DeclareTextCompositeCommand{\'}{OT1}{i}{\@tabacckludge'\i}
361 \DeclareTextCompositeCommand{\~}{OT1}{i}{\~\i}
362 \DeclareTextCompositeCommand{\"}{OT1}{i}{\~\i}
363 \end{OT1}

```

## 19.6 Definitions for the T1 encoding

The definitions for the ‘Extended T<sub>E</sub>X text’ (T1) encoding.

Declare the encoding.

```

364 \begin{OT1}
365 \DeclareFontEncoding{T1}{}{}
366 \DeclareTextAccent{\'}{T1}{0}
367 \DeclareTextAccent{\'}{T1}{1}
368 \DeclareTextAccent{\~}{T1}{2}
369 \DeclareTextAccent{\~}{T1}{3}
370 \DeclareTextAccent{\"}{T1}{4}
371 \DeclareTextAccent{\H}{T1}{5}
372 \DeclareTextAccent{\R}{T1}{6}
373 \DeclareTextAccent{\V}{T1}{7}
374 \DeclareTextAccent{\U}{T1}{8}
375 \DeclareTextAccent{\=}{T1}{9}
376 \DeclareTextAccent{\.}{T1}{10}

```

Some accents have to be built by hand. Note that `\oalign` and `\oalign` must be inside a group. In these definitions we no longer use the helper function `\sh@ft` from `plain.tex` since that now has two incompatible definitions.

```

377 \DeclareTextCommand{\b}{T1}[1]
378   {\hmode\bgroup\oalign{\relax#1\cr\hidewidth\ltx@sh@ft{-3ex}%
379     \vbox to.2ex{\hbox{\char9}\vss}\hidewidth}\egroup}
380 \DeclareTextCommand{\c}{T1}[1]
381   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent11 #1%
382     \else{\oalign{\unhbox\z@\cr
383       \hidewidth\char11\hidewidth}}\fi}
384 \DeclareTextCommand{\d}{T1}[1]
385   {\hmode\bgroup
386     \oalign{\relax#1\cr\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}
387 \DeclareTextCommand{\k}{T1}[1]
388   {\hmode\bgroup\oalign{\null#1\cr\hidewidth\char12}\egroup}
389 \DeclareTextCommand{\textogonekcentered}{T1}[1]
390   {\hmode\bgroup\oalign{%
391     \null#1\cr\hidewidth\char12\hidewidth}\egroup}

```

Some symbols are constructed.

Slot 24 contains a small circle intended for construction of these two glyphs.

```

392 \DeclareTextCommand{\textperthousand}{T1}
393   {\%\char 24 } % space or ‘relax as delimiter?’
394 \DeclareTextCommand{\textpertenthousand}{T1}
395   {\%\char 24\char 24 } % space or ‘relax as delimiter?’

```

Declare the text symbols.

```

396 \DeclareTextSymbol{\AA}{T1}{197}

```

```

397 \DeclareTextSymbol{\AE}{T1}{198}
398 \DeclareTextSymbol{\DH}{T1}{208}
399 \DeclareTextSymbol{\DJ}{T1}{208}
400 \DeclareTextSymbol{\L}{T1}{138}
401 \DeclareTextSymbol{\NG}{T1}{141}
402 \DeclareTextSymbol{\OE}{T1}{215}
403 \DeclareTextSymbol{\O}{T1}{216}
404 \DeclareTextSymbol{\SS}{T1}{223}
405 \DeclareTextSymbol{\TH}{T1}{222}
406 %\DeclareTextSymbol{\aa}{T1}{229}
407 \DeclareTextSymbol{\ae}{T1}{230}
408 \DeclareTextSymbol{\dh}{T1}{240}
409 \DeclareTextSymbol{\dj}{T1}{158}
410 \DeclareTextSymbol{\guillemotleft}{T1}{19}
411 \DeclareTextSymbol{\guillemotright}{T1}{20}
412 \DeclareTextSymbol{\guilsinglleft}{T1}{14}
413 \DeclareTextSymbol{\guilsinglright}{T1}{15}
414 \DeclareTextSymbol{\i}{T1}{25}
415 \DeclareTextSymbol{\j}{T1}{26}
416 \DeclareTextSymbol{\ij}{T1}{188}
417 \DeclareTextSymbol{\IJ}{T1}{156}
418 \DeclareTextSymbol{\l}{T1}{170}
419 \DeclareTextSymbol{\ng}{T1}{173}
420 \DeclareTextSymbol{\oe}{T1}{247}
421 \DeclareTextSymbol{\o}{T1}{248}
422 \DeclareTextSymbol{\quotedblbase}{T1}{18}
423 \DeclareTextSymbol{\quotesinglbase}{T1}{13}
424 \DeclareTextSymbol{\ss}{T1}{255}
425 \DeclareTextSymbol{\textasciicircum}{T1}{'\^}
426 \DeclareTextSymbol{\textasciitilde}{T1}{'\~}
427 \DeclareTextSymbol{\textbackslash}{T1}{'\\}
428 \DeclareTextSymbol{\textbar}{T1}{'|}
429 \DeclareTextSymbol{\textbraceleft}{T1}{'\{ }
430 \DeclareTextSymbol{\textbraceright}{T1}{'\} }
431 \DeclareTextSymbol{\textcompwordmark}{T1}{23}
432 \DeclareTextSymbol{\textdollar}{T1}{'\$}
433 \DeclareTextSymbol{\textemdash}{T1}{22}
434 \DeclareTextSymbol{\textendash}{T1}{21}
435 \DeclareTextSymbol{\textexclamdown}{T1}{189}
436 \DeclareTextSymbol{\textgreater}{T1}{'\>}
437 %\DeclareTextSymbol{\texthyphenchar}{T1}{127}
438 %\DeclareTextSymbol{\texthyphen}{T1}{'\-}
439 \DeclareTextSymbol{\textless}{T1}{'\<}
440 \DeclareTextSymbol{\textquestiondown}{T1}{190}
441 \DeclareTextSymbol{\textquotedblleft}{T1}{16}
442 \DeclareTextSymbol{\textquotedblright}{T1}{17}
443 \DeclareTextSymbol{\textquotedbl}{T1}{'\"}
444 \DeclareTextSymbol{\textquoteleft}{T1}{'\'}
445 \DeclareTextSymbol{\textquoteright}{T1}{'\'}
446 \DeclareTextSymbol{\textsection}{T1}{159}
447 \DeclareTextSymbol{\textsterling}{T1}{191}
448 \DeclareTextSymbol{\textunderscore}{T1}{95}
449 \DeclareTextSymbol{\textvisiblespace}{T1}{32}
450 \DeclareTextSymbol{\th}{T1}{254}

```

Declare the composites.

```

451 \DeclareTextComposite{\.}{T1}{i}{\i}
452 \DeclareTextComposite{\.}{T1}{\i}{\i}
"80 = 128

453 \DeclareTextComposite{\u}{T1}{A}{128}
454 \DeclareTextComposite{\k}{T1}{A}{129}
455 \DeclareTextComposite{\'}{T1}{C}{130}
456 \DeclareTextComposite{\v}{T1}{C}{131}
457 \DeclareTextComposite{\v}{T1}{D}{132}
458 \DeclareTextComposite{\v}{T1}{E}{133}
459 \DeclareTextComposite{\k}{T1}{E}{134}
460 \DeclareTextComposite{\u}{T1}{G}{135}
"88 = 136

461 \DeclareTextComposite{\'}{T1}{L}{136}
462 \DeclareTextComposite{\v}{T1}{L}{137}
463 \DeclareTextComposite{\'}{T1}{N}{139}
464 \DeclareTextComposite{\v}{T1}{N}{140}
465 \DeclareTextComposite{\H}{T1}{O}{142}
466 \DeclareTextComposite{\'}{T1}{R}{143}
"90 = 144

467 \DeclareTextComposite{\v}{T1}{R}{144}
468 \DeclareTextComposite{\'}{T1}{S}{145}
469 \DeclareTextComposite{\v}{T1}{S}{146}
470 \DeclareTextComposite{\c}{T1}{S}{147}
471 \DeclareTextComposite{\v}{T1}{T}{148}
472 \DeclareTextComposite{\c}{T1}{T}{149}
473 \DeclareTextComposite{\H}{T1}{U}{150}
474 \DeclareTextComposite{\r}{T1}{U}{151}
"98 = 152

475 \DeclareTextComposite{\.}{T1}{Y}{152}
476 \DeclareTextComposite{\'}{T1}{Z}{153}
477 \DeclareTextComposite{\v}{T1}{Z}{154}
478 \DeclareTextComposite{\.}{T1}{Z}{155}
479 \DeclareTextComposite{\.}{T1}{I}{157}
"A0 = 160

480 \DeclareTextComposite{\u}{T1}{a}{160}
481 \DeclareTextComposite{\k}{T1}{a}{161}
482 \DeclareTextComposite{\'}{T1}{c}{162}
483 \DeclareTextComposite{\v}{T1}{c}{163}
484 \DeclareTextComposite{\v}{T1}{d}{164}
485 \DeclareTextComposite{\v}{T1}{e}{165}
486 \DeclareTextComposite{\k}{T1}{e}{166}
487 \DeclareTextComposite{\u}{T1}{g}{167}
"A8 = 168

488 \DeclareTextComposite{\'}{T1}{l}{168}
489 \DeclareTextComposite{\v}{T1}{l}{169}
490 \DeclareTextComposite{\'}{T1}{n}{171}
491 \DeclareTextComposite{\v}{T1}{n}{172}
492 \DeclareTextComposite{\H}{T1}{o}{174}
493 \DeclareTextComposite{\'}{T1}{r}{175}

```

"B0 = 176

```

494 \DeclareTextComposite{\v}{T1}{r}{176}
495 \DeclareTextComposite{\'}{T1}{s}{177}
496 \DeclareTextComposite{\v}{T1}{s}{178}
497 \DeclareTextComposite{\c}{T1}{s}{179}
498 \DeclareTextComposite{\v}{T1}{t}{180}
499 \DeclareTextComposite{\c}{T1}{t}{181}
500 \DeclareTextComposite{\H}{T1}{u}{182}
501 \DeclareTextComposite{\r}{T1}{u}{183}

```

"B8 = 184

```

502 \DeclareTextComposite{\"}{T1}{y}{184}
503 \DeclareTextComposite{\'}{T1}{z}{185}
504 \DeclareTextComposite{\v}{T1}{z}{186}
505 \DeclareTextComposite{\.}{T1}{z}{187}

```

"C0 = 192

```

506 \DeclareTextComposite{\'}{T1}{A}{192}
507 \DeclareTextComposite{\'}{T1}{A}{193}
508 \DeclareTextComposite{\^}{T1}{A}{194}
509 \DeclareTextComposite{\~}{T1}{A}{195}
510 \DeclareTextComposite{\"}{T1}{A}{196}
511 \DeclareTextComposite{\r}{T1}{A}{197}
512 \DeclareTextComposite{\c}{T1}{C}{199}

```

"C8 = 200

```

513 \DeclareTextComposite{\'}{T1}{E}{200}
514 \DeclareTextComposite{\'}{T1}{E}{201}
515 \DeclareTextComposite{\^}{T1}{E}{202}
516 \DeclareTextComposite{\"}{T1}{E}{203}
517 \DeclareTextComposite{\'}{T1}{I}{204}
518 \DeclareTextComposite{\'}{T1}{I}{205}
519 \DeclareTextComposite{\^}{T1}{I}{206}
520 \DeclareTextComposite{\"}{T1}{I}{207}

```

"D0 = 208

```

521 \DeclareTextComposite{\~}{T1}{N}{209}
522 \DeclareTextComposite{\'}{T1}{0}{210}
523 \DeclareTextComposite{\'}{T1}{0}{211}
524 \DeclareTextComposite{\^}{T1}{0}{212}
525 \DeclareTextComposite{\~}{T1}{0}{213}
526 \DeclareTextComposite{\"}{T1}{0}{214}

```

"D8 = 216

```

527 \DeclareTextComposite{\'}{T1}{U}{217}
528 \DeclareTextComposite{\'}{T1}{U}{218}
529 \DeclareTextComposite{\^}{T1}{U}{219}
530 \DeclareTextComposite{\"}{T1}{U}{220}
531 \DeclareTextComposite{\'}{T1}{Y}{221}

```

"E0 = 224

```

532 \DeclareTextComposite{\'}{T1}{a}{224}
533 \DeclareTextComposite{\'}{T1}{a}{225}
534 \DeclareTextComposite{\^}{T1}{a}{226}
535 \DeclareTextComposite{\~}{T1}{a}{227}
536 \DeclareTextComposite{\"}{T1}{a}{228}

```

```

537 \DeclareTextComposite{\r}{T1}{a}{229}
538 \DeclareTextComposite{\c}{T1}{c}{231}
"E8 = 232
539 \DeclareTextComposite{\'}{T1}{e}{232}
540 \DeclareTextComposite{\'}{T1}{e}{233}
541 \DeclareTextComposite{\^}{T1}{e}{234}
542 \DeclareTextComposite{\"}{T1}{e}{235}
543 \DeclareTextComposite{\'}{T1}{i}{236}
544 \DeclareTextComposite{\'}{T1}{i}{236}
545 \DeclareTextComposite{\'}{T1}{i}{237}
546 \DeclareTextComposite{\'}{T1}{i}{237}
547 \DeclareTextComposite{\^}{T1}{i}{238}
548 \DeclareTextComposite{\^}{T1}{i}{238}
549 \DeclareTextComposite{\"}{T1}{i}{239}
550 \DeclareTextComposite{\"}{T1}{i}{239}
"F0 = 240
551 \DeclareTextComposite{\~}{T1}{n}{241}
552 \DeclareTextComposite{\'}{T1}{o}{242}
553 \DeclareTextComposite{\'}{T1}{o}{243}
554 \DeclareTextComposite{\^}{T1}{o}{244}
555 \DeclareTextComposite{\~}{T1}{o}{245}
556 \DeclareTextComposite{\"}{T1}{o}{246}
"F8 = 248
557 \DeclareTextComposite{\'}{T1}{u}{249}
558 \DeclareTextComposite{\'}{T1}{u}{250}
559 \DeclareTextComposite{\^}{T1}{u}{251}
560 \DeclareTextComposite{\"}{T1}{u}{252}
561 \DeclareTextComposite{\'}{T1}{y}{253}

562 \DeclareTextCompositeCommand{\k}{T1}{o}{\textogonekcentered{o}}
563 \DeclareTextCompositeCommand{\k}{T1}{0}{\textogonekcentered{0}}
564 </T1>

```

## 19.7 Definitions for the OMS encoding

The definitions for the ‘TeX math symbol’ (OMS) encoding. Even though this is meant to be a math font, it includes some of the standard L<sup>A</sup>T<sub>E</sub>X text symbols.

Declare the encoding.

```

565 (*OMS)
566 \DeclareFontEncoding{OMS}{-}{}

```

Declare the symbols.

```

567 % \changes{v1.99}{2004/02/02}{Added \cs{textbigcircle}}
568 %   Note that slot 13 has in places been named |\Orb|: please root
569 %   out and destroy this impolity wherever you find it!
570 %   \begin{macrocode}
571 \DeclareTextSymbol{\textasteriskcentered}{OMS}{3}      % "03
572 \DeclareTextSymbol{\textbackslash}{OMS}{110}           % "6E
573 \DeclareTextSymbol{\textbar}{OMS}{106}                 % "6A
574 \DeclareTextSymbol{\textbardbl}{OMS}{107}              % "6B
575 \DeclareTextSymbol{\textbraceleft}{OMS}{102}           % "66

```

```

576 \DeclareTextSymbol{\textbraceright}{OMS}{103}      % "67
577 \DeclareTextSymbol{\textbullet}{OMS}{15}           % "0F
578 \DeclareTextSymbol{\textdaggerdbl}{OMS}{122}       % "7A
579 \DeclareTextSymbol{\textdagger}{OMS}{121}          % "79
580 \DeclareTextSymbol{\textparagraph}{OMS}{123}       % "7B
581 \DeclareTextSymbol{\textperiodcentered}{OMS}{1}    % "01
582 \DeclareTextSymbol{\textsection}{OMS}{120}         % "78
583 \DeclareTextSymbol{\textbigcircle}{OMS}{13}        % "0D
584 \DeclareTextCommand{\textcircled}{OMS}[1]{\hmode@bgroup
585   \ooalign{%
586     \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
587     \char 13 % "0D
588   }%
589 \egroup}
590 \end{OMS}

```

## 19.8 Definitions for the OML encoding

The definitions for the ‘ $\TeX$  math italic’ (OML) encoding. Even though this is meant to be a math font, it includes some of the standard  $\LaTeX$  text symbols.

Declare the encoding.

```

591 \begin{OML}
592 \DeclareFontEncoding{OML}{}{}

```

Declare the symbols.

```

593 \DeclareTextSymbol{\textless}{OML}{'\<}
594 \DeclareTextSymbol{\textgreater}{OML}{'\>}
595 \DeclareTextAccent{\t}{OML}{127} % "7F
596 \end{OML}

```

## 19.9 Definitions for the OT4 encoding

These definitions are for the Polish extension to the ‘ $\TeX$  text’ (OT1) encoding. This encoding was created by B. Jackowski and M. Ryćko for use with the Polish version of Computer Modern and Computer Concrete. In positions 0–127 it is identical to OT1 but it contains some additional characters in the upper half. The  $\LaTeX$  support was developed by Mariusz Olko.

The PL fonts that use it are available as follows:

Metafont sources <ftp://ftp.gust.org.pl/TeX/language/polish/pl-mf.zip>;

Font files <ftp://ftp.gust.org.pl/TeX/language/polish/pl-tfm.zip>.

Declare the encoding.

```

597 \begin{OT4}
598 \DeclareFontEncoding{OT4}{}{}
599 \DeclareFontSubstitution{OT4}{cmr}{m}{n}

```

Declare the accents.

```

600 \DeclareTextAccent{"}{OT4}{127}
601 \DeclareTextAccent{'}{OT4}{19}
602 \DeclareTextAccent{.}{OT4}{95}
603 \DeclareTextAccent{=}{OT4}{22}
604 \DeclareTextAccent{^}{OT4}{94}
605 \DeclareTextAccent{'}{OT4}{18}
606 \DeclareTextAccent{~}{OT4}{126}

```

```

607 \DeclareTextAccent{\H}{OT4}{125}
608 \DeclareTextAccent{\u}{OT4}{21}
609 \DeclareTextAccent{\v}{OT4}{20}
610 \DeclareTextAccent{\r}{OT4}{23}

```

The ogonek accent is available only under a e A & E. But we have to provide some definition for \k. Some other accents have to be built by hand as in OT1:

```

611 \DeclareTextCommand{\k}{OT4}[1]{%
612   \TextSymbolUnavailable{\k{#1}}#1}

```

In these definitions we no longer use the helper function \sh@ft from plain.tex since that now has two incompatible definitions.

```

613 \DeclareTextCommand{\b}{OT4}[1]
614   {\hmode@bgroup\o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-3ex}%
615     \vbox to.2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
616 \DeclareTextCommand{\c}{OT4}[1]
617   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent24 #1%
618     \else{\oalign{\unhbox\z@\crrc\hidewidth\char24\hidewidth}}\fi}
619 \DeclareTextCommand{\d}{OT4}[1]
620   {\hmode@bgroup
621     \o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}

```

Declare the text symbols.

```

622 \DeclareTextSymbol{\AE}{OT4}{29}
623 \DeclareTextSymbol{\OE}{OT4}{30}
624 \DeclareTextSymbol{\O}{OT4}{31}
625 \DeclareTextSymbol{\L}{OT4}{138}
626 \DeclareTextSymbol{\ae}{OT4}{26}
627 \DeclareTextSymbol{\guillemotleft}{OT4}{174}
628 \DeclareTextSymbol{\guillemotright}{OT4}{175}
629 \DeclareTextSymbol{\i}{OT4}{16}
630 \DeclareTextSymbol{\j}{OT4}{17}
631 \DeclareTextSymbol{\l}{OT4}{170}
632 \DeclareTextSymbol{\o}{OT4}{28}
633 \DeclareTextSymbol{\oe}{OT4}{27}
634 \DeclareTextSymbol{\quotedblbase}{OT4}{255}
635 \DeclareTextSymbol{\ss}{OT4}{25}
636 \DeclareTextSymbol{\textendash}{OT4}{124}
637 \DeclareTextSymbol{\textendash}{OT4}{123}
638 \DeclareTextSymbol{\textexclamdown}{OT4}{60}
639 %\DeclareTextSymbol{\textthyphenchar}{OT4}{'\-}
640 %\DeclareTextSymbol{\textthyphen}{OT4}{'\-}
641 \DeclareTextSymbol{\textquestiondown}{OT4}{62}
642 \DeclareTextSymbol{\textquotedblleft}{OT4}{92}
643 \DeclareTextSymbol{\textquotedblright}{OT4}{'\'}
644 \DeclareTextSymbol{\textquoteleft}{OT4}{'\'}
645 \DeclareTextSymbol{\textquoteright}{OT4}{'\'}

```

Definition for Å as in OT1:

```

646 \DeclareTextCompositeCommand{\r}{OT4}{A}
647   {\leavevmode\setbox\z@\hbox{!}\dimen@ht\z@\advance\dimen@-1ex%
648     \rlap{\raise.67\dimen@\hbox{\char23}}A}

```

In the OT4 encoding, £ and \$ share a slot.

```

649 \DeclareTextCommand{\textdollar}{OT4}{\hmode@bgroup
650   \ifdim \fontdimen\@ne\font >\z@

```

```

651      \slshape
652    \else
653      \upshape
654    \fi
655    \char'\$ \egroup}
656 \DeclareTextCommand{\textsterling}{OT4}{\hmode@bgroup
657   \ifdim \fontdimen\@ne\font >\z@
658     \itshape
659   \else
660     \fontshape{ui}\selectfont
661   \fi
662   \char'\$ \egroup}

```

Declare the composites.

```

663 \DeclareTextComposite{\k}{OT4}{A}{129}
664 \DeclareTextComposite{\'}{OT4}{C}{130}
665 \DeclareTextComposite{\k}{OT4}{E}{134}
666 \DeclareTextComposite{\'}{OT4}{N}{139}
667 \DeclareTextComposite{\'}{OT4}{S}{145}
668 \DeclareTextComposite{\'}{OT4}{Z}{153}
669 \DeclareTextComposite{\.}{OT4}{Z}{155}
670 \DeclareTextComposite{\k}{OT4}{a}{161}
671 \DeclareTextComposite{\'}{OT4}{c}{162}
672 \DeclareTextComposite{\k}{OT4}{e}{166}
673 \DeclareTextComposite{\'}{OT4}{n}{171}
674 \DeclareTextComposite{\'}{OT4}{s}{177}
675 \DeclareTextComposite{\'}{OT4}{z}{185}
676 \DeclareTextComposite{\.}{OT4}{z}{187}
677 \DeclareTextComposite{\'}{OT4}{o}{211}
678 \DeclareTextComposite{\'}{OT4}{o}{243}
679 </OT4>

```

## 19.10 Definitions for the TS1 encoding

```

680 <*TS1>
681 \DeclareFontEncoding{TS1}{}{}
682 \DeclareFontSubstitution{TS1}{cmr}{m}{n}

```

Some accents have to be built by hand. Note that `\ooalign` and `\o@lign` must be inside a group.

```

683 \DeclareTextCommand{\capitalcedilla}{TS1}[1]
684   {\hmode@bgroup
685    \ooalign{\null#1\cr cr\hidewidth\char11\hidewidth}\egroup}
686 \DeclareTextCommand{\capitalogonek}{TS1}[1]
687   {\hmode@bgroup
688    \ooalign{\null#1\cr cr\hidewidth\char12\hidewidth}\egroup}

```

Accents for capital letters.

These commands can be used by the end user either directly or through definitions of the type

```
\DeclareTextCompositeCommand{\'}{T1}{X}{\capitalacute X}
```

None of the latter definitions are provided by default, since they are probably rarely used.

"00 = 0

```

689 \DeclareTextAccent{\capitalgrave}{TS1}{0}
690 \DeclareTextAccent{\capitalacute}{TS1}{1}
691 \DeclareTextAccent{\capitalcircumflex}{TS1}{2}
692 \DeclareTextAccent{\capitaltilde}{TS1}{3}
693 \DeclareTextAccent{\capitaldieresis}{TS1}{4}
694 \DeclareTextAccent{\capitalhungarumlaut}{TS1}{5}
695 \DeclareTextAccent{\capitalring}{TS1}{6}
696 \DeclareTextAccent{\capitalcaron}{TS1}{7}
"08 = 8
697 \DeclareTextAccent{\capitalbreve}{TS1}{8}
698 \DeclareTextAccent{\capitalmacron}{TS1}{9}
699 \DeclareTextAccent{\capitaldotaccent}{TS1}{10}

```

Tie accents.

The tie accent was borrowed from the `cmmi` font. The `tc` fonts now provide four tie accents, the first two are done in the classical way with assymetric glyphs hanging out of their boxes; the new ties are centered in their boxes like all other accents. They need a name: please tell us if you know what to call them.

" =

```

700 \DeclareTextAccent{\t}{TS1}{26}
701 \DeclareTextAccent{\capitaltie}{TS1}{27}
702 \DeclareTextAccent{\newtie}{TS1}{28}
703 \DeclareTextAccent{\capitalnewtie}{TS1}{29}

```

Compound word marks.

The text companion fonts contain two compound word marks of different heights, one has `cap_height`, the other `asc_height`.

```

704 \DeclareTextSymbol{\textcapitalcompwordmark}{TS1}{23}
705 \DeclareTextSymbol{\textascendercompwordmark}{TS1}{31}

```

The text companion symbols.

```

706 \DeclareTextSymbol{\textquotestraightbase}{TS1}{13}
"10 = 16
707 \DeclareTextSymbol{\textquotestraightdblbase}{TS1}{18}
708 \DeclareTextSymbol{\texttwelveudash}{TS1}{21}
709 \DeclareTextSymbol{\textthreequartersemdash}{TS1}{22}
"18 = 24
710 \DeclareTextSymbol{\textleftarrow}{TS1}{24}
711 \DeclareTextSymbol{\textrightarrow}{TS1}{25}
"20 = 32
712 \DeclareTextSymbol{\textblank}{TS1}{32}
713 \DeclareTextSymbol{\textdollar}{TS1}{36}
714 \DeclareTextSymbol{\textquotesingle}{TS1}{39}
"28 = 40
715 \DeclareTextSymbol{\textasteriskcentered}{TS1}{42}

```

Note that '054 is a comma and '056 is a full stop: these make numbers using oldstyle digits easier to input.

```

716 \DeclareTextSymbol{\textdblhyphen}{TS1}{45}
717 \DeclareTextSymbol{\textfractionsolidus}{TS1}{47}

```

Oldstyle digits.

"30 = 48

```
718 \DeclareTextSymbol{\textzerooldstyle}{TS1}{48}
719 \DeclareTextSymbol{\textoneoldstyle}{TS1}{49}
720 \DeclareTextSymbol{\texttwooldstyle}{TS1}{50}
721 \DeclareTextSymbol{\textthreeoldstyle}{TS1}{51}
722 \DeclareTextSymbol{\textfouroldstyle}{TS1}{52}
723 \DeclareTextSymbol{\textfiveoldstyle}{TS1}{53}
724 \DeclareTextSymbol{\textsixoldstyle}{TS1}{54}
725 \DeclareTextSymbol{\textsevenoldstyle}{TS1}{55}
```

"38 = 56

```
726 \DeclareTextSymbol{\texteightoldstyle}{TS1}{56}
727 \DeclareTextSymbol{\textnineoldstyle}{TS1}{57}
```

More text companion symbols.

```
728 \DeclareTextSymbol{\textlangle}{TS1}{60}
729 \DeclareTextSymbol{\textminus}{TS1}{61}
730 \DeclareTextSymbol{\textrangle}{TS1}{62}
```

"48 = 72

```
731 \DeclareTextSymbol{\textmho}{TS1}{77}
```

The big circle is here to define the command `\textcircled`. Formerly it was taken from the `cmsy` font.

```
732 \DeclareTextSymbol{\textbigcircle}{TS1}{79}
733 \DeclareTextCommand{\textcircled}{TS1}[1]{\hmode@bgroup
734   \oalign{%
735     \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
736     \char 79   % '117 = "4F
737   }%
738 \egroup}
```

More text companion symbols.

"50 = 80

```
739 \DeclareTextSymbol{\textohm}{TS1}{87}
```

"58 = 88

```
740 \DeclareTextSymbol{\textlbrackdbl}{TS1}{91}
741 \DeclareTextSymbol{\textrbrackdbl}{TS1}{93}
742 \DeclareTextSymbol{\textuparrow}{TS1}{94}
743 \DeclareTextSymbol{\textdownarrow}{TS1}{95}
```

"60 = 96

```
744 \DeclareTextSymbol{\textasciigrave}{TS1}{96}
745 \DeclareTextSymbol{\textborn}{TS1}{98}
746 \DeclareTextSymbol{\textdivorced}{TS1}{99}
747 \DeclareTextSymbol{\textdied}{TS1}{100}
```

"68 = 104

```
748 \DeclareTextSymbol{\textleaf}{TS1}{108}
749 \DeclareTextSymbol{\textmarried}{TS1}{109}
750 \DeclareTextSymbol{\textmusicalnote}{TS1}{110}
```

"78 = 120

```
751 \DeclareTextSymbol{\texttildelow}{TS1}{126}
```

This glyph, `\textdblhyphenchar` is hanging, like the hyphenchar of the ec fonts.

```
752 \DeclareTextSymbol{\textdblhyphenchar}{TS1}{127}
```

"80 = 128

```
753 \DeclareTextSymbol{\textasciibreve}{TS1}{128}
```

```
754 \DeclareTextSymbol{\textasciicaron}{TS1}{129}
```

This next glyph is *not* the same as `\textquotedbl`.

```
755 \DeclareTextSymbol{\textacutedbl}{TS1}{130}
```

```
756 \DeclareTextSymbol{\textgravedbl}{TS1}{131}
```

```
757 \DeclareTextSymbol{\textdagger}{TS1}{132}
```

```
758 \DeclareTextSymbol{\textdaggerdbl}{TS1}{133}
```

```
759 \DeclareTextSymbol{\textbardbl}{TS1}{134}
```

```
760 \DeclareTextSymbol{\textperthousand}{TS1}{135}
```

"88 = 136

```
761 \DeclareTextSymbol{\textbullet}{TS1}{136}
```

```
762 \DeclareTextSymbol{\textcelsius}{TS1}{137}
```

```
763 \DeclareTextSymbol{\textdollaroldstyle}{TS1}{138}
```

```
764 \DeclareTextSymbol{\textcentoldstyle}{TS1}{139}
```

```
765 \DeclareTextSymbol{\textflorin}{TS1}{140}
```

```
766 \DeclareTextSymbol{\textcolonmonetary}{TS1}{141}
```

```
767 \DeclareTextSymbol{\textwon}{TS1}{142}
```

```
768 \DeclareTextSymbol{\textnaira}{TS1}{143}
```

"90 = 144

```
769 \DeclareTextSymbol{\textguarani}{TS1}{144}
```

```
770 \DeclareTextSymbol{\textpeso}{TS1}{145}
```

```
771 \DeclareTextSymbol{\textlira}{TS1}{146}
```

```
772 \DeclareTextSymbol{\textrecipe}{TS1}{147}
```

```
773 \DeclareTextSymbol{\textinterrobang}{TS1}{148}
```

```
774 \DeclareTextSymbol{\textinterrobangdown}{TS1}{149}
```

```
775 \DeclareTextSymbol{\textdong}{TS1}{150}
```

```
776 \DeclareTextSymbol{\texttrademark}{TS1}{151}
```

"98 = 152

```
777 \DeclareTextSymbol{\textpertenthousand}{TS1}{152}
```

```
778 \DeclareTextSymbol{\textpilcrow}{TS1}{153}
```

```
779 \DeclareTextSymbol{\textbaht}{TS1}{154}
```

```
780 \DeclareTextSymbol{\textnumero}{TS1}{155}
```

This next name may change. For the following sign we know only a german name, which is abzüglich. The meaning is something like "commercial minus". An ASCII ersatz is `./.` (dot slash dot). The temporary English name is `\textdiscount`.

```
781 \DeclareTextSymbol{\textdiscount}{TS1}{156}
```

```
782 \DeclareTextSymbol{\textestimated}{TS1}{157}
```

```
783 \DeclareTextSymbol{\textopenbullet}{TS1}{158}
```

```
784 \DeclareTextSymbol{\textservicemark}{TS1}{159}
```

"A0 = 160

```
785 \DeclareTextSymbol{\textlquill}{TS1}{160}
```

```
786 \DeclareTextSymbol{\textrquill}{TS1}{161}
```

```
787 \DeclareTextSymbol{\textcent}{TS1}{162}
```

```
788 \DeclareTextSymbol{\textsterling}{TS1}{163}
```

```
789 \DeclareTextSymbol{\textcurrency}{TS1}{164}
```

```

790 \DeclareTextSymbol{\textyen}{TS1}{165}
791 \DeclareTextSymbol{\textbrokenbar}{TS1}{166}
792 \DeclareTextSymbol{\textsection}{TS1}{167}
" A8 = 168
793 \DeclareTextSymbol{\textasciidieresis}{TS1}{168}
794 \DeclareTextSymbol{\textcopyright}{TS1}{169}
795 \DeclareTextSymbol{\textordfeminine}{TS1}{170}
796 \DeclareTextSymbol{\textcopyleft}{TS1}{171}
797 \DeclareTextSymbol{\textlnot}{TS1}{172}
    The meaning of the circled-P is "sound recording copyright".
798 \DeclareTextSymbol{\textcircledP}{TS1}{173}
799 \DeclareTextSymbol{\textregistered}{TS1}{174}
800 \DeclareTextSymbol{\textasciimacron}{TS1}{175}
" B0 = 176
801 \DeclareTextSymbol{\textdegree}{TS1}{176}
802 \DeclareTextSymbol{\textpm}{TS1}{177}
803 \DeclareTextSymbol{\texttwosuperior}{TS1}{178}
804 \DeclareTextSymbol{\textthreesuperior}{TS1}{179}
805 \DeclareTextSymbol{\textasciicute}{TS1}{180}
806 \DeclareTextSymbol{\textmu}{TS1}{181} % micro sign
807 \DeclareTextSymbol{\textparagraph}{TS1}{182}
808 \DeclareTextSymbol{\textperiodcentered}{TS1}{183}
" B8 = 184
809 \DeclareTextSymbol{\textreferencemark}{TS1}{184}
810 \DeclareTextSymbol{\textonesuperior}{TS1}{185}
811 \DeclareTextSymbol{\textordmasculine}{TS1}{186}
812 \DeclareTextSymbol{\textsurd}{TS1}{187}
813 \DeclareTextSymbol{\textonequarter}{TS1}{188}
814 \DeclareTextSymbol{\textonehalf}{TS1}{189}
815 \DeclareTextSymbol{\textthreequarters}{TS1}{190}
816 \DeclareTextSymbol{\texteuro}{TS1}{191}
" E0 = 208
817 \DeclareTextSymbol{\texttimes}{TS1}{214}
" F0 = 240
818 \DeclareTextSymbol{\textdiv}{TS1}{246}
819 </TS1>

```

## 20 Package files

This file now also contains some packages that provide access to the more specialised encodings.

### 20.1 The fontenc package

This package allows authors to specify which encodings they will use. For each encoding F00, the package looks to see if the encoding F00 has already been declared. If it has not, the file `fooenc.def` is loaded. The default encoding is set to be F00.



```

864         produce further error messages.}}%
865 \let\reserved@f\relax

In case the current encoding is one of a list of known cyrillic ones we extend
the \@uclclist:
866 \expandafter\in@\expandafter{\CurrentOption}%
867                                     {T2A,T2B,T2C,X2,LCY,OT2}%
868 \ifin@

But only if it hasn't already been extended. This might happen if there are
several calls to fontenc loading one of the above encodings. If we don't do this check
the \@uclclist gets unnecessarily big, slowing down the processing at runtime.
869 \expandafter\in@\expandafter\cyr@\expandafter
870                                     {\@uclclist}%
871 \ifin@
872 \else
873 \update@uclc@with@cyrillic
874 \fi
875 \fi
876 }

877 \ProcessOptions*

878 \fontencoding\encodingdefault\selectfont

To save some space we get rid of the macro extending the \@uclclist (might
have happened already).
879 \let\update@uclc@with@cyrillic\relax

Finally we pretend that the fontenc package wasn't read in. This allows for
using it several times, e.g., in a class file and in the preamble (at the cost of not
getting any version info). That kind of hackery shows that using a general purpose
package just for loading an encoding is not the right kind of interface for setting
up encodings — it will get replaced at some point in the future.
880 \global\expandafter\let\csname ver@fontenc.sty\endcsname\relax
881 \global\expandafter\let\csname opt@fontenc.sty\endcsname\relax
882 \global\let\@ifl@ter@@\@ifl@ter
883 \def\@ifl@ter#1#2#3#4#5{\global\let\@ifl@ter\@ifl@ter@@}
884 \end{package}

```

## 20.2 The textcomp package

This one is for the TS1 encoding which contains text symbols for use with the T1-encoded text fonts. It therefore first inputs the file `TS1enc.def` and then sets (or resets) the defaults for the symbols it contains. The result of this is that when one of these symbols is accessed and the current encoding does not provide it, the symbol will be supplied by a silent, local change to this encoding.

```
885 \let\TS1sty\relax
```

Since many PostScript fonts only implement a subset of TS1 many commands only produce black blobs of ink. To resolve the resulting problems a number of options have been introduced and some code has been developed to distinguish sub-encodings.

The sub-encodings have a numerical id and are defined as follows for TS1:

**#5** those TS1 symbols that are also in the ISO-Adobe character set; without `textcurrency`, which is often misused for the Euro. Older Type1 fonts from the non-TeX world provide only this subset.

**#4** = **#5** + `\texteuro`. Most newer fonts provide this.

**#3** = **#4** + `\textomega`. Can also be described as  $TS1 \cap (ISO-Adobe \cup MacRoman)$ . (Except for the missing "currency".)

**#2** = **#3** + `\textestimated` + `textcurrency`. Can also be described as  $TS1 \cap Adobe-Western-2$ . This may be relevant for OpenType fonts, which usually show the Adobe-Western-2 character set.

**#1** = TS1 without `\textcircled` and `\t`. These two glyphs are often not implemented and if their kernel defaults are changed commands like `\copyright` unnecessarily fail.

**#0** = full TS1

And here a summary to go in the transcript file:

```
886 \PackageInfo{textcomp}{Sub-encoding information:\MessageBreak
887   \space\space 5 = only ISO-Adobe without
888                       \string\textcurrency\MessageBreak
889   \space\space 4 = 5 + \string\texteuro\MessageBreak
890   \space\space 3 = 4 + \string\textohm\MessageBreak
891   \space\space 2 = 3 + \noexpand\textestimated+
892                       \string\textcurrency\MessageBreak
893   \space\space 1 = TS1 - \noexpand\textcircled-
894                       \string\t\MessageBreak
895   \space\space 0 = TS1 (full)\MessageBreak
896   Font families with sub-encoding setting implement\MessageBreak
897   only a restricted character set as indicated.\MessageBreak
898   Family '?' is the default used for unknown fonts.\MessageBreak
899   See the documentation for details\@gobble}
```

`\DeclareEncodingSubset` An encoding subset to which a font family belongs is declared by the command `\DeclareEncodingSubset` that takes the major encoding as the first argument (e.g., TS1), the family name as the second argument (e.g., `cmr`), and the subset encoding id as a third, (e.g., 0 for `cmr`).

The default encoding subset to use when nothing is known about the current font family is named `?`.

```
900 \def\DeclareEncodingSubset#1#2#3{%
901   \@ifundefined{#1:#2}%
902     {\PackageInfo{textcomp}{Setting #2 sub-encoding to #1/#3}}%
903     {\PackageInfo{textcomp}{Changing #2 sub-encoding to #1/#3}}%
904   \@namedef{#1:#2}{#3}}
905 \onlypreamble\DeclareEncodingSubset
```

The options for the package are the following:

**safe** for unknown font families enables only symbols that are also in the ISO-Adobe character set; without "currency", which is often misused for the Euro. Older Type1 fonts from the non-TeX world provide only this subset.

**euro** enables the “safe” symbols plus the `\texteuro` command. Most newer fonts provide this.

**full** enables all TS1 commands; useful only with fonts like EC or CM bright.

**almostfull** same as “full”, except that `\textcircled` and `\t` are *not* redefined from their defaults to avoid that commands like `\copyright` suddenly no longer work.

**force** ignore all subset encoding definitions stored in the package itself or in the configuration file and always use the default subset as specified by one of the other options (seldom useful, only dangerous).

`\iftc@forced` Switch used to implement the **force** option

```
906 \newif\iftc@forced \tc@forcedfalse
```

This is implemented by defining the default subset:

```
907 \DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
908 \DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
909 \DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{4}}
910 \DeclareOption{safe}{\DeclareEncodingSubset{TS1}{?}{5}}
```

The default is “almostfull” which means that old documents will work except that `\textcircled` and `\t` will use the kernel defaults (with the advantage that this also works if the current font (as often the case) doesn’t implement these glyphs).

The “force” option simply sets the switch to true.

```
911 \DeclareOption{force}{\tc@forcedtrue}
```

The suggestions to user is to use the “safe” option always unless that balks in which case they could switch to “almostfull” but then better check their output manually.

```
912 \def\tc@errorwarn{\PackageError}
913 \DeclareOption{warn}{\gdef\tc@errorwarn#1#2#3{\PackageWarning{#1}{#2}}}
914 \ExecuteOptions{almostfull}
915 \ProcessOptions\relax
```

`\CheckEncodingSubset` The command `\CheckEncodingSubset` will check if the current font family has the right encoding subset to typeset a certain command. It takes five arguments as follows: first argument is either `\UseTextSymbol`, `\UseTextAccent` depending on whether or not the symbol is a text symbol or a text accent.

The second argument is the encoding from which this symbol should be fetched.

The third argument is either a fake accessor command or an error message. the code in that argument (if ever executed) receives two arguments: `#2` and `#5` of `\CheckEncodingSubset`.

Argument four is the subset encoding id to test against: if this value is higher than the subset id of the current font family then we typeset the symbol, i.e., execute `#1{#2}#5` otherwise it runs `#3#5`, e.g., to produce an error message or fake the glyph somehow.

Argument five is the symbol or accent command that is being checked.

For usage examples see definitions below.

```
916 \iftc@forced
```

If the “force” option was given we always use the default for testing against.

```

917 \def\CheckEncodingSubset#1#2#3#4#5{%
918     \ifnum #4>%
919         0\csname #2:\endcsname
920         \relax
921     \expandafter\@firstoftwo
922 \else
923     \expandafter\@secondoftwo
924 \fi
925     {#1{#2}}{#3}%
926     #5%
927 }

```

In normal circumstances the test is a bit more complicated: first check if there exists a macro  $\langle arg2 \rangle : \langle current-family \rangle$  and if so use that value to test against, otherwise use the default to test against.

```

928 \else
929 \def\CheckEncodingSubset#1#2#3#4#5{%
930     \ifnum #4>%
931         \expandafter\ifx\csname #2:\f@family\endcsname\relax
932             0\csname #2:\endcsname
933         \else
934             \csname #2:\f@family\endcsname
935         \fi
936     \relax
937     \expandafter\@firstoftwo
938 \else
939     \expandafter\@secondoftwo
940 \fi
941     {#1{#2}}{#3}%
942     #5%
943 }
944 \fi

```

tc@subst

```

945 \def\tc@subst#1{%
946     \tc@errorwarn{textcomp}% % should be latex error if general
947     {Symbol \string#1 not provided by\MessageBreak
948     font family \f@family\space
949     in TS1 encoding.\MessageBreak Default family used instead}\@eha
950     \bgroup\fontfamily\textcompsubstdefault\selectfont#1\egroup
951 }

```

\textcompsubstdefault

```

952 \def\textcompsubstdefault{cmr}

```

\tc@error \tc@error is going to be used in arg #3 of \CheckEncodingSubset when a symbol is not available in a certain font family. It gets pass the encoding it normally lives in (arg one) and the name of the symbol or accent that has a problem.

```

953 % error commands take argument:
954 % #1 symbol to be used
955 \def\tc@error#1{%
956     \PackageError{textcomp}% % should be latex error if general

```

```

957 {Accent \string#1 not provided by\MessageBreak
958   font family \f@family\space
959   in TS1 encoding}\@eha
960 }

```

`\tc@fake@euro` `\tc@fake@euro` is an example of a “fake” definition to use in arg #3 of `\CheckEncodingSubset` when a symbol is not available in a certain font family. Here we produce an Euro symbol by combining a “C” with a “=”.

```

961 \def\tc@fake@euro#1{%
962   \leavevmode
963   \PackageInfo{textcomp}{Faking \noexpand#1for font family
964               \f@family\MessageBreak in TS1 encoding}%
965   \valign{##\cr
966     \vfil\hbox to 0.07em{\dimen@\f@size\p@
967                           \math@fontsfalse
968                           \fontsize{.7\dimen@}\z@\selectfont=\hss}%
969     \vfil\cr%
970     \hbox{C}\crrcr
971   }%
972 }

```

`\tc@check@symbol` These are two abbreviations that we use below to check symbols and accents in  
`\tc@check@accent` TS1. Only there to save some space, e.g., we can then write

```
\DeclareTextCommandDefault{\textcurrency}{\tc@check@symbol3\textcurrency}
```

to ensure that `\textcurrency` is only typeset if the current font has a TS1 subset id of less than 3. Otherwise `\tc@error` is called telling the user that for this font family `\textcurrency` is not available.

```

973 \def\tc@check@symbol{\CheckEncodingSubset\UseTextSymbol{TS1}\tc@subst}
974 \def\tc@check@accent{\CheckEncodingSubset\UseTextAccent{TS1}\tc@error}

```

We start with the commands that are “safe” and which can be unconditionally set up, first the accents...

```

975 \DeclareTextAccentDefault{\capitalcedilla}{TS1}
976 \DeclareTextAccentDefault{\capitalogonek}{TS1}
977 \DeclareTextAccentDefault{\capitalgrave}{TS1}
978 \DeclareTextAccentDefault{\capitalacute}{TS1}
979 \DeclareTextAccentDefault{\capitalcircumflex}{TS1}
980 \DeclareTextAccentDefault{\capitaltilde}{TS1}
981 \DeclareTextAccentDefault{\capitaldieresis}{TS1}
982 \DeclareTextAccentDefault{\capitalhungarumlaut}{TS1}
983 \DeclareTextAccentDefault{\capitalring}{TS1}
984 \DeclareTextAccentDefault{\capitalcaron}{TS1}
985 \DeclareTextAccentDefault{\capitalbreve}{TS1}
986 \DeclareTextAccentDefault{\capitalmacron}{TS1}
987 \DeclareTextAccentDefault{\capitaldotaccent}{TS1}

```

...and then the other glyphs.

```

988 \DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
989 \DeclareTextSymbolDefault{\textascendercompwordmark}{TS1}
990 \DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
991 \DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
992 \DeclareTextSymbolDefault{\texttwelveudash}{TS1}

```

```

993 \DeclareTextSymbolDefault{\textthreequartersemdash}{TS1}
994 \DeclareTextSymbolDefault{\textdollar}{TS1}
995 \DeclareTextSymbolDefault{\textquotesingle}{TS1}
996 \DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
997 \DeclareTextSymbolDefault{\textfractionsolidus}{TS1}
998 \DeclareTextSymbolDefault{\textminus}{TS1}
999 \DeclareTextSymbolDefault{\textlbrackdbl}{TS1}
1000 \DeclareTextSymbolDefault{\textrbrackdbl}{TS1}
1001 \DeclareTextSymbolDefault{\textasciigrave}{TS1}
1002 \DeclareTextSymbolDefault{\texttildelow}{TS1}
1003 \DeclareTextSymbolDefault{\textasciibreve}{TS1}
1004 \DeclareTextSymbolDefault{\textasciicaron}{TS1}
1005 \DeclareTextSymbolDefault{\textgravedbl}{TS1}
1006 \DeclareTextSymbolDefault{\textacutedbl}{TS1}
1007 \DeclareTextSymbolDefault{\textdagger}{TS1}
1008 \DeclareTextSymbolDefault{\textdaggerdbl}{TS1}
1009 \DeclareTextSymbolDefault{\textbardbl}{TS1}
1010 \DeclareTextSymbolDefault{\textperthousand}{TS1}
1011 \DeclareTextSymbolDefault{\textbullet}{TS1}
1012 \DeclareTextSymbolDefault{\textcelsius}{TS1}
1013 \DeclareTextSymbolDefault{\textflorin}{TS1}
1014 \DeclareTextSymbolDefault{\texttrademark}{TS1}
1015 \DeclareTextSymbolDefault{\textcent}{TS1}
1016 \DeclareTextSymbolDefault{\textsterling}{TS1}
1017 \DeclareTextSymbolDefault{\textyen}{TS1}
1018 \DeclareTextSymbolDefault{\textbrokenbar}{TS1}
1019 \DeclareTextSymbolDefault{\textsection}{TS1}
1020 \DeclareTextSymbolDefault{\textasciidieresis}{TS1}
1021 \DeclareTextSymbolDefault{\textcopyright}{TS1}
1022 \DeclareTextSymbolDefault{\textordfeminine}{TS1}
1023 \DeclareTextSymbolDefault{\textlnot}{TS1}
1024 \DeclareTextSymbolDefault{\textregistered}{TS1}
1025 \DeclareTextSymbolDefault{\textasciimacron}{TS1}
1026 \DeclareTextSymbolDefault{\textdegree}{TS1}
1027 \DeclareTextSymbolDefault{\textpm}{TS1}
1028 \DeclareTextSymbolDefault{\texttwosuperior}{TS1}
1029 \DeclareTextSymbolDefault{\textthreesuperior}{TS1}
1030 \DeclareTextSymbolDefault{\textasciacute}{TS1}
1031 \DeclareTextSymbolDefault{\textmu}{TS1}
1032 \DeclareTextSymbolDefault{\textparagraph}{TS1}
1033 \DeclareTextSymbolDefault{\textperiodcentered}{TS1}
1034 \DeclareTextSymbolDefault{\textonesuperior}{TS1}
1035 \DeclareTextSymbolDefault{\textordmasculine}{TS1}
1036 \DeclareTextSymbolDefault{\textonequarter}{TS1}
1037 \DeclareTextSymbolDefault{\textonehalf}{TS1}
1038 \DeclareTextSymbolDefault{\textthreequarters}{TS1}
1039 \DeclareTextSymbolDefault{\texttimes}{TS1}
1040 \DeclareTextSymbolDefault{\textdiv}{TS1}

```

The `\texteuro` is only available for subsets with id 4 or less. Otherwise we fake the glyph using `\tc@fake@euro`

```

1041 \DeclareTextCommandDefault{\texteuro}
1042 {\CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro5\texteuro}

```

The `\textohm` is only available for subsets with id 3 or less. Otherwise we

produce an error.

```
1043 \DeclareTextCommandDefault{\textohm}{\tc@check@symbol4\textohm}
```

The `\textestimated` and `\textcurrency` are only provided for fonts with subset encoding with id 2 or less.

```
1044 \DeclareTextCommandDefault{\textestimated}%
```

```
1045   {\tc@check@symbol3\textestimated}
```

```
1046 \DeclareTextCommandDefault{\textcurrency}%
```

```
1047   {\tc@check@symbol3\textcurrency}
```

Nearly all of the remaining glyphs are provided only with fonts with id 1 or 0, i.e., are essentially complete.

```
1048 \DeclareTextCommandDefault{\capitaltie}%
```

```
1049   {\tc@check@accent2\capitaltie}
```

```
1050 \DeclareTextCommandDefault{\newtie}%
```

```
1051   {\tc@check@accent2\newtie}
```

```
1052 \DeclareTextCommandDefault{\capitalnewtie}%
```

```
1053   {\tc@check@accent2\capitalnewtie}
```

```
1054 \DeclareTextCommandDefault{\textleftarrow}%
```

```
1055   {\tc@check@symbol2\textleftarrow}
```

```
1056 \DeclareTextCommandDefault{\textrightarrow}%
```

```
1057   {\tc@check@symbol2\textrightarrow}
```

```
1058 \DeclareTextCommandDefault{\textblank}%
```

```
1059   {\tc@check@symbol2\textblank}
```

```
1060 \DeclareTextCommandDefault{\textdblhyphen}%
```

```
1061   {\tc@check@symbol2\textdblhyphen}
```

```
1062 \DeclareTextCommandDefault{\textzerooldstyle}%
```

```
1063   {\tc@check@symbol2\textzerooldstyle}
```

```
1064 \DeclareTextCommandDefault{\textoneoldstyle}%
```

```
1065   {\tc@check@symbol2\textoneoldstyle}
```

```
1066 \DeclareTextCommandDefault{\texttwooldstyle}%
```

```
1067   {\tc@check@symbol2\texttwooldstyle}
```

```
1068 \DeclareTextCommandDefault{\textthreeoldstyle}%
```

```
1069   {\tc@check@symbol2\textthreeoldstyle}
```

```
1070 \DeclareTextCommandDefault{\textfouroldstyle}%
```

```
1071   {\tc@check@symbol2\textfouroldstyle}
```

```
1072 \DeclareTextCommandDefault{\textfiveoldstyle}%
```

```
1073   {\tc@check@symbol2\textfiveoldstyle}
```

```
1074 \DeclareTextCommandDefault{\textsixoldstyle}%
```

```
1075   {\tc@check@symbol2\textsixoldstyle}
```

```
1076 \DeclareTextCommandDefault{\textsevenoldstyle}%
```

```
1077   {\tc@check@symbol2\textsevenoldstyle}
```

```
1078 \DeclareTextCommandDefault{\texteightoldstyle}%
```

```
1079   {\tc@check@symbol2\texteightoldstyle}
```

```
1080 \DeclareTextCommandDefault{\textnineoldstyle}%
```

```
1081   {\tc@check@symbol2\textnineoldstyle}
```

```
1082 \DeclareTextCommandDefault{\textlangle}%
```

```
1083   {\tc@check@symbol2\textlangle}
```

```
1084 \DeclareTextCommandDefault{\textrangle}%
```

```
1085   {\tc@check@symbol2\textrangle}
```

```
1086 \DeclareTextCommandDefault{\textmho}%
```

```
1087   {\tc@check@symbol2\textmho}
```

```
1088 \DeclareTextCommandDefault{\textbigcircle}%
```

```
1089   {\tc@check@symbol2\textbigcircle}
```

```
1090 \DeclareTextCommandDefault{\textuparrow}%
```

```

1091 {\tc@check@symbol2\textuparrow}%
1092 \DeclareTextCommandDefault{\textdownarrow}%
1093 {\tc@check@symbol2\textdownarrow}%
1094 \DeclareTextCommandDefault{\textborn}%
1095 {\tc@check@symbol2\textborn}%
1096 \DeclareTextCommandDefault{\textdivorced}%
1097 {\tc@check@symbol2\textdivorced}%
1098 \DeclareTextCommandDefault{\textdied}%
1099 {\tc@check@symbol2\textdied}%
1100 \DeclareTextCommandDefault{\textleaf}%
1101 {\tc@check@symbol2\textleaf}%
1102 \DeclareTextCommandDefault{\textmarried}%
1103 {\tc@check@symbol2\textmarried}%
1104 \DeclareTextCommandDefault{\textmusicalnote}%
1105 {\tc@check@symbol2\textmusicalnote}%
1106 \DeclareTextCommandDefault{\textdblhyphenchar}%
1107 {\tc@check@symbol2\textdblhyphenchar}%
1108 \DeclareTextCommandDefault{\textdollaroldstyle}%
1109 {\tc@check@symbol2\textdollaroldstyle}%
1110 \DeclareTextCommandDefault{\textcentoldstyle}%
1111 {\tc@check@symbol2\textcentoldstyle}%
1112 \DeclareTextCommandDefault{\textcolonmonetary}%
1113 {\tc@check@symbol2\textcolonmonetary}%
1114 \DeclareTextCommandDefault{\textwon}%
1115 {\tc@check@symbol2\textwon}%
1116 \DeclareTextCommandDefault{\textnaira}%
1117 {\tc@check@symbol2\textnaira}%
1118 \DeclareTextCommandDefault{\textguarani}%
1119 {\tc@check@symbol2\textguarani}%
1120 \DeclareTextCommandDefault{\textpeso}%
1121 {\tc@check@symbol2\textpeso}%
1122 \DeclareTextCommandDefault{\textlira}%
1123 {\tc@check@symbol2\textlira}%
1124 \DeclareTextCommandDefault{\textrecipe}%
1125 {\tc@check@symbol2\textrecipe}%
1126 \DeclareTextCommandDefault{\textinterrobang}%
1127 {\tc@check@symbol2\textinterrobang}%
1128 \DeclareTextCommandDefault{\textinterrobangdown}%
1129 {\tc@check@symbol2\textinterrobangdown}%
1130 \DeclareTextCommandDefault{\textdong}%
1131 {\tc@check@symbol2\textdong}%
1132 \DeclareTextCommandDefault{\textpertenthousand}%
1133 {\tc@check@symbol2\textpertenthousand}%
1134 \DeclareTextCommandDefault{\textpilcrow}%
1135 {\tc@check@symbol2\textpilcrow}%
1136 \DeclareTextCommandDefault{\textbaht}%
1137 {\tc@check@symbol2\textbaht}%
1138 \DeclareTextCommandDefault{\textnumero}%
1139 {\tc@check@symbol2\textnumero}%
1140 \DeclareTextCommandDefault{\textdiscount}%
1141 {\tc@check@symbol2\textdiscount}%
1142 \DeclareTextCommandDefault{\textopenbullet}%
1143 {\tc@check@symbol2\textopenbullet}%
1144 \DeclareTextCommandDefault{\textservicemark}%

```

```

1145 {\tc@check@symbol2\textservicemark}
1146 \DeclareTextCommandDefault{\textlquill}%
1147 {\tc@check@symbol2\textlquill}
1148 \DeclareTextCommandDefault{\textrquill}%
1149 {\tc@check@symbol2\textrquill}
1150 \DeclareTextCommandDefault{\textcopyleft}%
1151 {\tc@check@symbol2\textcopyleft}
1152 \DeclareTextCommandDefault{\textcircledP}%
1153 {\tc@check@symbol2\textcircledP}
1154 \DeclareTextCommandDefault{\textreferencemark}%
1155 {\tc@check@symbol2\textreferencemark}
1156 \DeclareTextCommandDefault{\textsurd}%
1157 {\tc@check@symbol2\textsurd}

```

The `\textcircled` and `\t` are handled specially, unless the current font has a subset id of 0 (i.e. full TS1) we pick the symbols up from the the math font encodings, i.e., the third argument to `\CheckEncodingSubset` uses `\UseTextAccent` to get them from there.

```

1158 \DeclareTextCommandDefault{\textcircled}
1159 {\CheckEncodingSubset\UseTextAccent{TS1}%
1160 {\UseTextAccent{OMS}}1\textcircled}
1161 \DeclareTextCommandDefault{\t}
1162 {\CheckEncodingSubset\UseTextAccent{TS1}%
1163 {\UseTextAccent{OML}}1\t}

```

Finally input the encoding-specific definitions for TS1 thus making the top-level definitions optimised for this encoding (and not for the default encoding, see section 19.2).

```

1164 \input{ts1enc.def}

```

Now having the new glyphs available we also want to make sure that they are used. For most cases this will automatically happen but for some glyphs there are inferior definitions already known to L<sup>A</sup>T<sub>E</sub>X which will prevent the usage of the TS1 versions (see section 19.1 above). So we better get rid of them:

```

1165 \UndeclareTextCommand{\textsterling}{OT1}
1166 \UndeclareTextCommand{\textdollar}{OT1}

```

Similar declarations should probably be made for other encodings like OT4 if they are in use.

```

1167 %\UndeclareTextCommand{\textsterling}{OT4}
1168 %\UndeclareTextCommand{\textdollar}{OT4}

```

From the T1 encoding there are two candidates for removal: `%0` and `%00` since these are both constructed from `%` followed by a tiny ‘o’ rather than being a single glyph. The problem with this approach is that in PostScript fonts this small zero is usually not available resulting in `%■` rather than `%0` while the real glyph (at least for `\textperthousand`) is available in the PostScript version of TS1. So for the moment we compromise by removing the T1 declaration for `\textperthousand` but keeping the one for `\textpertenthousand`. This will have the effect that with Computer Modern fonts everything will come out (although `%0` and `%00` are not taken from the same physical font) and with PostScript fonts `%0` will come out correctly while `%00` will most likely look like `%■` — which is probably an improvement over just getting a single ‘■’ to indicate a completely missing glyph, which would happen if we also ‘undeclared’ `\textpertenthousand`.

```

1169 \UndeclareTextCommand{\textperthousand}{T1}
1170 %\UndeclareTextCommand{\textpertenthousand}{T1}

```

### 20.2.1 Supporting oldstyle digits

```

1171 \DeclareRobustCommand\oldstylenums[1]{%
1172   \begingroup
1173   \ifmmode
1174     \mathgroup\symletters #1%
1175   \else
1176     \CheckEncodingSubset\@use@text@encoding{TS1}%
1177     {\PackageWarning{textcomp}%
1178       {Oldstyle digits unavailable for
1179        family \f@family.\MessageBreak
1180        Lining digits used instead}}%
1181     \tw@{#1}%
1182   \fi
1183 \endgroup
1184 }

```

### 20.2.2 Subset encoding defaults

For many font families commonly used in the T<sub>E</sub>X world we provide the subset encoding data here. Users can add additional font families in the file `textcomp.cfg` if they own other fonts.

However, if the option “forced” was given then all subset encoding specifications are ignored, so there is no point in setting any of them up:

```

1185 \iftc@forced \else

```

Computer modern based fonts (e.g., CM, CM-Bright, Concrete):

```

1186 \DeclareEncodingSubset{TS1}{cmr}      {0}
1187 \DeclareEncodingSubset{TS1}{cmss}     {0}
1188 \DeclareEncodingSubset{TS1}{cmtt}     {0}
1189 \DeclareEncodingSubset{TS1}{cmvtt}    {0}
1190 \DeclareEncodingSubset{TS1}{cmbr}     {0}
1191 \DeclareEncodingSubset{TS1}{cmtl}     {0}
1192 \DeclareEncodingSubset{TS1}{ccr}      {0}

```

PSNFSS fonts:

```

1193 \DeclareEncodingSubset{TS1}{ptm}      {4}
1194 \DeclareEncodingSubset{TS1}{pcr}      {4}
1195 \DeclareEncodingSubset{TS1}{phv}      {4}
1196 \DeclareEncodingSubset{TS1}{ppl}      {3}
1197 \DeclareEncodingSubset{TS1}{pag}      {4}
1198 \DeclareEncodingSubset{TS1}{pbk}      {4}
1199 \DeclareEncodingSubset{TS1}{pnc}      {4}
1200 \DeclareEncodingSubset{TS1}{pzc}      {4}
1201 \DeclareEncodingSubset{TS1}{bch}      {4}
1202 \DeclareEncodingSubset{TS1}{put}      {5}

```

Other CTAN fonts (probably not complete):

```

1203 \DeclareEncodingSubset{TS1}{uag}      {5}
1204 \DeclareEncodingSubset{TS1}{ugq}      {5}
1205 \DeclareEncodingSubset{TS1}{ul8}      {4}
1206 \DeclareEncodingSubset{TS1}{ul9}      {4} % (LuxiSans, one day)

```

```

1207 \DeclareEncodingSubset{TS1}{augie}    {5}
1208 \DeclareEncodingSubset{TS1}{dayrom}   {3}
1209 \DeclareEncodingSubset{TS1}{dayroms}  {3}
1210 \DeclareEncodingSubset{TS1}{pxr}      {0}
1211 \DeclareEncodingSubset{TS1}{pxss}     {0}
1212 \DeclareEncodingSubset{TS1}{pxtt}     {0}
1213 \DeclareEncodingSubset{TS1}{txr}      {0}
1214 \DeclareEncodingSubset{TS1}{txss}     {0}
1215 \DeclareEncodingSubset{TS1}{txtt}     {0}

```

Latin Modern and TeX Gyre:

```

1216 \DeclareEncodingSubset{TS1}{lmr}      {0}
1217 \DeclareEncodingSubset{TS1}{lmdh}     {0}
1218 \DeclareEncodingSubset{TS1}{lmss}     {0}
1219 \DeclareEncodingSubset{TS1}{lmssq}    {0}
1220 \DeclareEncodingSubset{TS1}{lmvtt}    {0}
1221 \DeclareEncodingSubset{TS1}{lmtt}     {0}

1222 \DeclareEncodingSubset{TS1}{qhv}     {0}
1223 \DeclareEncodingSubset{TS1}{qag}     {0}
1224 \DeclareEncodingSubset{TS1}{qbk}     {0}
1225 \DeclareEncodingSubset{TS1}{qcr}     {0}
1226 \DeclareEncodingSubset{TS1}{qcs}     {0}
1227 \DeclareEncodingSubset{TS1}{qpl}     {0}
1228 \DeclareEncodingSubset{TS1}{qtm}     {0}
1229 \DeclareEncodingSubset{TS1}{qzc}     {0}
1230 \DeclareEncodingSubset{TS1}{qhvc}    {0}

```

Fourier-GUTenberg:

```

1231 \DeclareEncodingSubset{TS1}{futs}     {4}
1232 \DeclareEncodingSubset{TS1}{futex}    {4}
1233 \DeclareEncodingSubset{TS1}{futj}     {4}

```

Y&Y's Lucida Bright

```

1234 \DeclareEncodingSubset{TS1}{hlh}     {3}
1235 \DeclareEncodingSubset{TS1}{hls}     {3}
1236 \DeclareEncodingSubset{TS1}{hlst}    {3}

```

The remaining settings for Lucida are conservative: the following fonts contain the `\textohm` character but not the `\texteuro`, i.e., belong to neither subset 4 nor subset 3. If you want to use the `\textohm` with these fonts copy these definition to `textcomp.cfg` and change the subset to 3. However in that case make sure that you do not use the `\texteuro`.

```

1237 \DeclareEncodingSubset{TS1}{hlct}    {5}
1238 \DeclareEncodingSubset{TS1}{hlx}     {5}
1239 \DeclareEncodingSubset{TS1}{hlce}    {5}
1240 \DeclareEncodingSubset{TS1}{hlcn}    {5}
1241 \DeclareEncodingSubset{TS1}{hlcw}    {5}
1242 \DeclareEncodingSubset{TS1}{hlcf}    {5}

```

Other commercial families...

```

1243 \DeclareEncodingSubset{TS1}{pplx}    {3}
1244 \DeclareEncodingSubset{TS1}{pplj}    {3}
1245 \DeclareEncodingSubset{TS1}{ptmx}    {4}
1246 \DeclareEncodingSubset{TS1}{ptmj}    {4}

```

If the file `textcomp.cfg` exists it will be loaded at this point. This allows to define further subset encodings for font families not covered by default.

```
1247 \InputIfFileExists{textcomp.cfg}
1248   {\PackageInfo{textcomp}{Local configuration file used}}{}
1249 \fi
1250 </TS1sty>
```

# File m

## ltcounts.dtx

### 21 Counters and Lengths

Commands for defining and using counters. This file defines:

<code>\newcounter</code>	To define a new counter.
<code>\setcounter</code>	To set the value of counters.
<code>\addtocounter</code>	Increase the counter #1 by the number #2.
<code>\stepcounter</code>	Increase a counter by one.
<code>\refstepcounter</code>	Increase a counter by one, also setting the value used by <code>\label</code> .
<code>\value</code>	For accessing the value of the counter as a T <sub>E</sub> X number (as opposed to <code>\the&lt;counter&gt;</code> which expands to the <i>printed</i> representation of <code>&lt;counter&gt;</code> )
<code>\arabic</code>	<code>\arabic{&lt;counter&gt;}</code> : 1, 2, 3, ...
<code>\roman</code>	<code>\roman{&lt;counter&gt;}</code> : i, ii, iii, ...
<code>\Roman</code>	<code>\Roman{&lt;counter&gt;}</code> : I, II, III, ...
<code>\alph</code>	<code>\alph{&lt;counter&gt;}</code> : a, b, c, ...
<code>\Alph</code>	<code>\Alph{&lt;counter&gt;}</code> : A, B, C, ...
<code>\fnsymbol</code>	<code>\fnsymbol{&lt;counter&gt;}</code> : *, †, ‡, ...

1 (\*2ekernel)

#### 21.1 Environment Counter Macros

An environment foo has an associated counter defined by the following control sequences:

<code>\c@foo</code>	Contains the counter's numerical value. It is defined by <code>\newcount\foocounter</code> .
<code>\thefoo</code>	Macro that expands to the printed value of <code>\foocounter</code> . For example, if sections are numbered within chapters, and section headings look like Section II-3. The Nature of Counters then <code>\thesection</code> might be defined by: <code>\def\thesection</code> <code>{\@Roman{\c@chapter}-\@arabic{\c@section}}</code>
<code>\p@foo</code>	Macro that expands to a printed 'reference prefix' of counter foo. Any <code>\ref</code> to a value created by counter foo will produce the expansion of <code>\p@foo\thefoo</code> when the <code>\label</code> command is executed. See file <code>ltxref.dtx</code> for an extension of this mechanism.
<code>\cl@foo</code>	List of counters to be reset when foo stepped. Has format <code>\@elt{countera}\@elt{counterb}\@elt{counterc}</code> .

#### NOTE:

`\thefoo` and `\p@foo` *must* be defined in such a way that `\edef\bar{\thefoo}` or `\edef\bar{\p@foo}` defines `\bar` so that it will evaluate to the counter value at the time of the `\edef`, even after `\foocounter` and any other counters have been changed. This will happen if you use the standard commands `\@arabic`, `\@Roman`, etc.

The following commands are used to define and modify counters.

`\refstepcounter{<foo>}`  
 Same as `\stepcounter`, but it also defines `\@currentreference` so that a subsequent `\label{<bar>}` command causes `\ref{<bar>}` to generate the current value of counter `<foo>`.

`\@definecounter{<foo>}`  
 Initializes counter `{<foo>}` (with empty reset list), defines `\p@foo` and `\thefoo` to be null. Also adds `<foo>` to `\cl@ckpt` – the reset list of a dummy counter `@ckpt` used for taking checkpoints for the `\include` system.

`\@addtoreset{<foo>}{<bar>}` : Adds counter `<foo>` to the list of counters `\cl@bar` to be reset when counter `<bar>` is stepped.

`\setcounter` `\setcounter{<foo>}{<val>}` : Globally sets `\foocounter` equal to `<val>`.

```

2 \def\setcounter#1#2{%
3   \@ifundefined{c@#1}%
4     {\@nocounterr{#1}}%
5     {\global\csname c@#1\endcsname#2\relax}}
```

`\addtocounter` `\addtocounter{<foo>}{<val>}` Globally increments `\foocounter` by `<val>`.

```

6 \def\addtocounter#1#2{%
7   \@ifundefined{c@#1}%
8     {\@nocounterr{#1}}%
9     {\global\advance\csname c@#1\endcsname #2\relax}}
```

`\newcounter` `\newcounter{<newctr>}[<oldctr>]` Defines `<newctr>` to be a counter, which is reset when counter `<oldctr>` is stepped. If `<newctr>` already defined produces ‘`c@newctr` already defined’ error.

```

10 \def\newcounter#1{%
11   \expandafter\ifdefinable \csname c@#1\endcsname
12     {\@definecounter{#1}}%
13     \@ifnextchar[{\@newctr{#1}}{-}]}
```

`\value` `\value{<ctr>}` produces the value of counter `<ctr>`, for use with a `\setcounter` or `\addtocounter` command.

```

14 \def\value#1{\csname c@#1\endcsname}
```

`\@newctr`

```

15 \def\@newctr#1[#2]{%
16   \@ifundefined{c@#2}{\@nocounterr{#2}}{\@addtoreset{#1}{#2}}}
```

`\stepcounter` `\stepcounterfoo` Globally increments counter `\c@F00` and resets all subsidiary counters.

```

17 \def\stepcounter#1{%
18   \addtocounter{#1}\@ne
19   \begingroup
20     \let\@elt\@stpelt
21     \csname cl@#1\endcsname
22   \endgroup}
```

`\@stpelt` Rather than resetting the “within” counter to zero we set it to `-1` and then run `\stepcounter` that moves it to `0` and also initiates resetting the next level down.

```

23 </2ekernel>
24 \ifx\@stpelt\@includeinrelease\@includeinrelease{2015/01/01}{\@stpelt}
```

```

25 \latexrelease)                                {Reset nested counters}%
26 (*2ekernel | latexrelease)
27 \def\@stpelt#1{\global\csname c@#1\endcsname \m@ne\stepcounter{#1}}%
28 \latexrelease)\EndIncludeInRelease
29 /2ekernel | latexrelease)
30 \latexrelease)\IncludeInRelease{0000/00/00}{\@stpelt}
31 \latexrelease)                                {Reset nested counters}%%
32 \latexrelease)\def\@stpelt#1{\global\csname c@#1\endcsname \z@}%
33 \latexrelease)\EndIncludeInRelease
34 (*2ekernel)

\cl@ckpt

35 \def\cl@ckpt{\@elt{page}}

\@definecounter

36 \def\@definecounter#1{\expandafter\newcount\csname c@#1\endcsname
37   \setcounter{#1}\z@
38   \global\expandafter\let\csname cl@#1\endcsname\@empty
39   \@addtoreset{#1}{@ckpt}%
40   \global\expandafter\let\csname p@#1\endcsname\@empty
41   \expandafter
42   \gdef\csname the#1\expandafter\endcsname\expandafter
43     {\expandafter\@arabic\csname c@#1\endcsname}}

\@addtoreset

44 \def\@addtoreset#1#2{\expandafter\@cons\csname cl@#2\endcsname {{#1}}

    Numbering commands for definitions of \theCOUNTER and \list arguments.
    All commands can now be used in text and math mode.

\arabic Representation of counter as arabic numerals. Changed 29 Apr 86 to make it
print the obvious thing it COUNTER not positive.
45 \def\arabic#1{\expandafter\@arabic\csname c@#1\endcsname}

\roman Representation of counter as lower-case Roman numerals.
46 \def\roman#1{\expandafter\@roman\csname c@#1\endcsname}

\Roman Representation of counter as upper-case Roman numerals.
47 \def\Roman#1{\expandafter\@Roman\csname c@#1\endcsname}

\alph Representation of counter as a lower-case letter: 1 = a, 2 = b, etc.
48 \def\alph#1{\expandafter\@alph\csname c@#1\endcsname}

\Alph Representation of counter as an upper-case letter: 1 = A, 2 = B, etc.
49 \def\Alph#1{\expandafter\@Alph\csname c@#1\endcsname}

\fnsymbol Representation of COUNTER as a footnote symbol: 1 = *, 2 = †, etc.
50 \def\fnsymbol#1{\expandafter\@fnsymbol\csname c@#1\endcsname}

\@arabic \@arabic\F00counter Representation of \F00counter as arabic numerals.
51 \def\@arabic#1{\number #1} %% changed 29 Apr 86

```

```

\@roman \@roman\F00counter Representation of \F00counter as lower-case Roman nu-
merals.
52 \def\@roman#1{\romannumeral #1}

\@Roman \@Roman\F00counter Representation of \F00counter as upper-case Roman nu-
merals.
53 \def\@Roman#1{\expandafter\@slowromancap\romannumeral #1@}

\@slowromancap Fully expandable macro to change a roman number to uppercase.
54 \def\@slowromancap#1{\ifx @#1% then terminate
55     \else
56     \if i#1I\else\if v#1V\else\if x#1X\else\if l#1L\else\if
57     c#1C\else\if d#1D\else \if m#1M\else#1\fi\fi\fi\fi\fi\fi\fi
58     \expandafter\@slowromancap
59     \fi
60 }

\@alph \@alph\F00counter Representation of \F00counter as a lower-case letter: 1 =
a, 2 = b, etc.
61 \def\@alph#1{%
62     \ifcase#1\or a\or b\or c\or d\or e\or f\or g\or h\or i\or j\or
63     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
64     y\or z\else\@ctrerr\fi}

\@Alph \@Alph\F00counter Representation of \F00counter as an upper-case letter: 1 =
A, 2 = B, etc.
65 \def\@Alph#1{%
66     \ifcase#1\or A\or B\or C\or D\or E\or F\or G\or H\or I\or J\or
67     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
68     Y\or Z\else\@ctrerr\fi}

\@fnsymbol Typesetting old fashioned footnote symbols. This can be done both in text or
math mode now.

This macro is another example of an ever recurring problem in TEX: Deter-
mining if something is text-mode or math-mode. It is imperative for the decision
between text and math to be delayed until the actual typesetting is done as the
code in question may go through an \edef or \write where an \ifmode test
would be executed prematurely. Hence in the implementation below, \@fnsymbol
is not robust in itself but the parts doing the actual typesetting are.

In the case of \@fnsymbol we make use of the robust command \TextOrMath
which takes two arguments and typesets the first if in text-mode and the second if
in math-mode. Note that in order for this command to make the correct decision,
it must insert a \relax token if run under regular TEX, which ruins any kerning
between the preceding characters and whatever awaits typesetting. If you use
eTEX as engine for LATEX (as recommended) this unfortunate side effect is not
present.
69 </2ekernel>
70 <latexrelease>\IncludeInRelease{2015/01/01}{\@fnsymbol}{Use \TextOrMath}%
71 <2ekernel | latexrelease>
72 \def\@fnsymbol#1{%
73     \ifcase#1\or \TextOrMath\textasteriskcentered *\or

```

```

74 \TextOrMath \textdagger \dagger\or
75 \TextOrMath \textdaggerdbl \ddagger \or
76 \TextOrMath \textsection \mathsection\or
77 \TextOrMath \textparagraph \mathparagraph\or
78 \TextOrMath \textbardbl \|\or
79 \TextOrMath {\textasteriskcentered\textasteriskcentered}{**}\or
80 \TextOrMath {\textdagger\textdagger}{\dagger\dagger}\or
81 \TextOrMath {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}\else
82 \ctrerr \fi
83 }%
84 </2ekernel | latexrelease>
85 <latexrelease>\EndIncludeInRelease
86 <latexrelease>\IncludeInRelease{0000/00/00}{\@fnsymbol}{Use \TextOrMath}%
87 <latexrelease>\def\@fnsymbol#1{\ensuremath{%
88 <latexrelease> \ifcase#1\or *\or \dagger\or \ddagger\or \mathsection\or
89 <latexrelease> \mathparagraph\or \|\or **\or \dagger\dagger
90 <latexrelease> \or \ddagger\ddagger \else\ctrerr\fi}}%
91 <latexrelease>\EndIncludeInRelease
92 <*2ekernel>

```

`\TextOrMath` When using regular  $\text{T}_{\text{E}}\text{X}$ , we make this command robust so that it always selects the correct branch in an `\ifmmode` switch with the usual disadvantage of ruining kerning. For the application we use it for here that shouldn't matter. The alternative would be to mimic `\IeC` from `inputenc` but then it will have the disadvantage of choosing the wrong branch if appearing at the beginning of an alignment cell. However, users of  $\text{eT}_{\text{E}}\text{X}$  will be pleasantly surprised to get the best of both worlds and no bad side effects.

First some code for checking if we are running  $\text{eT}_{\text{E}}\text{X}$  but making sure not to permanently turn `\TeXversion` into `\relax`.

```

93 </2ekernel>
94 <latexrelease>\IncludeInRelease{2015/01/01}{\TextOrMath}{\TextOrMath}%
95 <*2ekernel | latexrelease>
96 \begingroup\expandafter\expandafter\expandafter\endgroup
97 \expandafter\ifx\csname eTeXversion\endcsname\relax

```

In case of ordinary  $\text{T}_{\text{E}}\text{X}$  we define `\TextOrMath` as a robust command but make sure it always grabs its arguments. If we didn't do this it might very well gobble spaces in the input stream.

```

98 \DeclareRobustCommand\TextOrMath{%
99 \ifmmode \expandafter\@secondoftwo
100 \else \expandafter\@firstoftwo \fi}
101 \protected@edef\TextOrMath#1#2{\TextOrMath{#1}{#2}}
102 \else

```

For  $\text{eT}_{\text{E}}\text{X}$  the situation is similar. The robust macro is a hidden one so that we again avoid problems of gobbling spaces in the input.

```

103 \protected\expandafter\def\csname TextOrMath\space\endcsname{%
104 \ifmmode \expandafter\@secondoftwo
105 \else \expandafter\@firstoftwo \fi}
106 \edef\TextOrMath#1#2{%
107 \expandafter\noexpand\csname TextOrMath\space\endcsname
108 {#1}{#2}}
109 \fi
110 </2ekernel | latexrelease>

```

```

111 \latexrelease\EndIncludeInRelease
112 \latexrelease\IncludeInRelease{0000/00/00}{\TextOrMath}{\TextOrMath}%
113 \latexrelease\let\TextOrMath\@undefined
114 \latexrelease\EndIncludeInRelease
115 \*2ekernel)
116 \2ekernel)

```

# File n ltnlength.dtx

## 22 Lengths

```

\newlength Declare #1 to be a new length command.
\setlength Set the length command, #1, to the value #2.
\addtolength Increase the value of the length command, #1, by the value #2.
\settowidth Set the length, #1 to the width of a box containing #2.
\settoheight Set the length, #1 to the height of a box containing #2.
\settodepth Set the length, #1 to the depth of a box containing #2.

1 (*2ekernel)
2 \message{lengths,}

\newlength
3 \def\newlength#1{\@ifdefinable#1{\newskip#1}}

\setlength
4 (/2ekernel)
5 \langle latexrelease \rangle \IncludeInRelease{2015/01/01}%
6 \langle latexrelease \rangle \setlength{Using \setlength with \dimen0}%
7 (*2ekernel | latexrelease)

8 \def\setlength#1#2{#1 #2\relax}
9 (/2ekernel | latexrelease)
10 \langle latexrelease \rangle \EndIncludeInRelease
11 \langle latexrelease \rangle \IncludeInRelease{0000/00/00}%
12 \langle latexrelease \rangle \setlength{Using \setlength with \dimen0}%
13 \langle latexrelease \rangle \def\setlength#1#2{#1#2\relax}
14 \langle latexrelease \rangle \EndIncludeInRelease
15 (*2ekernel)

\addtolength \relax added 24 Mar 86
16 \def\addtolength#1#2{\advance#1 #2\relax}

\settoheight The obvious analogs of \settowidth.
\settodepth
\settowidth
\@settodim Clear the memory afterwards (which might be a lot).
17 \def\@settodim#1#2#3{\setbox\@tempboxa\hbox{#{#3}}#2#1\@tempboxa}
18 \setbox\@tempboxa\box\voidb@x}
19 \def\settoheight{\@settodim\ht}
20 \def\settodepth {\@settodim\dp}
21 \def\settowidth {\@settodim\wd}

\@settopoint This macro takes the contents of the skip register that is supplied as its argument
and removes the fractional part to make it a whole number of points. This can be
used in class files to avoid values like 345.4666666pt when calculating a dimension.

22 \def\@settopoint#1{\divide#1\p@\multiply#1\p@}
23 (/2ekernel)

```

## File o

# ltfssbas.dtx

This file contains the main implementation of the ‘low level’ font selection commands. See other parts of the L<sup>A</sup>T<sub>E</sub>X distribution, or *The L<sup>A</sup>T<sub>E</sub>X Companion* for higher level documentation of the L<sup>A</sup>T<sub>E</sub>X ‘New’ Font Selection Scheme.

**Warning:** The macro documentation is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

The ‘2ekernel’ code ensures that a `\usepackage{autofss1}` is essentially ignored if a ‘full’ format is being used that has picture mode already in the format. Note the `autofss2` loading is currently disabled.

```
1 (2ekernel)\expandafter\let\csname ver@autofss1.sty\endcsname\fmtversion
```

## 23 Preliminary macros

We define a number of macros that will be used later.

`\@nomath` `\@nomath` is used by most macros that will have no effect in math mode. It issues a warning message.

```
2 (*2ekernel)
3 \def\@nomath#1{\relax\ifmmode
4   \@font@warning{Command \noexpand#1invalid in math mode}\fi}
```

`\no@alphabet@error` The macro `\no@alphabet@error` is called whenever the user requests a math *alphabet* that is not available in the current *version*. In math mode an error message is produced otherwise the command keeps silent. The argument is the name of the control sequence that identifies the math *alphabet*. The `\relax` at the beginning is necessary to prevent T<sub>E</sub>X from scanning too far in certain situations.

```
5 \gdef\no@alphabet@error#1{\relax \ifmmode
6   \@latex@error{Math\space alphabet\space identifier\space
7     \noexpand#1is\space undefined\space in\space math\space
8     version\space ‘\math@version’}%
9   {Your\space requested\space math\space alphabet\space
10    is\space undefined\space in\space the\space current\space
11    math\space version.^^JCheck\space the\space spelling\space
12    or\space use\space the\space \noexpand\SetMathAlphabet\space
13    command.}
14   \fi}
```

`\new@mathgroup` We also give a new name to `\newfam` and `\fam` to avoid verbal confusion (see the introduction).<sup>2</sup>

```
\mathgroup
15 %\def\new@mathgroup{\alloc@8\mathgroup\chardef\sixt@@n}
16 \let\mathgroup\fam
17 %\let\newfam\new@mathgroup
18 \@onlypreamble\new@mathgroup
```

<sup>2</sup>For the same reason it seems advisable to `\let\fam` and `\newfam` equal to `\relax`, but this is commented out to retain compatibility to existing style files.

## 24 Macros for setting up the tables

`\DeclareFontShape` The macro `\DeclareFontShape` takes 6 arguments:

```
19 \def\DeclareFontShape{\begingroup
```

First we restore the catcodes of all characters used in the syntax.

```
20 \nfss@catcodes
```

We use `\expandafter \endgroup` to restore catcode in case something goes wrong with the argument parsing (suggested by Tim Van Zandt)

`\DeclareFontShape`

```
21 \expandafter\endgroup
22 \DeclareFontShape@}
23 \def\DeclareFontShape@#1#2#3#4#5#6{%
24 \expandafter\ifx\csname #1+#2\endcsname\relax
25 \@latex@error{Font family ‘#1+#2’ unknown}\@eha
26 \else
27 \expandafter
28 \xdef\csname#1/#2/#3/#4\endcsname{\expandafter\noexpand
29 \csname #5\endcsname}%
30 \def\reserved@a{#6}%
31 \global
32 \expandafter\let\csname#5\endcsname\expandafter\endcsname
33 \ifx\reserved@a@empty
34 \@empty
35 \else
36 \reserved@a
37 \fi
38 \fi
39 }
```

`\DeclareFixedFont` Define a direct font switch that avoids all overhead.

```
40 \def\DeclareFixedFont#1#2#3#4#5#6{%
41 \begingroup
42 \math@fontsfalse
43 \every@math@size{}%
44 \fontsize{#6}\z@
45 \usefont{#2}{#3}{#4}{#5}%
46 \global\expandafter\let\expandafter#1\the\font
47 \endgroup
48 }
```

`\do@subst@correction`

```
49 \def\do@subst@correction{%
50 \xdef\subst@correction{%
51 \font@name
52 \global\expandafter\font
53 \csname \curr@fontshape/\f@size\endcsname
54 \noexpand\fontname\font
55 \relax}%
```

Calling `\subst@correction` after the current group means calling it after we have loaded the substitution font which is done inside a group.

```
56 \aftergroup\subst@correction
57 }
```

`\DeclareFontFamily`

```
58 \def\DeclareFontFamily#1#2#3{%
```

If we want fast checking for the encoding scheme we can just check for `\T@.` being defined.

```
59 % \@tempwafalse
60 % \def\reserved@b{#1}%
61 % \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
62 %     \ifx\reserved@b\reserved@c \@tempwattrue\fi}%
63 % \cdp@list
64 % \if@tempswa
65 % \ifundefined{T@#1}%
66 %     {%
67 %         \@latex@error{Encoding scheme ‘#1’ unknown}\@eha
68 %     }%
69 % }
```

Now we have to define the macro `\<#1>+<#2>` to contain `#3`. But since most of the time `#3` will be empty we use `\let` in a tricky way rather than a simple `\def` since this will save internal memory. We store the argument `#3` in a temporary macro `\reserved@a`.

```
70     \def\reserved@a{#3}%
```

We compare `\reserved@a` with `\@empty`. If these two are the same we `\let` the ‘extra’ macro equal to `\@empty` which is not the same as doing a `\let` to `\reserved@a` — the latter would blow one extra memory location rather than reusing the one from `\@empty`.

```
71     \global
72     \expandafter\let\csname #1+#2\expandafter\endcsname
73         \ifx \reserved@a\@empty
74             \@empty
75         \else \reserved@a
76         \fi
77     }%
78 }
```

`\cdp@list` We initialize the code page list to be empty.

```
79 \let\cdp@list\@empty
80 \@onlypreamble\cdp@list
```

`\cdp@elt`

```
81 \let\cdp@elt\relax
82 \@onlypreamble\cdp@elt
```

`\DeclareFontEncoding`

```
83 \def\DeclareFontEncoding{%
```

First we start with ignoring all blanks and newlines since every surplus space in the second or third argument will come out in a weird place in the document.

```
84     \begingroup
85     \nfss@catcodes
86     \expandafter\endgroup
87     \DeclareFontEncoding@}
88 \@onlypreamble\DeclareFontEncoding
```

```

89 \def\DeclareFontEncoding#1#2#3{%
90   \expandafter
91   \ifx\csname T@#1\endcsname\relax
92     \def\cdp@elt{\noexpand\cdp@elt}%
93     \xdef\cdp@list{\cdp@list\cdp@elt{#1}%
94                   {\default@family}{\default@series}%
95                   {\default@shape}}}%

```

To support encoding dependent commands (like accents) we initialise the command `\<encoding>-cmd` to be `\@changed@cmd`. (See `ltoutenc.dtx` for details.)

```

96   \expandafter\let\csname#1-cmd\endcsname\@changed@cmd
97 \else
98   \font@info{Redeclaring font encoding #1}%
99 \fi
100 \global\@namedef{T@#1}{#2}%
101 \global\@namedef{M@#1}{\default@M#3}%

```

Keep a record of the last encoding being declared:

```

102 \xdef\LastDeclaredEncoding{#1}%
103 }
104 \@onlypreamble\DeclareFontEncoding@

```

`\LastDeclaredEncoding` The last encoding being declared by `\DeclareFontEncoding`.

```

105 \def\LastDeclaredEncoding{}

```

`\DeclareFontSubstitution`

```

106 \def\DeclareFontSubstitution#1#2#3#4{%
107   \expandafter
108   \ifx\csname T@#1\endcsname\relax
109     \@latex@error{Encoding scheme ‘#1’ unknown}\@eha
110   \else
111     \begingroup

```

We loop through the `\cdp@list` and rebuild it anew in `\toks@` thereby replacing the defaults for the encoding in question with the new defaults. It is important to store the encoding to test against expanded in `\reserved@a` since it might just be `\LastDeclaredEncoding` that is passed as `#1`.

```

112     \edef\reserved@a{#1}%
113     \toks@{}%
114     \def\cdp@elt##1##2##3##4{%
115       \def\reserved@b{##1}%
116       \ifx\reserved@a\reserved@b

```

Here we use the new defaults but we use `##1` (i.e., the encoding name already stored previously) since we know that it is expanded.

```

117       \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
118     \else

```

If `\reserved@a` and `\reserved@b` differ then we simply copy from the old list to the new.

```

119       \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
120     \fi}%
121     \cdp@list

```

```

122     \xdef\cdp@list{\the\toks@}%
123   \endgroup
124   \global
125   \@namedef{D@#1}{%
126     \def\default@family{#2}%
127     \def\default@series{#3}%
128     \def\default@shape{#4}%
129   }%
130 \fi
131 }
132 \@onlypreamble\DeclareFontSubstitution

```

\DeclareFontEncodingDefaults

```

133 \def\DeclareFontEncodingDefaults#1#2{%
134   \ifx\relax#1\else
135     \ifx\default@T\@empty\else
136       \@font@info{Overwriting encoding scheme text defaults}%
137     \fi
138     \gdef\default@T{#1}%
139   \fi
140   \ifx\relax#2\else
141     \ifx\default@M\@empty\else
142       \@font@info{Overwriting encoding scheme math defaults}%
143     \fi
144     \gdef\default@M{#2}%
145   \fi
146 }
147 \@onlypreamble\DeclareFontEncodingDefaults

```

\default@T

\default@M

```

148 \let\default@T\@empty
149 \let\default@M\@empty

```

\DeclarePreloadSizes

```

150 \def\DeclarePreloadSizes#1#2#3#4#5{%
151   \@ifundefined{T@#1}{%
152     {\@latex@error{Encoding scheme ‘#1’ unknown}\@eha}%
153   }%

```

Don't know at the moment what this group here does!

```

154   \begingroup

```

We define a macro `\reserved@f`<sup>3</sup> that grabs the next *size* and loads the corresponding font. This is done by delimiting `\reserved@f`'s only argument by the token `,` (comma).

```

155   \def\reserved@f##1,{%

```

The end of the list will be detected when there are no more elements, i.e. when `\reserved@f`'s argument is empty. The trick used here is explained in Appendix D of the *T<sub>E</sub>Xbook*: if the argument is empty the `\if` will select the first clause and `\let \reserved@f` equal to `\relax`. (We use the `>` character here since it cannot appear in font file names.)

```

156     \if>##1>%

```

---

<sup>3</sup>We cannot use `\@tempa` since it is needed in `\pickup@font`.

```

157         \let\reserved@f\relax
158     \else

```

Otherwise, we define `\font@name` appropriately and call `\pickup@font` to do the work. Note that the requested `\curr@fontshape` combination must have been defined, or you will get an error. The definition of `\font@name` is carried out globally to be consistent with the rest of the code in this file.

```

159         \xdef\font@name{\csname#1/#2/#3/#4/##1\endcsname}%
160         \pickup@font

```

Now we forget the name of the font just loaded. More precisely, we set the corresponding control sequence to `\relax`. This means that later on, when the font is first used, the macro `\define@newfont` is called again to execute the ‘extra’ macro for this font.

```

161         \global\expandafter\let\font@name\relax
162     \fi

```

Finally we call `\reserved@f` again to process the next *size*. If `\reserved@f` was `\let` equal to `\relax` this will end the macro.

```

163     \reserved@f}%

```

We finish with reinserting the list of sizes after the `\reserved@f` macro and appending an empty element so that the end of the list is recognized properly.

```

164     \reserved@f#5,,%
165     \endgroup
166 }%
167 }
168 \@onlypreamble\DeclarePreloadSizes

```

`\ifmath@fonts` We need a switch to decide if we have to switch math fonts. For this purpose we provide `\ifmath@fonts` that can be set to true or false by the `\S@...` macros depending on if math fonts are provided for this size or not. The default is of course to switch all fonts.

```

169 \newif\ifmath@fonts \math@fontstrue

```

`\DeclareMathSizes` `\DeclareMathSizes` takes the text size, math text size, math script size, and math scriptscript size as arguments and defines the right `\S@...` macro.

```

170 \def\DeclareMathSizes{%
171     \@ifstar{\@DeclareMathSizes\math@fontsfalse}%
172     {\@DeclareMathSizes{}}%
173 \@onlypreamble\DeclareMathSizes

```

`\@DeclareMathSizes` This modification by Michael J. Downes on `comp.text.tex` on 2002/10/17 allows the user to have settings such as

```

\DeclareMathSizes{9.5dd}{9.5dd}{7.4dd}{6.6dd}.

```

```

174 />2kernel)
175 \<latexrelease>\IncludeInRelease{2015/01/01}{\@DeclareMathSizes}%
176 \<latexrelease>           {Arbitrary units in \DeclareMathSizes}%
177 /*2kernel|<latexrelease>
178 \def\@DeclareMathSizes #1#2#3#4#5{%
179     \@defaultunits\dimen@ #2pt\relax\@nnil
180     \if $#3$%
181         \expandafter\let\csname S@<strip@pt>\dimen@</strip@pt>\endcsname\math@fontsfalse
182     \else

```

```

183 \@defaultunits\dimen@ii #3pt\relax\@nnil
184 \@defaultunits\@tempdima #4pt\relax\@nnil
185 \@defaultunits\@tempdimb #5pt\relax\@nnil
186 \toks@{#1}%
187 \expandafter\xdef\csname S@\strip@pt\dimen@\endcsname{%
188 \gdef\noexpand\tf@size{\strip@pt\dimen@ii}%
189 \gdef\noexpand\sf@size{\strip@pt\@tempdima}%
190 \gdef\noexpand\ssf@size{\strip@pt\@tempdimb}%
191 \the\toks@
192 }%
193 \fi
194 }%
195 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
196 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
197 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
198 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
199 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
200 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
201 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
202 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
203 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
204 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
205 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
206 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
207 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
208 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
209 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
210 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
211 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
212 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
213 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
214 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi
215 \ifx\@fontenc\@fontenc\relax\else\@fontenc\relax\fi

```

## 25 Selecting a new font

### 25.1 Macros for the user

`\fontencoding` As we said in the introduction a font is described by four parameters. We first define macros to specify the wanted *family*, *series*, or *shape*. These are simply recorded in internal macros `\f@family`, `\f@series`, and `\f@shape`, resp. We use `\edef`'s so that the arguments can also be macros.

```

216 \DeclareRobustCommand\fontencoding[1]{%
217 \expandafter\ifx\csname T@#1\endcsname\relax
218 \latexerror{Encoding scheme ‘#1’ unknown}\@eha
219 \else
220 \edef\f@encoding{#1}%
221 \ifx\cf@encoding\f@encoding

```

If the new encoding is the same as the old encoding we have nothing to do. However, in case we had a sequence of several encoding changes without a `\selectfont` in-between we can save processing by making sure that `\enc@update` is `\relax`.

```

222         \let\enc@update\relax
223     \else
    If current and new encoding differ we define the macro \enc@update to contain
    all updates necessary at \selectfont time.
224         \let\enc@update\@@enc@update
225     \fi
226 \fi
227 }

\@@enc@update
228 \def\@@enc@update{%
    When \@@enc@update is executed \f@encoding holds the encoding name for the
    new encoding and \cf@encoding the name of the last active encoding.
    We start by setting the init command for encoding dependent macros to
    \@changed@cmd.
229     \expandafter
230     \let
231     \csname\cf@encoding -cmd\endcsname
232     \@changed@cmd
    Then we turn the one for the new encoding to \@current@cmd (see ltoutenc.dtx
    for further explanations).
233     \expandafter
234     \let
235     \csname\f@encoding-cmd\endcsname
236     \@current@cmd
    We execute the default settings \default@T, followed by the one for the new
    encoding.
237     \default@T
238     \csname T@\f@encoding\endcsname
    Finally we change the default substitution values, disable \enc@update and make
    \f@encoding officially the current encoding.
239     \csname D@\f@encoding\endcsname
240     \let\enc@update\relax
241     \let\cf@encoding\f@encoding
242 }

\enc@update The default action in \selectfont is to do nothing.
243 \let\enc@update\relax

\fontfamily
  \f@family 244 \DeclareRobustCommand\fontfamily[1]{\edef\f@family{#1}}
\fontseries 245 \DeclareRobustCommand\fontseries[1]{\edef\f@series{#1}}
  \f@series 246 \DeclareRobustCommand\fontshape [1]{\edef\f@shape{#1}}
\fontshape
  \f@shape Some handy abbreviation if you want to get some particular font in the current
  size. If also the size should change one has to issue a \fontsize command first.
247 \def\usefont#1#2#3#4{\fontencoding{#1}\fontfamily{#2}%
248     \fontseries{#3}\fontshape{#4}\selectfont
249     \ignorespaces}

```

`\linespread` The command `\linespread` changes the current `\baselinestretch` by calling `\set@fontsize`. The values for `\f@size` and `\f@baselineskip` will be left unchanged.

```

250 \DeclareRobustCommand\linespread[1]
251   {\set@fontsize{#1}\f@size\f@baselineskip}

```

`\fontsize` We also define a macro that allows to specify a size. In this case, however, we also need the value of `\baselineskip`. As the first argument to `\set@fontsize` we pass the current value of `\baselinestretch`. This will either match the internal value (in which case nothing changes, or it will be an updated value due to a user change of that macro using `\renewcommand`. If we would pass the internal `\f@linespread` such a change would be effectively overwritten by a size change.

```

252 \DeclareRobustCommand\fontsize[2]
253   {\set@fontsize\baselinestretch{#1}{#2}}

```

`\f@linespread` This macro holds the current internal value for `\baselinestretch`.

```

254 \let\f@family\@empty
255 \let\f@series\@empty
256 \let\f@shape\@empty
257 \let\f@size\@empty
258 \let\f@baselineskip\@empty
259 \let\f@linespread\@empty

```

`\cf@encoding`

```

260 \let\f@encoding\@empty
261 \let\cf@encoding\@empty

```

`\@defaultunits` The function `\@defaultunits` when wrapped around a `dimen` or `skip` assignment supplies default units. Usage:

```

\@defaultunits\dimen@=#1pt\relax\@nnil

```

Note: the `\relax` is *important*. Other units can be substituted for the ‘pt’ if desired.

We use `\remove@to@nnil` as an auxiliary macros for `\@defaultunits`. It just has to gobble the supplied default unit ‘pt’ or whatever, if it wasn’t used in the assignment.

```

262 \def\@defaultunits{\afterassignment\remove@to@nnil}

```

`\strip@pt` This macro strips the characters `pt` produced by using `\the` on a `dimen` register.

`\rem@pt`

```

263 \begingroup
264   \catcode‘P=12
265   \catcode‘T=12
266   \lowercase{
267     \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}
268   \expandafter\endgroup\x
269 \def\strip@pt{\expandafter\rem@pt\the}

```

`\mathversion` `\mathversion` takes the math *version* name as argument, defines `\math@version` appropriately and switches to the font selected forcing a call to `\glb@settings` if the *version* is known to the system.

```

270 \DeclareRobustCommand\mathversion[1]
271   {\@nomath\mathversion

```

```

272         \expandafter\ifx\csname mv@#1\endcsname\relax
273         \@latex@error{Math version ‘#1’ is not defined}\@eha\else
274         \edef\math@version{#1}%

```

We need to force a math font setup both now and at the point where we return to the previous math version. Forcing a math font setup can simply be done by setting `\glb@currsiz` to an invalid value since this will trigger the setup when the formula starts.

```

275         \gdef\glb@currsiz{}%

```

When the scope of the current `\mathversion` ends we need to restore the old setup. However this time we need to force it directly at least if we are inside math, otherwise we could wait. Another way to enhance this code here is to do the setting only if the version really has changed after all. This might be interesting in case of `amstext` and `boldsymbol`.

```

276         \aftergroup\glb@settings
277         \fi}

```

If  $\text{\TeX}$  would support a hook just before the end of a formula (opposite of `\everymath` so to speak) the implementation of the algorithm would be much simpler because in that case we would set up the correct math fonts at this point without having to worry about incorrect settings due to nesting. The same would be true if in  $\text{\LaTeX}$  the use of `$` (as the primitive  $\text{\TeX}$  command) would be impossible and instead only a higher-level interface would be available. Note that this does not mean that a `$` couldn't be the short-hand for starting and stopping that higher-level interface, it only means that the direct  $\text{\TeX}$  function must be hidden.

Anyway, since we don't have this and won't have it in  $\text{\LaTeX 2}_\epsilon$  we need to implement it in a somewhat slower way.

We test for the current math font setup on entry of a formula, i.e., on the hooks `\everymath` and `\everydisplay`. But since these hooks may contain user data we provide ourselves with an internal version of these hooks which stays frozen.

```

\frozen@everymath New internal names for \everymath and \everydisplay.
\frozen@everydisplay 278 \let\frozen@everymath\everymath
279 \let\frozen@everydisplay\everydisplay

```

```

\everymath Now we provide now user hooks that will be called in the frozen internals.
\everydisplay 280 \newtoks\everymath
281 \newtoks\everydisplay

```

```

\frozen@everymath Now we define the behaviour of the frozen hooks: first check the math setup then
call the user hook.
282 \frozen@everymath = {\check@mathfonts
283                     \the\everymath}

\frozen@everydisplay Ditto for the display hook.
284 \frozen@everydisplay = {\check@mathfonts
285                     \the\everydisplay}

```

```

\curr@math@size This holds locally the current math size.
286 \let\curr@math@size\@empty

```

## 25.2 Macros for loading fonts

<code>\pickup@font</code>	<p>The macro <code>\pickup@font</code> which is used in <code>\selectfont</code> is very simple: if the font name is undefined (i.e. not known yet) it calls <code>\define@newfont</code> to load it.</p> <pre> 287 \def\pickup@font{% 288     \expandafter \ifx \font@name \relax 289     \define@newfont 290     \fi} </pre>
<code>\split@name</code>	<p><code>\pickup@font</code> assumes that <code>\font@name</code> is set but it is sometimes called when <code>\f@family</code>, <code>\f@series</code>, <code>\f@shape</code>, or <code>\f@size</code> may have the wrong settings (see, e.g., the definition of <code>\getanddefine@fonts</code>). Therefore we need a macro to extract font <i>family</i>, <i>series</i>, <i>shape</i>, and <i>size</i> from the font name. To this end we define <code>\split@name</code> which takes the font name as a list of characters of <code>\catcode 12</code> (without the backslash at the beginning) delimited by the special control sequence <code>\@nil</code>. This is not very complicated: we first ensure that <code>/</code> has the right <code>\catcode</code></p> <pre> 291 {\catcode'\/=12 </pre> <p>and define <code>\split@name</code> so that it will define our private <code>\f@encoding</code>, <code>\f@family</code>, <code>\f@series</code>, <code>\f@shape</code>, and <code>\f@size</code> macros.</p> <pre> 292 \gdef\split@name#1/#2/#3/#4/#5\@nil{\def\f@encoding{#1}% 293                                     \def\f@family{#2}% 294                                     \def\f@series{#3}% 295                                     \def\f@shape{#4}% 296                                     \def\f@size{#5}} </pre>
<code>\curr@fontshape</code>	<p>Abbreviation which may get removed again for speed.</p> <pre> 297 \def\curr@fontshape{\f@encoding/\f@family/\f@series/\f@shape} </pre>
<code>\define@newfont</code>	<p>Now we can tackle the problem of defining a new font.</p> <pre> 298 \def\define@newfont{% </pre> <p>We have already mentioned that the token list that <code>\split@name</code> will get as argument must not start with a backslash. To reach this goal we will set the <code>\escapechar</code> to <code>-1</code> so that the <code>\string</code> primitive will not generate an escape character. To keep this change local we open a group. We use <code>\begingroup</code> for this purpose since <code>\define@newfont</code> might be called in math mode, and an empty <code>\bgroup... \egroup</code> would add an empty Ord atom to the math list and thus affect the spacing.</p> <p>Also locally redefine <code>\typeout</code> so that ‘No file ...fd’ Warnings become Font Info message just sent to the log file.</p> <pre> 299 \begingroup 300     \let\typeout\@font@info 301     \escapechar\m@ne </pre> <p>Then we extract <i>encoding scheme</i>, <i>family</i>, <i>series</i>, <i>shape</i>, and <i>size</i> from the font name. Note the four <code>\expandafter</code>’s so that <code>\font@name</code> is expanded first, then <code>\string</code>, and finally <code>\split@name</code>.</p> <pre> 302     \expandafter\expandafter\expandafter 303     \split@name\expandafter\string\font@name\@nil </pre>

If the `\curr@fontshape` combination is not available, (i.e. undefined) we call the macro `\wrong@fontshape` to take care of this case. Otherwise `\extract@font` will load the external font for us.

```

304 % \expandafter\ifx
305 % \csname\curr@fontshape\endcsname \relax
306 \try@load@fontshape % try always
307 % \fi
308 \expandafter\ifx
309 \csname\curr@fontshape\endcsname \relax
310 \wrong@fontshape\else

```

To allow substitution we call the `\curr@fontshape` macro which usually will expand to `\relax` but may hold code for substitution (see `\subst@fontshape` definition).

```

311 % \csname\curr@fontshape\endcsname
312 \extract@font\fi

```

We are nearly finished and must only restore the `\escapechar` by closing the group.

```

313 \endgroup}
314 \def\try@load@fontshape{%
315 \expandafter
316 \ifx\csname \f@encoding+\f@family\endcsname\relax
317 \font@info{Try loading font information for
318 \f@encoding+\f@family}%

```

We predefine this combination to be `\@empty` which means that next time we don't try again unnecessary in case we don't find a `.fd` file. If the file contains a `\DeclareFontFamily` command than this setting will be overwritten.

```

319 \global\expandafter\let
320 \csname\f@encoding+\f@family\endcsname\@empty

```

Set the catcodes used in the syntax, but do it only once (this will be restored at the end of the font loading group).

```

321 \nfss@catcodes
322 \let\nfss@catcodes\relax

```

For increased portability make the external filename monospace, but look for the (old style) mixed case filename if the first attempt fails.

On any monospace system this means that the file is looked for twice which takes up time and string space, but at least for this release Check for both names to give people time to re-install their private `fd` files with lowercase names.

```

323 \edef\reserved@a{%
324 \lowercase{%
325 \noexpand\InputIfFileExists{\f@encoding\f@family.fd}}}%
326 \reserved@a\relax
327 {\input@{\f@encoding\f@family.fd}}%
328 \fi}

```

`\nfss@catcodes` This macro should contain the standard `\catcode` assignments to all characters which are used in the commands found in an `.fd` file and which might have special `\catcodes` in the middle of a document. If necessary, this list can be extended in a package file using a suitable number of `\expandafter`, i.e.,

```

\expandafter\def\expandafter\nfss@catcodes
\expandafter{\nfss@catcodes <additional settings>}

```

Note, that this macro might get executed several times since it is also called by `\DeclareFontShape`, thus it probably should not be misused as a general purpose hook.

```
329 \def\nfss@catcodes{%
```

We start by making @ a letter and ignoring all blanks and newlines.

```
330     \makeatletter
331     \catcode'\ 9%
332     \catcode'\^^I9%
333     \catcode'\^^M9%
```

Then we set up \, {, }, # and % in case an .fd file is loaded during a verbatim environment.

```
334     \catcode'\\\z@
335     \catcode'\{ \@ne
336     \catcode'\}\tw@
337     \catcode'\#6%
338     \catcode'\^7%
339     \catcode'\%14%
```

The we make sure that the important syntax parts have the right `\catcode`.

```
340     \@makeother\<%
341     \@makeother\>%
342     \@makeother\*%
343     \@makeother\.%
344     \@makeother\-%
345     \@makeother\/%
346     \@makeother\[%
347     \@makeother\]%
348     \@makeother\'%
349     \@makeother\'%
350     \@makeother\"%
351 }
```

`\DeclareErrorFont` Declare the last resort shape! We assume that in this fontshape there is a 10pt font but it doesn't really matter. We only loose one macro name if the assumption is false. But at least the font should be there!

```
352 \def\DeclareErrorFont#1#2#3#4#5{%
353     \xdef\error@fontshape{%
354         \noexpand\expandafter\noexpand\split@name\noexpand\string
355         \expandafter\noexpand\csname#1/#2/#3/#4/#5\endcsname
356         \noexpand\@nil}%
```

Initialize all those internal variables which may or may not have values in the first seconds of NFSS' bootstrapping process. Later on such values will be updated when an encoding is selected, etc.

We definitely don't want to set `\f@encoding`; we can set all the others since if they are left "blank" any selection would grap "error default values" as well. However, this probably should go also.

```
357 %     \gdef\f@encoding{#1}%
358     \gdef\default@family{#2}%
359     \gdef\default@series{#3}%
360     \gdef\default@shape{#4}%
361     \global\let\f@family\default@family
```

```

362     \global\let\f@series\default@series
363     \global\let\f@shape\default@shape
364     \gdef\f@size{#5}%
365     \gdef\f@baselineskip{#5pt}%
366 }
367 \@onlypreamble\DeclareErrorFont

```

`\wrong@fontshape` Before we come to the macro `\extract@font` we have to take care of unknown `\curr@fontshape` combinations. The general strategy is to issue a warning and to try a default *shape*, then a default *series*, and finally a default *family*. If this last one also fails T<sub>E</sub>X will go into an infinite loop. But if the defaults are set incorrectly one deserves nothing else!

```

368 (/2ekernel)
369 (latexrelease)\IncludeInRelease{2015/01/01}{\wrong@fontshape}%
370 (latexrelease)           {Font substitution in preamble}%
371 (*2ekernel|latexrelease)
372 \def\wrong@fontshape{%
373     \csname D@f@encoding\endcsname % install defaults if in math
374     \edef\reserved@a{\csname\curr@fontshape\endcsname}%
375     \ifx\last@fontshape\reserved@a
376         \errmessage{Corrupted NFSS tables}%
377         \error@fontshape
378     \else

```

Then we warn the user about the mess and set the shape to its default.

```

379     \let\f@shape\default@shape

```

If the combination is not known, try the default *series*.

```

380     \expandafter\ifx\csname\curr@fontshape\endcsname\relax
381         \let\f@series\default@series

```

If this is still undefined, try the default *family*. Otherwise give up. We never try to change the encoding scheme!

```

382     \expandafter
383     \ifx\csname\curr@fontshape\endcsname\relax
384         \let\f@family\default@family

```

If we change the font family and we are in the preamble then the corresponding `.fd` file may not been loaded yet. Therefore we try this now. Otherwise equating the requested font shape with the finally selected fontshape below will fail and can result in “NFSS tables corrupted”. After begin document that will not happen as all `.fd` files involved in substitution are loaded at `\begin{document}`.

```

385         \begingroup
386             \try@load@fontshape
387         \endgroup
388     \fi \fi
389 \fi

```

At this point a valid `\curr@fontshape` combination must have been found. We inform the user about this fact.

The `\expandafter\string` here stops T<sub>E</sub>X adding the space that it usually puts after command names in messages. The similar construction with `\@undefined` just produces ‘undefined’, but saves a few tokens.

`\@wrong@font@char` is locally redefined in `\UseTextSymbol` from its normal (empty) definition, to report the symbol generating the font switch.

```
390 \font@warning{Font shape '\expandafter\string\reserved@a'
391 \expandafter@gobble\string\undefined\MessageBreak
392 using '\curr@fontshape' instead@\wrong@font@char}%
393 \global\let\last@fontshape\reserved@a
```

We change `\@defaultsubs` to produce a warning at the end of the document.

The macro `\@defaultsubs` is initially `\relax` but gets changed here if some default font substitution happens. It is then executed in `\enddocument`.

```
394 \gdef\@defaultsubs{%
395 \font@warning{Some font shapes were not available, defaults
396 substituted.\@gobbletwo}}%
```

If we substitute a `\curr@fontshape` combination by the default one we don't want the warning to be printed out whenever this (unknown) combination is used. Therefore we globally `\let` the macro corresponding to the wanted combination equal to its substitution. This requires the use of four `\expandafter`'s since `\csname... \endcsname` has to be expanded before `\reserved@a` (i.e. the requested combination), and this must happen before the `\let` is executed.

```
397 \global\expandafter\expandafter\expandafter\let
398 \expandafter\reserved@a
399 \csname\curr@fontshape\endcsname
```

Now we can redefine `\font@name` accordingly. This *must* be done globally since it might occur in the group opened by `\define@newfont`. If we would this definition were local the closing `\endgroup` there would restore the old meaning of `\font@name` and then switch to the wrong font at the end of `\selectfont` although the correct font was loaded.

```
400 \xdef\font@name{%
401 \csname\curr@fontshape/\f@size\endcsname}%
```

The last thing this macro does is to call `\pickup@font` again to load the font if it is not defined yet. At this point this code will loop endlessly if the defaults are not well defined.

```
402 \pickup@font}
403 </2ekernel | latexrelease>
404 <latexrelease>\EndIncludeInRelease
405 <latexrelease>\IncludeInRelease{0000/00/00}{\wrong@fontshape}%
406 <latexrelease> {Font substitution in preamble}%
407 <latexrelease>\def\wrong@fontshape{%
408 <latexrelease> \csname D@\f@encoding\endcsname
409 <latexrelease> \edef\reserved@a{\csname\curr@fontshape\endcsname}%
410 <latexrelease> \ifx\last@fontshape\reserved@a
411 <latexrelease> \errmessage{Corrupted NFSS tables}%
412 <latexrelease> \error@fontshape
413 <latexrelease> \else
414 <latexrelease> \let\f@shape\default@shape
415 <latexrelease> \expandafter\ifx\csname\curr@fontshape\endcsname\relax
416 <latexrelease> \let\f@series\default@series
417 <latexrelease> \expandafter
418 <latexrelease> \ifx\csname\curr@fontshape\endcsname\relax
419 <latexrelease> \let\f@family\default@family
420 <latexrelease> \fi \fi
```

```

421 \latexrelease \fi
422 \latexrelease \font@warning{Font shape
423 \latexrelease '\expandafter\string\reserved@a'
424 \latexrelease \expandafter\@gobble\string\@undefined
425 \latexrelease \MessageBreak
426 \latexrelease using '\curr@fontshape' instead\@wrong@font@char}%
427 \latexrelease \global\let\last@fontshape\reserved@a
428 \latexrelease \gdef\@defaultsubs{%
429 \latexrelease \font@warning{Some font shapes were not available,
430 \latexrelease defaults substituted.\@gobbletwo}}%
431 \latexrelease \global\expandafter\expandafter\expandafter\let
432 \latexrelease \expandafter\reserved@a
433 \latexrelease \csname\curr@fontshape\endcsname
434 \latexrelease \xdef\font@name{%
435 \latexrelease \csname\curr@fontshape/\f@size\endcsname}%
436 \latexrelease \pickup@font}
437 \latexrelease \EndIncludeInRelease
438 \*2kernel)

```

`\@wrong@font@char` Normally empty but redefined in `\UseTextSymbol` so that the Font shape undefined message can refer to the symbol causing the problem.

```
439 \let\@wrong@font@char\@empty
```

`\@defaultsubs` See above.

```
\@defaultsubs 440 \let\@defaultsubs\relax
```

`\strip@prefix` In `\extract@font` we will need a way to recover the replacement text of a macro. This is done by the primitive `\meaning` together with the macro `\strip@prefix` (for the details see appendix D of the *T<sub>E</sub>Xbook*, p. 382).

```
441 \def\strip@prefix#1>{\}
```

## 26 Assigning math fonts to *versions*

`\install@mathalphabet` This is just another name for `\gdef` but we can redefine it if necessary later on.

```
442 \let\install@mathalphabet\gdef
```

`\math@fonts`

```
443 \let\math@fonts\@empty
```

`\select@group` `\select@group` has four arguments: the new *math alphabet identifier* (a control sequence), the *math group number*, the extra macro for math mode and the `\curr@fontshape` definition macro name. We first check if we are in math mode.

```
444 %\def\select@group#1#2#3{\relax\ifmmode
```

We do these things locally using `\begin@group` instead of `\bgroup` to avoid the appearance of an empty Ord atom on the math list.

```
445 % \begin@group
```

We set the math fonts for the *family* in question by calling `\getanddefine@fonts` in the correct environment.

```
446 % \escapechar\m@ne
```

```
447 % \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
```

We globally select the math fonts...

```
448 % \globaldefs\@ne \math@fonts
```

... and close the group to restore `\globaldefs` and `\escapechar`.

```
449 % \endgroup
```

As long as no *size* or *version* change occurs the  $\langle\textit{math alphabet identifier}\rangle$  should simply switch to the installed *math group* instead of calling `\select@group` unnecessarily. So we globally redefine the first argument (the new  $\langle\textit{math alphabet identifier}\rangle$ ) to expand into a `\mathgroup` switch and then select this *alphabet*. Note that this redefinition will be overwritten by the next call to a *version* macro.

The original code for the end of `\select@group` was

```
\gdef#1{#3\mathgroup #2}#1\fi}
```

i.e. first redefining the  $\langle\textit{math alphabet identifier}\rangle$  and then calling the new definition to switch to the wanted  $\langle\textit{math group}\rangle$ . Now we define the  $\langle\textit{math alphabet identifier}\rangle$  as a call to the `\use@mathgroup` command.

```
450 % \xdef#1{\noexpand\use@mathgroup\noexpand#2%
```

```
451 % \number\csname c@mv@\math@version\endcsname}}%
```

But this is not sufficient, as we learned the hard way. The problem here is that the loading of the fonts that comprise the alphabet identifier `#1`, as well as the necessary math font assignments is deferred until it is used. This is OK so far, but if the fonts are switched within the current formula (which may happen if a sub-formula is a box that contains a math version switch) the font assignments for `#1` are not restored unless `#1` is used again. This is disastrous since TeX sees the wrong fonts at the end of the math formula, when it converts the math list into a horizontal list.

This is taken into account as follows: When a math alphabet identifier is used for the first time in a certain version it modifies the corresponding macro `\mv@<version>` so that it calls `\getanddefine@fonts` directly in future as well. We use the macro `\extract@alph@from@version` to do this. It takes the math alphabet identifier `#1` and the math version macro as arguments.

```
452 % \expandafter\extract@alph@from@version
```

```
453 % \csname mv@\math@version\expandafter\endcsname
```

```
454 % \expandafter{\number\csname c@mv@\math@version\endcsname}}%
```

```
455 % #1%
```

```
456 % \stepcounter{mv@\math@version}%
```

Finally, it is not possible to simply call the new definition since we have an argument (the third argument of `\use@mathgroup` or more exactly the argument of `\math@egroup` if the `margid` option is in force) which would swallow our closing `\fi`. So we use the `\expandafter` technique to remove the `\fi` before the `\use@mathgroup` is expanded.

```
457 %\expandafter #1\fi}
```

`\extract@alph@from@version`

We proceed to the definition of the macro `\extract@alph@from@version`. As stated above, it takes a math alphabet identifier and a math version macro (e.g. `\mv@normal`) as its arguments.

```
458 \def\extract@alph@from@version#1#2#3{%
```

To extract and replace the definition of math alphabet identifier `#3` in macro `#1` we have to recall how this definition looks like: Somewhere in the replacement

text of #1 there is the sequence

```
\install@mathalphabet<math alphabet identifier> #3{%
  <Definitions for >#3}
```

Hence, the first thing we do is to extract the tokens preceding this definitions, the definition itself, and the tokens following it. To this end we define one auxiliary macro `\reserved@a`.

```
459      \def\reserved@a##1\install@mathalphabet#3##2##3\@nil{%
```

When `\reserved@a` is expanded, it will have the tokens preceding the definition in question in its first argument (`##1`), the following tokens in its third argument (`##3`), and the replacement text for the math alphabet identifier `#3` in its second argument. (`##2`). This is then recorded for later use in a temporary macro `\reserved@b`.

```
460      \def\reserved@b{##2}%
```

Additionally, we define a macro `\reserved@c` to reconstruct the definitions for the math version in question from the tokens that will remain unchanged (`##1` and `##3`) and the yet to build new definitions for the math alphabet identifier `#3`.

```
461      \def\reserved@c####1{\gdef#1{##1####1##3}}%
```

Then we execute our auxiliary macro.

```
462      \expandafter\reserved@a#1\@nil
```

OK, so now we have to build the new definition for `#3`. To do so, we first extract the interesting parts out of the old one. The old definition looks like:

```
\select@group<math alphabet identifier>
  <math group number><math extra part>
<curr@fontshape definition>
```

So we define a new temporary macro `\reserved@a` that extracts these parts.

```
463      \def\reserved@a\select@group#3##1##2\@nil{%
```

This macro can now directly rebuild the math version definition by calling `\reserved@c`:

```
464      \reserved@c{%
465      \getanddefine@fonts{#2}##2%
466      \install@mathalphabet#3{%
467      \relax\ifmmode \else \non@alpherr#3\fi
468      \use@mathgroup##1{#2}}%
```

In addition it defines the alphabet the way it should be used from now on.

```
469      \gdef#3{\relax\ifmmode \else \non@alpherr#3\fi
470      \use@mathgroup##1{#2}}%
```

Finally, we only have to call this macro `\reserved@a` on the old definitions recorded in `\reserved@b`:

```
471      \expandafter\reserved@a\reserved@b\@nil
472      }
```

`\math@bgroup` Here are the default definitions for `\math@bgroup` and `\math@egroup`. We use `\bgroup` instead of `\begingroup` to avoid ‘leaking out’ of style changes. This has the side effect of always producing mathord atoms.

```
473 \let\math@bgroup\bgroup
474 \def\math@egroup#1{#1\egroup}
```

`\calculate@math@sizes` Here is the default definition for `\calculate@math@sizes` a more elaborate interface is under testing in `mathscale.sty`.

```

475 \gdef\calculate@math@sizes{%
476   \@font@info{Calculating\space math\space sizes\space for\space
477     size\space <\f@size>}%
478   \dimen@ \f@size \p@
479   \@tempdimb \defaultscriptratio \dimen@
480   \dimen@ \defaultscriptscriptratio \dimen@
481   \expandafter\xdef\csname S@\f@size\endcsname{%
482     \gdef\noexpand\tf@size{\f@size}%
483     \gdef\noexpand\sfs@size{\strip@pt\@tempdimb}%
484     \gdef\noexpand\ssf@size{\strip@pt\dimen@}%
485     \noexpand\math@fontstrue}}

```

`\defaultscriptratio` The default ratio for math sizes is:  
`\defaultscriptscriptratio` 1 to `\defaultscriptratio` to `\defaultscriptscriptratio`.  
By default this is 1 to .7 to .5.

```

486 \def\defaultscriptratio{.7}
487 \def\defaultscriptscriptratio{.5}

```

`\noaccents@` If we don't have a definition for `\noaccents@` we provide a dummy.

```

488 \ifx\noaccents@\@undefined
489   \let\noaccents@\@empty
490 \fi

```

`\showhyphens` The `\showhyphens` command must be redefined since the version in `plain.tex` uses `\tenrm`. We have also made some further adjustments for its use in L<sup>A</sup>T<sub>E</sub>X.

```

491 \gdef\showhyphens#1{%
492   \setbox0\vbox{%
493     \color@begingroup
494     \everypar{}%
495     \parfillskip\z@skip\hsize\maxdimen
496     \normalfont
497     \pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\ #1%
498     \color@endgroup}}

```

`\addto@hook` We need a macro to add tokens to a hook.

```

499 \long\def\addto@hook#1#2{#1\expandafter{\the#1#2}}

```

`\@vpt`

```

500 \def\@vpt{5}

```

`\@vipt`

```

501 \def\@vipt{6}

```

`\@viipt`

```

502 \def\@viipt{7}

```

`\@viiipt`

```

503 \def\@viiipt{8}

```

`\@ixpt`

```

504 \def\@ixpt{9}

```

```

\@xpt
505 \def\@xpt{10}

\@xipt
506 \def\@xipt{10.95}

\@xiipt
507 \def\@xiipt{12}

\@xivpt
508 \def\@xivpt{14.4}

\@xvipt
509 \def\@xvipt{17.28}

\@xxpt
510 \def\@xxpt{20.74}

\@xxvpt
511 \def\@xxvpt{24.88}
512 </2ekernel>

```

# File p

## ltfsstrc.dtx

### 27 Introduction

This package contains the code for tracing font loading and font changes. It basically overlays some of the low-level functions of NFSS with additional code used for tracing.

The package accepts the following options:

**errorshow** Write all information about font changes etc. only to the transcript file unless an error happens. This means that information about font substitution will not be shown on the terminal.

**warningshow** Show all NFSS warnings on the terminal. This setting corresponds to the default behaviour of NFSS if the `tracefnt` package is *not* loaded!

**infoshow** Show all NFSS warning and all NFSS info messages (that are normally only written to the transcript file) also on the terminal. This is the default if the `tracefnt` package is loaded.

**debugshow** In addition to `infoshow` show also changing of math fonts as far as possible (this option can produce a large amount of output).

**loading** Show the name of external fonts when they are loaded. This option shows only “newly” loaded fonts not those already preloaded in the format or the class file before the `tracefnt` package became active.

**pausing** Turn all font warnings into errors so that L<sup>A</sup>T<sub>E</sub>X will stop.

### 28 A driver for this document

The next bit of code contains the documentation driver file for T<sub>E</sub>X, i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the DOCSTRIP program.

When this file is processed directly by L<sup>A</sup>T<sub>E</sub>X this will produce the documentation as well.

```
1 (*driver)
2 \documentclass{ltxdoc}
3
4
5 %\OnlyDescription % comment out for implementation details
6
7 \begin{document}
8   \DocInput{ltfsstrc.dtx}
9 \end{document}
10 /driver
```

## 29 The Implementation

**Warning:** Read the macro documentation with a grain of salt. It is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

If we are making a package file it is a good idea to test whether we are running under 2e. This code is actually placed at the very beginning of this file for easier maintenance, thus commented out here.

```
11 (*package)
12 %\NeedsTeXFormat{LaTeX2e}
13 %\ProvidesPackage{tracefnt}[??/??/?? v?.??]
14 %
15 %\Standard LaTeX package (font tracing)]
16 \end{package}
```

The `debug` module makes use of commands contained in a special package file named `trace.sty`.<sup>4</sup>

```
16 \ifdebug \input trace.sty
```

## 30 Handling Options

`\tracingfonts` Here is the definition of the integer register for the font trace. As a default in a package file we use 1 to give error messages if fonts are substituted. If this code is used for debugging or tracing reasons in the format file (i.e. in `fam.dtx`) we use 0 as the default. But if no font trace is used we build a definition that will produce a warning message.

```
17 (*2ekernel)
18 \def\tracingfonts{%
19   \@font@warning{Command \noexpand\tracingfonts
20     not provided.\MessageBreak
21     Use the ‘tracefnt’ package.\MessageBreak Command found:}%
22   \count@}
23 \end{2ekernel}
```

The `\count@` in the line above will remove the number after `\tracingfonts`. Note that this definition will be overwritten by the next line if one of these modules are included.

```
24 (*package, trace, debug)
25 \newcount\tracingfonts
26 \tracingfonts=0
27 \end{package, trace, debug}
```

The option `errorshow` turns off all warnings so that only real errors are shown. `warningshow` corresponds to the NFSS default (when `tracefnt` is not loaded). `info` is the default for this package here; and `debugshow`, `loading`, and `pausing` extend the amount of information even further.

```
28 (*package)
29 \DeclareOption{errorshow}{%
30   \def\@font@info#1{%
31     \GenericInfo{(Font)}\@spaces\@spaces\@spaces\space\space}%
32   }
33 \end{package}
```

---

<sup>4</sup>This package is not in distribution at the moment (and probably doesn't any longer work). Think of this part of the code as being historical artefacts.

```

32             {LaTeX Font Info: \space\space\space#1}}%
33   \def\@font@warning#1{%
34     \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
35     {LaTeX Font Warning: #1}}%
36   }
37 \DeclareOption{warningshow}{%
38   \def\@font@info#1{%
39     \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
40     {LaTeX Font Info: \space\space\space#1}}%
41   \def\@font@warning#1{%
42     \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
43     {LaTeX Font Warning: #1}}%
44   }
45 \DeclareOption{infoshow}{%
46   \def\@font@info#1{%
47     \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
48     {LaTeX Font Info: \space\space\space#1}}%
49   \def\@font@warning#1{%
50     \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
51     {LaTeX Font Warning: #1}}%
52   }
53 \DeclareOption{loading}{%
54   \tracingfonts\tw@
55   }
56 \DeclareOption{debugshow}{%
57   \ExecuteOptions{infoshow}%
58   \tracingfonts\thr@@
59   }
60 \DeclareOption{pausing}{%
61   \def\@font@warning#1{%
62     \GenericError
63     {(Font)\@spaces\@spaces\@spaces\space\space}%
64     {LaTeX Font Warning: #1}%
65     {See the LaTeX Companion for details.}%
66     {I'll stop for every LaTeX Font Warning because
67     you requested\MessageBreak the 'pausing' option
68     to the tracefmt package.}}%
69   }

```

We make `infoshow` the default, which in turn defines `\font@warning` and `\font@info`.

```

70 \ExecuteOptions{infoshow}
71 \ProcessOptions
72 \endpackage

```

We also need a default definition inside the kernel:

```

73 \ifx\kernel
74   \def\@font@info#1{%
75     \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
76     {LaTeX Font Info: \space\space\space#1}}%
77   \def\@font@warning#1{%
78     \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%

```

```

79                                     {LaTeX Font Warning: #1}}%
80 </2ekernel>

```

## 31 Macros common to fam.tex and tracefmt.sty

In the first versions of `tracefmt.dtx` some macros of `fam.dtx`<sup>5</sup> were redefined to included the extra tracing information. Now these macros are all defined in this file (i.e. removed from `fam.dtx`) and different production versions can be obtained simply by specifying a different set of modules to include when generating `lftss.dtx`.

### 31.1 General font loading

`\extract@font` This macro organizes the font loading. It first calls `\get@external@font` which will return in `\external@font` the name of the external font file (the .tfm) as it was determined by the NFSS tables.

```

81 <*2ekernel | package>
82 \def\extract@font{%
83   \get@external@font

```

Then the external font is loaded and assigned to the font identifier stored inside `\font@name` (for this reason we need `\expandafter`).

```

84   \global\expandafter\font\font@name\external@font\relax

```

When tracing we typeout the internal and external font name.

```

85 <*trace>
86   \ifnum \tracingfonts >\@ne
87     \@font@info{External font '\external@font'
88               loaded as\MessageBreak \font@name}\fi
89 </trace>

```

Finally we call the corresponding “loading action” macros to finish things. First the font is locally selected to allow the use of `\font` inside the loading action macros.

```

90   \font@name \relax

```

The next two lines execute the “loading actions” for the family and then for the individual font shape.

```

91   \csname \f@encoding+\f@family\endcsname
92   \csname\curr@fontshape\endcsname
93   \relax
94   }
95 </2ekernel | package>

```

The `\relax` at the end needs to be explained. This is inserted to prevent `TEX` from scanning too far when it is executing the replacement text of the loading code macros.

`\get@external@font` This function tries to find an external font name. It will place the name into the macro `\external@font`. If no font is found it will return the one that was defined via `\DeclareErrorFont`.

```

96 <*2ekernel>
97 \def\get@external@font{%

```

---

<sup>5</sup>This file is currently not distributed in documented form. Its code is part of `lftss.dtx`.

We don't know the external font name at the beginning.

```

98   \let\external@font\@empty
99   \edef\font@info{\expandafter\expandafter\expandafter\string
100     \csname \curr@fontshape \endcsname}%
101   \try@size@range

```

If this failed, we'll try to substitute another size of the same font. This is done by the `\try@size@substitution` macro. It “knows about” `\do@extract@font`, `\font@name`, `\f@size`, and so on.

```

102   \ifx\external@font\@empty
103     \try@size@substitution
104     \ifx\external@font\@empty
105       \@latex@error{Font \expandafter \string\font@name\space
106         not found}\@eha
107       \error@fontshape
108       \get@external@font
109     \fi\fi
110 }
111 /2ekernel)

```

`\selectfont` The macro `\selectfont` is called whenever a font change must take place.

```

112 (*2ekernel | package)
113 \DeclareRobustCommand\selectfont
114   {%

```

When `debug` is specified we actually want something like ‘`undebg`’. The font selection is now stable so that using `\tracingall` on some other macros will show us a lot of unwanted information about font loading. Therefore we disable tracing during font loading as long as `\tracingfonts` is less than 4.

```

115 (+debug) \pushtracing
116 (+debug) \ifnum\tracingfonts<4 \tracingoff
117 (+debug) \else \tracingon\p@selectfont \fi

```

If `\baselinestretch` was redefined by the user it will not longer match its internal counterpart `\f@linespread`. If so we call `\set@fontsize` to prepare `\size@update`.

```

118   \ifx\f@linespread\baselinestretch \else
119     \set@fontsize\baselinestretch\f@size\f@baselineskip \fi

```

Then we generate the internal name of the font by concatenating *family*, *series*, *shape*, and current *size*, with slashes as delimiters between them. This is much more readable than standard L<sup>A</sup>T<sub>E</sub>X's `\twfbf`, etc. We define `\font@name` globally, as always. The reason for this is explained later on.

```

120   \xdef\font@name{%
121     \csname\curr@fontshape/\f@size\endcsname}%

```

We call the macro `\pickup@font` which will load the font if necessary.

```

122   \pickup@font

```

Then we select the font.

```

123   \font@name

```

If `\tracingfonts` is greater than 2 we also show the font switch. We do this before `\glb@settings` is called since this macro might redefine `\font@name`.

```

124 (*trace)

```

```

125     \ifnum \tracingfonts>\tw@
126         \font@info{Switching to \font@name}\fi
127 </trace>

```

Finally we call `\size@update`. This macro is normally empty but will contain actions (like setting the `\baselineskip`) that have to be carried out when the font size, the base `\baselineskip` or the `\baselinestretch` have changed.

```

128     \size@update

```

A similar function is called to handle anything related to encoding updates. This one is changed from `\relax` by `\fontencoding`.

```

129     \enc@update

```

Just before ending this macro we have to pop the tracing stack if it was pushed before.

```

130 <+debug> \poptracing
131     }

```

`\set@fontsize` The macro `\set@fontsize` does the actual work. First it assigns new values to `\f@size`, `\f@baselineskip` and `\f@linespread`.

```

132 \def\set@fontsize#1#2#3{%
133     \@defaultunits\@tempdimb#2pt\relax\@nnil
134     \edef\f@size{\strip@pt\@tempdimb}%
135     \@defaultunits\@tempskipa#3pt\relax\@nnil
136     \edef\f@baselineskip{\the\@tempskipa}%
137     \edef\f@linespread{#1}%

```

For backward compatibility and for later testing within `\selectfont` the internal value of `\f@linespread` is passed back to `\baselinestretch`.

```

138     \let\baselinestretch\f@linespread

```

Additional processing will happen within `\selectfont`. For this reason the macro `\size@update` (which will be called in `\selectfont`) will be defined to be:

```

139     \def\size@update{%

```

First calculate the new `\baselineskip` and also store it in `normalbaselineskip`

```

140         \baselineskip\f@baselineskip\relax
141         \baselineskip\f@linespread\baselineskip
142         \normalbaselineskip\baselineskip

```

then to set up a new `\strutbox`

```

143         \setbox\strutbox\hbox{%
144             \vrule\@height.7\baselineskip
145             \@depth.3\baselineskip
146             \@width\z@}%

```

We end with a bit of tracing information.

```

147 <*trace>
148     \ifnum \tracingfonts>\tw@
149         \ifx\f@linespread\@empty
150             \let\reserved@a\@empty
151         \else
152             \def\reserved@a{\f@linespread x}%
153         \fi
154         \font@info{Changing size to \f@size/\reserved@a
155             \f@baselineskip}%
156         \aftergroup\type@restoreinfo \fi
157 </trace>

```

When all this is processed `\size@update` redefines itself to `\relax` so that in later calls of `\selectfont` no extra code will be executed.

```
158      \let\size@update\relax}%
159  }
```

Instead of defining this macro internally we might speed things up by placing the code into a separate macro and use `\let`!

`\size@update` Normally this macro does nothing; it will be redefined by `\set@fontsize` to initiate an update.

```
160 \let\size@update\relax
```

`\type@restoreinfo` This macro produces some info when a font size and/or baseline change will get restored.

```
161 (*trace)
162   \def\type@restoreinfo{%
163     \ifx\f@linespread\@empty
164       \let\reserved@a\@empty
165     \else
166       \def\reserved@a{\f@linespread x}%
167     \fi
168     \@font@info{Restoring size to
169                \f@size/\reserved@a\f@baselineskip}}
170 (/trace)
```

`\glb@settings` The macro `\glb@settings` globally selects all math fonts for the current size if necessary.

```
171 \def\glb@settings{%
```

When `\glb@settings` gains control a size change was requested and all previous font assignments need to be replaced. Therefore the old values of the fonts are no longer needed. For every *math group* the new assignments are appended to `\math@fonts`. But this happens only if the `math@fonts` switch is set to true. However, we always set up the correct math sizes for script and scriptscript fonts since they may be needed even if we don't set up the whole math machinery.

Here we set the math size, script size and scriptscript size. If the `S@...` macro is not defined we have to first calculate the three sizes.

```
172   \expandafter\ifx\csname S@\f@size\endcsname\relax
173     \calculate@math@sizes
174   \fi
```

The effect of this is that `\calculate@math@sizes` may or may not define the `S@...` macro. In the first case the next time the same size is requested this macro is used, otherwise `\calculate@math@sizes` is called again. This also sets the `math@fonts` switch. If it is true we must switch the math fonts.

```
175   \csname S@\f@size\endcsname
176   \ifmath@fonts
177 (*trace)
178   \ifnum \tracingfonts>\tw@
179     \@font@info{Setting up math fonts for
180                \f@size/\f@baselineskip}\fi
181 (/trace)
```

Inside a group we execute the macro for the current math *version*. This sets `\math@fonts` to a list of `\textfont...` assignments. `\getanddefine@fonts` (which may be called at this point) needs the `\escapechar` parameter to be set to `-1`.

```
182      \begingroup
183      \escapechar\m@ne
184      \csname mv@\math@version \endcsname
```

Then we set `\globaldefs` to 1 so that all following changes are done globally. The math font assignments recorded in `\math@fonts` are executed and `\glb@currsiz` is set equal to `\f@size`. This signals that the fonts for math in this size are set up.

```
185      \globaldefs\@ne
186      \math@fonts
187      \let \glb@currsiz \f@size
188      \endgroup
```

Finally we execute any code that is supposed to happen whenever the math font setup changes. This register will be executed in local mode which means that everything that is supposed to have any effect should be done globally inside. We can't execute it within `\globaldefs\@ne` as we don't know what ends up inside this register, e.g., it might contain calculations which use some local registers to calculate the final (global) value.

```
189      \the\every@math@size
```

Otherwise we announce that the math fonts are not set up for this size.

```
190 (*trace)
191      \else
192          \ifnum \tracingfonts>\tw@
193              \@font@info{No math setup for
194                          \f@size/\f@baselineskip}\fi
195      \end{trace}
196      \fi
197  }
198  \end{kernel} \end{package}
```

`\baselinestretch` In `\selectfont` we used `\baselinestretch` as a factor when assigning a value to `\baselineskip`. We use 1 as a default (i.e. no stretch).

```
199 (*2ekernel)
200 \def\baselinestretch{1}
```

`\every@math@size` We must still define the hook `\every@math@size` we used in `\glb@settings`. We initialize it to nothing. It is important to remember that everything that goes into this hook should to global updates, local changes will have weird effects.

```
201 \newtoks\every@math@size
202 \every@math@size={}
203 \end{kernel}
```

## 31.2 Math fonts setup

### 31.2.1 Outline of algorithm for math font sizes

$\TeX$  uses the the math fonts that are current when the end of a formula is reached. If we don't want to keep font setups local to every formula (which would result in

an enormous overhead, we have to be careful not to end up with the wrong setup in case formulas are nested, e.g., we need to be able to handle

```
$ a=b+c \mbox{ \small for all $b$ and $c\in Z$}$
```

Here the inner formulae `b` and `c\in Z` are typeset in `\small` but we have to return to `\normalsize` before we reach the closing `$` of the outer formula.

This is handled in the following way:

1. At any point in the document the global variable `\gbl@currsiz` contains the point size for which the math fonts currently are set up.
2. Whenever we start a formula we compare its value with the local variable `\f@size` that describes the current text font size.
3. If both are the same we assume that we can use the current math font setup without adjustment.
4. If they differ we call `\gbl@settings` which changes the math font setup and updates `\gbl@currsiz`.
  - (a) If we are recursively inside another formula (`\if@inmath`) we ensure that `\gbl@settings` is executed again in the outer formula, so that the old setup is automatically restored.
  - (b) Otherwise, we set the switch `@inmath` locally to `true` so that all nested formulae will be able to detect that they are nested in some outer formula.

The above algorithm has the following features:

- For sizes which are not containing any formula no math setup is done. Compared to the original algorithm of NFSS this results in the following savings:
  - No unnecessary loading of math fonts for sizes that are not used to typeset any math formulae (explicit or implicit ones).
  - No time overhead due to unnecessary changes of the math font setup on entrance and exit of the text font size.
- Math font setup changes for top-level formulae will survive (there is no restoration after the formula) thus any following formula in the same size will be directly typesettable. Compared to original implementation in NFSS2 the new algorithm has the overhead of one test per formula to see if the current math setup is valid (in the original algorithm the setup was always valid, thus no test was necessary).
- In nested formulae the math font setup is restored in the outer formula by a series of `\aftergroup` commands and checks. Compared to the original algorithm this involves additional checks ( $2 \times \langle \text{non-math levels} \rangle$  per inner formula).

### 31.2.2 Code for math font size setting

`\check@mathfonts` In the `\check@mathfonts` macros we implement the steps 2 to 4 except that instead of a switch the macro `\init@restore@glb@settings` is used.

```

204 (*2ekernel|package)
205 \def\check@mathfonts{%
206   \ifx \glb@currsiz \f@size
207   (*trace)
208     \ifnum \tracingfonts>\thr@@
209       \@font@info{*** MATH: no change \f@size\space
210         curr/global (\curr@math@size/\glb@currsiz)}\fi
211   (/trace)
212   \else
213   (*trace)
214     \ifnum \tracingfonts>\thr@@
215       \@font@info{*** MATH: setting up \f@size\space
216         curr/global (\curr@math@size/\glb@currsiz)}\fi
217   (/trace)
218     \glb@settings
219     \init@restore@glb@settings
220   \fi
221   \let\curr@math@size\f@size
222   \def\init@restore@glb@settings{\aftergroup\restglb@settings}%
223 }
```

`\init@restore@glb@settings` This macros does by default nothing but get redefined inside `\check@mathfonts` to initiate fontsize restoring in nested formulas.

```

224 (-trace)\let\init@restore@glb@settings\relax
225 (*trace)
226 \def\init@restore@glb@settings{%
227   \ifnum \tracingfonts>\thr@@
228     \@font@info{*** MATH: no resetting (not in
229       nested math)}\fi
230 }
231 (/trace)
```

`\restglb@settings` This macro will be executed the first time after the current formula.

```

232 \def\restglb@settings{%
233   (*trace)
234     \ifnum \tracingfonts>\thr@@
235       \@font@info{*** MATH: restoring}\fi
236   (/trace)
237     \begingroup
238       \let\f@size\curr@math@size
239       \ifx\glb@currsiz \f@size
240   (*trace)
241     \ifnum \tracingfonts>\thr@@
242       \@font@info{*** MATH: ... already okay (\f@size)}\fi
243   (/trace)
244     \else
245   (*trace)
246     \ifnum \tracingfonts>\thr@@
247       \@font@info{*** MATH: ... to \f@size}\fi
248   (/trace)
```

```

249         \glb@settings
250     \fi
251 \endgroup
252 }

```

### 31.2.3 Other code for math

`\use@mathgroup` The `\use@mathgroup` macro should be used in user macros to select a math group. Depending on whether or not the `margid` option is in force it has two or three arguments. For this reason it should be called as the last macro.

First we test if we are inside math mode since we don't want to apply a useless definition.

```

253 \def\use@mathgroup#1#2{\relax\ifmmode

```

```

254 (*trace)
255   \ifnum \tracingfonts>\tw@
256     \count@#2\relax
257     \@font@info{Using \noexpand\mathgroup
258               (\the\count@) #2}\fi
259 (/trace)

```

If so we first call the '=' macro (i.e. argument three) to set up special things for the selected math group. Then we call `\mathgroup` to select the group given by argument two and finally we place `#1` (i.e. the argument of the *math alphabet identifier*) at the end. This part of the code is surrounded by two commands which behave like `\begingroup` and `\endgroup` if we want *math alphabet identifier*s but will expand into `\empty` if we want simply switches to a new math group. Since argument number 2 may be a digit instead of a control sequence we add a `\relax`. Otherwise something like `\mit{1}` would switch to math group 11 (and back) instead of printing an oldstyle 1.

```

260     \math@bgroup
261     \expandafter\ifx\csname M@f@encoding\endcsname#1\else
262       #1\fi
263     \mathgroup#2\relax

```

Before we reinsert the swallowed token (arg. three) into the input stream, in the case that the *math alphabet identifier* isn't called in math mode, we remove the `\fi` with the `\expandafter` trick. This is necessary if the token is actually an macro with arguments. In such a case the `\fi` will be misinterpreted as the first argument which would be disastrous.

```

264     \expandafter\math@egroup\fi}%

```

The surrounding macros equal `\begingroup` and `\endgroup`. But using internal names makes it possible to overwrite their meaning in certain cases. This is for example used in  $\TeX$  macros for placing accents.

`\math@egroup` If the `margid` option is in force (which can be tested by looking at the definition of `\math@bgroup` we change the `\math@egroup` command a bit to display the current *math group number* after it closes the scope of *math alphabet* with `\endgroup`.

```

265 (*trace)
266   \ifx\math@bgroup\bgroup
267     \def\math@egroup#1{#1\egroup

```

```

268     \ifnum \tracingfonts>\tw@
269     \@font@info{Restoring \noexpand\mathgroup
270       (\ifnum\mathgroup=\m@ne default\else \the\mathgroup \fi)%
271       }\fi}
272   \fi
273 \end{tracing}

```

`\getanddefine@fonts` `\getanddefine@fonts` has two arguments: the *math group number* and the *family/series/shape* name as a control sequence.

```

274 \def\getanddefine@fonts#1#2{%
First we turn of tracing when \tracingfonts is less than 4.
275 \ifnum \tracingfonts<4 \tracingoff
276 \ifnum \tracingfonts<4 \tracingoff
277 \else \tracingon\getanddefine@fonts \fi
278 \end{tracing}
279 \ifnum \tracingfonts>\tw@
280 \count@#1\relax
281 \@font@info{\noexpand\mathgroup (\the\count@) #1 :=\MessageBreak
282   \string#2 \tf@size/\sf@size/\ssf@size}\fi
283 \end{tracing}

```

We append the current `\tf@size` to `#2` to obtain the font name.<sup>6</sup> Again, `font@name` is defined globally, for the reasons explained in the description of `\wrong@fontshape`.

```

284 \xdef\font@name{\csname \string#2/\tf@size\endcsname}%

```

Then we call `\pickup@font` to load it if necessary. We remember the internal name as `\textfont@name`.

```

285 \pickup@font \let\textfont@name\font@name

```

Same game for `\scriptfont` and `\scriptscriptfont`:

```

286 \xdef\font@name{\csname \string#2/\sf@size\endcsname}%
287 \pickup@font \let\scriptfont@name\font@name
288 \xdef\font@name{\csname \string#2/\ssf@size\endcsname}%
289 \pickup@font

```

Then we append the new `\textfont...` assignments to the `\math@fonts`.

```

290 \edef\math@fonts{\math@fonts
291   \textfont#1\textfont@name
292   \scriptfont#1\scriptfont@name
293   \scriptscriptfont#1\font@name}%

```

Just before ending this macro we have to pop the tracing stack if it was pushed before.

```

294 \ifnum \tracingfonts>\tw@ \poptracing
295 \fi
296 \end{2ekernel} \package

```

---

<sup>6</sup>One might ask why this expansion does not generate a macro name that starts with an additional `\` character. The solution is that `\escapechar` is set to `-1` before `\getanddefine@fonts` is called.

## 32 Scaled font extraction

`\ifnot@nil` We begin with a simple auxiliary macro. It checks whether its argument is the token `\@nil`. If so, it expands to `\@gobble` which discards the following argument, otherwise it expands to `\@firstofone` which reproduces its argument.

```
297 (*2kernel)
298 \def\ifnot@nil#1{\def\reserved@a{#1}%
299 \ifx\reserved@a\@nnil \expandafter\@gobble
300 \else \expandafter\@firstofone\fi}
```

`\remove@to@nnil` Three other auxiliary macros will be needed in the following: `\remove@to@nnil` gobbles up everything up to, and including, the next `\@nnil` token, and `\remove@angles` and `\remove@star` do the same for the character `>` and `*`, respectively, instead of `\@nnil`.

```
301 \def\remove@to@nnil#1\@nnil{}
302 \def\remove@angles#1>{\set@simple@size@args}
303 \def\remove@star#1*{#1}
```

`\extract@sizefn` This macro takes a size specification and parses it into size function and the optional and mandatory arguments.

```
304 \def\extract@sizefn#1*#2\@nnil{%
305 \if>#2>\set@size@func@args#1\@nnil
306 \let\sizefn@info\@empty
307 \else\expandafter\set@size@func@args\remove@star#2\@nnil
308 \def\sizefn@info{#1}\fi
309 }
```

`\try@simple@size` This function tries to extract the given size (specified by `\f@size`) for the requested font shape. The font information must already be present in `\font@info`. The central macro that does the real work is `\extract@fontinfo`. We will first give a simple example how this macro works, and describe it in full generality later.

Assume that the requested parameters are: *encoding scheme* ‘OT1’, *family* ‘cm’, *series* ‘sansserif’, *shape* ‘normal’, and *size* ‘12’. The corresponding font definitions have already been extracted from the macro `\OT1/cm/sansserif/normal` and stored in `font@info`. (Otherwise `\extract@fontinfo` doesn’t get called.) This information consists of a token list made of characters of category code 12 of the form

```
<10*>cmss10<12*>cmss12<17*>cmss17
```

For reasonable packages one usually needs more sizes but this is sufficient to get the flavour. We will define a macro `\extract@fontinfo` to find the external font name (‘cmss12’) for us:

```
\def\extract@fontinfo#1<12*#2>#3<#4\@nnil{%
\set@simple@size@args#3<#4\@nnil
\execute@size@function{#2}}
```

so that when it gets called via

```
\extract@fontinfo<10*>cmss10<12*>cmss12<17*>cmss17\@nnil
```

#1 will contain all characters before <12\*>, #2 will be empty, #3 will be exactly cmss12, and #3 will be 17>cmss17. The expansion is therefore

```
\set@simple@size@args cmss12<17*>cmss17\@nnil
\execute@size@function{}
```

This means: the default (empty) size function will be executed, with its optional argument argument set to empty and its mandatory argument set to cmss12 by \set@simple@size@args. As we discussed earlier, the effect of the default size function is to load the given external font (cmss12) at the specified size (12)—which is exactly what was intended.

But this is only part of the whole story. It may be that the size requested does not occur in the token list \font@info. And the simple definition of \extract@fontinfo we gave above does not allow to specify give more than one size specification in front of the external font name.

Let's address these two problems separately. The first one is solved with the following trick: We define \extract@fontinfo as follows:

```
\def\extract@fontinfo#1<12*#2>#3<#4\@nnil{%
  \ifnot@nil{#3}%
    {\set@simple@size@args#3<#4\@nnil
      \execute@size@function{#2}%
    }}%
```

How does this work? We call \extract@fontinfo via

```
\expandafter\extract@fontinfo\font@info<12*>\@nil<\@nnil
```

i.e. by appending <12\*>\@nil<\@nnil. If the size ('12' in this case) appears in \font@info everything works as explained above, the only difference being that argument #4 of \extract@fontinfo additionally gets the tokens <12\*>\@nil<\@nnil. However, if the size is not found everything up to the final <12\*> is in argument #1, #3 gets \@nil, and #2 and #4 are empty. The macro \ifnot@nil will discard the calls to \set@simple@size@args and execute@size@function, and hence \font@info will continue to be equal to \@empty. This means that no simple size specification matching the requested size could be found.

The second problem (more than one simple size specification for one external font name) will be addressed in \set@simple@size@args below.

The macros are hidden inside other control sequences so that we have to build \extract@fontinfo in several steps.

So here's the actual definition of \extract@font in \try@simple@size.

```
310 % % this could be replaced by \try@size@range making the subst slower!
```

```
311 \def\try@simple@size{%
```

\reserved@a is made an abbreviation for the head of the definition of the macro \extract@fontinfo.

```
312   \def\reserved@a{\def\extract@fontinfo###1}%
```

Now we can define \extract@fontinfo. Here we handle a small but convenient variation: in case of the default (empty) size function it is allowed to omit the \* character.

```
313   \expandafter\reserved@a\expandafter<\f@size>##2<##3\@nnil{%
```

```
314     \ifnot@nil{##2}%
```

```

315         {\set@simple@size@args##2<##3\@nnil
316         \execute@size@function\sizefn@info
317         }}%

```

Now we call `\extract@fontinfo`. Note the `<\@nil` tokens at the end.

```

318     \expandafter\expandafter
319     \expandafter\extract@fontinfo\expandafter\font@info
320     \expandafter<\f@size>\@nil<\@nnil
321 }

```

`\set@simple@size@args` As promised above, the macro `\set@simple@size@args` will handle the case of several size specifications in a row. If another size specification follows, the very first token of its argument list is the character `<`. By starting the definition as follows,

```

322 \def\set@simple@size@args#1<{%

```

parameter `#1` is empty in this case, and contains the size function's arguments otherwise. We distinguish these two cases (Note that the character `<` cannot appear in `#1`) by calling `\remove@angles` for empty `#1` and `\extract@sizefn` otherwise. In the latter case we have to take care of the remaining character tokens and discard them. This is done by `\remove@to@nnil`. Note also the use of Kabelschacht's method.

```

323         \if<#1<%
324         \expandafter\remove@angles
325         \else
326         \extract@sizefn#1*\@nil
327         \expandafter\remove@to@nnil
328         \fi}

```

Now, we are through with the case of a simple size, except for calling the size function. This will be handled later, as it is the same mechanism for all types of size specification. We will now proceed to macros for extraction of size range specification.

`\extract@rangefontinfo` `\extract@rangefontinfo` goes through a font shape definition in the input until it recognizes the tokens `<\@nil->`. It looks for font ranges with font size functions. Its operation is rather simple: it discards everything up to the next size specification and passes this on to `\is@range` for inspection. The specification (parameter `#2` is inserted again, in case it is needed later.

```

329 \def\extract@rangefontinfo#1<#2>{%
330     \is@range#2->\@nil#2>}

```

`\is@range` `\is@range` is again a sort of dispatcher macro: if the size specification it is looking at is not a range specification it discards it and calls `\extract@rangefontinfo` to continue the search. Otherwise it calls `\check@range` to check the requested size against the specified range.

From the way `\is@range` is called inside `\extract@rangefontinfo` we see that `#2` is the character `>` if the size specification found is a simple one (as it does not contain a `-` character. This is checked easily enough and `\extract@rangefontinfo` called again. Note that the extra tokens inserted after the `\@nil` in the call to `\is@range` appear at the beginning of the first argument to `\extract@rangefontinfo` and are hence ignored.

```

331 \def\is@range#1-#2\@nil{%
332   \if>#2\expandafter\check@single\else
333     \expandafter\check@range\fi}

```

**\check@range** \check@range takes lower bound as parameter #1, upper bound as #2, size function as #3 and the size function's arguments as #4. If #3 is the special token \@nil \font@info is exhausted and we can stop searching.

```

334 \def\check@range#1-#2>#3<#4\@nnil{%
335   \ifnot@nil{#3}{%

```

If #3 wasn't \@nil we have a range. We start by assuming that we have to recurse. Note that we have to reinsert an < as it was already removed by scanning.

```

336     \def\reserved@f{\extract@rangefontinfo<#4\@nnil}%

```

We have to make sure that both boundaries are present, if not we have to set them. Here we check the upper bound. If \upper@bound is zero after the assignment we set it to \maxdimen (upper open range). We need to use a *<dimen>* register for the scan since we may have a decimal number as the boundary.

```

337     \upper@bound0#2\p@
338     \ifdim\upper@bound=\z@ \upper@bound\maxdimen\fi

```

Now we check the upper boundary against \f@size. If it is larger or equal than \f@size this range is no good and we have to recurse.

```

339     \ifdim \f@size \p@<\upper@bound

```

Otherwise we have to check the lower bound. This time it is not necessary to scan the boundary value into a register because if it is empty we get zero as desired. We could even omit the 0 which would result in 1pt as default lower boundary. If \f@size is smaller than the boundary we have to recurse.

```

340         \lower@bound0#1\p@
341         \ifdim \f@size \p@<\lower@bound
342         \else

```

If both tests are passed we can try executing the size function.

```

343         \set@simple@size@args#3<#4\@nnil
344         \execute@size@function\sizefn@info

```

If the function was successful it should have left an external font name in \external@font. We use this to see if we can stop scanning. Otherwise we recurse.

```

345         \ifx\external@font\@empty
346         \else
347         \let\reserved@f\@empty
348         \fi
349     \fi
350 \fi
351 \reserved@f}}

```

**\lower@bound** We use two dimen registers \lower@bound and \upper@bound to store the lower  
**\upper@bound** and upper endpoints of the range we found.

```

352 \newdimen\lower@bound
353 \newdimen\upper@bound

```

`\check@single` `\check@single` takes the size as parameter #1, size function as #2 and the size function's arguments as #3. We can assume that there is always something in the pipeline since the very last entry is a faked range (see above).

```
354 \def\check@single#1>#2<#3\@nnil{%
```

We start by assuming that we have to recurse. Note that we have to reinsert an < as it was already removed by scanning.

```
355 \def\reserved@f{\extract@rangefontinfo<#3\@nnil}%
```

Now we check the the size against `\f@size`. If it is not equal `\f@size` it is no good and we have to recurse.

```
356 \ifdim \f@size \p@=#1\p@
```

Otherwise if this test is passed we can try executing the size function.

```
357 \set@simple@size@args#2<#3\@nnil
```

```
358 \execute@size@function\sizefn@info
```

If the function was successful it should have left an external font name in `\external@font`. We use this to see if we can stop scanning. Otherwise we recurse.

```
359 \ifx\external@font\@empty
```

```
360 \else
```

```
361 \let\reserved@f\@empty
```

```
362 \fi
```

```
363 \fi
```

```
364 \reserved@f}
```

`\set@size@funct@args` This macro sets the optional and mandatory arguments for a size function. If the optional argument is not present it is set to the empty token list. The mandatory argument is delimited by the token `\@nil`.

```
365 \def\set@size@funct@args{\@ifnextchar[%
```

```
366 \set@size@funct@args@{\set@size@funct@args@[]}]}
```

```
367 \def\set@size@funct@args@[#1]#2\@nil{%
```

```
368 \def\mandatory@arg{#2}%
```

```
369 \def\optional@arg{#1}}}
```

```
370 \</2kernel>
```

`\DeclareSizeFunction` This function defines a new size function hiding the internal from the designer. The body of the size function may use `\optional@arg` and `\mandatory@arg` denoting the optional and mandatory argument that may follow the size specification `<...>`.

```
371 \<2kernel>
```

```
372 \def\DeclareSizeFunction#1#2{\@namedef{s@fct@#1}{#2}}
```

```
373 \@onlypreamble\DeclareSizeFunction
```

```
374 \</2kernel>
```

`\execute@size@function` This macro is very simple. The only point worth noting is that calling an undefined size function will do nothing (actually execute a `\relax`).

```
375 \<2kernel | package>
```

```
376 \def\execute@size@function#1{%
```

```
377 \<trace>
```

```
378 \ifundefined{s@fct@#1}%
```

```
379 {\errmessage{Undefined font size function #1}}%
```

```

380          \s@fct@}%
381          {\csname s@fct@#1\endcsname}%
382 \trace)
383 (-trace)      \csname s@fct@#1\endcsname
384 }
385 \endkernel | package)

```

**\try@size@range** This macro tries to find a suitable range for requested size (specified by `\f@size`) in `\font@info`. All the relevant action is done in `\extract@rangefontinfo`. All that needs to be done is to stuff in the token list in `\font@info` so that `\extract@rangefontinfo` can inspect it. Note the `<-*\@nil>` token at the end to stop scanning.

```

386 (*2kernel)
387 \def\try@size@range{%
388     \expandafter\extract@rangefontinfo\font@info <-*\@nil<\@nnil
389 }

```

**\try@size@substitution** This is the last thing that can be tried. If the desired `\f@size` is found neither among the simple size specifications nor in one of the ranges the whole list of size specifications is searched for a nearby simple size.

```

390 \gdef\try@size@substitution{%

```

First we do some initializations. `\@tempdimb` will hold the difference between the wanted size and the best solution found so far, so we initialise it with `\maxdimen`. The macro `\best@size` will hold the best size found, nothing found is indicated by the empty value.

```

391     \@tempdimb \maxdimen
392     \let \best@size \@empty

```

Now we loop over the specification

```

393     \expandafter \try@simples \font@info <\number\@M>\@nil<\@nnil
394 }

```

**\font@submax** The macro `\font@submax` records the maximal deviation from the desired size encountered so far. Its value is used in a warning message at `\end{document}`. The macro `\fontsubfuzz` contains the amount that will not cause terminal warnings (warnings still go into the transcript file).

```

395 \def\font@submax{0pt}
396 \def\fontsubfuzz{.4pt}
397 \endkernel)
398 (+package)\def\fontsubfuzz{0pt}

```

**\try@simples** `\try@simples` goes through a font shape definition in the input until it recognizes the tokens `<*\@nil>`. It looks for simple sizes to determine the two closest sizes. It is assumed that simple sizes are in increasing order.

```

399 (*2kernel)
400 \gdef\try@simples#1<#2>{%
401     \tryif@simple#2->\tryif@simple}

```

**\tryis@simple** `\tryis@simple` is similar to `\is@range`. If it sees a simple size, it checks it against the value of `\f@size` and sets `\lower@font@size` or `\higher@font@size`. In the latter case, it stops the iteration. By adding `<\number\@M>` at the end of the line we always have an end point. This is a hack which probably should be corrected.

First it checks whether it is finished already, then whether the size specification in question is a simple one.

```
402 \gdef\tryif@simple#1-#2\tryif@simple{%
```

Most common case for \reserved@f first:

```
403 \let \reserved@f \try@simples
```

```
404 \if>#2%
```

If so, it compares it to the value of \f@size. This is done using a dimen register since there may be fractional numbers.

```
405 \dimen@ #1\p@
```

```
406 \ifdim \dimen@<\M\p@
```

If \dimen@ is \M\p@ we have reached the end of the fontspec (hopefully) otherwise we compare the value with \f@size and compute in \@tempdimc the absolute value of the difference between the two values.

```
407 \ifdim \f@size\p@<\dimen@
```

```
408 \@tempdimc \dimen@
```

```
409 \advance\@tempdimc -\f@size\p@
```

```
410 \else
```

```
411 \@tempdimc \f@size\p@
```

```
412 \advance\@tempdimc -\dimen@
```

```
413 \fi
```

The result is then compared with the smallest difference we have encountered, if the new value (in \@tempdimc is smaller) we have found a size which is a better approximation so we make it the \best@size and adjust \@tempdimb.

```
414 \ifdim \@tempdimc<\@tempdimb
```

```
415 \@tempdimb \@tempdimc
```

```
416 \def \best@size{#1}%
```

```
417 \fi
```

When we have reached the end of the fontspec we substitute the best size found (if any). We code this inline to save macro space; in the past this was done by a macro called \subst@size.

```
418 \else
```

\subst@size This macro substitutes the size recorded in \best@size for the unavailable size \f@size. \font@submax records the maximum difference between desired size and selected size in the whole run.

```
419 % \subst@size %% coded inline
```

```
420 % \def\subst@size{%
```

```
421 \ifx \external@font\@empty
```

```
422 \ifx \best@size\@empty
```

```
423 \else
```

```
424 \ifdim \@tempdimb>\font@submax \relax
```

```
425 \xdef \font@submax {\the\@tempdimb}%
```

```
426 \fi
```

```
427 \let \f@user@size \f@size
```

```
428 \let \f@size \best@size
```

```
429 \ifdim \@tempdimb>\fontsubfuzz\relax
```

```
430 \@font@warning{Font\space shape\space
```

```
431 'curr@fontshape'\space in\space size\space
```

```
432 <\f@user@size>\space not\space available\MessageBreak
```

```
433 size\space <\f@size>\space substituted}%
```

```

434      \fi
435      \try@simple@size
436      \do@subst@correction
437      \fi
438      \fi
439      % %}

```

This brings us back into the main part of `\tryif@simple`. Finally we get rid of any rubbish left over on the input stack.

```

440      \let \reserved@f \remove@to@nnil
441      \fi
442      \fi

```

If it's a range iterate also.

```

443      \reserved@f}

```

## 32.1 Sizefunctions

In the following we define some useful size functions.

`\sfct@` This is the default size function. Mandatory argument is an external font name, optional argument a scale factor. The font is scaled to `\f@size` if no optional argument is present, and to `\f@size` multiplied by the optional argument otherwise.

```

444 \DeclareSizeFunction{}\empty@sfcnt\@font@warning}
445 \DeclareSizeFunction{s}\empty@sfcnt\@font@info}
446 \def\empty@sfcnt#1{%
447     \@tempdimb \f@size\p@
448     \ifx\optional@arg\empty
449     \else
450         \@tempdimb \optional@arg\@tempdimb
451         #1{Font\space shape\space '\curr@fontshape'\space
452             will\space be\MessageBreak
453             scaled\space to\space size\space \the\@tempdimb}%
454     \fi
455     \edef\external@font{\mandatory@arg\space at\the\@tempdimb}}

```

`\sfct@gen` This size function generates the external name from the mandatory argument and  
`\sfct@sgen` the requested user size, and thus can be used for external names where the size is encoded in the font name. The optional argument a scale factor. The font is scaled to `\f@size` if no optional argument is present, and to `\f@size` multiplied by the optional argument otherwise.

```

456 \DeclareSizeFunction{gen}\gen@sfcnt\@font@warning}
457 \DeclareSizeFunction{sgen}\gen@sfcnt\@font@info}
458 \def\gen@sfcnt{%
459     \edef\mandatory@arg{\mandatory@arg\f@size}%
460     \empty@sfcnt}

```

`\sfct@genb` This size function is similar to `gen`, but for fonts where the size is encoded in  
`\sfct@sgenb` the font name in centipoints, as in the DC fonts version 1.2. The font is scaled to `\f@size` if no optional argument is present, and to `\f@size` multiplied by the optional argument otherwise.

```

461 \DeclareSizeFunction{genb}{\genb@sfcnt\@font@warning}
462 \DeclareSizeFunction{sgenb}{\genb@sfcnt\@font@info}

463 \def\genb@sfcnt{%
464     \edef\mandatory@arg{\mandatory@arg\expandafter\genb@x\f@size..\@{}}%
465     \empty@sfcnt}

\genb@x The auxiliary macros \genb@x and \genb@y are used to convert the \f@size into
\genb@y centipoints.
466 \def\genb@x#1.#2.#3\@{\two@digits{#1}\genb@y#200\@{}}
467 \def\genb@y#1#2#3\@{\@{#1#2}}

\s@fct@sub This size function handles font substitution. The mandatory argument is a fam-
ily/series/shape combination, the optional argument (if present) is ignored. The
font encoding scheme cannot be changed. Therefore, the first thing we do is to
prepend the encoding scheme.
468 \DeclareSizeFunction{sub}{\sub@sfcnt\@font@warning}
469 \DeclareSizeFunction{ssub}{\sub@sfcnt\@font@info}

470 \def\sub@sfcnt#1{%
471     \edef\mandatory@arg{\f@encoding/\mandatory@arg}%

Next action is split the arg into its individual components and allow for a late font
shape load.
472     \begingroup
473     \expandafter\split@name\mandatory@arg/\@nil
474     \try@load@fontshape
475     \endgroup

Then we record the current \f@size since it may get clobbered.
476     \let\f@user@size\f@size

Then we check whether this new combination is defined and give an error message
if not. In this case we also switch to \error@fontshape.
477     \expandafter
478     \ifx\csname\mandatory@arg\endcsname\relax
479         \errmessage{No\space declaration\space for\space
480             shape\space \mandatory@arg}%
481         \error@fontshape
482     \else

Otherwise we warn the user about the substitution taking place.
483     #1{Font\space shape\space '\curr@fontshape'\space in\space
484         size\space <\f@size>\space not\space available\MessageBreak
485         Font\space shape\space '\mandatory@arg'\space tried\space
486         instead}%
487     \expandafter\split@name\mandatory@arg/\@nil
488     \fi

Then we restart the font specification scan by calling \get@external@font.
489     \edef\f@size{\f@user@size}%
490     \get@external@font

Finally \do@subst@correction is called to get the font name right.
491     \do@subst@correction
492 }

```

`\s@fct@subf` The `subf` size function allows substitution of another font. The mandatory argument is the external name of the font to be substituted, the optional argument a size scaling factor like in the default size function. The main difference to the default size function is the warning message.

```

493 \DeclareSizeFunction{subf}{\subf@sfcnt\@font@warning}
494 \DeclareSizeFunction{ssubf}{\subf@sfcnt\@font@info}

495 \def\subf@sfcnt#1{%
496     #1{Font\space shape\space '\curr@fontshape'\space in\space
497         size\space \f@size\space not\space available\MessageBreak
498         external\space font\space '\mandatory@arg'\space used}%
499     \empty@sfcnt#1%
500 }
```

`\s@fct@fixed` The `fixed` size function is for using a font at a different size than requested. A warning message is printed, and the external font to be used is taken from the mandatory argument. If an optional argument is present it is used as the ‘at’ size for the font. Otherwise the font is loaded at its design size.

```

501 \DeclareSizeFunction{fixed}{\fixed@sfcnt\@font@warning}
502 \DeclareSizeFunction{sfixed}{\fixed@sfcnt\@font@info}

503 \def\fixed@sfcnt#1{%
504     \ifx\optional@arg\@empty
505         \let\external@font\mandatory@arg
506     \else
507         \edef\external@font{\mandatory@arg\space at\optional@arg pt}%
508     \fi
509     #1{External\space font\space '\external@font'\space loaded\space
510         for\space size\MessageBreak
511         <\f@size>}%
512 }
513 /2ekernel)
```

## File q

# ltfsscmp.dtx

This file contains the implementation of commands giving compatibility with the original ‘NFSS1’ release of the Font Selection Scheme.

**Warning:** The macro documentation is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

Version 1 of NFSS is obsolete now for about 20 years (and was “current” only for a short intermediate time) so with the 2015 release these internal interface commands are removed from the kernel and made available via `latexrelease` package so that backward compatibility remains ensured for very old documents.

```

1 (*latexrelease)
2 \IncludeInRelease{2015/01/01}{\new@fontshape}%
3                               {NFSS version1 commands}%
4 \let\new@fontshape\@undefined
5 \let\warn@rel@i\@undefined
6 \let\scan@fontshape\@undefined
7 \let\scan@@fontshape\@undefined
8 \let\subst@fontshape\@undefined
9 \let\extra@def\@undefined
10 \let\default@mextra\@undefined
11 \let\preload@sizes\@undefined
12 \let\err@rel@i\@undefined
13 \let\newmathalphabet\@undefined
14 \let\newmathalphabet@\@undefined
15 \let\newmathalphabet@@\@undefined
16 \let@if@no@font@opt\@undefined
17 \let@no@font@optfalse\@undefined
18 \let\define@mathalphabet\@undefined
19 \let\define@mathgroup\@undefined
20 \let\addtoversion\@undefined
21 \EndIncludeInRelease

```

In older releases we provide the original definitions.

```

22 \IncludeInRelease{0000/00/00}{\new@fontshape}%
23                               {NFSS version1 commands}%

```

`\new@fontshape` The interface is now `\DeclareFontShape`.

```

24 \gdef\new@fontshape#1#2#3#4{%
25     \warn@rel@i\new@fontshape\DeclareFontShape
26     \expandafter\scan@fontshape\@gobble#4<\@nil><<%
27     \DeclareFontShape U{#1}{#2}{#3}\reserved@f}%
28 \@onlypreamble\new@fontshape

```

`\warn@rel@i` The warning message used above.

```

29 \gdef\warn@rel@i#1#2{%
30     \@font@warning{*** NFSS release 1 command
31                     \noexpand#1found\MessageBreak
32     *** Update by using release 2 command

```

```

33      \string#2.\MessageBreak
34      *** Recovery is probably possible}%
35 }%
36 \@onlypreamble\warn@rel@i

```

\scan@fontshape This will scan the old font shape definition syntax.

```

37 \gdef\scan@fontshape{%
38   \let\reserved@f\@empty
39   \let\reserved@e\@empty %      holds last info
40   \scan@@fontshape
41 }%
42 \@onlypreamble\scan@fontshape

```

\scan@@fontshape

```

43 \gdef\scan@@fontshape#1>#2#3<{%
44   \ifx\@nil#1%
45     \edef\reserved@f{\reserved@f\reserved@e}%
46   \else
47     \def\reserved@b{#1}%      nick names
48     \def\reserved@c{#3}%
49     \in@{ at}{#3}%
50     \ifin@
51       \in@{pt}{#3}% not a proof but a good chance
52     \ifin@

```

We grap also everything after pt and discard it if people have forgotten to place a percent sign there.

```

53     \def\reserved@a##1 at##2pt##3\@nil{%
54       \def\reserved@b{##2}%
55       \def\reserved@c{##1}%
56     }%
57     \reserved@a#3\@nil
58   \fi
59 \fi
60 \ifnum 0<0#2
61   \edef\reserved@d{subf*\reserved@c}%
62   \ifcase #2\or
63     \or
64   \else
65     \errmessage{*** What's this? NFSS release 0? ***}%
66   \fi
67 \else
68   \edef\reserved@d{#2\reserved@c}%
69 \fi
70 \ifx\reserved@d\reserved@e
71   \edef\reserved@f{\reserved@f<\reserved@b>}%
72 \else
73   \edef\reserved@f{\reserved@f\reserved@e<\reserved@b>}%add old info
74   \let\reserved@e\reserved@d
75 \fi
76 \expandafter\scan@@fontshape
77 \fi
78 }%
79 \@onlypreamble\scan@@fontshape

```

`\subst@fontshape` This is now also handled by the extend syntax of `\DeclareFontShape`.

```

80 \gdef\subst@fontshape#1#2#3#4#5#6{%
81     \warn@rel@i\subst@fontshape\DeclareFontShape
82     \DeclareFontShape{U}{#1}{#2}{#3}{<->sub*#4/#5/#6}{}}%
83 \@onlypreamble\subst@fontshape

```

`\extra@def` This was replaced by `\DeclareFontFamily`.

```

84 \gdef\extra@def#1#2#3{%
85     \warn@rel@i\extra@def\DeclareFontFamily
86     \DeclareFontFamily{U}{#1}{}%
87 }%
88 \@onlypreamble\extra@def

```

`\default@mextra` The new name is `\DeclareFontEncodingDefaults` but in this case we don't feel comfortable with this either.

```

89 \gdef\default@mextra{%
90     \warn@rel@i\default@mextra\DeclareFontEncodingDefaults

```

We pick up the argument to `\default@mextra` implicitly as the second argument of `\DeclareFontEncodingDefaults`.

```

91     \DeclareFontEncodingDefaults\relax
92 }%
93 \@onlypreamble\default@mextra

```

`\preload@sizes` The new interface is `\DeclarePreloadSizes`.

```

94 \gdef\preload@sizes{%
95     \warn@rel@i\preload@sizes\DeclarePreloadSizes
96     \DeclarePreloadSizes U%
97 }%
98 \@onlypreamble\preload@sizes

```

`\err@rel@i` This macro is used in cases where emulation with NFSS2 features is not really possible.

```

99 \gdef\err@rel@i#1#2{%
100     \@latex@error{*** NFSS release 1 command \noexpand#1found%
101         ^^J*** Recovery not possible. Use \string#2}%
102     {The new release of NFSS doesn't support the
103         \noexpand#1command^^Jany longer.
104         Please upgrade your file to the syntax of NFSS
105         release 2^^Jusing the \noexpand#2command.}%

```

Let's die.

```

106     \batchmode\input.\relax
107 }%
108 \@onlypreamble\err@rel@i

```

`\newmathalphabet` `\newmathalphabet` is the old form.

`\newmathalphabet@`

`\newmathalphabet@@`

```

109 \gdef\newmathalphabet{%
110     \if@no@font@opt
111         \@latex@error{*** NFSS release 1 command
112             \noexpand\newmathalphabet found%
113             ^^J \space*** Automatic recovery not possible.%
114             ^^J \space*** TYPE H for Help%
115             }%

```

```

116         {Please look at the file usrguide.tex for hints on
117         how to resolve this problem.}%
118     \else
119         \warn@rel@i\newmathalphabet\DeclareMathAlphabet
120     \fi
121     \@ifstar\newmathalphabet@@@
122         \newmathalphabet@@}%
123 \gdef\newmathalphabet@@#1{\DeclareMathAlphabet#1{U}{-}{-}}%
124 \gdef\newmathalphabet@@@#1#2#3#4{%
125     \DeclareMathAlphabet{#1}{U}{#2}{#3}{#4}}%
126 \@onlypreamble\newmathalphabet
127 \@onlypreamble\newmathalphabet@@
128 \@onlypreamble\newmathalphabet@@@

\if@no@font@opt
\@no@font@optfalse 129 \global\let\if@no@font@opt\iftrue
130 \gdef\@no@font@optfalse{\let\if@no@font@opt\iffalse}%

\define@mathalphabet This is a case where dying is best.
131 \gdef\define@mathalphabet{%
132     \err@rel@i\define@mathalphabet\DeclareMathAlphabet
133 }%
134 \@onlypreamble\define@mathalphabet

\define@mathgroup And here is another one
135 \gdef\define@mathgroup{%
136     \err@rel@i\define@mathgroup\DeclareSymbolFont
137 }%
138 \@onlypreamble\define@mathgroup

\addtoversion \addtoversion is the old form.
139 \def\addtoversion#1#2{%
140     \warn@rel@i\addtoversion\SetMathAlphabet
141     \SetMathAlphabet#2{#1}{U}}%
142 \@onlypreamble\addtoversion

    Finishing off this huge \IncludeInRelease argument:
143 \EndIncludeInRelease
144 </latexrelease>

```

## File r

# ltxssdcl.dtx

This file contains the main implementation of the font selection scheme commands. See other parts of the L<sup>A</sup>T<sub>E</sub>X distribution, or *The L<sup>A</sup>T<sub>E</sub>X Companion* for higher level documentation of these commands.

**Warning:** The macro documentation is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

## 33 Interface Commands

`\in@` `\in` is a utility macro with two arguments. It determines whether its first argument occurs in its second and sets the switch `\ifin@` accordingly. The first argument may not contain braces nor # (more precisely, tokens of category code 1, 2, or 6).

```

1 (*2kernel)
2 \def\in@#1#2%
3 {%
4   \begingroup
5     \def\in@@##1#1{%
6       \toks@{\expandafter{\in@@#2}{}}#1}%
7     \edef\in@@{\the\toks@}%
8     \expandafter\endgroup
9     \ifx\in@@\@empty
10      \in@false
11    \else
12      \in@true
13    \fi
14  }
15 \newif\ifin@

```

Before the `\begin{document}` command several *math versions* and *math alphabet identifiers* may be declared. In principle, there should be exactly one family/series/shape combination be declared for each version/alphabet pair. But we want to allow for defaults as well for automagical filling of holes.

While building the tables for math alphabet identifiers and math versions we keep several lists:

- the list of all math versions, `\version@list`, each entry prefixed by the control sequence `\version@elt`, i.e. this list has the following form

$$\backslash\text{version@elt}\langle\text{version}_1\rangle\backslash\text{version@elt}\langle\text{version}_2\rangle\ldots\backslash\text{version@elt}\langle\text{version}_n\rangle$$

- the list of all math alphabet identifiers. Here every entry has the form:

$$\backslash\text{group@elt}\langle\text{math group number}\rangle\{\{\langle\text{default family}\rangle\}\{\langle\text{default series}\rangle\}\{\langle\text{default shape}\rangle\}\}.$$

- Each defined math alphabet identifier holds a list containing information about the *versions* for which it is defined. This list has a more complicated structure: it looks as follows:

```

\set@alpha<the alphabet identifier itself>
\reserved@c<math version><font info>
...
\@nil

```

where *<font info>* is either `\reserved@e` (if the combination is not defined yet) or

```
{\family}{\series}{\shape}}
```

`\version@list` We initialize the version list to be empty.

```

16 \let\version@list=\@empty
17 \@onlypreamble\version@list

```

`\version@elt`

```

18 \let\version@elt\relax
19 \@onlypreamble\version@elt

```

`\new@mathversion` The macro `\new@mathversion` is called with the version control sequence as its argument.

```
20 %\def\new@mathversion#1{%
```

The first thing this macro does is to check if the version identifier is already present in `\version@list`. We enclose `\version@list` in braces since it might be empty (if no *version* is defined yet). But this means that we need a suitable number of `\expandafter` primitives.

```

21 % \expandafter\in@\expandafter#1\expandafter{\version@list}%
22 % \ifin@

```

If so it prints an error message. The `\next` macro is used to get rid of the four characters `\mv@` that would otherwise appear at the begin of the version name in the error message.

```

23 % \latex@error{Math version
24 % \expandafter\@gobblefour\string#1'
25 % already defined}\@eha

```

Otherwise we have a new version, and we can proceed with entering it into the tables. We add it to `\version@list`. This is very easy: we define `\version@elt` (which is the delimiter in `\version@list`) to protect itself and the following token from being expanded and simply redefine `\version@list`.

```

26 % \else
27 % \global\expandafter\newcount\c@\expandafter
28 % \gobble\string#1\endcsname
29 % \global\c@\expandafter
30 % \gobble\string#1\endcsname\@ne
31 % \def\version@elt{\noexpand\version@elt\noexpand}%
32 % \edef\version@list{\version@list\version@elt#1}%

```

Then we prepare to enter the new version into all math alphabet identifier lists. Remember that these lists use `\reserved@c` as delimiter, and that there appears the control sequence `\reserved@e` that must not be expanded. Therefore we take suitable precautions.

```
33 %      \def\reserved@c{\noexpand\reserved@c\noexpand}%
34 %      \let\reserved@e\relax
```

We will now go through the `\alpha@list` to process every  $\langle\textit{math alphabet identifier}\rangle$  in turn. Since this list has `\group@elt` as a delimiter we define this control sequence. It has three arguments as every entry consists of three items (as explained above).

```
35 %      \def\group@elt##1##2##3{%
```

The first of these arguments is the  $\langle\textit{math alphabet identifier}\rangle$ . We redefine it by appending the information about the new version at the end of the list contained in it. However, there is one subtlety: the definitions for `\reserved@c` and `\reserved@e` made above prevent the main part of the list from being expanded. But we still have to take care of the header and the trailer. To do this we remove the trailer by means of the macro `\remove@nil` which also protect the header from being expanded. Its definition is given below. Now we can prepare to add the new version.

```
36 %          \edef##1{\expandafter\remove@nil##1%
37 %              \reserved@c
38 %              #1%
39 %              \reserved@e
40 %              \noexpand\@nil}}%
```

Finally we call `\alpha@list` which will now execute the macro `\group@elt` once for every defined  $\langle\textit{math alphabet identifier}\rangle$ . And that's all for now.

```
41 %      \alpha@list
42 %      \fi}
```

`\alpha@list` As we explained above every entry in `\alpha@list` has the form

```
\alpha@elt
\alphabet identifier\internal group number\default font assignments}...
```

We initialize it to `\@empty`.

```
43 \let\alpha@list\@empty
44 \@onlypreamble\alpha@list
```

`\alpha@elt`

```
45 \let\alpha@elt\relax
46 \@onlypreamble\alpha@elt
```

`\newgroup` Start the group (fam) allocation at 0. (Doesn't belong here.)

```
47 \count18=-1
```

`\stepcounter`

`\select@group` We surround `\select@group` with braces so that functions using it can be used directly after `_` or `^`. However, if we use oldstyle syntax where the math alphabet doesn't have arguments (ie if `\math@bgroup` is not `\bgroup`) we need to get rid of the extra group.

```

48 </2ekernel>
49 <latexrelease>\IncludeInRelease{2015/01/01}
50 <latexrelease>          {\select@group}{\select@group}%
51 <*2ekernel | latexrelease>
52 \def\select@group#1#2#3#4{%
53   \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
54   {%
55     \ifmmode
56       \ifnum\csname c@mv@\math@version\endcsname<\e@mathgroup@top
57         \begingroup
58           \escapechar\m@ne
59           \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
60           \globaldefs\@ne \math@fonts
61         \endgroup
62         \init@restore@version
63         \xdef#1{\noexpand\use@mathgroup\noexpand#2%
64           {\number\csname c@mv@\math@version\endcsname}}%
65         \global\advance\csname c@mv@\math@version\endcsname\@ne
66       \else
67         \let#1\relax
68         \@latex@error{Too many math alphabets used in
69           version \math@version}%
70         \@eha
71       \fi
72     \else \expandafter\non@alpherr\fi
73     #1{#4}%
74   }%
75 }
76 </2ekernel | latexrelease>
77 <latexrelease>\EndIncludeInRelease
78 <latexrelease>\IncludeInRelease{0000/00/00}
79 <latexrelease>          {\select@group}{\select@group}%
80 <latexrelease>\def\select@group#1#2#3#4{%
81 <latexrelease> \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
82 <latexrelease> {%
83 <latexrelease> \ifmmode
84 <latexrelease>   \ifnum\csname c@mv@\math@version\endcsname<\sixt@n
85 <latexrelease>     \begingroup
86 <latexrelease>       \escapechar\m@ne
87 <latexrelease>       \getanddefine@fonts
88 <latexrelease>         {\csname c@mv@\math@version\endcsname}#3%
89 <latexrelease>       \globaldefs\@ne \math@fonts
90 <latexrelease>     \endgroup
91 <latexrelease>     \init@restore@version
92 <latexrelease>     \xdef#1{\noexpand\use@mathgroup\noexpand#2%
93 <latexrelease>       {\number\csname c@mv@\math@version\endcsname}}%
94 <latexrelease>     \global\advance\csname c@mv@\math@version\endcsname\@ne
95 <latexrelease>   \else
96 <latexrelease>     \let#1\relax
97 <latexrelease>     \@latex@error{Too many math alphabets used in
98 <latexrelease>       version \math@version}%
99 <latexrelease>     \@eha
100 <latexrelease>   \fi
101 <latexrelease> \else \expandafter\non@alpherr\fi

```

```

102 <latexrelease> #1{#4}%
103 <latexrelease> }%
104 <latexrelease>}
105 <latexrelease>\EndIncludeInRelease
106 <*2ekernel>

107 \@onlypreamble\restore@mathversion

\init@restore@version
108 \def\init@restore@version{%
109     \global\let\init@restore@version\relax
110     \xdef\restore@mathversion
111         {\expandafter\noexpand\csname mv@\math@version\endcsname
112         \global\csname c@mv@\math@version\endcsname
113         \number\csname c@mv@\math@version\endcsname\relax}%
114     \aftergroup\dorestore@version
115 }
116 \@onlypreamble\init@restore@version

\non@alpherr
117 \gdef\non@alpherr#1{\@latex@error{%
The command here will have a space at the end of its name, so we make sure not
to insert an extra one.
118     \string#1allowed only in math mode}\@ehd}

\dorestore@version
119 \def\dorestore@version
120 {\ifmmode
121     \aftergroup\dorestore@version
122 \else
123     \gdef\init@restore@version{%
124         \global\let\init@restore@version\relax
125         \xdef\restore@mathversion
126             {\expandafter\noexpand\csname mv@\math@version\endcsname
127             \global\csname c@mv@\math@version\endcsname
128             \number\csname c@mv@\math@version\endcsname\relax}%
129         \aftergroup\dorestore@version
130     }%
131     \begingroup
132         \let\getanddefine@fonts\@gobbletwo
133         \restore@mathversion
134     \endgroup
135     \fi}%
136 \@onlypreamble\dorestore@version

\document@select@group We surround \select@group with braces so that functions using it can be used
directly after _ or ^.
137 </2ekernel>
138 <latexrelease>\IncludeInRelease{2015/01/01}
139 <latexrelease> {\document@select@group}{\document@select@group}%
140 <*2ekernel | latexrelease>
141 \def\document@select@group#1#2#3#4{%
142     \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi

```

```

143 {%
144 \ifmmode
145 \ifnum\csname c@mv@\math@version\endcsname<\e@mathgroup@top
146 \begingroup
147 \escapechar\m@ne
148 \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
149 \globaldefs\@ne \math@fonts
150 \endgroup
151 \expandafter\extract@alph@from@version
152 \csname mv@\math@version\expandafter\endcsname
153 \expandafter{\number\csname
154 c@mv@\math@version\endcsname}%
155 #1%
156 \global\advance\csname c@mv@\math@version\endcsname\@ne
157 \else
158 \let#1\relax
159 \@latex@error{Too many math alphabets used
160 in version \math@version}%
161 \@eha
162 \fi
163 \else \expandafter\non@alpherr\fi
164 #1{#4}%
165 }%
166 }
167 </2ekernel | latexrelease>
168 <latexrelease>\EndIncludeInRelease
169 <latexrelease>\IncludeInRelease{0000/00/00}
170 <latexrelease> {\document@select@group}{\document@select@group}%
171 <latexrelease>\def\document@select@group#1#2#3#4{%
172 <latexrelease> \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
173 <latexrelease> {%
174 <latexrelease> \ifmmode
175 <latexrelease> \ifnum\csname c@mv@\math@version\endcsname<\sist@n
176 <latexrelease> \begingroup
177 <latexrelease> \escapechar\m@ne
178 <latexrelease> \getanddefine@fonts
179 <latexrelease> {\csname c@mv@\math@version\endcsname}#3%
180 <latexrelease> \globaldefs\@ne \math@fonts
181 <latexrelease> \endgroup
182 <latexrelease> \expandafter\extract@alph@from@version
183 <latexrelease> \csname mv@\math@version\expandafter\endcsname
184 <latexrelease> \expandafter{\number\csname
185 <latexrelease> c@mv@\math@version\endcsname}%
186 <latexrelease> #1%
187 <latexrelease> \global\advance\csname c@mv@\math@version\endcsname\@ne
188 <latexrelease> \else
189 <latexrelease> \let#1\relax
190 <latexrelease> \@latex@error{Too many math alphabets used
191 <latexrelease> in version \math@version}%
192 <latexrelease> \@eha
193 <latexrelease> \fi
194 <latexrelease> \else \expandafter\non@alpherr\fi
195 <latexrelease> #1{#4}%
196 <latexrelease> }%

```

```

197 \latexrelease}
198 \latexrelease\EndIncludeInRelease
199 \*2ekernel)

```

`\process@table`

```

200 \def\process@table{%
201   \def\cdp@elt##1##2##3##4{%
202     \@font@info{Checking defaults for
203       ##1/##2/##3/##4}%
204     \expandafter
205     \ifx\csname##1/##2/##3/##4\endcsname\relax

```

Grouping is important for two reasons, first `\cdp@elt` will get redefined if `\Declare...` functions are executed within the external `.fd` file and secondly `\try@load@fontshape` changes a lot of catcodes without surrounding itself with a group.

```

206       \begingroup
207       \def\f@encoding{##1}\def\f@family{##2}%
208       \try@load@fontshape
209       \endgroup
210     \fi
211     \expandafter
212     \ifx\csname##1/##2/##3/##4\endcsname\relax
213       \@latex@error{This NFSS system isn't set up properly}%
214       {For encoding scheme ##1 the defaults
215         ##2/##3/##4 do not form a valid font shape}%
216     \else
217       \@font@info{... okay}%
218     \fi}%
219 \cdp@list

```

Now we make sure that `\error@fontshape` is okay.

```

220 \begingroup
221 \escapechar\m@ne
222 \error@fontshape
223 \expandafter\ifx\csname \curr@fontshape\endcsname\relax
224   \begingroup
225   \try@load@fontshape
226   \endgroup
227 \fi
228 \expandafter\ifx\csname \curr@fontshape\endcsname\relax
229   \@latex@error{This NFSS system isn't set up properly}%
230   {The system maintainer forgot to specify a suitable
231     substitution
232     font shape using the \noexpand\DeclareErrorFont
233     command}%
234 \fi
235 \endgroup

```

Set `\select@group` to its meaning used within the document body.

```

236 \let\select@group\document@select@group

```

Install the default font attributes they are currently pointing to error font shape. Don't use `\reset@font` since that would trigger `\selectfont`.

```

237 \fontencoding{\encodingdefault}%

```

```

238 \fontfamily{\familydefault}%
239 \fontseries{\seriesdefault}%
240 \fontshape{\shapedefault}%
kill all macros not longer needed. we need to add many more!!!!!!
241 \everyjob{}%
242 }
243 \@onlypreamble\process@table
244 %\@onlypreamble\set@mathradical

```

`\DeclareMathVersion`

```

245 \def\DeclareMathVersion#1{%
246 \expandafter\new@mathversion\csname mv@#1\endcsname}
247 \@onlypreamble\DeclareMathVersion

```

`\new@mathversion`

```

248 \def\new@mathversion#1{%
249 \expandafter\in@\expandafter#1\expandafter{\version@list}%
250 \ifin@
251 \font@info{Redefining math version
252 \expandafter\@gobblefour\string#1'}%
253 \else
254 \expandafter\newcount\csname c@\expandafter
255 \expandafter\@gobble\string#1\endcsname
256 \def\version@elt{\noexpand\version@elt\noexpand}%
257 \edef\version@list{\version@list\version@elt#1}%
258 \fi

```

`\toks@` is used to gather all tokens for the math version. `\count@` will be used to count the math groups we add to this version.

```

259 \toks@{}%
260 \count@\z@

```

Now we loop over `\group@list` to add all math groups defined so far to the version and at the same time to count them.

```

261 \def\group@elt##1##2{%
262 \advance\count@\@ne
263 \addto@hook\toks@{\getanddefine@fonts##1##2}%
264 }%
265 \group@list

```

We set the counter for this math version to the number of math groups found in `\group@list`.

```

266 \global\csname c@\expandafter\@gobble\string#1\endcsname\count@

```

Now we loop over `\alpha@list` to add all math alphabets known so far. We have to distinguish the case that an alphabet by default should produce an error in new versions.

```

267 \def\alpha@elt##1##2##3{%
268 \ifx##2\no@alphabet@error
269 \toks@\expandafter{\the\toks@\install@mathalphabet##1%
270 {\no@alphabet@error##1}}%
271 \else
272 \toks@\expandafter{\the\toks@\install@mathalphabet##1%
273 {\select@group##1##2##3}}%

```

```

274     \fi
275   }%
276   \alpha@list

```

Finally we define the math version to expand to the contents of \toks@.

```

277   \xdef#1{\the\toks@}%
278 }
279 \@onlypreamble\new@mathversion

```

\DeclareSymbolFont

```

280 \def\DeclareSymbolFont#1#2#3#4#5{%
281   \@tempswafalse
282   \edef\reserved@b{#2}%
283   \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
284     \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
285   \cdp@list
286   \if@tempswa
287     \ifundefined{sym#1}{%
288       \ifnum\count18<15 %
289         \expandafter\new@mathgroup\csname sym#1\endcsname
290         \expandafter\new@symbolfont\csname sym#1\endcsname
291           {#2}{#3}{#4}{#5}%
292       \else
293         \@latex@error{Too many symbol fonts declared}\@eha
294       \fi
295     }%
296     {%
297       \@font@info{Redeclaring symbol font ‘#1’}%

```

Update the group list.

```

298     \def\group@elt##1##2{%
299       \noexpand\group@elt\noexpand##1%
300       \expandafter\ifx\csname sym#1\endcsname##1%
301         \expandafter\noexpand\csname#2/#3/#4/#5\endcsname
302       \else
303         \noexpand##2%
304       \fi}%
305     \xdef\group@list{\group@list}%

```

Update the version list.

```

306     \def\version@elt##1{%
307       \expandafter
308       \SetSymbolFont@expandafter##1\csname#2/#3/#4/#5\endcsname
309       \endcsname \csname sym#1\endcsname
310     }%
311     \version@list
312   }%
313   \else
314     \@latex@error{Encoding scheme ‘#2’ unknown}\@eha
315   \fi
316 }
317 \@onlypreamble\DeclareSymbolFont

```

\group@list

```

318 \let\group@list\@empty
319 \@onlypreamble\group@list

\group@elt
320 \let\group@elt\relax
321 \@onlypreamble\group@elt

\new@symbolfont
322 \def\new@symbolfont#1#2#3#4#5{%
323   \toks@\expandafter{\group@list}%
324   \edef\group@list{\the\toks@\noexpand\group@elt\noexpand#1%
325     \expandafter\noexpand\csname#2/#3/#4/#5\endcsname}%
326   \def\version@elt##1{\toks@\expandafter{##1}%
327     \edef##1{\the\toks@\noexpand\getanddefine@fonts
328       #1\expandafter\noexpand\csname#2/#3/#4/#5\endcsname}%
329     \global\advance\csname c@\expandafter
330       \@gobble\string##1\endcsname\@ne
331   }%
332   \version@list
333 }
334 \@onlypreamble\new@symbolfont

\SetSymbolFont
335 \def\SetSymbolFont#1#2#3#4#5#6{%
336   \@tempswafalse
337   \edef\reserved@b{#3}%
338   \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
339     \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
340   \cdp@list
341   \if@tempswa
342     \expandafter\SetSymbolFont@
343     \csname mv@#2\expandafter\endcsname\csname#3/#4/#5/#6\expandafter
344       \endcsname \csname sym#1\endcsname
345   \else
346     \@latex@error{Encoding scheme ‘#3’ unknown}\@eha
347   \fi
348 }
349 \@onlypreamble\SetSymbolFont

\SetSymbolFont@
350 \def\SetSymbolFont@#1#2#3{%
351   \expandafter\in@\expandafter#1\expandafter{\version@list}%
352   \ifin@
353     \expandafter\in@\expandafter#3\expandafter{\group@list}%
354     \ifin@
355       \begingroup
356       \expandafter\get@cdp\string#2\@nil\reserved@a
357       \toks@{}%
358       \def\install@mathalphabet##1##2{%
359         \addto@hook\toks@{\install@mathalphabet##1{##2}}%
360       }%
361       \def\getanddefine@fonts##1##2{%
362         \ifnum##1=#3%
363           \addto@hook\toks@{\getanddefine@fonts#3#2}%

```

```

364         \expandafter\get@cdp\string##2\@nil\reserved@b
365         \ifx\reserved@a\reserved@b\else
366             \@font@info{Encoding '\reserved@b' has changed
367                 to '\reserved@a' for symbol font\MessageBreak
368                 '\expandafter\@gobblefour\string#3' in the
369                 math version '\expandafter
370                 \@gobblefour\string#1'}}%
371         \fi
372         \@font@info{%
373             Overwriting symbol font
374             '\expandafter\@gobblefour\string#3' in
375             version '\expandafter
376             \@gobblefour\string#1'\MessageBreak
377             \@spaces \expandafter\@gobble\string##2 -->
378                 \expandafter\@gobble\string#2}%
379         \else
380             \addto@hook\toks@{\getanddefine@fonts##1##2}%
381             \fi}%
382         #1%
383         \xdef#1{\the\toks@}%
384     \endgroup
385 \else
386     \@latex@error{Symbol font '\expandafter\@gobblefour\string#3'
387         not defined}\@eha
388 \fi
389 \else
390     \@latex@error{Math version '\expandafter\@gobblefour\string#1'
391         is not
392         defined}{You probably misspelled the name of the math
393         version.^^JOr you have to specify an additional package.}%
394 \fi
395 }
396 \@onlypreamble\SetSymbolFont@

```

\get@cdp

```

397 \def\get@cdp#1#2/#3\@nil#4{\def#4{#2}}
398 \@onlypreamble\get@cdp

```

\DeclareMathAlphabet

```

399 \def\DeclareMathAlphabet#1#2#3#4#5{%
400     \@tempswafalse
401     \edef\reserved@b{#2}%
402     \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
403         \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
404     \cdp@list
405     \if@tempswa
406         \expandafter\ifx
407             \csname\expandafter\@gobble\string#1\endcsname
408             \relax
409             \new@mathalphabet#1{#2}{#3}{#4}{#5}%
410         \else
411             Check if it is already a math alphabet.
412             \edef\reserved@a{\noexpand\in@\string\select@group}%

```

```

412         {\expandafter\meaning\csname \expandafter
413         \@gobble\string#1\space\endcsname}}%
414     \reserved@a
415     \ifin@
416         \@font@info{Redeclaring math alphabet \string#1}%
417         \def\version@elt##1{%
418             \expandafter\SetMathAlphabet@\expandafter
419             ##1\csname#2/#3/#4/#5\expandafter\endcsname
420
421             \csname M@#2\expandafter\endcsname
422             \csname \expandafter\@gobble\string#1\space\endcsname#1}%
423         \version@list
424     \else

```

Check if it is a math alphabet defined via `\DeclareSymbolFontAlphabet`.

```

424     \edef\reserved@a{\noexpand\in@{\string\use@mathgroup}%
425     {\expandafter\meaning\csname \expandafter
426     \@gobble\string#1\space\endcsname}}%
427     \reserved@a
428     \ifin@

```

In that case overwriting is simple since there is nothing inserted in the math version macros.

```

429         \@font@info{Redeclaring math alphabet \string#1}%
430         \new@mathalphabet#1{#2}{#3}{#4}{#5}%

```

Otherwise panic.

```

431     \else
432         \@latex@error{Command ‘\string#1’ already defined}\@eha
433     \fi
434 \fi
435 \fi
436 \else
437     \@latex@error{Encoding scheme ‘#2’ unknown}\@eha
438 \fi
439 }
440 \@onlypreamble\DeclareMathAlphabet

```

`\new@mathalphabet`

```

441 \def\new@mathalphabet#1#2#3#4#5{%
442     \toks@{\expandafter{\alpha@list}%
443     \edef#1{\expandafter\noexpand\csname \expandafter
444         \@gobble\string#1\space\endcsname
445         \if/#5/%
446             \noexpand\no@alphabet@error
447             \noexpand\no@alphabet@error
448         \else
449             \expandafter\noexpand\csname M@#2\endcsname
450             \expandafter\noexpand\csname#2/#3/#4/#5\endcsname
451         \fi
452     }%
453     \toks2\expandafter{#1}%
454     \edef\alpha@list{\the\toks@\noexpand\alpha@elt\the\toks2}%
455     \def\version@elt##1{\toks@\expandafter{##1}%
456         \edef##1{\the\toks@\install@mathalphabet

```

```

457 \expandafter\noexpand
458 \csname \expandafter\@gobble
459 \string#1\space\endcsname
460 {\if/#5/%
461 \noexpand\no@alphabet@error
462 \noexpand#1%
463 \else
464 \noexpand\select@group\the\toks2
465 \fi}}%
466 }%
467 \version@list
468 \expandafter\edef\csname \expandafter\@gobble
469 \string#1\space\endcsname{\if/#5/%
470 \noexpand\no@alphabet@error
471 \noexpand#1%
472 \else
473 \noexpand\select@group\the\toks2
474 \fi}%
475 \edef#1{\noexpand\protect
476 \expandafter\noexpand\csname \expandafter
477 \@gobble\string#1\space\endcsname}%
478 }
479 \@onlypreamble\new@mathalphabet

```

\SetMathAlphabet

```

480 \def\SetMathAlphabet#1#2#3#4#5#6{%
481 \@tempswafalse
482 \edef\reserved@b{#3}%
483 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
484 \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
485 \cdp@list
486 \if@tempswa
487 \expandafter\SetMathAlphabet@
488 \csname mv@#2\expandafter\endcsname\csname#3/#4/#5/#6\expandafter
489 \endcsname \csname M@#3\expandafter\endcsname
490 \csname \expandafter\@gobble\string#1\space\endcsname#1%
491 \else
492 \@latex@error{Encoding scheme ‘#3’ unknown}\@eha
493 \fi
494 }
495 \@onlypreamble\SetMathAlphabet

```

\SetMathAlphabet@

```

496 \def\SetMathAlphabet@#1#2#3#4#5{%
497 \expandafter\in@\expandafter#1\expandafter{\version@list}%
498 \ifin@
499 \expandafter\in@\expandafter#4\expandafter{\alpha@list}%
500 \ifin@
501 \begingroup
502 \toks@{}%
503 \def\getanddefine@fonts##1##2{%
504 \addto@hook\toks@{\getanddefine@fonts##1##2}%
505 }%
506 \def\reserved@c##1##2##3##4{% % for message below

```

```

507         \expandafter\@gobble\string##4}%
508 \def\install@mathalphabet##1##2{%
509     \ifx##1#4%
510         \addto@hook\toks@
511         {\install@mathalphabet#4{\select@group#4#3#2}}%
512         \@font@info{Overwriting math alphabet
513         '\string#5' in version '\expandafter
514         \@gobblefour\string#1'\MessageBreak
515         \@spaces \reserved@c##2 -->
516         \expandafter\@gobble\string#2}%
517     \else
518         \addto@hook\toks@{\install@mathalphabet##1{##2}}%
519     \fi
520 }%
521 #1%
522 \xdef#1{\the\toks@}%
523 \endgroup
524 \else

```

If the math alphabet was defined via `\DeclareSymbolFontAlphabet` we have remove its external definition and add it as a normal math alphabet to every version before trying to change it in one version.

```

525     \edef\reserved@a{%
526         \noexpand\in@{\string\use@mathgroup}{\meaning#4}}%
527 \reserved@a
528 \ifin@
529     \def\reserved@b##1\use@mathgroup##2##3{%
530         \def\reserved@b{##3}\def\reserved@c{##2}}%
531     \expandafter\reserved@b#4%
532     \begingroup
533     \def\install@mathalphabet##1##2{%
534         \addto@hook\toks@{\install@mathalphabet##1{##2}}%
535     }%
536     \def\getanddefine@fonts##1##2{%
537         \addto@hook\toks@{\getanddefine@fonts##1##2}%
538         \ifnum##1=\reserved@b
539             \expandafter
540             \addto@hook\expandafter\toks@
541             \expandafter{\expandafter\install@mathalphabet
542             \expandafter#4\expandafter
543             {\expandafter\select@group\expandafter
544             #4\reserved@c##2}}}%
545         \fi
546     }%
547     \def\version@elt##1{%
548         \toks@{}%
549         ##1%
550         \xdef##1{\the\toks@}%
551     }%
552     \version@list
553 \endgroup

```

Put it into the `\alpha@list` with default ‘error’

```

554     \expandafter\gdef\expandafter\alpha@list\expandafter
555     {\alpha@list

```

```

556         \alpha@elt #4\no@alphabet@error \no@alphabet@error}%
557         \gdef#4{\no@alphabet@error #5}% fake things :-)

```

Then call the internal setting routine again:

```

558         \SetMathAlphabet@{#1}{#2}{#3}#4#5%
559     \else
560         \@latex@error{Command '\string#5' not defined as a
561             math alphabet}%
562         {Use \noexpand\DeclareMathAlphabet to define it.}%
563     \fi
564 \fi
565 \else
566     \@latex@error{Math version '\expandafter\@gobblefour\string#1'
567         is not
568         defined}{You probably misspelled the name of the math
569         version.^^JOr you have to specify an additional package.}%
570 \fi
571 }
572 \@onlypreamble\SetMathAlphabet@

```

`\DeclareMathAlphabet` could do with more checks like allowing single number in #4 lowercase in #4 etc

```

573 \def\DeclareMathAccent#1#2#3#4{%
574     \expandafter\in@\csname sym#3\expandafter\endcsname
575     \expandafter{\group@list}%
576     \ifin@
577         \begin@group
578         \count\z@=#4\relax
579         \count\tw@\count\z@
580         \divide\count\z@\sist@@n
581         \count@\count\z@
582         \multiply\count@\sist@@n
583         \advance\count\tw@-\count@
584         \if\relax\noexpand#1% is command?
585             \edef\reserved@a{\noexpand\in@{\string\mathaccent}{\meaning#1}}%
586             \reserved@a
587         \ifin@
588             \expandafter\set@mathaccent
589             \csname sym#3\endcsname#1#2%
590             {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
591             \@font@info{Redeclaring math accent \string#1}%
592         \else
593             \expandafter\ifx
594             \csname\expandafter\@gobble\string#1\endcsname
595             \relax
596                 \expandafter\set@mathaccent
597                 \csname sym#3\endcsname#1#2%
598                 {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
599             \else
600                 \@latex@error{Command '\string#1' already defined}\@eha
601             \fi
602         \fi
603     \else
604         \@latex@error{Not a command name: '\noexpand#1'}\@eha
605     \fi

```

```

606     \endgroup
607   \else
608     \@latex@error{Symbol font ‘#3’ is not defined}\@eha
609   \fi
610 }
611 \@onlypreamble\DeclareMathAccent

\set@mathaccent

612 \def\set@mathaccent#1#2#3#4{%
613   \xdef#2{\mathaccent"\mathchar@type#3\hexnumber@#1#4\relax}}
614 \@onlypreamble\set@mathaccent

\DeclareMathSymbol

615 \def\DeclareMathSymbol#1#2#3#4{%
616   \expandafter\in@\csname sym#3\expandafter\endcsname
617   \expandafter{\group@list}%
618   \ifin@
619     \begingroup
620       \count\z@=#4\relax
621       \count\tw@\count\z@
622       \divide\count\z@\sist@@n
623       \count@\count\z@
624       \multiply\count@\sist@@n
625       \advance\count\tw@-\count@
626       \if\relax\noexpand#1% is command?
627         \edef\reserved@a{\noexpand\in@{\string\mathchar}{\meaning#1}}%
628         \reserved@a
629       \ifin@
630         \expandafter\set@mathsymbol
631           \csname sym#3\endcsname#1#2%
632           {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
633           \@font@info{Redeclaring math symbol \string#1}%
634       \else
635         \expandafter\ifx
636           \csname\expandafter\@gobble\string#1\endcsname
637           \relax
638         \expandafter\set@mathsymbol
639           \csname sym#3\endcsname#1#2%
640           {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
641       \else
642         \@latex@error{Command ‘\string#1’ already defined}\@eha
643       \fi
644     \fi
645   \else
646     \expandafter\set@mathchar
647       \csname sym#3\endcsname#1#2
648       {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
649     \fi
650   \endgroup
651   \else
652     \@latex@error{Symbol font ‘#3’ is not defined}\@eha
653   \fi
654 }
655 \@onlypreamble\DeclareMathSymbol

```

`\set@mathchar`

```
656 \def\set@mathchar#1#2#3#4{%
657   \global\mathcode'#2="\mathchar@type#3\hexnumber@#1#4\relax}
658 \@onlypreamble\set@mathchar
```

`\set@mathsymbol`

```
659 \def\set@mathsymbol#1#2#3#4{%
660   \global\mathchardef#2="\mathchar@type#3\hexnumber@#1#4\relax}
661 \@onlypreamble\set@mathsymbol

662 %\def\mathsymbol#1#2#3{%
663 %   \@tempcnta=#3\relax
664 %   \@tempcntb\@tempcnta
665 %   \divide\@tempcnta\sixt@@n
666 %   \count@\@tempcnta
667 %   \multiply\count@\sixt@@n
668 %   \advance\@tempcntb-\count@
669 %   \mathchar"\mathchar@type#1\hexnumber@#2%
670 %       \hexnumber@\@tempcnta\hexnumber@\@tempcntb\relax}
671 %
672 %\def\DeclareMathAlphabetCharacter#1#2#3{%
673 %   \DeclareMathSymbol{#1}7{#2}{#3}}
```

`\DeclareMathDelimiter`

```
674 \def\DeclareMathDelimiter#1{%
675   \if\relax\noexpand#1%
676     \expandafter\@DeclareMathDelimiter
677   \else
678     \expandafter\@xxDeclareMathDelimiter
679   \fi
680   #1}
681 \@onlypreamble\DeclareMathDelimiter
```

`\@xxDeclareMathDelimiter`

This macro checks if the second arg is a “math type” such as `\mathopen`. The undocumented original code didn’t use math types when the delimiter was a single letter. For this reason the coding is a bit strange as it tries to support the undocumented syntax for compatibility reasons.

```
682 \def\@xxDeclareMathDelimiter#1#2#3#4{%
```

7 is the default value returned in the case that `\mathchar@type` is passed something unexpected, like a math symbol font name. We locally move `\mathalpha` out of the way so if you use that the right branch is taken. This will still fail if an explicit number 7 is used!

```
683   \begingroup
684   \let\mathalpha\mathord
685   \ifnum7=\mathchar@type{#2}%
686     \endgroup
```

If this branch is taken we have old syntax (5 arguments).

```
687     \expandafter\@firstofone
688   \else
```

If this branch is taken `\mathchar@type` is different from 7 so we assume new syntax. In this case we also use the arguments to set up the letter as a math symbol for the case where it is not used as a delimiter.

```

689 \endgroup
690 \DeclareMathSymbol#1{#2}{#3}{#4}%

```

Then we arrange that `\@xDeclareMathDelimiter` only gets #1, #3, #4 ... as it does not expect a math type as argument.

```

691 \expandafter\@firstoftwo
692 \fi
693 {\@xDeclareMathDelimiter#1}{#2}{#3}{#4}}
694 \@onlypreamble\@xxDeclareMathDelimiter

```

`\@DeclareMathDelimiter`

```

695 \def\@DeclareMathDelimiter#1#2#3#4#5#6{%
696 \expandafter\in@\csname sym#3\expandafter\endcsname
697 \expandafter{\group@list}%
698 \ifin@
699 \expandafter\in@\csname sym#5\expandafter\endcsname
700 \expandafter{\group@list}%
701 \ifin@
702 \begingroup
703 \count\z@=#4\relax
704 \count\tw@\count\z@
705 \divide\count\z@\sist@n
706 \count@\count\z@
707 \multiply\count@\sist@n
708 \advance\count\tw@-\count@
709 \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
710 %
711 \count\z@=#6\relax
712 \count\tw@\count\z@
713 \divide\count\z@\sist@n
714 \count@\count\z@
715 \multiply\count@\sist@n
716 \advance\count\tw@-\count@
717 \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
718 %
719 \edef\reserved@a{\noexpand\in@{\string\delimiter}{\meaning#1}}%
720 \reserved@a
721 \ifin@
722 \expandafter\set@mathdelimiter
723 \csname sym#3\expandafter\endcsname
724 \csname sym#5\endcsname#1#2%
725 \reserved@c\reserved@d
726 \font@info{Redeclaring math delimiter \string#1}%
727 \else
728 \expandafter\ifx
729 \csname\expandafter\@gobble\string#1\endcsname
730 \relax
731 \expandafter\set@mathdelimiter
732 \csname sym#3\expandafter\endcsname
733 \csname sym#5\endcsname#1#2%
734 \reserved@c\reserved@d
735 \else
736 \@latex@error{Command '\string#1' already defined}\@eha
737 \fi

```

```

738     \fi
739   \endgroup
740   \else
741     \@latex@error{Symbol font ‘#5’ is not defined}\@eha
742   \fi
743   \else
744     \@latex@error{Symbol font ‘#3’ is not defined}\@eha
745   \fi
746 }
747 \@onlypreamble\@DeclareMathDelimiter

```

`\@xDeclareMathDelimiter`

```

748 \def\@xDeclareMathDelimiter#1#2#3#4#5{%
749   \expandafter\in@\csname sym#2\expandafter\endcsname
750   \expandafter{\group@list}%
751   \ifin@
752     \expandafter\in@\csname sym#4\expandafter\endcsname
753     \expandafter{\group@list}%
754   \ifin@
755     \begingroup
756       \count\z@=#3\relax
757       \count\tw@\count\z@
758       \divide\count\z@\sist@@n
759       \count@\count\z@
760       \multiply\count@\sist@@n
761       \advance\count\tw@-\count@
762       \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
763       %
764       \count\z@=#5\relax
765       \count\tw@\count\z@
766       \divide\count\z@\sist@@n
767       \count@\count\z@
768       \multiply\count@\sist@@n
769       \advance\count\tw@-\count@
770       \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
771       \expandafter\set@mathdelimiter
772       \csname sym#2\expandafter\endcsname\csname sym#4\endcsname#1%
773       \reserved@c\reserved@d
774     \endgroup
775   \else
776     \@latex@error{Symbol font ‘#4’ is not defined}\@eha
777   \fi
778   \else
779     \@latex@error{Symbol font ‘#2’ is not defined}\@eha
780   \fi
781 }
782 \@onlypreamble\@xDeclareMathDelimiter

```

`\set@mathdelimiter` We have to end the definition of a math delimiter like `\lfloor` with a space and not with `\relax` as we did before, because otherwise constructs involving `\abovewithdelims` will prematurely end (pr/1329)

```

783 \def\set@mathdelimiter#1#2#3#4#5#6{%
784   \xdef#3{\delimiter"\mathchar@type#4\hexnumber@#1#5%
785             \hexnumber@#2#6 }}

```

```

786 \@onlypreamble\set@mathdelimiter

\set@mathdelimiter

787 \def\set@mathdelimiter#1#2#3#4#5{%
788   \global\delcode'#3="\hexnumber@#1#4\hexnumber@#2#5\relax}
789 \@onlypreamble\set@mathdelimiter

\DeclareMathRadical

790 \def\DeclareMathRadical#1#2#3#4#5{%

Below is a crude fix to make this macro work if #1 is undefined or \relax. Should
be improved!

791   \expandafter\ifx
792     \csname\expandafter\@gobble\string#1\endcsname
793     \relax
794     \let#1\radical
795   \fi
796   \edef\reserved@a{\noexpand\in@{\string\radical}{\meaning#1}}%
797   \reserved@a
798   \ifin@
799     \expandafter\in@\csname sym#2\expandafter\endcsname
800     \expandafter{\group@list}%
801   \ifin@
802     \expandafter\in@\csname sym#4\expandafter\endcsname
803     \expandafter{\group@list}%
804   \ifin@
805     \begingroup
806       \count\z@=#3\relax
807       \count\tw@\count\z@
808       \divide\count\z@\sist@@n
809       \count@\count\z@
810       \multiply\count@\sist@@n
811       \advance\count\tw@-\count@
812       \edef\reserved@c{%
813         \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
814       \count\z@=#5\relax
815       \count\tw@\count\z@
816       \divide\count\z@\sist@@n
817       \count@\count\z@
818       \multiply\count@\sist@@n
819       \advance\count\tw@-\count@
820       \edef\reserved@d{%
821         \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
Coded inline instead of using \set@mathradical
822 %       \expandafter\set@mathradical
823 %       \csname sym#2\expandafter\endcsname
824 %       \csname sym#4\endcsname#1%
825 %       \reserved@c\reserved@d
826       \xdef#1{\radical"\expandafter\hexnumber@
827         \csname sym#2\endcsname\reserved@c
828         \expandafter\hexnumber@
829         \csname sym#4\endcsname\reserved@d
830         \relax}%
831     \endgroup

```

```

832     \else
833     \latex@error{Symbol font ‘#4’ is not defined}\@eha
834     \fi
835     \else
836     \latex@error{Symbol font ‘#2’ is not defined}\@eha
837     \fi
838     \else
839     \latex@error{Command ‘\string#1’ already defined}\@eha
840     \fi
841 }
842 \@onlypreamble\DeclareMathRadical

```

Definition below was wrong it contained `\delimiter !`

```

\def\set@mathradical#1#2#3#4#5{%
  \xdef#3{\radical"\hexnumber@#1#4\hexnumber@#2#5\relax}}

```

```

\mathalpha just a dummy currently
843 \let\mathalpha\relax

```

`\mathchar@type`

```

844 \def\mathchar@type#1{%
845   \ifodd 2#1#1 #1\else           % is this non-negative number?
846   \ifx#1\mathord 0\else
847   \ifx#1\mathop 1\else
848   \ifx#1\mathbin 2\else
849   \ifx#1\mathrel 3\else
850   \ifx#1\mathopen 4\else
851   \ifx#1\mathclose 5\else
852   \ifx#1\mathpunct 6\else
853   7%                             % anything else is variable ord
854   \fi
855   \fi
856   \fi
857   \fi
858   \fi
859   \fi
860   \fi
861   \fi}
862 \@onlypreamble\mathchar@type

```

`\DeclareSymbolFontAlphabet`

```

863 \def\DeclareSymbolFontAlphabet#1#2{%
864   \expandafter\DeclareSymbolFontAlphabet@
865   \csname \expandafter\@gobble\string#1\space\endcsname{#2}#1}
866 \@onlypreamble\DeclareSymbolFontAlphabet

```

`\DeclareSymbolFontAlphabet@`

```

867 \def\DeclareSymbolFontAlphabet@#1#2#3{%
We use the switch \if@tempswa to decide if we can declare this symbol font
alphabet.
868   \@tempwattrue

```

First check if #2 is known to be a symbol font

```
869 \expandafter\in@\csname sym#2\expandafter\endcsname
870 \expandafter{\group@list}%
871 \ifin@
```

Check if #1 is defined as a math alphabet defined via \DeclareMathAlphabet:

```
872 \expandafter\in@\expandafter#1\expandafter{\alpha@list}%
873 \ifin@
```

If so remove it from the \alpha@list and from all math version macros.

```
874 \@font@info{Redeclaring math alphabet \string#3}%
875 \toks@{}%
876 \def\alpha@elt##1##2##3{%
877 \ifx##1#1\else\addto@hook\toks@{\alpha@elt##1##2##3}\fi}%
878 \alpha@list
879 \xdef\alpha@list{\the\toks@}%
```

Now we loop over all versions and remove the math alphabet:

```
880 \def\version@elt##1{%
881 \begingroup
882 \toks@{}%
883 \def\getanddefine@fonts####1####2{%
884 \addto@hook\toks@{\getanddefine@fonts####1####2}}%
885 \def\install@mathalphabet####1####2{%
886 \ifx####1#1\else
887 \addto@hook\toks@{\install@mathalphabet
888 ####1{####2}}\fi}%
889 ##1%
890 \xdef##1{\the\toks@}%
891 \endgroup
892 }%
893 \version@list
894 \else
```

If #3 is not defined as a math alphabet check if it is defined at all:

```
895 \expandafter\ifx
896 \csname\expandafter\@gobble\string#1\space\endcsname
897 \relax
```

If it is undefined, fine otherwise check if it is a math alphabet defined via \DeclareSymbolFontAlphabet:

```
898 \else
899 \edef\reserved@a{%
900 \noexpand\in@{\string\use@mathgroup}{\meaning#1}}%
901 \reserved@a
902 \ifin@
903 \@font@info{Redeclaring math alphabet \string#3}%
904 \else
```

Since the command #3 is defined to be something which is not a math alphabet we have to skip redefining it.

```
905 \@tempswafalse
906 \@latex@error{Command '\string#3' already defined}\@eha
907 \fi
908 \fi
909 \fi
910 \else
```

Since the symbol font is not known we better skip defining this alphabet.

```

911     \@tempswafalse
912     \@latex@error{Unknown symbol font '#2'}\@eha
913     \fi
914     \if@tempswa

```

When we reach this point we are allowed to define #1 to be a symbol font math alphabet. This means that we have to set it to

```
\use@mathgroup <math-settings> \sym<name>
```

The  $\langle math\text{-}settings \rangle$  are the one for the encoding that is used in the font shape where  $\backslash\mathrm{sym}\langle name \rangle$  is pointing to. This means that we have to get it from the information stored in  $\backslash\mathrm{group@list}$ . Thus we loop through that list after defining  $\backslash\mathrm{group@elt}$  in a suitable way.

```

915     \def\group@elt##1##2{%
916         \expandafter\ifx\csname sym#2\endcsname##1%
917         \expandafter\reserved@a\string##2\@nil
918         \fi}%
919     \def\reserved@a##1##2/##3\@nil{%
920         \def\reserved@a{##2}}%
921     \group@list
922     \toks@{\relax\ifmmode \else \non@alpherr#1\fi}%
923     \edef#1{\the\toks@
924         \noexpand\use@mathgroup
925         \expandafter\noexpand\csname M@\reserved@a\endcsname
926         \csname sym#2\endcsname}%
927     \def#3{\protect#1}%
928     \fi
929 }
930 \@onlypreamble\DeclareSymbolFontAlphabet@
931 /2ekernel)

```

## File s

# ltfssini.dtx

This file contains the top level L<sup>A</sup>T<sub>E</sub>X interface to the font selection scheme commands. See other parts of the L<sup>A</sup>T<sub>E</sub>X distribution, or *The L<sup>A</sup>T<sub>E</sub>X Companion* for higher level documentation of these commands.

## 34 NFSS Initialisation

Finally, there are six commands that are to be used in L<sup>A</sup>T<sub>E</sub>X and that we will therefore protect against expansion at the wrong point: `\fontfamily`, `\fontseries`, `\fontshape`, `\fontsize`, `\selectfont`, and `\mathversion`.

```
1 \langle *2kernel\rangle
```

### 34.1 Providing math *versions*

L<sup>A</sup>T<sub>E</sub>X provides two *versions*. We call them *normal* and *bold*, respectively.

```
2 \DeclareMathVersion{normal}
3 \DeclareMathVersion{bold}
```

Now we define the standard font change commands. We don't allow the use of `\rmfamily` etc. in math mode.

First the changes to another *family*:

```
4 \DeclareRobustCommand\rmfamily
5     {\not@math@alphabet\rmfamily\mathrm
6      \fontfamily\rmdefault\selectfont}
7 \DeclareRobustCommand\sffamily
8     {\not@math@alphabet\sffamily\mathsf
9      \fontfamily\sfdefault\selectfont}
10 \DeclareRobustCommand\ttfamily
11     {\not@math@alphabet\ttfamily\mathtt
12      \fontfamily\ttdefault\selectfont}
```

Then the commands changing the *series*:

```
13 \DeclareRobustCommand\bfseries
14     {\not@math@alphabet\bfseries\mathbf
15      \fontseries\bfdefault\selectfont}
16 \DeclareRobustCommand\mdseries
17     {\not@math@alphabet\mdseries\relax
18      \fontseries\mddefault\selectfont}
19 \DeclareRobustCommand\upshape
20     {\not@math@alphabet\upshape\relax
21      \fontshape\updefault\selectfont}
```

Then the commands changing the *shape*:

```
22 \DeclareRobustCommand\slshape
23     {\not@math@alphabet\slshape\relax
24      \fontshape\sldefault\selectfont}
25 \DeclareRobustCommand\scshape
26     {\not@math@alphabet\scshape\relax
27      \fontshape\scdefault\selectfont}
```

```

28 \DeclareRobustCommand\itshape
29     {\not@math@alphabet\itshape\mathit
30      \fontshape\itdefault\selectfont}

```

`\em` We also have to define the *emphasize* font change command (i.e. `\em`). This command will look is the current font is sloped (i.e. has a positive `\fontdimen1`) and will then select either `\upshape` or `\itshape`.

```

31 \ifdim\fontdimen1>0pt\relax
32 \ifdim\fontdimen1>0pt\relax
33 \ifdim\fontdimen1>0pt\relax
34 \DeclareRobustCommand\em
35     {\@nomath\em \ifdim \fontdimen1>0pt\font >\z@
36      \emminnershape \else \itshape \fi}%
37 \def\emminnershape{\upshape}%
38 \ifdim\fontdimen1>0pt\relax
39 \ifdim\fontdimen1>0pt\relax
40 \ifdim\fontdimen1>0pt\relax
41 \ifdim\fontdimen1>0pt\relax
42 \ifdim\fontdimen1>0pt\relax
43 \ifdim\fontdimen1>0pt\relax
44 \ifdim\fontdimen1>0pt\relax
45 \ifdim\fontdimen1>0pt\relax
46 \ifdim\fontdimen1>0pt\relax

```

`\not@math@alphabet` This function generates an error message when it is called in math mode. The same function should be defined in `newlfont.sty`.

```

47 \def\not@math@alphabet#1#2{%
48     \relax
49     \ifmmode
50         \@latex@error{Command \noexpand#1invalid in math mode}%
51         {%
52             Please
53             \ifx#2\relax
54                 define a new math alphabet^^J%
55                 if you want to use a special font in math mode%
56             \else

```

We have to a `\noexpand` below to prevent expansion of `#2`. In case of `#1` we can omit this (due to the current definition of robust commands since they do come out right there :-).

```

57         use the math alphabet \noexpand#2instead of
58         the #1command%
59     \fi
60     .
61 }%
62 \fi}

```

Finally we provide two abbreviations to switch to the *L<sup>A</sup>T<sub>E</sub>X* versions.

```

63 \def\boldmath{\@nomath\boldmath
64     \mathversion{bold}}
65 \def\unboldmath{\@nomath\unboldmath
66     \mathversion{normal}}

```

Here we switch to the default math version by defining the internal macro `\math@version`. We dare not to call `\mathversion` at this place because this would call `\glb@settings`.

```
67 \def\math@version{normal}
```

## 34.2 Miscellaneous

`\newfont` We start by defining a few macros that are part of standard L<sup>A</sup>T<sub>E</sub>X's user interface.  
`\symbol` The use of these functions is not encouraged, but they will allow to process older documents without changes to the source.

```
68 \def\newfont#1#2{\@ifdefinable#1{\font#1=#2\relax}}
69 \def\symbolsymbol#1{\char #1\relax}
```

`\@setfontsize` This abbreviation is used by L<sup>A</sup>T<sub>E</sub>X's user level size changing commands, such as  
`\@setsize` `\large`.

```
70 \def\@setfontsize#1#2#3{\@nomath#1%
```

For the benefit of people relying on keeping the name of the current font command saved in `\@currsiz` we define it. To ensure that `\@setfontsize` keeps being robust we omit this assignment during times where `\protect` differs from `\@typeset@protect`.

```
71 \ifx\protect\@typeset@protect
72 \let\@currsiz#1%
73 \fi
74 \fontsize{#2}{#3}\selectfont}
```

For compatibility we also define `\@setsize` the 209 command

```
75 \*compat
76 \def\@setsize#1#2#3#4{\@setfontsize#1{#4}{#2}}
77 \compat
```

`\oldstylenums` This macro implements old style numerals but only works if we assume that the standard math fonts are used. Thus it needs changing in case other math encodings are used.

```
78 \def\oldstylenums#1{%
79 \begingroup
```

Provide spacing using the interword space of the current font.

```
80 \spaceskip\fontdimen\tw@\font
```

Then switch to the math italic font. We don't change the current value of `\f@series` which means that you can use bold numerals if `\bfseries` is in force. As family we use `\rmdefault` which means that this only works if there exist an OML encoded version of that font or rather a corresponding `.fd` file (which is the case for standard L<sup>A</sup>T<sub>E</sub>X fonts even though they only contain substitutions).

```
81 \usefont{OML}{\rmdefault}{\f@series}{it}%
82 \mathgroup\symbols #1%
83 \endgroup
84 }
```

`\hexnumber@` To set up L<sup>A</sup>T<sub>E</sub>X's special math character definitions we first provide a macro to generate hexadecimal numbers. It is a rather simple `\ifcase`.

```
85 \def\hexnumber@#1{\ifcase\number#1
```

```

86 0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or
87 9\or A\or B\or C\or D\or E\or F\fi}

```

`\nfss@text` In its simplest form `\nfss@text` is an `\mbox`. This will produce unbreakable text outside math and inside math you will get text with the same fonts as outside. The only drawback is that such item won't change sizes in subscripts. But this behavior can be easily changed. With the `amstex` style option one will get a sub style called `amstext` which will redefine the `\nfss@text` macro to produce correct text in all sizes.

We have to use `\def` instead of the shorter `\let` since `\mbox` is undefined when we reach this point.

```
88 \def\nfss@text#1{{\mbox{#1}}}
```

`\copyright` The definition of `\copyright` was changed so that it works in other type styles, and to make it robust. We leave the family untouched so that the copyright notice will come out differently if a different font family is in use. This command is commented out, since it is now defined in `ltoutenc.dtx`.

```

89 %\DeclareRobustCommand\copyright
90 %    {{\oalign{\hfil
91 %        \raise.07ex\hbox{\mdseries\upshape c}\hfil\crcr
92 %        \mathhexbox20D}}}

```

`\normalfont` The macro `\reset@font` is used in  $\text{\LaTeX}$  to switch to a standard font, in order to initialize the current font in situations where typesetting is done in a new visual context (e.g. in a footnote). We define it here to allow the test for the new  $\text{\LaTeX}$  version above but nevertheless are able to run all kind of mixtures.

`\reset@font`

`\p@reset@font`

The user interface name for `\reset@font` is `\normalfont`:

```

93 \DeclareRobustCommand\normalfont
94     {\usefont\encodingdefault
95         \familydefault
96         \seriesdefault
97         \shapedefault
98         \relax}
99 \let\reset@font\normalfont

```

We left out the special  $\text{\LaTeX}$  fonts which are not automatically included in the base version of the font selection since these fonts contain only a few characters which are also included in the AMS fonts so anybody who is using these fonts doesn't need them. But for compatibility reasons we will define these symbols.

```

100 \def\not@base#1{\@latex@error
101   {Command \noexpand#1not provided in base LaTeX2e}%
102   {Load the latexsym or the amsfonts package to
103   define this symbol}}
104 \def\mho{\not@base\mho}
105 \def\Join{\not@base\Join}
106 \def\Box{\not@base\Box}
107 \def\Diamond{\not@base\Diamond}
108 \def\leadsto{\not@base\leadsto}
109 \def\squsubset{\not@base\squsubset}
110 \def\sqsupset{\not@base\sqsupset}
111 \def\lhd{\not@base\lhd}

```

```

112 \def\unlhd{\not@base\unlhd}
113 \def\rhd{\not@base\rhd}
114 \def\unrhd{\not@base\unrhd}

```

We now initialize all variables set by `\DeclareErrorFont`. These values are not really important since they will be overwritten later on by the definition in `fontdef.ltx`.

However, if `fontdef.cfg` is corrupted then at least a hopefully suitable error font is present.

```

115 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}  %% don't modify this setting
116                                         %% overwrite it in fontdef.cfg
117                                         %% if necessary

```

We now load the customizable parts of NFSS.

```

118 \ifnum\inputlineno=\m@ne

```

Still using  $\TeX$ 2. need a configuration file to avoid setting the 8bit characters.

```

119 \InputIfFileExists{fonttext.cfg}
120     {\typeout{=====^^J%
121             ^^J%
122             Local config file fonttext.cfg used^^J%
123             ^^J%
124             =====}%
125     \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
126     }
127     {\typeout{!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
128             !^^J%
129             ! You MUST use a fonttext.cfg file!^^J%
130             ! As you are still using TeX2!!!!^^J%
131             !^^J%
132             ! See the documentation file tex2.txt^^J%
133             !^^J%
134             !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!}%
135     \batchmode \@@end}
136 \else

```

With  $\TeX$ 3 can use the standard `ltx` file if no configuration file exists.

```

137 \InputIfFileExists{fonttext.cfg}
138     {\typeout{=====^^J%
139             ^^J%
140             Local config file fonttext.cfg used^^J%
141             ^^J%
142             =====}%
143     \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
144     }
145     {\input{fonttext.ltx}}
146 \fi
147 \let\@addtofilelist\@gobble

```

Ditto for math although I don't think that we will get a lot of customisation :-)

```

148 \InputIfFileExists{fontmath.cfg}
149     {\typeout{=====^^J%
150             ^^J%
151             Local config file fontmath.cfg used^^J%

```

```

152             ^^J%
153             =====}%
154             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
155             }
156             {\input{fontmath.ltx}}
157 \let\@addtofilelist\@gobble

```

Then we preload several fonts. This file might be customized *without* changing the behavior of the format (i.e. necessary font definitions will be loaded at runtime if they are not preloaded). This is done in the file `preload.ltx`.

```

158 \InputIfFileExists{preload.cfg}
159     {\typeout{=====^^J%
160             ^^J%
161             Local config file preload.cfg used^^J%
162             ^^J%
163             =====}%
164             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
165             }
166             {\input{preload.ltx}}
167 \let\@addtofilelist\@gobble

```

```

\@acci We also save the values of some accents in \@acci, \@accii and \@acciii so they
\@accii can be restored by a minipage inside a tabbing environment.
\@acciii 168 \let\@acci\' \let\@accii\' \let\@acciii\=

```

```

\cal Here were the two old ⟨alphabet identifiers⟩.
\mit
169 </2ekernel>

```

# File t

## fontdef.dtx

### 35 Introduction

This file is used to generate the files `fonttext.ltx` (text font declarations) and `fontmath.ltx` (math font declarations), which are used during the format generation. It contains the declaration of the standard text encodings used at the site as well as a minimal subset of font shape groups that NFSS will look at to ensure that the specified encodings are valid.

The math part contains the setup for math encodings as well as the default math symbol declarations that belong to the encoding.

It is possible to change this setup (by using other fonts, or defaults) without losing the ability to process documents written at other sites. Portability in this sense means that a document will compile without errors. It does not mean, however, that identical output will be produced. For this it is necessary that the distributed setup is used at both installations.

### 36 Customization

You are not allowed to change this source file! If you want to change the default encodings and/or the font shape groups preloaded you should create a copy of `fonttext.ltx` under the name `fonttext.cfg` and change this copy. If  $\text{\LaTeX}$  2 $\epsilon$  finds a file of this name it will use it, otherwise it uses the standard file which is `fontdef.ltx`.

If you don't plan to use Computer Modern much or at all, it might (!) be a good idea to make your own `fonttext.cfg`. Look at the comments below (docstrip module 'text') to see what should go into such a file.

To change the math font setup use a copy of `fontmath.ltx` under the name `fontmath.cfg` and change this copy. However, dealing with this interface is even more a job for an expert than changing the text font setup — in short, we don't encourage either.

**Warning:** please note that we don't support customised  $\text{\LaTeX}$  versions. Thus, before sending in a bug report please try your test file with a  $\text{\LaTeX}$  format which is not customised and send in the log from that version (unless the problem goes away).

Please note: the following standard encodings have to be defined in all local variants of `font....cfg` to guarantee that all  $\text{\LaTeX}$  installations behave in the same way.

T1	Cork $\text{\TeX}$ text encoding
OT1	old $\text{\TeX}$ text encoding
U	unknown encoding
OML	old $\text{\TeX}$ math letters encoding
OMS	old $\text{\TeX}$ math symbols encoding
OMX	old $\text{\TeX}$ math extension symbols encoding

Notice that some of these encodings are ‘old’ in the sense that we hope that they will be superseded soon by encoding standards defined by the T<sub>E</sub>X user community. Therefore this set of default encodings may change in the future.

The first candidate is OT1 which will soon be replaced by T1, the official T<sub>E</sub>X text encoding.

**Warning:** If you add additional encodings to this file there is no guarantee any longer that files processable at your installation will also be processable at other installations. Thus, if you make use of such an encoding in your document, e.g. if you intend to typeset in Cyrillic (OT2 encoding), you need to specify this encoding in the preamble of your document prior to sending it to another installation. Once the encoding is specified in that place in your document, the document is processable at all L<sup>A</sup>T<sub>E</sub>X installations (provided they have suitable fonts installed).

For this reason we suggest that you define a short package file that sets up an additional encoding used at your site (rather than putting the encoding into this file) since this package can easily be shipped with your document.

## 37 The docstrip modules

The following modules are used to direct `docstrip` in generating external files:

<code>driver</code>	produce a documentation driver file
<code>text</code>	produce the file <code>fonttext.ltx</code>
<code>math</code>	produce the file <code>fontmath.ltx</code>
<code>cfgtext</code>	produce a dummy <code>fonttext.cfg</code> file
<code>cfgmath</code>	produce a dummy <code>fontmath.cfg</code> file

A typical `docstrip` command file would then have entries like:

```
\generateFile{fonttext.ltx}{t}{\from{fontdef.dtx}{text}}
```

## 38 A driver for this document

The next bit of code contains the documentation driver file for T<sub>E</sub>X, i.e. the file that will produce the documentation you are currently reading. It will be extracted from this file by the DOCSTRIP program.

```
1 (*driver)
2 \documentclass{ltxdoc}
3 \GetFileInfo{fontdef.dtx}
4 \begin{document}
5   \DocInput{fontdef.dtx}
6 \end{document}
7 \end{driver}
```

## 39 The fonttext.ltx file

The identification is done earlier on with a `\ProvidesFile` declaration.

```
8 (*text)
9 \typeout{=== Don't modify this file, use a .cfg file instead ===^J}
```

### 39.1 Encodings

This file declares the standard encodings for text and math fonts. All others should be declared in packages or in the documents directly.

For every text encoding there are normally a number of encoding specific commands, e.g. accents, special characters, etc. (The definition for such a command might have to change when the encoding is changed, because the character is in a different position, or not available at all, or the accent is produced in a different way.) This is handled by a general mechanism which is described in `ltoutenc.dtx`.

By convention, text encoding specific declarations, including the declaration `\DeclareFontEncoding`, are kept in separate file of the form `<enc>enc.def`, e.g. `ot1enc.def`. This allows other applications to make use of the declarations as well.

Similar to the default encoding, the loading of the encoding files for the two major text encodings shouldn't be changed. In particular, the `inputenc` package depends on this.

```
10 \input {omlenc.def}
11 \input {t1enc.def}
12 \input {ot1enc.def}      % <- should come after T1 for speed
13 \input {omsenc.def}
```

We then set the default text font encoding. This will hopefully change some day to T1. This setting should *not* be changed to produce a portable format.

```
14 \fontencoding{OT1}
```

If different encodings for text fonts are in use one could put the common setup into `\DeclareFontEncodingDefaults`. There is now a better mechanism so using this interface is discouraged!

```
15 \DeclareFontEncodingDefaults{}{}
```

Then we define the default substitution for every encoding. This release of  $\text{\LaTeX 2}_{\epsilon}$  assumes that the ec fonts are available. It is possible to change this to point to some other font family (e.g., Times with the appropriate encoding if it is available) without making documents non-portable. However, in such a case documents will produce different page breaks at other sites. The substitution defaults can all be changed without losing portability as long as there are font shape definitions for the selected substitutions.

```
16 \DeclareFontSubstitution{T1}{cmr}{m}{n}
17 \DeclareFontSubstitution{OT1}{cmr}{m}{n}
```

For every encoding declaration,  $\text{\LaTeX 2}_{\epsilon}$  will try to verify that the given substitution information makes sense, i.e. that it is impossible to go into an endless loop if font substitution happens. This is done at the moment the `\begin{document}` is encountered.  $\text{\LaTeX 2}_{\epsilon}$  will then check that for every encoding the substitution defaults form a valid font shape group, which means that it will check if there is a `\DeclareFontShape` declaration for this combination. We will therefore load the

corresponding .fd files now. If we don't do this they would be loaded at verification time (i.e. at `\begin{document}`) which would delay processing unnecessarily.

**Warning:** Please note that this means that you have to regenerate the format whenever you change any of these .fd files since L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> will not read .fd files if it already knows about the encoding/family combination.

The `\nfss@catcodes` ensures that white space is ignored in any definitions made in the fd files.

```
18 \begingroup
19 \nfss@catcodes
20 \input {t1cmr.fd}
21 \input {ot1cmr.fd}
22 \endgroup
```

We also load some other font definition files which are normally needed in a document. This is only done for processing speed and you can comment the next two lines out to save some memory. If necessary these files are then loaded when your document is processed. (Loading .fd files is a less drastic step compared to preloading fonts because the number of fonts is limited 255 at (nearly) every T<sub>E</sub>X installation, while the amount of main memory is not a limiting factor at most installations.)

```
23 \begingroup
24 \nfss@catcodes
25 \input {ot1cmss.fd}
26 \input {ot1cmtt.fd}
27 \endgroup
```

Even with all the precautions it is still possible that NFSS will run into problems, for example, when a .fd file contains corrupted data. To guard against such cases NFSS has a very low-level fallback font that is installed with the following line.

```
28 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
```

This means, “if everything else fails use Computer Modern Roman normal shape at 10pt in the old text encoding”. You can change the font used but the encoding should be the same as the one specified with `\fontencoding` above.

## 39.2 Defaults

To allow the use of `\rmfamily`, `\sffamily`, etc. in documents even if non-standard families are used we provide nine macros which hold the name of the corresponding families, series, and so on. This makes it easy to use other font families (like Times Roman, etc.). One simply has to redefine these defaults.

All these hooks have to be defined in this file but you can change their meaning (except for `\encodingdefault`) without making documents non-portable.

```
\rmdefault The following three definitions set up the meaning for \rmfamily, \sffamily, and
\sffdefault \ttfamily.
\ttdefault
29 \newcommand\rmdefault{cmr}
30 \newcommand\sffdefault{cmss}
31 \newcommand\ttdefault{cmtt}
```

```

\bfdefault Series changing commands are influenced by the following hooks.
\mddefault 32 \newcommand\bfdefault{bx}
           33 \newcommand\mddefault{m}

\itdefault Shape changing commands use the following hooks.
\sldefault 34 \newcommand\itdefault{it}
\scdefault 35 \newcommand\sldefault{sl}
\updefault 36 \newcommand\scdefault{sc}
           37 \newcommand\updefault{n}

\encodingdefault Finally we have the hooks that describe the behaviour of the \normalfont com-
\familydefault mand. To stay portable, the definition of \encodingdefault should not be
\seriesdefault changed and should match the setting above for \fontencoding. All other values
\shapedefault can be set according to your taste.
           38 \newcommand\encodingdefault{OT1}
           39 \newcommand\familydefault{\rmdefault}
           40 \newcommand\seriesdefault{\mddefault}
           41 \newcommand\shapedefault{\updefault}

           This finishes the low-level setup in fonttext.ltx.
           42 \</text>

```

## 40 The fontmath.ltx file

The identification is done earlier on with a `\ProvidesFile` declaration.

```

43 (*math)
44 \typeout{=== Don't modify this file, use a .cfg file instead ===^^J}

```

### 40.1 The font encodings used

```

45 \DeclareFontEncoding{OML}{-}{-}
46 \DeclareFontEncoding{OMS}{-}{-}
47 \DeclareFontEncoding{OMX}{-}{-}

```

Finally a declaration for U encoding which serves for all fonts that do not fit standard encodings. For math this sets up `\noaccents@` providing for AMS- $\LaTeX$ . This macro is used therein to handle accented characters if they are not supported by the font. In other words, if fonts with U encoding are used in math, all accents (like from `\breve`) are obtained from some other font that has them.

```

48 \DeclareFontEncoding{U}{-}{\noaccents@}

```

The encodings for math are next:

```

49 \DeclareFontSubstitution{OML}{cmm}{m}{it}
50 \DeclareFontSubstitution{OMS}{cmsy}{m}{n}
51 \DeclareFontSubstitution{OMX}{cmex}{m}{n}
52 \DeclareFontSubstitution{U}{cmr}{m}{n}

53 \begingroup
54 \nfss@catcodes
55 \input {omlcmm.fd}
56 \input {omscmsy.fd}
57 \input {omxcmex.fd}
58 \input {ucmr.fd}

```

59 \endgroup

### 40.1.1 Symbolfont and Alphabet declarations

We now define the basic symbol fonts used by L<sup>A</sup>T<sub>E</sub>X. These four symbol fonts must be defined by this file.

It is possible to make the symbol fonts point to other external fonts without losing the ability to process documents written at other sites, as long as one defines the same symbol font names with the same encodings, e.g. `operators` with `OT1` etc. If other encodings are used documents become non-portable. Such a change should therefore be done in a package file.

```
60 \DeclareSymbolFont{operators}    {OT1}{cmr}{m}{n}
61 \DeclareSymbolFont{letters}      {OML}{cmm}{m}{it}
62 \DeclareSymbolFont{symbols}      {OMS}{cmsy}{m}{n}
63 \DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}

64 \SetSymbolFont{operators}{bold}{OT1}{cmr}{bx}{n}
65 \SetSymbolFont{letters}{bold}{OML}{cmm}{b}{it}
66 \SetSymbolFont{symbols}{bold}{OMS}{cmsy}{b}{n}
```

Below are the seven math alphabets which are defined by NFSS. Again they must be defined by this file. However, as before you can change the fonts used without losing portability, but you should be careful when changing the encoding since that may make documents come out wrong.

```
67 \DeclareSymbolFontAlphabet{\mathrm}    {operators}
68 \DeclareSymbolFontAlphabet{\mathnormal}{letters}
69 \DeclareSymbolFontAlphabet{\mathcal}    {symbols}
70 \DeclareMathAlphabet          {\mathbf} {OT1}{cmr}{bx}{n}
71 \DeclareMathAlphabet          {\mathsf} {OT1}{cmss}{m}{n}
72 \DeclareMathAlphabet          {\mathit} {OT1}{cmr}{m}{it}
73 \DeclareMathAlphabet          {\mathtt} {OT1}{cmtt}{m}{n}
```

Given the currently available fonts we cannot bold-en `\mathbf` and `\mathtt` but in principle one could use ‘ultra bold’ or something. The alphabets defined via `\DeclareSymbolFontAlphabet` will change automatically in a new math version if the corresponding symbol font changes.

```
74 \SetMathAlphabet\mathsf{bold}{OT1}{cmss}{bx}{n}
75 \SetMathAlphabet\mathit{bold}{OT1}{cmr}{bx}{it}
```

## 40.2 Math font sizes

The declarations below declare the text, script and scriptscript size to be used for each text font size.

All occurrences of sizes longer than a single character are replaced with the macro name that holds them, saving a number of tokens (but losing a bit of speed, so this may not stay this way).

```
76 \DeclareMathSizes{5}{5}{5}{5}
77 \DeclareMathSizes{6}{6}{5}{5}
78 \DeclareMathSizes{7}{7}{5}{5}
79 \DeclareMathSizes{8}{8}{6}{5}
80 \DeclareMathSizes{9}{9}{6}{5}
81 \DeclareMathSizes{\@xpt}{\@xpt}{7}{5}
82 \DeclareMathSizes{\@xipt}{\@xipt}{8}{6}
83 \DeclareMathSizes{\@xipt}{\@xipt}{8}{6}
```

```

84 \DeclareMathSizes{\@xivpt}{\@xivpt}{\@xpt}{7}
85 \DeclareMathSizes{\@xvipt}{\@xvipt}{\@xipt}{\@xpt}
86 \DeclareMathSizes{\@xxpt}{\@xxpt}{\@xivpt}{\@xipt}
87 \DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xxpt}{\@xvipt}

```

## 40.3 The math symbol assignments

We start by setting up math codes for most of the characters typed in directly from the keyboard. Most of them are normally already setup up in the same way by  $\text{Init}\text{T}_{\text{E}}\text{X}$ . However, we repeat them here to have a complete setup which can be exchanged with another if desired.

### 40.3.1 The letters

```

88 \DeclareMathSymbol{a}{\mathalpha}{letters}{'a}
89 \DeclareMathSymbol{b}{\mathalpha}{letters}{'b}
90 \DeclareMathSymbol{c}{\mathalpha}{letters}{'c}
91 \DeclareMathSymbol{d}{\mathalpha}{letters}{'d}
92 \DeclareMathSymbol{e}{\mathalpha}{letters}{'e}
93 \DeclareMathSymbol{f}{\mathalpha}{letters}{'f}
94 \DeclareMathSymbol{g}{\mathalpha}{letters}{'g}
95 \DeclareMathSymbol{h}{\mathalpha}{letters}{'h}
96 \DeclareMathSymbol{i}{\mathalpha}{letters}{'i}
97 \DeclareMathSymbol{j}{\mathalpha}{letters}{'j}
98 \DeclareMathSymbol{k}{\mathalpha}{letters}{'k}
99 \DeclareMathSymbol{l}{\mathalpha}{letters}{'l}
100 \DeclareMathSymbol{m}{\mathalpha}{letters}{'m}
101 \DeclareMathSymbol{n}{\mathalpha}{letters}{'n}
102 \DeclareMathSymbol{o}{\mathalpha}{letters}{'o}
103 \DeclareMathSymbol{p}{\mathalpha}{letters}{'p}
104 \DeclareMathSymbol{q}{\mathalpha}{letters}{'q}
105 \DeclareMathSymbol{r}{\mathalpha}{letters}{'r}
106 \DeclareMathSymbol{s}{\mathalpha}{letters}{'s}
107 \DeclareMathSymbol{t}{\mathalpha}{letters}{'t}
108 \DeclareMathSymbol{u}{\mathalpha}{letters}{'u}
109 \DeclareMathSymbol{v}{\mathalpha}{letters}{'v}
110 \DeclareMathSymbol{w}{\mathalpha}{letters}{'w}
111 \DeclareMathSymbol{x}{\mathalpha}{letters}{'x}
112 \DeclareMathSymbol{y}{\mathalpha}{letters}{'y}
113 \DeclareMathSymbol{z}{\mathalpha}{letters}{'z}

114 \DeclareMathSymbol{A}{\mathalpha}{letters}{'A}
115 \DeclareMathSymbol{B}{\mathalpha}{letters}{'B}
116 \DeclareMathSymbol{C}{\mathalpha}{letters}{'C}
117 \DeclareMathSymbol{D}{\mathalpha}{letters}{'D}
118 \DeclareMathSymbol{E}{\mathalpha}{letters}{'E}
119 \DeclareMathSymbol{F}{\mathalpha}{letters}{'F}
120 \DeclareMathSymbol{G}{\mathalpha}{letters}{'G}
121 \DeclareMathSymbol{H}{\mathalpha}{letters}{'H}
122 \DeclareMathSymbol{I}{\mathalpha}{letters}{'I}
123 \DeclareMathSymbol{J}{\mathalpha}{letters}{'J}
124 \DeclareMathSymbol{K}{\mathalpha}{letters}{'K}
125 \DeclareMathSymbol{L}{\mathalpha}{letters}{'L}
126 \DeclareMathSymbol{M}{\mathalpha}{letters}{'M}

```

```

127 \DeclareMathSymbol{N}{\mathalpha}{letters}{'N}
128 \DeclareMathSymbol{O}{\mathalpha}{letters}{'O}
129 \DeclareMathSymbol{P}{\mathalpha}{letters}{'P}
130 \DeclareMathSymbol{Q}{\mathalpha}{letters}{'Q}
131 \DeclareMathSymbol{R}{\mathalpha}{letters}{'R}
132 \DeclareMathSymbol{S}{\mathalpha}{letters}{'S}
133 \DeclareMathSymbol{T}{\mathalpha}{letters}{'T}
134 \DeclareMathSymbol{U}{\mathalpha}{letters}{'U}
135 \DeclareMathSymbol{V}{\mathalpha}{letters}{'V}
136 \DeclareMathSymbol{W}{\mathalpha}{letters}{'W}
137 \DeclareMathSymbol{X}{\mathalpha}{letters}{'X}
138 \DeclareMathSymbol{Y}{\mathalpha}{letters}{'Y}
139 \DeclareMathSymbol{Z}{\mathalpha}{letters}{'Z}

```

#### 40.3.2 The digits

```

140 \DeclareMathSymbol{0}{\mathalpha}{operators}{'0}
141 \DeclareMathSymbol{1}{\mathalpha}{operators}{'1}
142 \DeclareMathSymbol{2}{\mathalpha}{operators}{'2}
143 \DeclareMathSymbol{3}{\mathalpha}{operators}{'3}
144 \DeclareMathSymbol{4}{\mathalpha}{operators}{'4}
145 \DeclareMathSymbol{5}{\mathalpha}{operators}{'5}
146 \DeclareMathSymbol{6}{\mathalpha}{operators}{'6}
147 \DeclareMathSymbol{7}{\mathalpha}{operators}{'7}
148 \DeclareMathSymbol{8}{\mathalpha}{operators}{'8}
149 \DeclareMathSymbol{9}{\mathalpha}{operators}{'9}

```

#### 40.3.3 Punctuation, brace, etc. keys

```

150 \DeclareMathSymbol{!}{\mathclose}{operators}{"21}
151 \DeclareMathSymbol{*}{\mathbin}{symbols}{"03} % \ast
152 \DeclareMathSymbol{+}{\mathbin}{operators}{"2B}
153 \DeclareMathSymbol{,}{\mathpunct}{letters}{"3B}
154 \DeclareMathSymbol{-}{\mathbin}{symbols}{"00}
155 \DeclareMathSymbol{.}{\mathord}{letters}{"3A}
156 \DeclareMathSymbol{:}{\mathrel}{operators}{"3A}
157 \DeclareMathSymbol{;}{\mathpunct}{operators}{"3B}
158 \DeclareMathSymbol{=}{\mathrel}{operators}{"3D}
159 \DeclareMathSymbol{?}{\mathclose}{operators}{"3F}

```

The following symbols are defined as delimiters below which automatically defines them as math symbols.

```

160 %\DeclareMathSymbol{(}{\mathopen}{operators}{"28}
161 %\DeclareMathSymbol{)}{\mathclose}{operators}{"29}
162 %\DeclareMathSymbol{/}{\mathord}{letters}{"3D}
163 %\DeclareMathSymbol{[}{\mathopen}{operators}{"5B}
164 %\DeclareMathSymbol{]}{\mathclose}{operators}{"5D}
165 %\DeclareMathSymbol{|}{\mathord}{symbols}{"6A}
166 %\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}
167 %\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}

```

Should all of the following being activated by default? Probably not.

```

168 %\DeclareMathSymbol{\}{\mathopen}{symbols}{"66}
169 %\DeclareMathSymbol{\'}{\mathclose}{symbols}{"67}
170 %\DeclareMathSymbol{\}\}{\mathord}{symbols}{"6E} % \backslash
171 \mathcode'\ =8000 % \space
172 \mathcode\' =8000 % ^\prime

```

```
173 \mathcode'\_="8000 % \_
```

#### 40.3.4 Delimitercodes for characters

[to be completed]

Finally, `IniTeX` sets all `\delcode` values to -1, except `\delcode'.`=0

```
174 \DeclareMathDelimiter{({}\mathopen}{operators}{28}{largesymbols}{00}
175 \DeclareMathDelimiter{)}{\mathclose}{operators}{29}{largesymbols}{01}
176 \DeclareMathDelimiter{[]\mathopen}{operators}{5B}{largesymbols}{02}
177 \DeclareMathDelimiter{[]\mathclose}{operators}{5D}{largesymbols}{03}
```

The next two are considered to be relations when not used in the context of a delimiter! And worse, they do even represent different glyphs when being used as delimiter and not as delimiter. This is a user level syntax inherited from plain `TeX`. Therefore we explicitly redefine the math symbol definitions for these symbols afterwards.

```
178 \DeclareMathDelimiter{<}\mathopen{symbols}{68}{largesymbols}{0A}
179 \DeclareMathDelimiter{>}\mathclose{symbols}{69}{largesymbols}{0B}
180 \DeclareMathSymbol{<}\mathrel{letters}{3C}
181 \DeclareMathSymbol{>}\mathrel{letters}{3E}
```

And here is another case where the non-delimiter version produces a glyph different from the delimiter version.

```
182 \DeclareMathDelimiter{/}\mathord{operators}{2F}{largesymbols}{0E}
183 \DeclareMathSymbol{/}\mathord{letters}{3D}

184 \DeclareMathDelimiter{|}\mathord{symbols}{6A}{largesymbols}{0C}

185 \expandafter\DeclareMathDelimiter\@backslashchar
186 \mathord{symbols}{6E}{largesymbols}{0F}
```

N.B. `{` and `}` should NOT get delcodes; otherwise parameter grouping fails!

### 40.4 Symbols accessed via control sequences

#### 40.4.1 Greek letters

```
187 \DeclareMathSymbol{\alpha}\mathord{letters}{0B}
188 \DeclareMathSymbol{\beta}\mathord{letters}{0C}
189 \DeclareMathSymbol{\gamma}\mathord{letters}{0D}
190 \DeclareMathSymbol{\delta}\mathord{letters}{0E}
191 \DeclareMathSymbol{\epsilon}\mathord{letters}{0F}
192 \DeclareMathSymbol{\zeta}\mathord{letters}{10}
193 \DeclareMathSymbol{\eta}\mathord{letters}{11}
194 \DeclareMathSymbol{\theta}\mathord{letters}{12}
195 \DeclareMathSymbol{\iota}\mathord{letters}{13}
196 \DeclareMathSymbol{\kappa}\mathord{letters}{14}
197 \DeclareMathSymbol{\lambda}\mathord{letters}{15}
198 \DeclareMathSymbol{\mu}\mathord{letters}{16}
199 \DeclareMathSymbol{\nu}\mathord{letters}{17}
200 \DeclareMathSymbol{\xi}\mathord{letters}{18}
201 \DeclareMathSymbol{\pi}\mathord{letters}{19}
202 \DeclareMathSymbol{\rho}\mathord{letters}{1A}
203 \DeclareMathSymbol{\sigma}\mathord{letters}{1B}
204 \DeclareMathSymbol{\tau}\mathord{letters}{1C}
205 \DeclareMathSymbol{\upsilon}\mathord{letters}{1D}
206 \DeclareMathSymbol{\phi}\mathord{letters}{1E}
```

```

207 \DeclareMathSymbol{\chi}{\mathord}{letters}{1F}
208 \DeclareMathSymbol{\psi}{\mathord}{letters}{20}
209 \DeclareMathSymbol{\omega}{\mathord}{letters}{21}
210 \DeclareMathSymbol{\varepsilon}{\mathord}{letters}{22}
211 \DeclareMathSymbol{\vartheta}{\mathord}{letters}{23}
212 \DeclareMathSymbol{\varpi}{\mathord}{letters}{24}
213 \DeclareMathSymbol{\varrho}{\mathord}{letters}{25}
214 \DeclareMathSymbol{\varsigma}{\mathord}{letters}{26}
215 \DeclareMathSymbol{\varphi}{\mathord}{letters}{27}
216 \DeclareMathSymbol{\Gamma}{\mathalpha}{operators}{00}
217 \DeclareMathSymbol{\Delta}{\mathalpha}{operators}{01}
218 \DeclareMathSymbol{\Theta}{\mathalpha}{operators}{02}
219 \DeclareMathSymbol{\Lambda}{\mathalpha}{operators}{03}
220 \DeclareMathSymbol{\Xi}{\mathalpha}{operators}{04}
221 \DeclareMathSymbol{\Pi}{\mathalpha}{operators}{05}
222 \DeclareMathSymbol{\Sigma}{\mathalpha}{operators}{06}
223 \DeclareMathSymbol{\Upsilon}{\mathalpha}{operators}{07}
224 \DeclareMathSymbol{\Phi}{\mathalpha}{operators}{08}
225 \DeclareMathSymbol{\Psi}{\mathalpha}{operators}{09}
226 \DeclareMathSymbol{\Omega}{\mathalpha}{operators}{0A}

```

#### 40.4.2 Ordinary symbols

```

227 \DeclareMathSymbol{\aleph}{\mathord}{symbols}{40}
228 \def\hbar{{\mathchar'26\mkern-9mu h}}
229 \DeclareMathSymbol{\imath}{\mathord}{letters}{7B}
230 \DeclareMathSymbol{\jmath}{\mathord}{letters}{7C}
231 \DeclareMathSymbol{\ell}{\mathord}{letters}{60}
232 \DeclareMathSymbol{\wp}{\mathord}{letters}{7D}
233 \DeclareMathSymbol{\Re}{\mathord}{symbols}{3C}
234 \DeclareMathSymbol{\Im}{\mathord}{symbols}{3D}
235 \DeclareMathSymbol{\partial}{\mathord}{letters}{40}
236 \DeclareMathSymbol{\infty}{\mathord}{symbols}{31}
237 \DeclareMathSymbol{\prime}{\mathord}{symbols}{30}
238 \DeclareMathSymbol{\emptyset}{\mathord}{symbols}{3B}
239 \DeclareMathSymbol{\nabla}{\mathord}{symbols}{72}
240 \def\surd{{\mathchar"1270}}
241 \DeclareMathSymbol{\top}{\mathord}{symbols}{3E}
242 \DeclareMathSymbol{\bot}{\mathord}{symbols}{3F}
243 \def\angle{{\vbox{\ialign{$\m@th\scriptstyle##$\crrc
244 \not\mathrel{\mkern14mu}\crrc
245 \noalign{\nointerlineskip}
246 \mkern2.5mu\leaders\hrule \@height.34pt\hfill\mkern2.5mu\crrc}}}}
247 \DeclareMathSymbol{\triangle}{\mathord}{symbols}{34}
248 \DeclareMathSymbol{\forall}{\mathord}{symbols}{38}
249 \DeclareMathSymbol{\exists}{\mathord}{symbols}{39}
250 \DeclareMathSymbol{\neg}{\mathord}{symbols}{3A}
251 \let\not=\neg
252 \DeclareMathSymbol{\flat}{\mathord}{letters}{5B}
253 \DeclareMathSymbol{\natural}{\mathord}{letters}{5C}
254 \DeclareMathSymbol{\sharp}{\mathord}{letters}{5D}
255 \DeclareMathSymbol{\clubsuit}{\mathord}{symbols}{7C}
256 \DeclareMathSymbol{\diamondsuit}{\mathord}{symbols}{7D}
257 \DeclareMathSymbol{\heartsuit}{\mathord}{symbols}{7E}
258 \DeclareMathSymbol{\spadesuit}{\mathord}{symbols}{7F}

```

### 40.4.3 Large Operators

```

259 \DeclareMathSymbol{\coprod}{\mathop}{largesymbols}{60}
260 \DeclareMathSymbol{\bigvee}{\mathop}{largesymbols}{57}
261 \DeclareMathSymbol{\bigwedge}{\mathop}{largesymbols}{56}
262 \DeclareMathSymbol{\biguplus}{\mathop}{largesymbols}{55}
263 \DeclareMathSymbol{\bigcap}{\mathop}{largesymbols}{54}
264 \DeclareMathSymbol{\bigcup}{\mathop}{largesymbols}{53}
265 \DeclareMathSymbol{\intop}{\mathop}{largesymbols}{52}
266 \def\int{\intop\nolimits}
267 \DeclareMathSymbol{\prod}{\mathop}{largesymbols}{51}
268 \DeclareMathSymbol{\sum}{\mathop}{largesymbols}{50}
269 \DeclareMathSymbol{\bigotimes}{\mathop}{largesymbols}{4E}
270 \DeclareMathSymbol{\bigoplus}{\mathop}{largesymbols}{4C}
271 \DeclareMathSymbol{\bigodot}{\mathop}{largesymbols}{4A}
272 \DeclareMathSymbol{\ointop}{\mathop}{largesymbols}{48}
273 \def\oint{\ointop\nolimits}
274 \DeclareMathSymbol{\bigsqcup}{\mathop}{largesymbols}{46}
275 \DeclareMathSymbol{\smallint}{\mathop}{symbols}{73}

```

### 40.4.4 Binary symbols

```

276 \DeclareMathSymbol{\triangleleft}{\mathbin}{letters}{2F}
277 \DeclareMathSymbol{\triangleright}{\mathbin}{letters}{2E}
278 \DeclareMathSymbol{\bigtriangleup}{\mathbin}{symbols}{34}
279 \DeclareMathSymbol{\bigtriangledown}{\mathbin}{symbols}{35}
280 \let \varbigtriangledown \bigtriangledown
281 \let \varbigtriangleup \bigtriangleup

```

These last two synonyms are needed because the `stamryrd` package redefines them as Operators.

```

282 \DeclareMathSymbol{\wedge}{\mathbin}{symbols}{5E}
283 \let\land=\wedge
284 \DeclareMathSymbol{\vee}{\mathbin}{symbols}{5F}
285 \let\lor=\vee
286 \DeclareMathSymbol{\cap}{\mathbin}{symbols}{5C}
287 \DeclareMathSymbol{\cup}{\mathbin}{symbols}{5B}
288 \DeclareMathSymbol{\ddagger}{\mathbin}{symbols}{7A}
289 \DeclareMathSymbol{\dagger}{\mathbin}{symbols}{79}
290 \DeclareMathSymbol{\sqcap}{\mathbin}{symbols}{75}
291 \DeclareMathSymbol{\sqcup}{\mathbin}{symbols}{74}
292 \DeclareMathSymbol{\uplus}{\mathbin}{symbols}{5D}
293 \DeclareMathSymbol{\amalg}{\mathbin}{symbols}{71}
294 \DeclareMathSymbol{\diamond}{\mathbin}{symbols}{05}
295 \DeclareMathSymbol{\bullet}{\mathbin}{symbols}{0F}
296 \DeclareMathSymbol{\wr}{\mathbin}{symbols}{6F}
297 \DeclareMathSymbol{\div}{\mathbin}{symbols}{04}
298 \DeclareMathSymbol{\odot}{\mathbin}{symbols}{0C}
299 \DeclareMathSymbol{\oslash}{\mathbin}{symbols}{0B}
300 \DeclareMathSymbol{\otimes}{\mathbin}{symbols}{0A}
301 \DeclareMathSymbol{\ominus}{\mathbin}{symbols}{09}
302 \DeclareMathSymbol{\oplus}{\mathbin}{symbols}{08}
303 \DeclareMathSymbol{\mp}{\mathbin}{symbols}{07}
304 \DeclareMathSymbol{\pm}{\mathbin}{symbols}{06}
305 \DeclareMathSymbol{\circ}{\mathbin}{symbols}{0E}
306 \DeclareMathSymbol{\bigcirc}{\mathbin}{symbols}{0D}

```

```

307 \DeclareMathSymbol{\setminus}{\mathbin}{symbols}{"6E}
308 \DeclareMathSymbol{\cdot}{\mathbin}{symbols}{"01}
309 \DeclareMathSymbol{\ast}{\mathbin}{symbols}{"03}
310 \DeclareMathSymbol{\times}{\mathbin}{symbols}{"02}
311 \DeclareMathSymbol{\star}{\mathbin}{letters}{"3F}

```

#### 40.4.5 Relations

```

312 \DeclareMathSymbol{\propto}{\mathrel}{symbols}{"2F}
313 \DeclareMathSymbol{\sqsubseteq}{\mathrel}{symbols}{"76}
314 \DeclareMathSymbol{\sqsupseteq}{\mathrel}{symbols}{"77}
315 \DeclareMathSymbol{\parallel}{\mathrel}{symbols}{"6B}
316 \DeclareMathSymbol{\mid}{\mathrel}{symbols}{"6A}
317 \DeclareMathSymbol{\dashv}{\mathrel}{symbols}{"61}
318 \DeclareMathSymbol{\vdash}{\mathrel}{symbols}{"60}
319 \DeclareMathSymbol{\nearrow}{\mathrel}{symbols}{"25}
320 \DeclareMathSymbol{\searrow}{\mathrel}{symbols}{"26}
321 \DeclareMathSymbol{\nwarrow}{\mathrel}{symbols}{"2D}
322 \DeclareMathSymbol{\swarrow}{\mathrel}{symbols}{"2E}
323 \DeclareMathSymbol{\Leftrightarrow}{\mathrel}{symbols}{"2C}
324 \DeclareMathSymbol{\Leftarrow}{\mathrel}{symbols}{"28}
325 \DeclareMathSymbol{\Rightarrow}{\mathrel}{symbols}{"29}
326 \def\neq{\not=} \let\ne=\neq
327 \DeclareMathSymbol{\leq}{\mathrel}{symbols}{"14}
328 \let\le=\leq
329 \DeclareMathSymbol{\geq}{\mathrel}{symbols}{"15}
330 \let\ge=\geq
331 \DeclareMathSymbol{\succ}{\mathrel}{symbols}{"1F}
332 \DeclareMathSymbol{\prec}{\mathrel}{symbols}{"1E}
333 \DeclareMathSymbol{\approx}{\mathrel}{symbols}{"19}
334 \DeclareMathSymbol{\succeq}{\mathrel}{symbols}{"17}
335 \DeclareMathSymbol{\preceq}{\mathrel}{symbols}{"16}
336 \DeclareMathSymbol{\supset}{\mathrel}{symbols}{"1B}
337 \DeclareMathSymbol{\subset}{\mathrel}{symbols}{"1A}
338 \DeclareMathSymbol{\supseteq}{\mathrel}{symbols}{"13}
339 \DeclareMathSymbol{\subseteq}{\mathrel}{symbols}{"12}
340 \DeclareMathSymbol{\in}{\mathrel}{symbols}{"32}
341 \DeclareMathSymbol{\ni}{\mathrel}{symbols}{"33}
342 \let\owns=\ni
343 \DeclareMathSymbol{\gg}{\mathrel}{symbols}{"1D}
344 \DeclareMathSymbol{\ll}{\mathrel}{symbols}{"1C}
345 \DeclareMathSymbol{\not}{\mathrel}{symbols}{"36}
346 \DeclareMathSymbol{\leftrightharpoonup}{\mathrel}{symbols}{"24}
347 \DeclareMathSymbol{\leftarrow}{\mathrel}{symbols}{"20}
348 \let\gets=\leftarrow
349 \DeclareMathSymbol{\rightarrow}{\mathrel}{symbols}{"21}
350 \let\to=\rightarrow
351 \DeclareMathSymbol{\mapstochar}{\mathrel}{symbols}{"37}
352 \def\mapsto{\mapstochar\rightarrow}
353 \DeclareMathSymbol{\sim}{\mathrel}{symbols}{"18}
354 \DeclareMathSymbol{\simeq}{\mathrel}{symbols}{"27}
355 \DeclareMathSymbol{\perp}{\mathrel}{symbols}{"3F}
356 \DeclareMathSymbol{\equiv}{\mathrel}{symbols}{"11}
357 \DeclareMathSymbol{\asymp}{\mathrel}{symbols}{"10}
358 \DeclareMathSymbol{\smile}{\mathrel}{letters}{"5E}

```

```

359 \DeclareMathSymbol{\frown}{\mathrel}{letters}{"5F}
360 \DeclareMathSymbol{\leftharpoonup}{\mathrel}{letters}{"28}
361 \DeclareMathSymbol{\leftharpoondown}{\mathrel}{letters}{"29}
362 \DeclareMathSymbol{\rightharpoonup}{\mathrel}{letters}{"2A}
363 \DeclareMathSymbol{\rightharpoondown}{\mathrel}{letters}{"2B}

```

Here cometh much profligate robustification of math constructs. Warning: some of these commands may become non-robust if an AMS package is loaded.

Further potential problems: some math font packages may make unfortunate assumptions about some of these definitions that are not true of the robust versions we need.

```

364 \DeclareRobustCommand
365   \cong{\mathrel{\mathpalette\@vereq\sim}} % congruence sign
366 \def\@vereq#1#2{\lower.5\p@\vbox{\lineskiplimit\maxdimen\lineskip-.5\p@
367   \ialign{${\m@th#1\hfil##\hfil$\crcr#2\crcr=\crcr$}}}
368 \DeclareRobustCommand
369   \notin{\mathrel{\m@th\mathpalette@cncel\in}}
370 \def@cncel#1#2{\m@th\oalign{${\hfil#1\mkern1mu/\hfil$\crcr$#1#2$}}
371 \DeclareRobustCommand
372   \rightleftharpoons{\mathrel{\mathpalette\rlh@{}}}
373 \def\rlh@#1{\vcenter{\m@th\hbox{\oalign{\raise2pt
374   \hbox{${\hfil#1\rightharpoonup$}\crcr
375   ${\hfil#1\leftharpoondown$}\crcr}}}
376 \DeclareRobustCommand
377   \doteq{\buildrel\textstyle.\over=}

```

#### 40.4.6 Arrows

```

378 \DeclareRobustCommand
379   \joinrel{\mathrel{\mkern-3mu}}
380 \DeclareRobustCommand
381   \relbar{\mathrel{\smash-}} % \smash, because -
382                               % has the same height as +

```

In contrast to `plain.tex` `\Relbar` got braces around the equal sign to guard against it being “math active” expanding to `\futurelet...`. This might be the case when packages are implementing shorthands for math, e.g. `=>` meaning `\Rightarrow` etc. It would actually be better not to use `=` in such definitions but instead define something like `\mathequalsign` and use this. However we can’t do this now as it would break other math layouts where characters are in different places (since those wouldn’t know about the need for a new command name).

```

383 \DeclareRobustCommand
384   \Relbar{\mathrel{=}}
385 \DeclareMathSymbol{\lhook}{\mathrel}{letters}{"2C}
386 \def\hookrightarrow{\lhook\joinrel\rightarrow}
387 \DeclareMathSymbol{\rhook}{\mathrel}{letters}{"2D}
388 \def\hookleftarrow{\leftarrow\joinrel\rhook}
389 \DeclareRobustCommand
390   \bowtie{\mathrel{\triangleright\joinrel\mathrel{\triangleleft}}}
391 \DeclareRobustCommand
392   \models{\mathrel{||}\joinrel\Relbar}
393 \DeclareRobustCommand
394   \Longrightarrow{\Relbar\joinrel\rightarrow}

```

LaTeX Change: `\longrightarrow` and `\longleftarrow` redefined to make then robust.

```

395 \DeclareRobustCommand\longrightarrow
396     {\relbar\joinrel\rightarrow}
397 \DeclareRobustCommand\longleftarrow
398     {\leftarrow\joinrel\relbar}
399 \DeclareRobustCommand
400     \Llongleftarrow{\Leftarrow\joinrel\Relbar}
401 \DeclareRobustCommand
402     \longmapsto{\mapstochar\longrightarrow}
403 \DeclareRobustCommand
404     \longleftrightarrow{\leftarrow\joinrel\rightarrow}
405 \DeclareRobustCommand
406     \Llongleftrightarrow{\Leftarrow\joinrel\Rightarrow}
407 \DeclareRobustCommand
408     \iff{\;\Longleftrightarrow\;}

```

#### 40.4.7 Punctuation symbols

```

409 \DeclareMathSymbol{\ldotp}{\mathpunct}{letters}{"3A}
410 \DeclareMathSymbol{\cdotp}{\mathpunct}{symbols}{"01}
411 \DeclareMathSymbol{\colon}{\mathpunct}{operators}{"3A}

This is commented out, since \ldots is now defined in ltoutenc.dtx.
412 %\def\@ldots{\mathinner{\ldotp\ldotp\ldotp}}
413 %\DeclareRobustCommand\ldots
414 %    {\relax\ifmmode\@ldots\else\mbox{$\m@th\@ldots$,}\fi}
415 \DeclareRobustCommand
416     \cdots{\mathinner{\cdotp\cdotp\cdotp}}
417 \DeclareRobustCommand
418     \vdots{\vbox{\baselineskip4p@ \lineskiplimit\z@
419         \kern6p@\hbox{.}\hbox{.}\hbox{.}}}
420 \DeclareRobustCommand
421     \ddots{\mathinner{\mkern1mu\raise7p@
422         \vbox{\kern7p@\hbox{.}}\mkern2mu
423         \raise4p@\hbox{.}\mkern2mu\raise\p@\hbox{.}\mkern1mu}}

```

#### 40.4.8 Math accents

```

424 \DeclareMathAccent{\acute}{\mathalpha}{operators}{"13}
425 \DeclareMathAccent{\grave}{\mathalpha}{operators}{"12}
426 \DeclareMathAccent{\ddot}{\mathalpha}{operators}{"7F}
427 \DeclareMathAccent{\tilde}{\mathalpha}{operators}{"7E}
428 \DeclareMathAccent{\bar}{\mathalpha}{operators}{"16}
429 \DeclareMathAccent{\breve}{\mathalpha}{operators}{"15}
430 \DeclareMathAccent{\check}{\mathalpha}{operators}{"14}
431 \DeclareMathAccent{\hat}{\mathalpha}{operators}{"5E}
432 \DeclareMathAccent{\vec}{\mathord}{letters}{"7E}
433 \DeclareMathAccent{\dot}{\mathalpha}{operators}{"5F}
434 \DeclareMathAccent{\widetilde}{\mathord}{largesymbols}{"65}
435 \DeclareMathAccent{\widehat}{\mathord}{largesymbols}{"62}

```

For some reason plain T<sub>E</sub>X never bothered to provide a ring accent in math (although it is available in the fonts), but since we got a request for it here we go:

```

436 \DeclareMathAccent{\mathring}{\mathalpha}{operators}{"17}

```

#### 40.4.9 Radicals

```
437 \DeclareMathRadical{\sqrtsign}{symbols}{"70}{largesymbols}{"70}
```

#### 40.4.10 Over and under something, etc

```
438 \def\overrightarrow#1{\vbox{\m@th\ialign{##\crrc
439   \rightarrowfill\crrc\noalign{\kern-\p@\nointerlineskip}
440   $\hfil\displaystyle{#1}\hfil$\crrc}}}%
441 \def\overleftarrow#1{\vbox{\m@th\ialign{##\crrc
442   \leftarrowfill\crrc\noalign{\kern-\p@\nointerlineskip}}%
443   $\hfil\displaystyle{#1}\hfil$\crrc}}}%
444 \def\overbrace#1{\mathop{\vbox{\m@th\ialign{##\crrc\noalign{\kern3\p@}%
445   \downbracefill\crrc\noalign{\kern3\p@\nointerlineskip}}%
446   $\hfil\displaystyle{#1}\hfil$\crrc}}}\limits}
447 \def\underbrace#1{\mathop{\vtop{\m@th\ialign{##\crrc
448   $\hfil\displaystyle{#1}\hfil$\crrc
449   \noalign{\kern3\p@\nointerlineskip}}%
450   \upbracefill\crrc\noalign{\kern3\p@}}}}\limits}
```

(quite a waste of tokens, IMHO — Frank)

```
451 \def\skew#1#2#3{{\muskip\z@#1mu\divide\muskip\z@\tw@ \mkern\muskip\z@
452   #2{\mkern-\muskip\z@#3}\mkern\muskip\z@}\mkern-\muskip\z@}{}}
453 \def\rightarrowfill{$\m@th\smash-\mkern-7mu%
454   \cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill
455   \mkern-7mu\mathord\rightarrow$}
456 \def\leftarrowfill{$\m@th\mathord\leftarrow\mkern-7mu%
457   \cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill
458   \mkern-7mu\smash-$}
459 \DeclareMathSymbol{\braceld}{\mathord}{largesymbols}{"7A}
460 \DeclareMathSymbol{\bracerd}{\mathord}{largesymbols}{"7B}
461 \DeclareMathSymbol{\bracelu}{\mathord}{largesymbols}{"7C}
462 \DeclareMathSymbol{\braceru}{\mathord}{largesymbols}{"7D}
463 \def\downbracefill{$\m@th \setbox\z@\hbox{$\braceld$}%
464   \braceld\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru
465   \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd$}
466 \def\upbracefill{$\m@th \setbox\z@\hbox{$\braceld$}%
467   \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd
468   \braceld\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru$}
```

#### 40.4.11 Delimiters

```
469 \DeclareMathDelimiter{\lmoustache} % top from (, bottom from )
470   {\mathopen}{largesymbols}{"7A}{largesymbols}{"40}
471 \DeclareMathDelimiter{\rmoustache} % top from ), bottom from (
472   {\mathclose}{largesymbols}{"7B}{largesymbols}{"41}
473 \DeclareMathDelimiter{\arrowvert} % arrow without arrowheads
474   {\mathord}{symbols}{"6A}{largesymbols}{"3C}
475 \DeclareMathDelimiter{\Arrowvert} % double arrow without arrowheads
476   {\mathord}{symbols}{"6B}{largesymbols}{"3D}
477 \DeclareMathDelimiter{\Vert}
478   {\mathord}{symbols}{"6B}{largesymbols}{"0D}
479 \let\|=\Vert
480 \DeclareMathDelimiter{\vert}
481   {\mathord}{symbols}{"6A}{largesymbols}{"0C}
482 \DeclareMathDelimiter{\uparrow}
483   {\mathrel}{symbols}{"22}{largesymbols}{"78}
484 \DeclareMathDelimiter{\downarrow}
```

```

485 {\mathrel}{symbols}{"23}{largesymbols}{"79}
486 \DeclareMathDelimiter{\updownarrow}
487 {\mathrel}{symbols}{"6C}{largesymbols}{"3F}
488 \DeclareMathDelimiter{\Uparrow}
489 {\mathrel}{symbols}{"2A}{largesymbols}{"7E}
490 \DeclareMathDelimiter{\Downarrow}
491 {\mathrel}{symbols}{"2B}{largesymbols}{"7F}
492 \DeclareMathDelimiter{\Updownarrow}
493 {\mathrel}{symbols}{"6D}{largesymbols}{"77}
494 \DeclareMathDelimiter{\backslash} % for double coset G\backslash H
495 {\mathord}{symbols}{"6E}{largesymbols}{"0F}
496 \DeclareMathDelimiter{\rangle}
497 {\mathclose}{symbols}{"69}{largesymbols}{"0B}
498 \DeclareMathDelimiter{\langle}
499 {\mathopen}{symbols}{"68}{largesymbols}{"0A}
500 \DeclareMathDelimiter{\rbrace}
501 {\mathclose}{symbols}{"67}{largesymbols}{"09}
502 \DeclareMathDelimiter{\lbrace}
503 {\mathopen}{symbols}{"66}{largesymbols}{"08}
504 \DeclareMathDelimiter{\rceil}
505 {\mathclose}{symbols}{"65}{largesymbols}{"07}
506 \DeclareMathDelimiter{\lceil}
507 {\mathopen}{symbols}{"64}{largesymbols}{"06}
508 \DeclareMathDelimiter{\rfloor}
509 {\mathclose}{symbols}{"63}{largesymbols}{"05}
510 \DeclareMathDelimiter{\lfloor}
511 {\mathopen}{symbols}{"62}{largesymbols}{"04}

```

`\lgroup` There are three plain  $\TeX$  delimiters which are not fully supported by NFSS, since they partly point into a bold `cmr` font. Allocating a full symbol font, just `\rgroup` to have three delimiters seems a bit too much given the limited space available. `\bracevert` For this reason only the extensible sizes are supported. If this is not desired one can use, without losing portability, define `\mathbf` and `\mathtt` as font symbol alphabet (setting up `cmr/bx/n` and `cmth/m/n` as symbol fonts first) and modify the delimiter declarations to point with their small variant to those symbol fonts. (This is done in `oldlfont.dtx` so look there for examples.)

```

512 \DeclareMathDelimiter{\lgroup} % extensible ( with sharper tips
513 {\mathopen}{largesymbols}{"3A}{largesymbols}{"3A}
514 \DeclareMathDelimiter{\rgroup} % extensible ) with sharper tips
515 {\mathclose}{largesymbols}{"3B}{largesymbols}{"3B}
516 \DeclareMathDelimiter{\bracevert} % the vertical bar that extends braces
517 {\mathord}{largesymbols}{"3E}{largesymbols}{"3E}

```

## 40.5 Math versions of text commands

The `\mathunderscore` here is really a text definition, so it has been put back into `ltoutenc.dtx` (by Chris, 30/04/97) and should be removed from here.

These symbols are the math versions of text commands such as `\P`, `\$`, etc.

```

\mathparagraph These math symbols are not in plain  $\TeX$ .
\mathsection 518 \DeclareMathSymbol{\mathparagraph}{\mathord}{symbols}{"7B}
\mathdollar 519 \DeclareMathSymbol{\mathsection}{\mathord}{symbols}{"78}
\mathsterling 520 \DeclareMathSymbol{\mathdollar}{\mathord}{operators}{"24}
\mathunderscore

```

```

521 \def\mathsterling{\mathit{\mathchar"7024}}
522 \def\mathunderscore{\kern.06em\vbox{\hrule\@width.3em}}

```

`\mathellipsis` This is plain T<sub>E</sub>X's `\ldots`.

```

523 \def\mathellipsis{\mathinner{\ldotp\ldotp\ldotp}}%

```

## 40.6 Other special functions and parameters

### 40.6.1 Biggggg

```

524 \def\big#1{{\hbox{$\left#1\vbox to8.5\p@{\right.\n@space$}}}}
525 \def\Big#1{{\hbox{$\left#1\vbox to11.5\p@{\right.\n@space$}}}}
526 \def\bigg#1{{\hbox{$\left#1\vbox to14.5\p@{\right.\n@space$}}}}
527 \def\Bigg#1{{\hbox{$\left#1\vbox to17.5\p@{\right.\n@space$}}}}
528 \def\n@space{\null\delimiterspace\z@ \m@th}

```

### 40.6.2 The log-like functions

`\operator@font` The `\operator@font` determines the symbol font used for log-like functions.

```

529 \def\operator@font{\mathgroup\symoperators}

```

### 40.6.3 Parameters

```

530 \thinmuskip=3mu
531 \medmuskip=4mu plus 2mu minus 4mu
532 \thickmuskip=5mu plus 5mu

```

This finishes the low-level setup in `fontmath.ltx`.

```

533 \end{math}

```

## 41 Default cfg files

We provide default `cfg` files here to ensure that on installations that search large file trees we do not pick up some strange customisation files from somewhere.

```

534 (*cfgtext | cfgmath | cfgprel)
535 %%
536 %%
537 %%
538 %% Load the standard setup:
539 %%
540 (+cfgtext)\input{fonttext.ltx}
541 (+cfgmath)\input{fontmath.ltx}
542 (+cfgprel)\input{preload.ltx}
543 %%
544 %% Small changes could go here; see documentation in cfgguide.tex for
545 %% allowed modifications.
546 %%
547 %% In particular it is not allowed to misuse this configuration file
548 %% to modify internal LaTeX commands!
549 %%
550 %% If you use this file as the basis for configuration please change
551 %% the \ProvidesFile lines to clearly identify your modification, e.g.,
552 %%
553 (+cfgtext)%% \ProvidesFile{fonttext.cfg}[2001/06/01
554 (+cfgmath)%% \ProvidesFile{fontmath.cfg}[2001/06/01

```

```

555 <+cfgprel>%%    \ProvidesFile{preload.cfg}[2001/06/01
556 %%                                Customised local font setup]
557 %%
558 %%
559 </cfgtext | cfgmath | cfgprel>

```

# File u

## preload.dtx

### 42 Overview

This file contains an number of possible settings for preloading fonts during installation of NFSS2 (which is used by L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>). It will be used to generate the following files:

preload.min	minimal subset of fonts necessary to run NFSS2
preload.ori	preload of CM fonts similar to the old <code>lfonts.tex</code>
preload.ltx	The standard selection of preloads
cmpreloa.xpt	preload of CM fonts for 10pt document size
cmpreloa.xip	preload of CM fonts for 11pt document size
cmpreloa.xii	preload of CM fonts for 12pt document size
dcpreloa.xpt	preload of DC fonts for 10pt size
dcpreloa.xip	preload of DC fonts for 11pt size
dcpreloa.xii	preload of DC fonts for 12pt size

These files are for installations that make use of Computer Modern fonts either old encoding (OT1) or Cork encoding (T1). The Computer Modern fonts with Cork encoding are known as DC-fonts.

Most important is `preload.ltx` which is used during format generation. You are *not* allowed to change this file.

### 43 Customization

You can customize the preloaded fonts in your L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> system by installing a file with the name `preload.cfg`. If this file exists it will be used in place of the system file `preload.ltx`. You can, for example, copy one of the files mentioned above (that can be generated from this source) to `preload.cfg`.

Or you can define completely other preloads. In that case start from `preload.min` since that contains the fonts that have to be preloaded by \*all\* L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> systems.

Avoid using `preload.ori`, it will load so many fonts that on most installations it is nearly impossible to load other font families afterwards. This file is only generated to show what fonts have been preloaded by L<sup>A</sup>T<sub>E</sub>X 2.09.

If you normally use other fonts than Computer Modern `preload.min` might be best.

**Warning:** If you preload fonts with encodings other than the normally supported encodings you have to declare that encoding in a `fontdef.cfg` configuration file (see the documentation in the file `fontdef.dtx`). Adding an extra encoding to the format might produce non-portable documents, thus this should be avoided if possible.

## 44 Module switches for the DOCSTRIP program

The DOCSTRIP will generate the above file from this source using the following module directives:

driver	produce a documentation driver file
preload	produce a preload... file
cm	for OT1 encoded Computer Modern
dc	for T1 encoded Computer Modern
min	produce minimal subset
xpt	produce 10pt preloads
xipt	produce 11pt preloads
xipt	produce 12pt preloads
ori	produce preloads similar to old <code>lfonts.tex</code>
tex	produce <code>preload.ltx</code>

A typical DOCSTRIP command file would then have entries like:

```
\generateFile{preload.min}{t}{\from{preload.dtx}{preload,min}}
```

for generating preload files.

## 45 A driver for this document

The next bit of code contains the documentation driver file for  $\text{\TeX}$ , i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the DOCSTRIP program.

```
1 (*driver)
2 \documentclass{ltxdoc}
3 %\OnlyDescription % comment out for implementation details
4 \begin{document}
5   \DocInput{preload.dtx}
6 \end{document}
7 \end{driver}
```

## 46 The code

We begin by loading the math extension font (`cmex10`) and the  $\text{\LaTeX}$  line and circle fonts. It is necessary to do this explicitly since these are used by `lplain.tex` and `latex.tex`. Since the internal font name contains / characters and digits we construct the name via `\csname`. These are the only fonts (!) that must be loaded in this file.

All `\DeclarePreloadSizes` can be removed or others can be added, they only influence the processing speed.

```
8 \expandafter\font\csname OMX/cmex/m/n/10\endcsname=cmex10\relax
9 \font\tenln =line10 \font\tenlnw =linew10\relax
10 \font\tenclrc=lcircle10 \font\tenclrcw=lcirclew10\relax
```

The above fonts should not be touched but anything below this point here in the preload suggestions can be modified without any problems.

```
11 (-tex)%*****
```

```

12 <-tex>% Start any modification below this point **
13 <-tex>%*****
14 <-tex>
15 %%
16 %% Computer Modern Roman:
17 %%-----
18 <*ori>
19 \DeclarePreloadSizes{OT1}{cmr}{m}{n}
20     {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74,24.88}
21 \DeclarePreloadSizes{OT1}{cmr}{bx}{n}{9,10,10.95,12,14.4,17.28}
22 \DeclarePreloadSizes{OT1}{cmr}{m}{sl}{10,10.95,12}
23 \DeclarePreloadSizes{OT1}{cmr}{m}{it}{7,8,9,10,10.95,12}
24 </ori>
25 <+xpt & cm> \DeclarePreloadSizes{OT1}{cmr}{m}{n}{5,7,10}
26 <+xpt & dc> \DeclarePreloadSizes{T1}{cmr}{m}{n}{5,7,10}
27 <+xipt & cm> \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,10.95}
28 <+xipt & dc> \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,10.95}
29 <+xiipt & cm> \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,12}
30 <+xiipt & dc> \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,12}
31 %%
32 %% Computer Modern Sans:
33 %%-----
34 <+ori> \DeclarePreloadSizes{OT1}{cmss}{m}{n}{10,10.95,12}
35 %%
36 %% Computer Modern Typewriter:
37 %%-----
38 <+ori> \DeclarePreloadSizes{OT1}{cmtt}{m}{n}{9,10,10.95,12}
39 %%
40 %% Computer Modern Math:
41 %%-----
42 <*ori>
43 \DeclarePreloadSizes{OML}{cmm}{m}{it}
44     {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
45 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}
46     {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
47 </ori>

```

The math fonts are the same for both DC and CM fonts. So far there isn't an agreed on standard.

```

48 <*xpt>
49 \DeclarePreloadSizes{OML}{cmm}{m}{it}{5,7,10}
50 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{5,7,10}
51 </xpt>
52 <*xipt>
53 \DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,10.95}
54 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,10.95}
55 </xipt>
56 <*xiipt>
57 \DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,12}
58 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,12}
59 </xiipt>
60 %%
61 %% LaTeX symbol fonts:
62 %%-----

```

```
63 <*ori>
64 \DeclarePreloadSizes{U}{\lasy}{m}{n}
65      {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
66 </ori>
67 </preload>
```

# File v

## ltfntcmd.dtx

### Abstract

The commands defined in this file `ltfntcmd` are part of the kernel code for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>/NFSS2.

It is also meant to serve as documentation for package writers since it demonstrates how to define high-level font changing commands using a small number of creator functions.

## 47 Introduction

Font changes such as `\bfseries`, `\sffamily`, etc. are declarations; this means that their scope is delimited by the grouping structure, either by the next `\end` of some environment or by explicitly using a group, e.g., writing something like `{\bfseries...}` in the source. If you make the mistake of writing `\bfseries{...}` (thinking of `\bfseries` as a command with one argument) then the result is rather striking.

Font declarations are an artifact of the T<sub>E</sub>X system and for several reasons it is better to avoid them on the user level whenever possible. In L<sup>A</sup>T<sub>E</sub>X3 they will probably all be replaced by environments and by font commands taking one argument.

This file defines a creator function for such declarative font switches. This function creates commands which can be used in both math and text.

This file also defines a number of high-level commands (all starting with `\text..`) that have one argument and typeset this argument in the requested way. Thus these commands are for typesetting short pieces of text in a specific family, series or shape. These are all produced as examples of the use of a creator function which is itself also defined in this file.

Table 1 shows all these high-level commands in action. A further advantage of using these commands is that they automatically take care of any necessary italic correction on either side of their argument.

Thus, when using such commands, one does not have to worry about forgetting the italic correction when changing fonts. Only in very few situations is this additional space wrong but, for example, most typographers recommend omitting the italic correction if a small punctuation character, like a comma, directly follows the font change. Since the amount of correction required is partly a matter of taste, you can define in what situations the italic correction should be suppressed. This is done by putting the characters that should cancel a preceding italic correction in the list `\nocorrlist`.<sup>7</sup> The default definition for this list is produced by the following.

```
\newcommand \nocorrlist {,.}
```

---

<sup>7</sup>Any package that changes the `\catcode` of a character inside `\nocorrlist` must then explicitly reset the list. Otherwise the changed character will no longer be recognized by the suppression algorithm.

<i>Command</i>	<i>Corresponds to</i>	<i>Action</i>
<code>\textrm{..}</code>	<code>\rmfamily</code>	Typeset argument in roman family
<code>\textsf{..}</code>	<code>\sffamily</code>	Typeset argument in <b>sans serif</b> family
<code>\texttt{..}</code>	<code>\ttfamily</code>	Typeset argument in <b>typewriter</b> family
<code>\textmd{..}</code>	<code>\mdseries</code>	Typeset argument in medium series
<code>\textbf{..}</code>	<code>\bfseries</code>	Typeset argument in <b>bold</b> series
<code>\textup{..}</code>	<code>\upshape</code>	Typeset argument in normal shape
<code>\textit{..}</code>	<code>\itshape</code>	Typeset argument in <i>italic</i> shape
<code>\textsl{..}</code>	<code>\slshape</code>	Typeset argument in <i>slanted</i> shape
<code>\textsc{..}</code>	<code>\scshape</code>	Typeset argument in SMALL CAPS shape
<code>\emph{..}</code>	<code>\em</code>	Typeset argument <i>emphasized</i>

Table 1: Font-change commands with arguments

The font change commands provided here all start with `\text..` to emphasize that they are for use in normal text and to be easily memorable. They automatically take care of any necessary italic correction on either side of the argument.

It is best to declare the most often used characters first, because this will make the processing slightly faster. For example,

```
\emph{When using the \NFSS{ } high-level commands,
the \emph{proper} use of italic corrections is
automatically taken care of}. Only
\emph{sometimes} one has to help \LaTeX{ } by
adding a \verb=\nocorr= command.
```

which results in:

*When using the NFSS high-level commands, the proper use of italic corrections is automatically taken care of. Only sometimes one has to help L<sup>A</sup>T<sub>E</sub>X by adding a \nocorr command.*

In contrast, the use of the declaration forms is often more appropriate when you define your own commands or environments.

```
\newenvironment{bfitemize}{\begin{itemize}\normalfont\bfseries}
                          {\end{itemize}}
\begin{bfitemize}
\item This environment produces boldface items.
\item It is defined in terms of \LaTeX's
      \texttt{itemize} environment and NFSS
      declarations.
\end{bfitemize}
```

This gives:

- This environment produces boldface items.

- It is defined in terms of L<sup>A</sup>T<sub>E</sub>X's `itemize` environment and NFSS declarations.

In addition to global customization of when to insert the italic correction, it is of course sometimes necessary to explicitly insert one with `\/`.

It is also possible to suppress the italic correction in individual instances. For this, the command `\nocorr` is provided.

The `\nocorr` must appear as the first or last token inside the braces of the argument of the `\text...` commands, at that end of the text where you wish to suppress the italic correction.

It is worth pointing out here that inserting a `\/` in places where it can have no function (i.e. anywhere except immediately after a slanted letter) is not an error—it will just be silently ignored. Unfortunately this is not true if the redefinition of `\/` in `amstex.sty` is used as this version can cause space to be removed immediately before the `\/`.

## 48 The implementation

`\DeclareTextFontCommand` This is the creator function for `\text..` commands. It gives a warning if `\foo` or `\fragfoo` is already defined.

In math mode it simply puts the font declaration and text into a box (possibly an automagically sized one).

Otherwise it first scans the text to see where `\nocorr` occurs within it. This sets the `\check@ic` commands to do what is necessary concerning the italic correction at both ends.

The algorithm for deciding whether to put in an italic correction is not very subtle: one is added whenever the newly current font is not itself positively sloped, unless the next token is a character in the ‘nocorr’ list. At the end of the text this is done after closing the group so as to check the ‘outer font’. Note that this will often result in adding an italic correction token after a character in an unsloped font; we believe (in early 2003) that this is perhaps inefficient but not dangerous.

It also now checks for empty contents of the text command and optimises this case. Some care is also taken to check that doing dangerous things in vertical mode is avoided.

The italic correction token is added to the horizontal list before (in the list) an immediately preceding non-zero glob of glue (skip) and any non-zero penalty preceding that since, in the typical case, this puts it immediately after the last character in the preceding word.

Note that it is necessary to put in the `\aftergroup\maybe@ic` at the end of the group so that it comes after any other `aftergroup` tokens and immediately before the following tokens. It is also necessary to remove the `\fi` from the token list before the group ends; this is done by adding an `\expandafter` just before the closing brace.

```

1 (*2kernel)
2 \def \DeclareTextFontCommand #1#2{%
3   \DeclareRobustCommand#1[1]{%
4     \ifmmode
5       \nfss@text{#2##1}%
6     \else
7       \hmode@bgroup

```

```

8      \text@command{##1}%
9      #2\check@icl ##1\check@icr
10     \expandafter
11     \egroup
12     \fi
13         }%
14 }

\textrm Now we define the \text{family} commands in terms of the above; \texttt does
\textsf not look very nice!
\texttt 15 \DeclareTextFontCommand{\textrm}{\rmfamily}
\textnormal 16 \DeclareTextFontCommand{\textsf}{\sffamily}
17 \DeclareTextFontCommand{\texttt}{\ttfamily}
18 \DeclareTextFontCommand{\textnormal}{\normalfont}

\textbf For the series attribute:
\textmd 19 \DeclareTextFontCommand{\textbf}{\bfseries}
20 \DeclareTextFontCommand{\textmd}{\mdseries}

\textit And for the shapes:
\textsl 21 \DeclareTextFontCommand{\textit}{\itshape}
\textsc 22 \DeclareTextFontCommand{\textsl}{\slshape}
\textup 23 \DeclareTextFontCommand{\textsc}{\scshape}
24 \DeclareTextFontCommand{\textup}{\upshape}

\emph Finally we have the \em font change declaration of LATEX. The corresponding
definition with argument is
25 \DeclareTextFontCommand{\emph}{\em}

\nocorr This is just a label, so it does nothing; it should also be unexpandable.
26 \let \nocorr \relax

\check@icl We define these defaults in case some error causes them to be expanded at the
\check@icr wrong time.
27 \let \check@icl \@empty
28 \let \check@icr \@empty

\text@command This checks for a \nocorr as the first token in its argument and also for one in
\check@nocorr@ any other position not protected within braces (the latter is treated as if it were
at the end of the argument).
Is this the correct action in the ‘empty’ case? It is efficient but typographically
it is, strictly, incorrect!
29 \def \text@command #1{%
30   \def \reserved@a {#1}%
31   \ifx \reserved@a \@empty
32     \let \check@icl \@empty
33     \let \check@icr \@empty
34   \else
\space is a reserved word in LATEX or actually already in plain TEX. If somebody
really redefines it so many things will break that I don’t see any reason to make
this routine here slower than necessary.
35 %   \def \reserved@b { }%

```

```

36 %    \ifx \reserved@a \reserved@b
37    \ifx \reserved@a \space
38        \let \check@icl \@empty
39        \let \check@icr \@empty
40    \else
41        \check@nocorr@ #1\nocorr\@nil
42    \fi
43 \fi
44 }
45 \def \check@nocorr@ #1#2\nocorr#3\@nil {%

```

The two checks are initialised here to their values in the normal case.

```

46    \let \check@icl \maybe@ic
47    \def \check@icr {\ifvmode \else \aftergroup \maybe@ic \fi}%
48    \def \reserved@a {\nocorr}%
49    \def \reserved@b {#1}%
50    \def \reserved@c {#3}%
51    \ifx \reserved@a \reserved@b
52        \ifx \reserved@c \@empty

```

In this case there is a `\nocorr` at the start but not at the end, so `\check@icl` should be empty.

```

53        \let \check@icl \@empty
54    \else

```

Otherwise there is a `\nocorr` both at the start and elsewhere, so no italic corrections should be added.

```

55        \let \check@icl \@empty
56        \let \check@icr \@empty
57    \fi
58    \else
59        \ifx \reserved@c \@empty

```

In this case there is no `\nocorr` anywhere, so we need to check for an italic correction at both the beginning and the end. This has been set up as the default so no code is needed here.

```

60    \else

```

In this case there is no `\nocorr` at the start but there is one elsewhere, so no `\aftergroup` is needed.

```

61        \let \check@icr \@empty
62    \fi
63 \fi
64 }

```

`\ifmaybe@ic` Switch used solely within `\maybe@ic` not interfering with other switches.

```

65 \newif\ifmaybe@ic

```

`\maybe@ic` These macros implement the italic correction.

```

\maybe@ic@ 66 \def \maybe@ic {\futurelet\@let@token\maybe@ic@}
67 \def \maybe@ic@ {%

```

We first check to see if the current font is positively sloped. (But do not forget the message Rainer sent about an upright font with non-zero slope! Or is this an urban myth?) It has been suggested that this should test against a small positive value, but what?

```

68 \ifdim \fontdimen\@ne\font>\z@
69 \else
70 \maybe@ictrue

```

It would be possible, but probably not worthwhile, to continue the forward scan beyond any closing braces.

```

71 \expandafter\@tfor\expandafter\reserved@a\expandafter:\expandafter=%
72 \nocorrlist

```

We have to hide the `\@let@token` in the macro `\t@st@ic` rather than testing it directly in the loop since it might be `\let` to a `\fi` or `\else`, which would result in chaos.

```

73 \do \t@st@ic

```

Frank thinks that the next bit it is inefficient if done after the second change. Chris thinks that most all of this is inefficient for the commonest cases: but that is the price of a cleverer algorithm. It is certainly needed to deal with the use of `\nolinebreak`.

```

74 \ifmaybe@ic \sw@slant \fi
75 \fi
76 }

```

`\t@st@ic` The next token in the input stream is stored in `\@let@token` via a `\let`, the current token from `\nocorrlist` is stored via `\def` in `\reserved@a`. To compare them we have to fiddle around a bit.

If the only things to check were characters then this could be done via an `\if` thus their catcodes would not matter; but this will not work whilst `\futurelet` is used above.

```

77 \def \t@st@ic {%
78 \expandafter\let\expandafter\reserved@b\expandafter=\reserved@a\relax
79 \ifx\reserved@b\@let@token

```

If they are the same we record the fact and jump out of the loop.

```

80 \maybe@icfalse
81 \@break@tfor
82 \fi
83 }

```

`\sw@slant` The definition of the mysterious `\sw@slant` command is as follows.  
`\fix@penalty` 84 \def \sw@slant {%

It is surely correct to put in an italic correction when there is no skip. If the last thing on the list is actually a zero skip (including things whose dimension part is zero, such as `\hfill`), or anything other than a character, then the italic correction will have no effect.

In order to work correctly with unbreakable spaces from `~` (and other common forms of line-breaking control) we also move back across a penalty before the glue.

```

85 \ifdim \lastskip=\z@
86 \fix@penalty
87 \else
88 \skip@ \lastskip
89 \unskip
90 \fix@penalty
91 \hskip \skip@

```

```

92 \fi
93 }

```

The above code means: “If there is a non-zero space just before the current position (`\ifdim...`) save the amount of that space (`\skip@\lastskip`), remove it (`\unskip`), then do a similar thing if there is a penalty just before the skip, and finally put the space back in.”

Since zero glue cannot be distinguished in this context from no glue, we dare not put in an `\hskip` in this case as this may produce an unwanted breakpoint. This is not satisfactory.

The penalty before the glue is handled similarly, with the same caveats concerning the zero case. Is this the first recorded use of `\unpenalty` in standard L<sup>A</sup>T<sub>E</sub>X code?

```

94 \def \fix@penalty {%
95   \ifnum \lastpenalty=\z@
96     \@@italiccorr
97   \else
98     \count@ \lastpenalty
99     \unpenalty
100    \@@italiccorr
101    \penalty \count@
102  \fi
103 }

```

`\nocorrlist` This holds the list of characters that should prevent italic correction. They should be ordered by decreasing frequency of use. If any such character is made active later on one needs to redefine the list so that the active character becomes part of it.

```

104 \def \nocorrlist {,.}

```

`\nfss@text` This command will by default behave like a L<sup>A</sup>T<sub>E</sub>X `\mbox` but may be redefined by packages such as `amstext.sty` to be a bit cleverer.

```

105 \ifx \nfss@text\undefined
106   \def \nfss@text {\leavevmode\hbox}
107 \fi

```

`\DeclareOldFontCommand` This is the function used to create declarative font-changing commands that can also be used to change alphabets in math-mode.

Usage: `\DeclareOldFontCommand \fn{<font-change decls>} <math-alphabet>`

Here `\fn` is the font-declaration command being defined, `<font-change decls>` is the declaration it will expand to in text-mode, and `<math-alphabet>` is the (single) math alphabet specifier which is to be used in math-mode.

It does not care whether the command being defined already exists but it does give a warning if it redefines anything.

Here are some typical examples of its use in conjunction with more basic NFSS2 font commands.

```

\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}

```

```

108 \def \DeclareOldFontCommand #1#2#3{%
109   \DeclareRobustCommand #1{\@fontswitch {#2}{#3}}%
110 }

```

`\@fontswitch` These two commands actually do the necessary tests and declarative font- or  
`\@@math@egroup` alphabet-changing.

```

\@@math@egroup
111 \def \@fontswitch #1#2{%
112   \ifmmode
113     \let \math@bgroup \relax
114     \def \math@egroup {\let \math@bgroup \@@math@bgroup
115                        \let \math@egroup \@@math@egroup}%

```

We need to have a `\relax` in the following line in case the `#2` is something like `\mathsf` grabbing the next token as an argument. For this reason the code also uses explicit arguments again (see pr/1275).

```

116     #2\relax
117   \else
118     #1%
119   \fi
120 }
121 \let \@@math@bgroup \math@bgroup
122 \let \@@math@egroup \math@egroup

```

These commands are available only in the preamble.

```

123 \@onlypreamble \DeclareTextFontCommand
124 \@onlypreamble \DeclareOldFontCommand

```

## 49 Initialization

`\normalsize` This is defined to produce an error.

```

125 \def\normalsize{%
126   \@latex@error {The font size command \protect\normalsize\space
127                 is not defined:\MessageBreak
128                 there is probably something wrong with
129                 the class file}\@eha
130 }
131 </2ekernel>

```

File w

## ltpageno.dtx

### 50 Page Numbering

Page numbers are produced by a page counter, used just like any other counter. The only difference is that `\c@page` contains the number of the next page to be output (the one currently being produced), rather than one minus it. Thus, it is normally initialized to 1 rather than 0. `\c@page` is defined to be `\count0`, rather than a count assigned by `\newcount`.

`\pagenumbering` The user sets the pagenumber style with the `\pagenumbering{<foo>}` command, which sets the page counter to 1 and defines `\thepage` to be `\foo`. For example, `\pagenumbering{roman}` causes pages to be numbered i, ii, etc.

```
1 <*2kernel>
2 \message{page nos.,}

3 \countdef\c@page=0 \c@page=1
4 \def\c1@page{}
5 \def\pagenumbering#1{%
6   \global\c@page \@ne \gdef\thepage{\csname @#1\endcsname
7     \c@page}}
8 </2kernel>
```

## File x

# ltxref.dtx

## 51 Cross Referencing

The user writes `\label{foo}` to define the following cross-references:

`\ref{foo}`: value of most recently incremented referencable counter. in the current environment. (Chapter, section, theorem and enumeration counters counters are referencable, footnote counters are not.)

`\pageref{foo}`: page number at which `\label{foo}` command appeared. where foo can be any string of characters not containing ‘\’, ‘{’ or ‘}’.

Note: The scope of the `\label` command is delimited by environments, so `\begin{theorem} \label{foo} ... \end{theorem} \label{bar}` defines `\ref{foo}` to be the theorem number and `\ref{bar}` to be the current section number.

Note: `\label` does the right thing in terms of spacing – i.e., leaving a space on both sides of it is equivalent to leaving a space on either side.

### 51.1 Cross Referencing

```
1 (*2kernel)
2 \message{x-ref,}
```

This is implemented as follows. A referencable counter CNT is incremented by the command `\refstepcounter{CNT}`, which sets `\@currentlabel == {CNT}{eval(\p@cnt\theCNT)}`. The command `\label{FOO}` then writes the following on file `\@auxout`:

```
\newlabel{FOO}{{eval(\@currentlabel)}{eval(\thepage)}}
```

```
\ref{FOO} ==
BEGIN
  if \r@foo undefined
  then  @refundefined := G T
        ??
        Warning: 'reference foo on page ... undefined'
  else  \@car \eval(\r@FOO)\@nil
  fi
END
```

```
\pageref{foo} =
BEGIN
  if \r@foo undefined
  then  @refundefined := G T
        ??
        Warning: 'reference foo on page ... undefined'
  else  \@cdr \eval(\r@FOO)\@nil
  fi
END
```

`\G@refundefinedtrue` This does not save on name-space (since `\G@refundefinedfalse` was never needed) but it does make the implementation of such one-way switches more consistent. The extra macro to make the change is used since this change appears several times.

**Note** despite its name, `\G@refundefinedtrue` does *not* correspond to an `\if` command, and there is no matching `...false`. It would be more natural to call the command `\G@refundefined` (as inspection of the change log will reveal) but unfortunately such a change would break any package that had defined a `\ref`-like command that mimicked the definition of `\ref`, calling `\G@refundefinedtrue`. Inspection of the T<sub>E</sub>X archives revealed several such packages, and so this command has been named `...true` so that the definition of `\ref` need not be changed, and the packages will work without change.

```

3 % \newif\ifG@refundefined
4 % \def\G@refundefinedtrue{\global\let\ifG@refundefined\iftrue}
5 % \def\G@refundefinedfalse{\global\let\ifG@refundefined\iffalse}
6 \def\G@refundefinedtrue{%
7   \gdef\@refundefined{%
8     \@latex@warning@no@line{There were undefined references}}
9 \let\@refundefined\relax

```

`\ref` Referencing a `\label`. RmS 91/10/25: added a few extra `\reset@font`, as suggested by Bernd Raichle

`\pageref`

`\setref` RmS 92/08/14: made `\ref` and `\pageref` robust  
RmS 93/09/08: Added setting of `refundefined` switch.

```

10 \def\@setref#1#2#3{%
11   \ifx#1\relax
12     \protect\G@refundefinedtrue
13     \nfss@text{\reset@font\bfseries ??}%
14     \@latex@warning{Reference ‘#3’ on page \thepage \space
15                   undefined}%
16   \else
17     \expandafter#2#1\null
18   \fi}
19 \def\ref#1{\expandafter\@setref\csname r@#1\endcsname\@firstoftwo{#1}}
20 \def\pageref#1{\expandafter\@setref\csname r@#1\endcsname
21               \@secondoftwo{#1}}

```

`\newlabel` This command will be written to the `.aux` file to pass label information from one run to another.

`\@newl@bel` The internal form of `\newlabel` and `\bibcite`. Note that this macro does its work inside a group. That way the local assignments it needs to do don't clutter the save stack. This prevents large documents with many labels to run out of save stack.

```

22 \def\@newl@bel#1#2#3{%
23   \@ifundefined{#1#2}%
24     \relax
25     {\gdef \@multiplelabels {%
26       \@latex@warning@no@line{There were multiply-defined labels}}%
27       \@latex@warning@no@line{Label ‘#2’ multiply defined}}%
28   \global\@namedef{#1#2}{#3}}

```

```

29 \def\newlabel{\@newl@bel r}
30 \@onlypreamble\@newl@bel

\if@multiplelabels This is redefined to produce a warning if at least one label is defined more than
\@multiplelabels once. It is executed by the \enddocument command.
31 \let \@multiplelabels \relax

\label The commands \label and \refstepcounter have been changed to allow
\refstepcounter \protect'ed commands to work properly. For example,

\def\thechapter{\protect\foo{\arabic{chapter}.\roman{section}}}}

will cause a \label{bar} command to define \ref{bar} to expand to something
like \foo{4.d}. Change made 20 Jul 88.

32 \def\label#1{\@bsphack
33   \protected@write\@auxout{}%
34     {\string\newlabel{#1}{\@currentlabel}\thepage}}%
35   \@esphack}

36 \def\refstepcounter#1{\stepcounter{#1}%
37   \protected@edef\@currentlabel
38     {\csname p@#1\endcsname\csname the#1\endcsname}%
39 }

\@currentlabel For \label commands that come before any environment

40 \def\@currentlabel{}

41 </2kernel>

```

## 51.2 An extension of counter referencing

At the moment a reference to a counter `foo` will generate the equivalent of `\p@foo\thefoo` although not quite in this form. For some applications it would be nice if one could have `\thefoo` being an argument to `\p@foo` to be able to put material before and after the number generated by `\thefoo`. This can be easily achieved with a small change to one of the kernel commands as follows:

```

\def\refstepcounter#1{\stepcounter{#1}%
  \protected@edef\@currentlabel
    {\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%
}

```

The trick is to ensure that `\csname the#1\endcsname` is turned into a single token before `\p@...` is expanded further. This way, if the `\p@...` command is a macro with one argument it will receive `\the...`. With the kernel code (i.e., without the `\expandafter`) it will instead pick up `\csname` which would be disastrous.

Using `\expandafter` instead of braces delimiting the argument is better because, assuming that the `\p@...` command is not defined as a macro with one argument, the braces will stay and prohibit kerning that might otherwise happen between the glyphs generated by `\the...` and surrounding glyphs.

We have refrained from making this change in the kernel code although for existing documents it would be 100% backward compatible. The reason being

that any class or package making use of this functionality would then horribly fail with older L<sup>A</sup>T<sub>E</sub>X installations.

Instead we suggest that people who are interested in using this functionality in a document class or package add the redefinition to the class file. To ensure that this redefinition is properly applied they might want to test for the original definition first, e.g.

```
\CheckCommand*\refstepcounter[1]{\stepcounter{#1}%
  \protected@edef\@currentlabel
    {\csname p@#1\endcsname\csname the#1\endcsname}%
}
\renewcommand*\refstepcounter[1]{\stepcounter{#1}%
  \protected@edef\@currentlabel
    {\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%
}
```

# File y

## ltmiscen.dtx

### 52 Miscellaneous Environments

This section implements the basic environment mechanism, and also a few specific environments including `document`, The math environments and related commands, the ‘flushing’ environments, (`center`, `flushleft`, `flushright`), and `verbatim`.

```
1 (*2ekernel)
2 \message{environments,}
```

#### 52.1 Environments

`\begin{foo}` and `\end{foo}` are used to delimit environment `foo`.

`\begin{foo}` starts a group and calls `\foo` if it is defined, otherwise it does nothing.

`\end{foo}` checks to see that it matches the corresponding `\begin` and if so, it calls `\endfoo` and does an `\endgroup`. Otherwise, `\end{foo}` does nothing.

If `\end{foo}` needs to ignore blanks after it, then `\endfoo` should globally set the `@ignore` switch true with `\@ignoretrue` (this will automatically be global).

NOTE: `@@end` is defined to be the `\end` command of T<sub>E</sub>X82.

`\enddocument` is the user’s command for ending the manuscript file.

`\stop` is a panic button — to end T<sub>E</sub>X in the middle.

```
\enddocument ==
BEGIN
  \@checkend{document}    %% checks for unmatched \begin
  \clearpage
  \begingroup
    if @filesw = true
    then close file @mainaux
    if G@refundefined = true
    then LaTeX Warning: 'There are undefined references.' fi
    if @multiplelabels = true
    then LaTeX Warning:
      'One or more label(s) multiply defined.'
    else
      \@setckpt {ARG1}{ARG2} == null
      \newlabel{LABEL}{VAL} ==
      BEGIN
        \reserved@a == VAL
        if def(\reserved@a) = def(\r@LABEL)
        else @tempwa := true          fi
      END
      \bibcite{LABEL}{VAL} == null
      BEGIN
        \reserved@a == VAL
        if def(\reserved@a) = def(\g@LABEL)
        else @tempwa := true          fi
    fi
  fi
END
```

```

                                END
                                @tempswa := false
                                make @ a letter
                                \input \jobname.AUX
                                if @tempswa = true
                                    then LaTeX Warning: 'Label may have changed.
                                                Rerun to get cross-references right.'
                                fi
                                fi
                                fi
                                \endgroup
                                finish up
                                END

                                \@writefile{EXT}{ENTRY} ==
                                    if tf@EXT undefined
                                        else \write\tf@EXT{ENTRY}
                                    fi

\@currentvir The name of the current environment.  Initialized to document to so that
\end{document} works correctly.
3 \def\@currentvir{document}

\if@ignore
\@ignoretrue 4 \def\@ignorefalse{\global\let\if@ignore\iffalse}
\@ignorefalse 5 \def\@ignoretrue {\global\let\if@ignore\iftrue}
6 \@ignorefalse

\ignorespacesafterend

7 \let\ignorespacesafterend\@ignoretrue

\enddocument

8 \def\enddocument{%
The \end{document} hook is executed first.  If necessary it can contain a
\clearpage to output dangling floats first.  In this position it can also contain
something like \end{foo} so that the whole document effectively starts and ends
with some special environment.  However, this must be used with care, eg if two
applications would use this without knowledge of each other the order of the en-
vironments will be wrong after all. \AtEndDocument is redefined at this point so
that and such commands that get into the hook do not chase their tail...

9 \let\AtEndDocument\@firstofone
10 \@enddocumenthook
11 \@checkend{document}%
12 \clearpage
13 \begingroup
14 \if@filesw
15 \immediate\closeout\@mainaux
16 \let\@setckpt\@gobbletwo
17 \let\@newl@bel\@testdef
The previous line is equiv to setting

\def\newlabel{\@testdef r}%
\def\bibcite{\@testdef b}%

```

We use `\@@input` to load the `.aux` file, so that it doesn't show up in the list of files produced by `\listfiles`.

```
18      \@tempswafalse
19      \makeatletter \@@input\jobname.aux
20      \fi

21      \@dofilelist
```

First we check for font size substitution bigger than `\fontsubfuzz`. The `\relax` is necessary because this is a macro not a register.

```
22      \ifdim \font@submax >\fontsubfuzz\relax
```

In case you wonder about the `\@gobbletwo` inside the message below, this is a horrible hack to remove the tokens `\on@line`. that are added by `\@font@warning` at the end.

```
23      \@font@warning{Size substitutions with differences\MessageBreak
24                      up to \font@submax\space have occurred.\@gobbletwo}%
25      \fi
```

The macro `\@defaultsubs` is initially `\relax` but gets redefined to produce a warning if there have been some default font substitutions.

```
26      \@defaultsubs
```

The macro `\@refundefined` is initially `\relax` but gets redefined to produce a warning if there are undefined refs.

```
27      \@refundefined
```

If a label is defined more than once, `\@tempswa` will always be true and thus produce a “Label(s) may ...” warning. But since a rerun will not solve that problem (unless one uses a package like `varioref` that generates labels on the fly), we suppress this message.

```
28      \if@filesw
29          \ifx \@multiplelabels \relax
30              \if@tempswa
31                  \@latex@warning@no@line{Label(s) may have changed.
32                      Rerun to get cross-references right}%
33              \fi
34          \else
35              \@multiplelabels
36          \fi
37      \fi
38      \endgroup
39      \deadcycles\z@\@@end}
```

```
\@testdef
```

```
40 \def\@testdef #1#2#3{%
41     \def\reserved@a{#3}\expandafter \ifx \csname #1@#2\endcsname
42     \reserved@a \else \@tempwatrue \fi}
```

```
\@writefile
```

```
43 \long\def\@writefile#1#2{%
44     \ifundefined{tf@#1}\relax
45     {\@temptokena{#2}%
46         \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
47     }%
48 }
```

```

\stop
49 \def\stop{\clearpage\deadcycles\z@\let\par\@par\@end}

50 \everypar{\@nodocument} %% To get an error if text appears before the
51 \nullfont                %% \begin{document}

\begin, \end, and \@checkend changed so \end{document} will catch
an unmatched \begin.  Changed 24 May 89 as suggested by
Frank Mittelbach and Rainer Sch\"opf.

\begin{NAME} ==
BEGIN
  IF \NAME undefined THEN \reserved@a == BEGIN report error
END
                                ELSE \reserved@a ==
                                (\@currenvir :=L NAME) \NAME
  FI
  @ignore :=G F                %% Added 30 Nov 88
  \begingroup
  \@endpe := F
  \@currenvir :=L NAME
  \NAME
END

\end{NAME} ==
BEGIN
  \endNAME
  \@checkend{NAME}
  \endgroup
  IF @endpe = T                %% @endpe set True by \@endparenv
    THEN \@doendpe            %% \@doendpe redefines \par and
\everypar                    %% to suppress paragraph indentation in
                                %% immediately following text
  FI
  IF @ignore = T
    THEN @ignore :=G F
    \ignorespaces
  FI
END

\@checkend{NAME} ==
BEGIN
  IF \@currenvir = NAME
    ELSE \@badend{NAME}
  FI
END

```

```

\begin
52 \def\begin#1{%
53   \ifundefined{#1}%
54     {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
55     {\def\reserved@a{\def\@currenvir{#1}%
56       \edef\@currenvline{\on@line}%
57       \csname #1\endcsname}}%
58   \@ignorefalse
59   \begingroup\@endpfalse\reserved@a}

\end
60 \def\end#1{%
61   \csname end#1\endcsname\@checkend{#1}%
62   \expandafter\endgroup\if@endpe\@doendpe\fi
63   \if@ignore\@ignorefalse\ignorespaces\fi}

\@checkend
64 \def\@checkend#1{\def\reserved@a{#1}\ifx
65   \reserved@a\@currenvir \else\@badend{#1}\fi}

\@currenvline We do need a default value for \@currenvline on top-level since the document
environment cancels the brace group. This means that a mismatch with \begin
{document} will not produce a line number. Thus the outer default must be
\@empty or we will end up with two spaces.
66 \let\@currenvline\@empty

```

## 52.2 Center, Flushright, Flushleft

```

67 \message{center,}

\center, \flushright and \flushleft set
\rightskip = 0pt or \@flushglue (as appropriate)
\leftskip  = 0pt or \@flushglue (as appropriate)
\parindent = 0pt
\parfillskip = 0pt. (except \flushleft)
\\          == \par \vskip -\parskip
\\[LENGTH] == \\ \vskip LENGTH
\\*         == \par \penalty 10000 \vskip -\parskip
\\*[LEN]    == \\* \vskip LENGTH

```

They invoke the `trivlist` environment to handle vertical spacing before and after them.

`\centering`, `\raggedright` and `\raggedleft` are the declaration analogs of the above.

`\raggedright` has a more universal effect, however. It sets `\@rightskip := flushglue`. Every environment, like the list environments, that set `\rightskip` to its 'normal' value set it to `\@rightskip`

```

\@centercr
68 \def\@centercr{\ifhmode \unskip\else \@nolnerr\fi
69     \par\@ifstar{\nobreak\@xcentercr}\@xcentercr}

\@xcentercr
70 \def\@xcentercr{\addvspace{-\parskip}\@ifnextchar
71     [\@icentercr\ignorespaces}

\@icentercr
72 \def\@icentercr[#1]{\vskip #1\ignorespaces}

center We use \relax to prevent \item scanning too far.
73 \def\center{\trivlist \centering\item\relax}
74 \def\endcenter{\endtrivlist}

\centering
75 \def\centering{%
76     \let\\\@centercr
77     \rightskip\@flushglue\leftskip\@flushglue
78     \parindent\z@\parfillskip\z@skip}

\@rightskip
79 \newskip\@rightskip \@rightskip \z@skip

flushleft We use \relax to prevent \item scanning too far.
80 \def\flushleft{\trivlist \raggedright\item\relax}
81 \def\endflushleft{\endtrivlist}

\raggedright
82 \def\raggedright{%
83     \let\\\@centercr\@rightskip\@flushglue \rightskip\@rightskip
84     \leftskip\z@skip
85     \parindent\z@}

flushright We use \relax to prevent \item scanning too far.
86 \def\flushright{\trivlist \raggedleft\item\relax}
87 \def\endflushright{\endtrivlist}

\raggedleft
88 \def\raggedleft{%
89     \let\\\@centercr
90     \rightskip\z@skip\leftskip\@flushglue
91     \parindent\z@\parfillskip\z@skip}

92 \message{verbatim,}

```

## 52.3 Verbatim

The verbatim environment uses the fixed-width `\ttfamily` font, turns blanks into spaces, starts a new line for each carriage return (or sequence of consecutive carriage returns), and interprets *every* character literally. I.e., all special characters `\`, `{`, `$`, etc. are `\catcode'd` to 'other'.

The command `\verb` produces in-line verbatim text, where the argument is delimited by any pair of characters. E.g., `\verb #...#` takes '...' as its argument, and sets it verbatim in `\ttfamily` font.

The \*-variants of these commands are the same, except that spaces print as the T<sub>E</sub>Xbook's space character instead of as blank spaces.

`\@vobeyspaces`

```
93 {\catcode'\ =\active%
94 \gdef\@vobeyspaces{\catcode'\ \active\let \@xobeysp}}
```

`\@xobeysp`

`\@xverbatim`

```
\@sxverbatim 95 \begingroup \catcode '|=0 \catcode '['= 1
96 \catcode']=2 \catcode '\{=12 \catcode '\}=12
97 \catcode'\|=12 \gdef\@xverbatim#1\end{verbatim} [#1\end[verbatim]]
98 \gdef\@sxverbatim#1\end{verbatim*} [#1\end[verbatim*]]
99 \endgroup
```

`\@verbatim` Real start of verbatim environment We use `\relax` to prevent `\item` scanning too far.

```
100 \def\@verbatim{\trivlist \item\relax
101 \if@minipage\else\vskip\parskip\fi
102 \leftskip\@totalleftmargin\rightskip\z@skip
103 \parindent\z@\parfillskip\@flushglue\parskip\z@skip
```

Added `\@@par` to clear possible `\parshape` definition from a surrounding list (the verbatim guru says).

```
104 \@@par
105 \@tempswafalse
106 \def\par{%
107 \if@tempswa
```

A `\leavevmode` added: needed if, for example, a blank verbatim line is the first thing in a list item (wow!).

```
108 \leavevmode \null \@@par\penalty\interlinepenalty
109 \else
110 \@tempswatrue
111 \ifhmode\@@par\penalty\interlinepenalty\fi
112 \fi}%
```

To allow customization we hide the font used in a separate macro.

```
113 \let\do\@makeother \dospecials
114 \obeylines \verbatim@font \@noligs
115 \hyphenchar\font\m@ne
```

To avoid a breakpoint after the labels box, we remove the penalty put there by the list macros: another use of `\unpenalty`!

```
116 \everypar \expandafter{\the\everypar \unpenalty}%
117 }
```

```

\verbatim (RmS 93/09/19) Protected against 'missing item' error message triggered by
\endverbatim empty verbatim environment.
118 \def\verbatim{\@verbatim \frenchspacing\@vobeyspaces \@xverbatim}
119 \def\endverbatim{\if@newlist \leavevmode\fi\endtrivlist}

\verbatim@font Macro to select the font used for verbatim typesetting. It also does other work if
necessary for the font used.
120 \def\verbatim@font{\normalfont\ttfamily}

verbatim*
121 \@namedef{verbatim*}{\@verbatim\@sxverbatim}
122 \expandafter\let\csname endverbatim*\endcsname =\endverbatim

\@makeother
123 \def\@makeother#1{\catcode'#12\relax}

\verb@balance@group
124 \let\verb@balance@group\@empty

\verb@egroup
125 \def\verb@egroup{\global\let\verb@balance@group\@empty\egroup}

\verb@eol@error
126 \begingroup
127 \obeylines%
128 \gdef\verb@eol@error{\obeylines%
129 \def~M{\verb@egroup\@latex@error{%
130 \noexpand\verb ended by end of line}\@ehc}}%
131 \endgroup

\verb Typesetting a small piece verbatim.
132 \def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
133 \bgroup
134 \verb@eol@error \let\do\@makeother \dospecials
135 \verbatim@font\@noligs
136 \@ifstar\@sverb\@verb}

\@sverb Definitions of \@sverb and \@verb changed so \verb+ foo+ does not lose lead-
ing blanks when it comes at the beginning of a line. Change made 24 May 89.
Suggested by Frank Mittelbach and Rainer Schöpf.
137 \def\@sverb#1{%
138 \catcode'#1\active
139 \lccode'\~'#1%
140 \gdef\verb@balance@group{\verb@egroup
141 \@latex@error{\noexpand\verb illegal in command argument}\@ehc}%
142 \aftergroup\verb@balance@group
143 \lowercase{\let~\verb@egroup}}%

\@verb
144 \def\@verb{\@vobeyspaces \frenchspacing \@sverb}

\verbatim@nolig@list
145 \def\verbatim@nolig@list{\do\~\do\<\do\>\do\,\do\''\do\~}

```

```

\do@noligs
146 \def\do@noligs#1{%
147   \catcode'#1\active
148   \begingroup
149     \lccode'\~'#1\relax
150     \lowercase{\endgroup\def~{\leavevmode\kern\z@\char'#1}}}}

\@noligs To stay compatible with packages that use \@noligs we keep it.
151 \def\@noligs{\let\do\do@noligs \verbatim@nolig@list}

152 \</2kernel>

```

File z

# ltmath.dtx

## 53 Math setup

This file contains a lot of the original plain T<sub>E</sub>X code, as well as the L<sup>A</sup>T<sub>E</sub>X environments for math. It still needs sorting out.

```
1 (*2kernel)
2 \message{math definitions,}
```

### 53.1 Math commands based on plain T<sub>E</sub>X

#### 53.1.1 The log-like functions

```
\log The standard operators:
3 \def\log{\mathop{\operator@font log}\nolimits}
4 \def\lg{\mathop{\operator@font lg}\nolimits}
5 \def\ln{\mathop{\operator@font ln}\nolimits}
6 \def\lim{\mathop{\operator@font lim}\nolimits}
7 \def\limsup{\mathop{\operator@font lim}\nolimits,\sup}
8 \def\liminf{\mathop{\operator@font lim}\nolimits,\inf}
9 \def\sin{\mathop{\operator@font sin}\nolimits}
10 \def\arcsin{\mathop{\operator@font arcsin}\nolimits}
11 \def\sinh{\mathop{\operator@font sinh}\nolimits}
12 \def\cos{\mathop{\operator@font cos}\nolimits}
13 \def\arccos{\mathop{\operator@font arccos}\nolimits}
14 \def\cosh{\mathop{\operator@font cosh}\nolimits}
15 \def\tan{\mathop{\operator@font tan}\nolimits}
16 \def\arctan{\mathop{\operator@font arctan}\nolimits}
17 \def\tanh{\mathop{\operator@font tanh}\nolimits}
18 \def\cot{\mathop{\operator@font cot}\nolimits}
19 \def\coth{\mathop{\operator@font coth}\nolimits}
20 \def\sec{\mathop{\operator@font sec}\nolimits}
21 \def\csc{\mathop{\operator@font csc}\nolimits}
22 \def\max{\mathop{\operator@font max}\nolimits}
23 \def\min{\mathop{\operator@font min}\nolimits}
24 \def\sup{\mathop{\operator@font sup}\nolimits}
25 \def\inf{\mathop{\operator@font inf}\nolimits}
26 \def\arg{\mathop{\operator@font arg}\nolimits}
27 \def\ker{\mathop{\operator@font ker}\nolimits}
28 \def\dim{\mathop{\operator@font dim}\nolimits}
29 \def\hom{\mathop{\operator@font hom}\nolimits}
30 \def\det{\mathop{\operator@font det}\nolimits}
31 \def\exp{\mathop{\operator@font exp}\nolimits}
32 \def\Pr{\mathop{\operator@font Pr}\nolimits}
33 \def\gcd{\mathop{\operator@font gcd}\nolimits}
34 \def\deg{\mathop{\operator@font deg}\nolimits}

\bmod And some operators have to be done by hand:
35 \def\bmod{%
36 \nonscript\mskip-\medmuskip\mkern5mu%
```

```

37 \mathbin{\operator@font mod}\penalty900\mkern5mu%
38 \nonscript\mskip-\medmuskip}

```

`\pmod`

```

39 \def\pmod#1{%
40 \allowbreak\mkern18mu({\operator@font mod}\,\,\,#1)}

```

### 53.1.2 Biggggg

`\big` Variants on `\big` and friends for use with delimiters:

```

41 \def\bigl{\mathopen\big}
42 \def\bigm{\mathrel\big}
43 \def\biggr{\mathclose\big}
44 \def\Bigl{\mathopen\Big}
45 \def\Bigm{\mathrel\Big}
46 \def\Bigr{\mathclose\Big}
47 \def\biggl{\mathopen\bigg}
48 \def\biggm{\mathrel\bigg}
49 \def\biggr{\mathclose\bigg}
50 \def\Biggl{\mathopen\Bigg}
51 \def\Biggm{\mathrel\Bigg}
52 \def\Biggr{\mathclose\Bigg}

```

### 53.1.3 The UNSORTED Rest

The other math commands are lifted from plain T<sub>E</sub>X.

`\jot`

```

53 \newdimen\jot
54 \jot=3pt

```

`\interdisplaylinepenalty`

```

55 \newcount\interdisplaylinepenalty
56 \interdisplaylinepenalty=100

```

`\choose`

```

57 \def\choose{\atopwithdelims()}

```

`\brack`

```

58 \def\brack{\atopwithdelims[]}

```

`\brace`

```

59 \def\brace{\atopwithdelims\{\}}

```

`\mathpalette`

```

60 \def\mathpalette#1#2{%
61 \mathchoice
62 {#1\displaystyle{#2}}%
63 {#1\textstyle{#2}}%
64 {#1\scriptstyle{#2}}%
65 {#1\scriptscriptstyle{#2}}

```

```

\root
\rootbox 66 \newbox\rootbox
\root 67 \def\root#1\of{%
68 \setbox\rootbox\hbox{$\m@th\scriptscriptstyle{#1}$}%
69 \mathpalette\root@t}

70 \def\root#1#2{%
71 \setbox\z@\hbox{$\m@th#1\sqrt{#2}$}%
72 \dimen@ \ht\z@ \advance\dimen@-\dp\z@
73 \mkern5mu\raise.6\dimen@\copy\rootbox
74 \mkern-10mu\box\z@}

\phantom
\hphantom 75 \newif\ifv@
\vphantom 76 \newif\ifh@

77 \def\vphantom{\v@true\h@false\ph@nt}
78 \def\hphantom{\v@false\h@true\ph@nt}
79 \def\phantom{\v@true\h@true\ph@nt}

80 \def\ph@nt{%
81 \ifmmode
82 \expandafter\mathpalette\expandafter\mathph@nt
83 \else
84 \expandafter\makeph@nt
85 \fi}

86 \def\makeph@nt#1{%
87 \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finph@nt}

88 \def\mathph@nt#1#2{%
89 \setbox\z@\hbox{$\m@th#1{#2}$}\finph@nt}

90 \def\finph@nt{%
91 \setbox\zw@null
92 \ifv@ \ht\zw@\ht\z@ \dp\zw@\dp\z@\fi
93 \ifh@ \wd\zw@\wd\z@\fi \box\zw@}

\mathstrut
94 \def\mathstrut{\vphantom{}}

\smash
95 \def\smash{%
96 \relax % \relax, in case this comes first in \halign
97 \ifmmode
98 \expandafter\mathpalette\expandafter\mathsm@sh
99 \else
100 \expandafter\makesm@sh
101 \fi}

102 \def\makesm@sh#1{%
103 \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finsm@sh}
104 \def\mathsm@sh#1#2{%
105 \setbox\z@\hbox{$\m@th#1{#2}$}\finsm@sh}
106 \def\finsm@sh{\ht\z@\z@ \dp\z@\z@ \box\z@}

```

```

\buildrel
107 \def\buildrel#1\over#2{\mathrel{\mathop{\kern\z@#2}\limits^{#1}}}

\cases
108 \def\cases#1{\left\{\,\,\vcenter{\normalbaselines\m@th
109 \ialign{\hfil$&\quad{##}\hfil\crr#1\crr}\right.}

\matrix
110 \def\matrix#1{\null\,\vcenter{\normalbaselines\m@th
111 \ialign{\hfil$##$\hfil&\quad\hfil$##$\hfil\crr
112 \mathstrut\crr\noalign{\kern-\baselineskip}
113 #1\crr\mathstrut\crr\noalign{\kern-\baselineskip}}\,}

\pmatrix
114 \def\pmatrix#1{\left(\matrix{#1}\right)}

\bordermatrix
115 \def\bordermatrix#1{\begingroup \m@th
116 \@tempdima 8.75\p@
117 \setbox\z@\vbox{%
118 \def\cr{\crr\noalign{\kern2\p@\global\let\cr\endline}}%
119 \ialign{##$\hfil\kern2\p@\kern-\@tempdima&\thinspace\hfil$##$\hfil
120 &\quad\hfil$##$\hfil\crr
121 \omit\strut\hfil\crr\noalign{\kern-\baselineskip}%
122 #1\crr\omit\strut\cr}}%
123 \setbox\@tw@\vbox{\unvcopy\z@\global\setbox\@ne\lastbox}%
124 \setbox\@tw@\hbox{\unhbox\@ne\unskip\global\setbox\@ne\lastbox}%
125 \setbox\@tw@\hbox{$\kern\wd\@ne\kern-\@tempdima\left(\kern-\wd\@ne
126 \global\setbox\@ne\vbox{\box\@ne\kern2\p@}%
127 \vcenter{\kern-\ht\@ne\unvbox\z@\kern-\baselineskip}\,,\right)$}%
128 \null;\vbox{\kern\ht\@ne\box\@tw@}\endgroup}

\openup
129 \def\openup{\afterassignment\openup\dimen@}
130 \def\@openup{\advance\lineskip\dimen@
131 \advance\baselineskip\dimen@
132 \advance\lineskiplimit\dimen@}

\displaylines
133 \newif\ifdt@p
134 \def\display{\global\dt@ptrue\openup\jot\m@th
135 \everycr{\noalign{\ifdt@p \global\dt@pfalse \ifdim\prevdepth>-1000\p@
136 \vskip-\lineskiplimit \vskip\normallineskiplimit \fi
137 \else \penalty\interdisplaylinepenalty \fi}}
138 \def\@lign{\tabskip\z@skip\everycr{}} % restore inside \display
139 \def\displaylines#1{\display \tabskip\z@skip
140 \halign{\hb@xt@\displaywidth{$\@lign\hfil\displaystyle##\hfil$}\crr
141 #1\crr}}

\sp
\sb
142 \let\sp=^
143 \let\sb=_

```

```

\>
\; 144 %\def\,{\mskip\thinmuskip}      % already defined in ltspase
\! 145 \def\>{\mskip\medmuskip}
    146 \def\;{\mskip\thickmuskip}
    147 \def\!{\mskip-\thinmuskip}

\*
    148 \def\*{\discretionary{\thinspace\the\textfont2\char2}{\{}}{\}}

\: Nickname for the medium space since \> is not available inside tabbing.
    149 \let\:=\>

\active@math@prime This is the definition of the active math prime.
    150 \def\active@math@prime{^{\bgroup\prim@s}

\prime@s
    151 {\catcode'\='=\active \global\let'\active@math@prime}

    152 \def\prim@s{%
    153   \prime\futurelet\@let@token\pr@m@s}

    154 \def\pr@m@s{%
    155   \ifx'\@let@token
    156     \expandafter\pr@@@s
    157   \else
    158     \ifx~\@let@token
    159       \expandafter\expandafter\expandafter\pr@@@t
    160     \else
    161       \egroup
    162     \fi
    163   \fi}

    164 \def\pr@@@s#1{\prim@s}
    165 \def\pr@@@t#1#2{#2\egroup}

    166 {\catcode'\_=\active \gdef_{\_}} % _ in math is
    167                                     % either subscript or \_

```

## 53.2 Math Environments

\( Produces  $\dots$  with checks that \(( isn't used in math mode, and that \) is only used in math mode begun with \(.

```

168 </2ekernel>
169 <latexrelease>\IncludeInRelease{2015/01/01}{\{()\{Make \(\ robust}%
170 <*2ekernel | latexrelease>
171 \DeclareRobustCommand\({%
172   \relax\ifmmode\@badmath\else$\fi}%
173 \DeclareRobustCommand\){%
174   \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
175 </2ekernel | latexrelease>
176 <latexrelease>\EndIncludeInRelease
177 <latexrelease>\IncludeInRelease{0000/00/00}{\{()\{Make \(\ robust}%
178 <latexrelease>\def\({%

```

```

179 <latexrelease> \relax\ifmmode\@badmath\else$\fi}%
180 <latexrelease>\def\}%
181 <latexrelease> \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
182 <latexrelease>\EndIncludeInRelease
183 (*2ekernel)

\l Produces $$...$$ with checks that \l isn't used in math mode, and that \l is
\l only used in display math mode (though there is no real test that this display
math started with \l and not with $$).

184 /2ekernel)
185 <latexrelease>\IncludeInRelease{2015/01/01}{\l}{Make \l robust}%
186 (*2ekernel | latexrelease)
187 \DeclareRobustCommand\l{%
188     \relax\ifmmode
189         \@badmath
190     \else
191         \ifvmode
192             \nointerlineskip
193             \makebox[.6\linewidth]{}%
194         \fi
195         $$$%$$ BRACE MATCH HACK
196     \fi
197 }%

198 \DeclareRobustCommand\l{%
199     \relax\ifmmode
200         \ifinner
201             \@badmath
202         \else
203             $$$%$$ BRACE MATCH HACK
204         \fi
205     \else
206         \@badmath
207     \fi
208     \ignorespaces
209 }%

210 /2ekernel | latexrelease)
211 <latexrelease>\EndIncludeInRelease
212 <latexrelease>\IncludeInRelease{0000/00/00}{\l}{Make \l robust}%
213 <latexrelease>\def\l{%
214 <latexrelease> \relax\ifmmode
215 <latexrelease> \@badmath
216 <latexrelease> \else
217 <latexrelease> \ifvmode
218 <latexrelease> \nointerlineskip
219 <latexrelease> \makebox[.6\linewidth]{}%
220 <latexrelease> \fi
221 <latexrelease> $$$%$$ BRACE MATCH HACK
222 <latexrelease> \fi
223 <latexrelease>}%

224 <latexrelease>\def\l{%
225 <latexrelease> \relax\ifmmode
226 <latexrelease> \ifinner
227 <latexrelease> \@badmath

```

```

228 \latexrelease \else
229 \latexrelease $$$%$$$ BRACE MATCH HACK
230 \latexrelease \fi
231 \latexrelease \else
232 \latexrelease \badmath
233 \latexrelease \fi
234 \latexrelease \ignorespaces
235 \latexrelease}%
236 \latexrelease\EndIncludeInRelease
237 (*2ekernel)

math Disguises for \(...\) and \[...\].
displaymath 238 \let\math=\(
239 \let\endmath=\)

240 \def\displaymath{\[}
241 \def\enddisplaymath{\]\@ignoretrue}

equation Numbered equations, using the counter \c@equation. Note: The document style
\c@equation must define \theequation etc., and do the appropriate \@addtoreset. It should
also redefine \@eqnnum if another format for the equation number is desired other
than the standard (...), or to move the equation numbers to the flushleft. (See
comment on the \def of \@eqnnum.)

242 \@definecounter{equation}
243 \def\equation{$$\refstepcounter{equation}}
244 \def\endequation{\eqno \hbox{\@eqnnum}$$\@ignoretrue}

\@eqnnum Produces the equation number for equation and eqnarray environments. The
following definition is for flushright numbers; for flushleft numbers, see leqno.clo.
The equation number is set in black roman type even if an eqnarray environment
appears in an italic environment.

245 \def\@eqnnum{\normalfont \normalcolor (\theequation)}

\stackrel A disguise for plain TEX's buildrel.

246 \def\stackrel#1#2{\mathrel{\mathop{#2}\limits^{#1}}}

\frac A disguise for plain TEX's \over.

247 \def\frac#1#2{\begingroup#1\endgroup\over#2}

\sqrtn Add an optional argument to plain's \sqrt to give the nth root of an expression
\@sqrtn  $\sqrt[n]{e}$ .

248 \DeclareRobustCommand\sqrtn{\ifnextchar[\@sqrtn\sqrtnsign}
249 \def\@sqrtn[#1]{\root #1\of}

eqnarray Here's the eqnarray environment: Default is for left-hand side of equations to be
\@eqcnt flushright. To make them flushleft, \let\@eqnset = \hfil.
\@eqpen 250 \newcount\@eqcnt
\if@eqnsw 251 \newcount\@eqpen
\@eqnset 252 \newif\if@eqnsw\@eqnswtrue
253 \newskip\@centering
254 \@centering = 0pt plus 1000pt

```

To get a proper `\@currentlabel` we have to redefine it for the whole display. Note that we can't use `\refstepcounter` as this results in `\@currentlabel` getting restored at the wrong and thus always writing the first label to the `.aux` file.

```

255 \def\eqnarray{%
256   \stepcounter{equation}%
257   \def\@currentlabel{\p@equation\theequation}%
258   \global\@eqnswtrue
259   \m@th
260   \global\@eqcnt\z@
261   \tabskip\@centering
262   \let\\\@eqnocr
263   $$\everycr{}\halign to\displaywidth\bgroup
264     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnscr
265     &\global\@eqcnt\@ne\hskip \tw@\arraycolsep \hfil${##}$\hfil
266     &\global\@eqcnt\tw@ \hskip \tw@\arraycolsep
267     $\displaystyle{##}$\hfil\tabskip\@centering
268     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
269     \tabskip\z@skip
270   \cr
271 }

272 \def\endeqnarray{%
273   \@eqnocr
274   \egroup
275   \global\advance\c@equation\m@ne
276   $$\@ignoretrue
277 }

278 \let\@eqnscr=\relax

\nonumber Switches off equation numbering.
279 \def\nonumber{\global\@eqnswfalse}

\@eqnocr
\@xeqnocr 280 \def\@xeqnocr{%
\@yeqnocr 281   {\ifnum0='}\fi
282   \@ifstar{%
283     \global\@eqpen\@M\@yeqnocr
284   }{%
285     \global\@eqpen\interdisplaylinepenalty \@yeqnocr
286   }%
287 }

288 \def\@yeqnocr{\@testopt\@xeqnocr\z@skip}

289 \def\@xeqnocr[#1]{%
290   \ifnum0='{ \fi}%
291   \@eqnocr
292   \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}%
293 }

\@@eqnocr
294 \def\@@eqnocr{\let\reserved@a\relax
295   \ifcase\@eqcnt \def\reserved@a{& &}\or \def\reserved@a{& &}%
296   \or \def\reserved@a{&}\else
297     \let\reserved@a\@empty

```

```

298      \@latex@error{Too many columns in eqnarray environment}\@ehc\fi
299      \reserved@a \if@eqnsw\@eqnnum\stepcounter{equation}\fi
300      \global\@eqnswtrue\global\@eqcnt\z@\cr}

eqnarray* Here's the eqnarray* environment:
\@seqnocr 301 \let\@seqnocr=\@eqnocr

302 \@namedef{eqnarray*}{\def\@eqnocr{\nonumber\@seqnocr}\eqnarray}
303 \@namedef{endeqnarray*}{\nonumber\endeqnarray}

\lefteqn \lefteqn{FORMULA} typesets FORMULA in display math style flushleft in a box of
width zero.
304 \def\lefteqn#1{\rlap{$\displaystyle #1$}}

\ensuremath In math mode, \ensuremath{text} is equivalent to text; in LR or paragraph
mode, it is equivalent to $text$. \relax is not needed in front of the \ifmmode as
\protect will be \let to \relax. This version (due to Donald Arseneau) avoids
duplicating its argument in the 'then' and 'else' part of the \ifmath which is
necessary in nested 'tabular' like environments. See amslatex/2104.
305 \DeclareRobustCommand{\ensuremath}{%
306   \ifmmode
307     \expandafter\@firstofone
308   \else
309     \expandafter\@ensuredmath
310   \fi}

\@ensuredmath The \relax stops \ensuremath{} starting display math.
311 \long\def\@ensuredmath#1{$\relax#1$}

312 </2ekernel>

```

## 53.3 External options to the standard document classes

### 53.3.1 Left equation numbering

`\@eqnnum` To put the equation number on the left side of an equation we have to use a little trick. The number is shifted `\displaywidth` to the left inside a box of (approximately) zero width. This fails when the equation is too wide, the equation number than may overprint the equation itself.

```

313 <*/leqno>
314 \renewcommand\@eqnnum{\hb@xt@.01\p@{}%
315      \rlap{\normalfont\normalcolor
316      \hskip -\displaywidth(\theequation)}}
317 </leqno>

```

### 53.3.2 Flush left equations

To get the displayed math environments to print the contents flush left (with an indentation) we have to redefine all of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>'s displayed math environments.

`\mathindent` The amount of indentation of the equations is stored in a register.

```

318 <*/fleqn>
319 \newdimen\mathindent

```

The setting of `\mathindent` has to be deferred until the class file has been processed, because `\leftmargin` is still 0pt wide at the moment `fileqn.clo` is read in.

```
320 \AtEndOfClass{\mathindent\leftmargin}
```

`\[` Begin display math;

```
321 \IncludeInRelease{2015/01/01}{\[]}{Make \[ robust}%
322 \DeclareRobustCommand\[\relax
323     \ifmmode\@badmath
324     \else
325         \begin{trivlist}%
326             \@beginparpenalty\predisplaypenalty
327             \@endparpenalty\postdisplaypenalty
328             \item[]\leavevmode
329             \hb@xt@\linewidth\bgroup $\m@th\displaystyle %$
330             \hskip\mathindent\bgroup
331         \fi}
332 \EndIncludeInRelease

333 \IncludeInRelease{0000/00/00}{\[]}{Make \[ robust}%
334 \renewcommand\[\relax
335     \ifmmode\@badmath
336     \else
337         \begin{trivlist}%
338             \@beginparpenalty\predisplaypenalty
339             \@endparpenalty\postdisplaypenalty
340             \item[]\leavevmode
341             \hb@xt@\linewidth\bgroup $\m@th\displaystyle %$
342             \hskip\mathindent\bgroup
343         \fi}
344 \EndIncludeInRelease
```

`\]` end display math;

```
345 \IncludeInRelease{2015/01/01}{\] }{Make \] robust}%
346 \DeclareRobustCommand\] \relax
347     \ifmmode
348         \egroup $\hfil% $
349     \egroup
350     \end{trivlist}%
351     \else \@badmath
352     \fi}
353 \EndIncludeInRelease

354 \IncludeInRelease{0000/00/00}{\] }{Make \] robust}%
355 \renewcommand\] \relax
356     \ifmmode
357         \egroup $\hfil% $
358     \egroup
359     \end{trivlist}%
360     \else \@badmath
361     \fi}
362 \EndIncludeInRelease
```

`equation` The equation environment

```

363 \renewenvironment{equation}%
364   {\@beginparpenalty\predisplaypenalty
365    \@endparpenalty\postdisplaypenalty
366    \refstepcounter{equation}%
367    \trivlist \item[]\leavevmode
368     \hb@xt@\linewidth\bgroup $\m@th% $
369     \displaystyle
370     \hskip\mathindent}%
371   {$\hfil % $
372    \displaywidth\linewidth\hbox{\@eqnnum}%
373   \egroup
374   \endtrivlist}

```

eqnarray The eqnarray environment

```

375 \renewenvironment{eqnarray}{%
376   \stepcounter{equation}%
377   \def\@currentlabel{\p@equation\theequation}%
378   \global\@eqnswtrue\m@th
379   \global\@eqcnt\z@
380   \tabskip\mathindent
381   \let\@=\@eqnocr
382   \setlength\abovedisplayskip{\topsep}%
383   \ifvmode
384     \addtolength\abovedisplayskip{\partopsep}%
385   \fi

```

When the documentclass uses a non-zero \parskip setting the \topsep might have a negative value to compensate for that. Therefore we add \parskip to \abovedisplayskip.

```

386   \addtolength\abovedisplayskip{\parskip}%
387   \setlength\belowdisplayskip{\abovedisplayskip}%
388   \setlength\belowdisplayshortskip{\abovedisplayskip}%
389   \setlength\abovedisplayshortskip{\abovedisplayskip}%
390   $$\everycr{}\halign to\linewidth% $$
391   \bgroup
392   \hskip\@centering
393   $\displaystyle\tabskip\z@skip{##}$\@eqnsele%
394   \global\@eqcnt\@ne \hskip \tw@\arraycolsep \hfil${##}$\hfil%
395   \global\@eqcnt\tw@ \hskip \tw@\arraycolsep
396   $\displaystyle{##}$\hfil \tabskip\@centering%
397   \global\@eqcnt\thr@@
398   \hb@xt@\z@\bgroup\hss##\egroup\tabskip\z@skip\cr}%
399   {\@eqnocr
400   \egroup
401   \global\advance\c@equation\m@ne$$$ $$
402   \@ignoretrue
403   }
404 \fleqn

```

## File A

# ltlists.dtx

### 54 List, and related environments

The generic commands for creating an indented environment – `enumerate`, `itemize`, `quote`, etc – are:

```
\list{<LABEL>}{<COMMANDS>} ... \endlist
```

which can be invoked by the user as the list environment. The LABEL argument specifies item labeling. COMMANDS contains commands for changing the horizontal and vertical spacing parameters.

Each item of the environment is begun by the command `\item[ITEMLABEL]` which produces an item labeled by ITEMLABEL. If the argument is missing, then the LABEL argument of the `\list` command is used as the item label.

The label is formed by putting `\makelabel{<ITEMLABEL>}` in an hbox whose width is either its natural width or else `\labelwidth`, whichever is larger. The `\list` command defines `\makelabel` to have the default definition:

```
\makelabel{<ARG>} == BEGIN \hfil ARG END
```

which, for a label of width less than `\labelwidth`, puts the label flushright, `\labelsep` to the left of the item's text. However, `\makelabel` can be `\let` to another command by the `\list`'s COMMANDS argument.

A `\usecounter{<foo>}` command in the second argument causes the counter *foo* to be initialized to zero, and stepped by every `\item` command without an argument. (`\label` commands within the list refer to this counter.)

When you leave a list environment, returning either to an enclosing list or normal text mode, LaTeX begins a new paragraph if and only if you leave a blank line after the `\end` command. This is accomplished by the `\@endparenv` command.

Blank lines are ignored every other reasonable place–i.e.:

- Between the `\begin{list}` and the first `\item`,
- Between the `\item` and the text of that item.
- Between the end of the last item and the `\end{list}`.

For an environment like quotation, in which items are not labeled, the entire environment is a single item. It is defined by letting `\quotation == \list{}{...}\item\relax`. (Note the `\relax`, there in case the first character in the environment is a '['.) The spacing parameters provide a great deal of flexibility in designing the format, including the ability to let the indentation of the first paragraph be different from that of the subsequent ones.

The trivlist environment is equivalent to a list environment whose second argument sets the following parameter values:

`\leftmargin = 0`: causes no indentation of left margin

`\labelwidth = 0`: see below for precise effect this has.

`\itemindent = 0`: with a null label, makes first paragraph have no indentation. Succeeding paragraphs have `\parindent` indentation. To give first paragraph same indentation, set `\itemindent = \parindent` before the `\item[]`.

Every `\item` in a trivlist environment must have an argument—in many cases, this will be the null argument (`\item[]`). The trivlist environment is mainly used for paragraphing environments, like `verbatim`, in which there is no margin change. It provides the same vertical spacing as the list environment, and works reasonably well when it occurs immediately after an `\item` command in an enclosing list.

## 54.1 List and Trivlist

The following variables are used inside a list environment:

`\totalleftmargin` The distance that the prevailing left margin is indented from the outermost left margin,

`\linewidth` The width of the current line. Must be initialized to `\hsize`.

`\listdepth` A count for holding current list nesting depth.

`\makelabel` A macro with a single argument, used to generate the label from the argument (given or implied) of the `\item` command. Initialized to `\mklab` by the `\list` command. This command must produce some stretch—i.e., an `\hfil`.

`\inlabel` A switch that is false except between the time an `\item` is encountered and the time that `TeX` actually enters horizontal mode. Should be tested by commands that can be messed up by the list environment's use of `\everypar`.

`\box\@labels` When `\inlabel = true`, it holds the labels to be put out by `\everypar`.

`\noparitem` A switch set by `\list` when `\inlabel = true`. Handles the case of a `\list` being the first thing in an item.

`\noparlist` A switch set true for a list that begins an item. No `\topsep` space is added before or after `\item`'s such a list.

`\newlist` Set true by `\list`, set false by the first text (by `\everypar`).

`\noitemarg` Set true when executing an `\item` with no explicit argument. Used to save space. To save time, make two separate `\@item` commands.

`\nmbrlist` Set true by `\usecounter` command, causes list to be numbered.

`\listctr` `\def`'ed by `\usecounter` to name of counter.

`\noskipsec` A switch set true by a sectioning command when it is creating an in-text heading with `\everypar`.

Throughout a list environment, `\hsize` is the width of the current line, measured from the outermost left margin to the outermost right margin. Environments like `tabbing` should use `\linewidth` instead of `\hsize`.

Here are the parameters of a list that can be set by commands in the `\list`'s COMMANDS argument. These parameters are all TeX skips or dimensions (defined by `\newskip` or `\newdimen`), so the usual TeX or L<sup>A</sup>T<sub>E</sub>X commands can be used to set them. The commands will be executed in vmode if and only if the `\list` was preceded by a `\par` (or something like an `\end{list}`), so the spacing parameters can be set according to whether the list is inside a paragraph or is its own paragraph.

## 54.2 Vertical Spacing (skips)

`\topsep`: Space between first item and preceding paragraph.

`\partopsep`: Extra space added to `\topsep` when environment starts a new paragraph (is called in vmode).

`\itemsep`: Space between successive items.

`\parsep`: Space between paragraphs within an item – the `\parskip` for this environment.

## 54.3 Penalties

`\@beginparpenalty`: put at the beginning of a list

`\@endparpenalty`: put at end of list

`\@itempenalty`: put between items.

## 54.4 Horizontal Spacing (dimens)

`\leftmargin`: space between left margin of enclosing environment (or of page if top level list) and left margin of this list. Must be nonnegative.

`\rightmargin`: analogous.

`\listparindent`: extra indentation at beginning of every paragraph of a list except the one started by the `\item` command. May be negative! Usually, labeled lists have `\listparindent` equal to zero.

`\itemindent`: extra indentation added right BEFORE an item label.

`\labelwidth`: nominal width of box that contains the label. If the natural width of the label  $\leq$  `\labelwidth`, then the label is flushed right inside a box of width `\labelwidth` (with an `\hfil`). Otherwise, a box of the natural width is employed, which causes an indentation of the text on that line.

`\labelsep`: space between end of label box and text of first item.

## 54.5 Default Values

Defaults for the list environment are set as follows. First, `\rightmargin`, `\listparindent` and `\itemindent` are set to 0pt. Then, one of the commands `\@listi`, `\@listii`, ... , `\@listvi` is called, depending upon the current level of the list. The `\@list ...` commands should be defined by the document style. A convention that the document style should follow is to set `\leftmargin` to `\leftmargini`, ..., `\leftmarginvi` for the appropriate level. Items that aren't changed may be left alone, but everything that could possibly be changed must be reset.

```
\list{LABEL}{COMMANDS} ==
BEGIN
  if \@listdepth > 5
    then LaTeX error: 'Too deeply nested'
    else \@listdepth :=G \@listdepth + 1
  fi
  \rightmargin      := 0pt
  \listparindent    := 0pt
  \itemindent       := 0pt
  \eval{@list \romannumeral\the\@listdepth} %% Set default values:
  \@itemlabel       :=L LABEL
  \makelabel        == \mklab
  @nmbrlist         :=L false
  COMMANDS

  \@trivlist          % commands common to \list and
\trivlist

  \parskip           :=L \parsep
  \parindent         :=L \listparindent
  \linewidth         :=L \linewidth - \rightmargin - \leftmargin
  \@totalleftmargin :=L \@totalleftmargin + \leftmargin
  \parshape 1 \@totalleftmargin \linewidth
  \ignorespaces      % gobble space up to \item
END

\endlist == BEGIN \@listdepth :=G \@listdepth -1
              \endtrivlist
              END

\@trivlist ==
BEGIN
  if @newlist = T then \@noitemerr fi
                      %% This command removed for some forgotten
reason.
  \@topsepadd :=L \topsep
  if @noskipsec then leave vertical mode fi %% Added 11 Jun 85
  if vertical mode
    then \@topsepadd :=L \@topsepadd + \partopsep
    else \unskip \par          % remove glue from end of last line
```

```

    fi
    if @inlabel = true
        then @noparitem :=L true
            @noparlist :=L true
        else @noparlist :=L false
            \@topsep :=L \@topsepadd
    fi
    \@topsep :=L \@topsep + \parskip %% Change 4 Sep 85
    \leftskip :=L 0pt % Restore paragraphing
parameters
    \rightskip :=L \@rightskip
    \parfillskip :=L 0pt + 1fil

NOTE: \@setpar called on every \list in case \par has been
temporarily munged before the \list command.
    \@setpar{if @newlist = false then {\@@par} fi}
    \@newlist :=G T
    \@outerparskip :=L \parskip
END

\trivlist ==
BEGIN
    \parsep := \parskip
    @nmbrlist := F
    \@trivlist
    \labelwidth := 0
    \leftmargin := 0
    \itemindent := \parindent
    \@itemlabel :=L "empty" %% added 93/12/13
    \makelabel{LABEL} == LABEL
END

\endtrivlist ==
BEGIN
    if @inlabel = T then \indent fi
    if horizontal mode then \unskip \par fi
    if @noparlist = true
        else if \lastskip > 0
            then \@tempskipa := \lastskip
                \vskip - \lastskip
                \vskip \@tempskipa - \@outerparskip + \parskip
            fi
        \@endparenv
    fi
END

\@endparenv ==
BEGIN
    \addpenalty{@endparpenalty}
    \addvspace{\@topsepadd}

```

```

\endgroup    %% ends the \begin command's \begingroup
\par == BEGIN
            \@restorepar
            \everypar{}
            \par
            END
\everypar == BEGIN remove \lastbox \everypar{} END
\begin group %% to match the \end commands \endgroup
END

\item == BEGIN if math mode then WARNING fi
            if next char = [
            then \@item
            else @noitemarg := true
                \@item[@itemlabel]
            END

\@item[LAB] ==
BEGIN
  if @noperitem = true
  then @noperitem := false
      % NOTE: then clause hardly every taken,
      % so made a macro \@donoperitem
      \box\@labels :=G \hbox{\hskip -\leftmargin
                          \box\@labels
                          \hskip \leftmargin }
  if @minipage = false then
      \@tempskipa := \lastskip
      \vskip -\lastskip
      \vskip \@tempskipa + \@outerparskip - \parskip
  fi
  else if @inlabel = true
      then \indent \par % previous item empty.
  fi
  if hmode then 2 \unskip's
      % To remove any space at end of prev.
      % paragraph that could cause a blank line.
      \par
  fi
  if @newlist = T
      then if @nobreak = T % Kludge if list follows \section
          then \addvspace{\@outerparskip - \parskip}
          else \addpenalty{\@beginparpenalty}
              \addvspace{\@topsep}
              \addvspace{-\parskip} %% added 4 Sep 85
          fi
      else \addpenalty{\@itempenalty}
          \addvspace{\itemsep}
      fi
  @inlabel :=G true

```

```

fi

\everypar{ @minipage :=G F
            @newlist :=G F
            if @inlabel = true
            then @inlabel :=G false
                \hskip -\parindent
                \box\@labels
                \penalty 0
                %% 3 Oct 85 - allow line break here
                \box\@labels :=G null
            fi
        \everypar{} }
@nobreak :=G false
if @noitemarg = true
then @noitemarg := false
    if @nmbrlist
    then \refstepcounter{\@listctr}
fi
fi
\@tempboxa :=L \hbox{\makelabel{LAB}}
\box\@labels :=G \@labels \hskip \itemindent
\hskip - (\labelwidth + \labelsep)
if \wd \@tempboxa > \labelwidth
then \box\@tempboxa
else \hbox to \labelwidth
{\unhbox\@tempboxa}
fi
\hskip\labelsep
\ignorespaces %gobble space up to text
END

\makelabel{LABEL} == ERROR %% default to catch lonely \item

\usecounter{CTR} == BEGIN @nmbrlist :=L true
                        \@listctr == CTR
                        \setcounter{CTR}{0}
                        END

DEFINE \dimen's and \count

\topskip
\partopsep 1 (*2ekernel)
\itemsep 2 \newskip\topsep
\parsep 3 \newskip\partopsep
\@topsep 4 \newskip\itemsep
\@topsepadd 5 \newskip\parsep
\outerparskip 6 \newskip\@topsep
7 \newskip\@topsepadd
8 \newskip\@outerparskip

```

```

\leftmargin
\rightmargin      9 \newdimen\leftmargin
\listparindent    10 \newdimen\rightmargin
\itemindent       11 \newdimen\listparindent
\labelwidth       12 \newdimen\itemindent
\labelsep         13 \newdimen\labelwidth
\@totalleftmargin 14 \newdimen\labelsep
                  15 \newdimen\linewidth
                  16 \newdimen\@totalleftmargin \@totalleftmargin=\z@

\leftmargini
\leftmarginii     17 \newdimen\leftmargini
\leftmarginiii    18 \newdimen\leftmarginii
\leftmarginiv     19 \newdimen\leftmarginiii
\leftmarginv      20 \newdimen\leftmarginiv
\leftmarginvi     21 \newdimen\leftmarginv
                  22 \newdimen\leftmarginvi

\@listdepth
\@itempenalty     23 \newcount\@listdepth \@listdepth=0
\@beginparpenalty 24 \newcount\@itempenalty
\@endparpenalty   25 \newcount\@beginparpenalty
                  26 \newcount\@endparpenalty

\@labels
                  27 \newbox\@labels

\if@inlabel
\@inlabelfalse   28 \newif\if@inlabel \@inlabelfalse
\@inlabeltrue
\if@newlist
\@newlistfalse   29 \newif\if@newlist \@newlistfalse
\@newlisttrue
\if@noparitem
\@noparitemfalse 30 \newif\if@noparitem \@noparitemfalse
\@noparitemtrue
\if@noparlist
\@noparlistfalse 31 \newif\if@noparlist \@noparlistfalse
\@noparlisttrue
\if@noitemarg
\@noitemargfalse 32 \newif\if@noitemarg \@noitemargfalse
\@noitemargtrue
\if@newlist
\@newlistfalse   33 \newif\if@nmbrlist \@nmbrlistfalse
\@newlisttrue
\list
                34 \def\list#1#2{%
                35   \ifnum \@listdepth >5\relax
                36     \@toodeep
                37   \else
                38     \global\advance\@listdepth\@ne
                39   \fi
                40   \rightmargin\z@

```

```

41 \listparindent\z@
42 \itemindent\z@
43 \csname @list\romannumeral\the\@listdepth\endcsname
44 \def\@itemlabel{#1}%
45 \let\makelabel\@mklab
46 \@nmbrrlistfalse
47 #2\relax
48 \@trivlist
49 \parskip\parsep
50 \parindent\listparindent
51 \advance\linewidth -\rightmargin
52 \advance\linewidth -\leftmargin
53 \advance\@totalleftmargin \leftmargin
54 \parshape \@ne \@totalleftmargin \linewidth
55 \ignorespaces}

```

\par@deathcycles

```

56 \newcount\par@deathcycles

```

\@trivlist Because \par is sometimes made a no-op it is possible for a missing \item to produce a loop that does not fill memory and so never gets trapped by T<sub>E</sub>X. We thus need to trap this here by setting \par to count the number of times a paragraph ii is called with no progress being made started.

```

57 \def\@trivlist{%
58   \if@noskipsec \leavevmode \fi
59   \@topsepadd \topsep
60   \ifvmode
61     \advance\@topsepadd \partopsep
62   \else
63     \unskip \par
64   \fi
65   \if@inlabel
66     \@nparitemtrue
67     \@nparlisttrue
68   \else
69     \if@newlist \@noitemerr \fi
70     \@nparlistfalse
71     \@topsep \@topsepadd
72   \fi
73   \advance\@topsep \parskip
74   \leftskip \z@skip
75   \rightskip \@rightskip
76   \parfillskip \@flushglue
77   \par@deathcycles \z@
78   \@setpar{\if@newlist
79             \advance\par@deathcycles \@ne
80             \ifnum \par@deathcycles >\@m
81               \@noitemerr
82               {\@par}%
83             \fi
84             \else
85               {\@par}%
86             \fi}%
87   \global \@newlisttrue

```

```
88 \@outerparskip \parskip}
```

```
\trivlist
```

```
89 \def\trivlist{%
90   \parsep\parskip
91   \@nmbrlistfalse
92   \@trivlist
93   \labelwidth\z@
94   \leftmargin\z@
95   \itemindent\z@
```

We initialise \@itemlabel so that a `trivlist` with an `\item` not having an optional argument doesn't produce an error message.

```
96   \let\@itemlabel\@empty
97   \def\makelabel##1{##1}}
```

```
\endlist
```

```
98 \def\endlist{%
99   \global\advance\@listdepth\m@ne
100  \endtrivlist}
```

The definition of `\trivlist` used to be in `ltspace.dtx` so that other commands could be 'let to it'. They now use `\def`.

```
\endtrivlist
```

```
101 \def\endtrivlist{%
102   \if@inlabel
103     \leavevmode
104     \global \@inlabelfalse
105   \fi
106   \if@newlist
107     \noitemerr
108     \global \@newlistfalse
109   \fi
110   \ifhmode\unskip \par
```

We also check if we are in math mode and issue an error message if so (hoping that \@currenvir resolves suitably). Otherwise the usual "perhaps a missing item" error will get triggered later which is confusing.

```
111   \else
112     \inmatherr{\end{\@currenvir}}}%
113   \fi
114   \if@noparlist \else
115     \ifdim\lastskip >\z@
116       \@tempskipa\lastskip \vskip -\lastskip
117       \advance\@tempskipa\parskip \advance\@tempskipa -\@outerparskip
118       \vskip\@tempskipa
119     \fi
120     \@endparenv
121   \fi
122 }
```

```
\@endparenv To suppress the paragraph indentation in text immediately following a paragraph-
\@doendpe    making environment, \everypar is changed to remove the space, and \par is
```

redefined to restore `\everypar`. Instead of redefining `\par` and `\everypar`, `\@endparenv` was changed to set the `@endpe` switch, letting `\end` redefine `\par` and `\everypar`.

This allows paragraph-making environments to work right when called by other environments. (Changed 27 Oct 86)

```
123 \def\@endparenv{%
124   \addpenalty\@endparpenalty\addvspace\@topsepadd\@endpetrue}

125 <latexrelease>\IncludeInRelease{2015/01/01}{\@doendpe}{clubpenalty fix}%
126 \def\@doendpe{\@endpetrue
127   \def\par{\@restorepar
```

If a section heading changes `\clubpenalty` to keep lines after it together then this modification is restored via the `\everypar` mechanism at the start of the next paragraph. As we destroy the contents of this token here we explicitly set `\clubpenalty` back to its default.

```
128   \clubpenalty\@clubpenalty
129   \everypar{\par\@endpefalse}\everypar
```

Use `\setbox0=\lastbox` instead of `\hskip -\parindent` so that a `\noindent` becomes a no-op when used before a line immediately following a list environment(23 Oct 86).

```
130   {\setbox\z@\lastbox}%
131   \everypar{\@endpefalse}}
132 <latexrelease>\EndIncludeInRelease

133 <latexrelease>\IncludeInRelease{0000/00/00}{\@doendpe}{clubpenalty fix}%
134 <latexrelease>\def\@doendpe{\@endpetrue
135 <latexrelease>   \def\par{\@restorepar\everypar{\par\@endpefalse}\everypar
136 <latexrelease>     {\setbox\z@\lastbox}\everypar{\@endpefalse}}
137 <latexrelease>\EndIncludeInRelease
```

```
\if@endpe
\@endpefalse 138 \newif\if@endpe
\@endpeltrue 139 \@endpefalse
```

```
\@mklab
140 \def\@mklab#1{\hfil #1}
```

```
\item
141 \def\item{%
142   \@inmatherr\item
143   \@ifnextchar [\@item{\@noitemargtrue \@item[\@itemlabel]}}
```

```
\@donoparitem
144 \def\@donoparitem{%
145   \@noparitemfalse
146   \global\setbox\@labels\hbox{\hskip -\leftmargin
147     \unhbox\@labels
148     \hskip \leftmargin}%
149   \if@minipage\else
150     \@tempskipa\lastskip
151     \vskip -\lastskip
```

```

152     \advance\@tempskipa\@outerparskip
153     \advance\@tempskipa -\parskip
154     \vskip\@tempskipa
155     \fi}

```

\@item

```

156 \def\@item[#1]{%
157   \if@noperitem
158     \@donoperitem
159   \else
160     \if@inlabel
161       \indent \par
162     \fi
163     \ifhmode
164       \unskip\unskip \par
165     \fi
166     \if@newlist
167       \if@nbreak
168         \@nbitem
169       \else
170         \addpenalty\@beginparpenalty
171         \addvspace\@topsep
172         \addvspace{-\parskip}%
173       \fi
174     \else
175       \addpenalty\@itempenalty
176       \addvspace\itemsep
177     \fi
178     \global\@inlabeltrue
179   \fi
180   \everypar{%
181     \@minipagefalse
182     \global\@newlistfalse

```

This \if@inlabel check is needed in case an item starts of inside a group so that \everypar does not become empty outside that group. nobreakfalse, etc etc.

```

183     \if@inlabel
184       \global\@inlabelfalse

```

The paragraph indent is now removed by using \setbox... since this makes \noindent a no-op here, as it should be. Thus the following comment is redundant but is left here for the sake of future historians: this next command was changed from an hskip to a kern to avoid a break point after the parindent box: the skip could cause a line-break if a very long label occurs in raggedright setting.

If \noindent was used after \item want to cancel the \itemindent skip. This case can be detected as the indentation box will be void.

```

185     {\setbox\z@\lastbox
186     \ifvoid\z@
187       \kern-\itemindent
188     \fi}%
189     \box\@labels
190     \penalty\z@
191   \fi

```

This code is intended to prevent a page break after the first line of an item that comes immediately after a section title. It may be sensible to always forbid a page break after one line of an item? As with all such settings of `\clubpenalty` it is local so will have no effect if the item starts in a group.

Only resetting `\@nbreak` when it is true is now essential since now it is sometimes set locally.

```

192   \if@nbreak
193     \@nbreakfalse
194     \clubpenalty \@M
195   \else
196     \clubpenalty \@clubpenalty
197     \everypar{}%
198   \fi}%

199   \if@noitemarg
200     \@noitemargfalse
201     \if@nmbrrlist

202     \refstepcounter\@listctr
203     \fi
204   \fi

```

We use `\sbox` to support colour commands.

```

205   \sbox\@tempboxa{\makelabel{#1}}%
206   \global\setbox\@labels\hbox{%
207     \unhbox\@labels
208     \hskip \itemindent
209     \hskip -\labelwidth
210     \hskip -\labelsep
211     \ifdim \wd\@tempboxa >\labelwidth
212       \box\@tempboxa

213   \else
214     \hbox to\labelwidth {\unhbox\@tempboxa}%
215   \fi
216   \hskip \labelsep}%
217   \ignorespaces}

```

`\makelabel`

```

218 \def\makelabel#1{%
219   \@latex@error{Lonely \string\item--perhaps a missing
220     list environment}\@ehc}

```

`\@nbitem`

```

221 \def\@nbitem{%
222   \@tempskipa\@outerparskip
223   \advance\@tempskipa -\parskip
224   \addvspace\@tempskipa}

```

`\usecounter`

```

225 \def\usecounter#1{\@nmbrrlisttrue\def\@listctr{#1}\setcounter{#1}\z@}

```

## 54.6 Itemize and Enumerate

Enumeration is done with four counters: `enumi`, `enumii`, `enumiii` and `enumiv`, where `enumN` controls the numbering of the Nth level enumeration. The label is generated by the commands `\labelenumi` ... `\labelenumiv`, which should be defined by the document style. Note that `\p@enumN\theenumN` defines the output of a `\ref` command. A typical definition might be:

```
\def\theenumii{\alph{enumii}}
\def\p@enumii{\theenumi}
\def\labelenumii{(\theenumii)}
```

which will print the labels as ‘(a)’, ‘(b)’, ... and print a `\ref` as ‘3a’.

The item numbers are moved to the right of the label box, so they are always a distance of `\labelsep` from the item.

`\@enumdepth` holds the current enumeration nesting depth.

Itemization is controlled by four commands: `\labelitemi`, `\labelitemii`, `\labelitemiii`, and `\labelitemiv`. To cause the second-level list to be bulleted, you just define `\labelitemii` to be `•`. `\@itemspacing` and `\@itemdepth` are the analogs of `\@enumspacing` and `\@enumdepth`.

```
\enumerate ==
BEGIN
  if \@enumdepth > 3
  then errormessage: “Too deeply nested”.
  else \@enumdepth :=L \@enumdepth + 1
      \@enumctr :=L eval(enum@\romannumeral\the\@enumdepth)
      \list{\label{\@enumctr}}
        {\usecounter{\@enumctr}
         \makelabel{LABEL} == \hss \llap{LABEL}}
    fi
END

\endenumerate == \endlist
```

```
\@enumdepth
226 \newcount\@enumdepth \@enumdepth = 0

\c@enumi
\c@enumii 227 \@definecounter{enumi}
\c@enumii 228 \@definecounter{enumii}
\c@enumiv 229 \@definecounter{enumiii}
230 \@definecounter{enumiv}

enumerate
231 \def\enumerate{%
232   \ifnum \@enumdepth >\thr@@\toodeep\else
233     \advance\@enumdepth\@ne
234     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%

235     \expandafter
236     \list
237     \csname label\@enumctr\endcsname
```

```

238         {\usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
239     \fi}

240 \let\endenumerate =\endlist

\itemize ==
BEGIN
    if \@itemdepth > 3
    then errormessage: 'Too deeply nested'.
    else \@itemdepth :=L \@itemdepth + 1
        \@itemitem ==
eval(labelitem\romannumeral\the\@itemdepth)
        \list{\@nameuse{\@itemitem}}
            {\makelabel{LABEL} == \hss \llap{LABEL}}
    fi
END

\enditemize == \endlist

\@itemdepth
241 \newcount\@itemdepth \@itemdepth = 0

itemize
242 \def\itemize{%
243     \ifnum \@itemdepth >\thr@@\toodeep\else
244         \advance\@itemdepth\@ne
245         \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%

246         \expandafter
247         \list
248         \csname\@itemitem\endcsname
249         {\def\makelabel##1{\hss\llap{##1}}}%
250     \fi}

251 \let\enditemize =\endlist
252 /2kernel)

```

# File B

## ltboxes.dtx

### 55 L<sup>A</sup>T<sub>E</sub>X Box commands

<code>\makebox</code>	<code>\makebox[⟨wid⟩][⟨pos⟩]{⟨obj⟩}</code> Puts <code>⟨obj⟩</code> in an <code>\hbox</code> of width <code>⟨wid⟩</code> , positioned by <code>⟨pos⟩</code> . The possible <code>⟨pos⟩</code> are: <code>s</code> stretched, <code>l</code> flushleft, <code>r</code> flushright, <code>c</code> (default) centred. If <code>⟨wid⟩</code> is missing, then <code>⟨pos⟩</code> is also missing and <code>⟨obj⟩</code> is put in an <code>\hbox</code> of its natural width. <code>\makebox(⟨x⟩,⟨y⟩)[⟨pos⟩]{⟨obj⟩}</code> Puts <code>⟨obj⟩</code> in an <code>\hbox</code> of width <code>x*\unitlength</code> and height <code>y*\unitlength</code> . <code>⟨pos⟩</code> arguments are <code>s</code> , <code>l</code> , <code>r</code> or <code>c</code> (default) for stretched, flushleft, flushright or centred, and <code>t</code> or <code>b</code> for top, bottom – or combinations like <code>tr</code> or <code>rb</code> . Default for horizontal and vertical are centered. Note that in this picture mode version of <code>\makebox</code> a <code>[b]</code> aligns on the <i>bottom</i> of the text as documented. If you want to align on the <i>baseline</i> use <code>\makebox( , ) [b]{\raisebox{0pt}{\height}[0pt]{xyz}}</code> or <code>\makebox( , ) [b]{\smash{xyz}}</code>
<code>\mbox</code>	<code>\mbox{⟨obj⟩}</code> The same as <code>\makebox{⟨obj⟩}</code> , but is more efficient as no checking for optional arguments is done.
<code>\newsavebox</code>	<code>\newsavebox{⟨cmd⟩}</code> : If <code>⟨cmd⟩</code> is undefined, then defines it to be a T <sub>E</sub> X box register.
<code>\savebox</code>	<code>\savebox{⟨cmd⟩} ...</code> : <code>⟨cmd⟩</code> is defined to be a T <sub>E</sub> X box register, and the ‘...’ are any <code>\makebox</code> arguments. It is like <code>\makebox</code> , except it doesn’t produce text but saves the value in <code>\box ⟨cmd⟩</code> .
<code>\sbox</code>	<code>\sbox{⟨cmd⟩}{⟨obj⟩}</code> is an efficient abbreviation for <code>\savebox{⟨cmd⟩}{⟨obj⟩}</code> .
<code>\lrbox</code>	<code>\begin{lrbox}{⟨cmd⟩}⟨text⟩\end{lrbox}</code> is equivalent to <code>\sbox{⟨cmd⟩}{⟨text⟩}</code> except that any white space at the beginning and end of <code>⟨text⟩</code> is ignored.
<code>\framebox</code>	<code>\framebox ...</code> : like <code>\makebox</code> , except it puts a ‘frame’ around the box. The frame is made of lines of thickness <code>\fboxrule</code> , separated by space <code>\fboxsep</code> from the text – except for <code>\framebox(X,Y) ...</code> , where the thickness of the lines is as for the picture environment, and there is no separation added.
<code>\fbox</code>	<code>\fbox{⟨obj⟩}</code> is an abbreviation for <code>\framebox{⟨obj⟩}</code> .
<code>\parbox</code>	<code>\parbox[⟨pos⟩][⟨height⟩][⟨inner-pos⟩]{⟨width⟩}{⟨text⟩}</code> : Makes a box with <code>\hsize ⟨width⟩</code> , positioned by <code>⟨pos⟩</code> as follows: <code>c</code> : <code>\vcenter</code> (placed in <code>\$. . . \$</code> if not in math mode) <code>b</code> : <code>\vbox</code> <code>t</code> : <code>\vtop</code> default value is <code>c</code> . Sets <code>\hsize := ⟨width⟩</code> and calls <code>\@parboxrestore</code> , which does the following: Restores the original definitions of:

```

\par
\\
\--
\'
\'
\=
Resets the following parameters:
\parindent      = 0pt
\parskip        = 0pt
\linewidth      = \hsize
\@totalleftmargin = 0pt
\leftskip       = 0pt
\rightskip      = 0pt
\@rightskip     = 0pt
\parfillskip    = 0pt plus 1fil
\lineskip       = \normallineskip
\baselineskip   = \normalbaselineskip
Calls \sloppy
Note: \arrayparboxrestore same as \parboxrestore but it doesn't re-
store \\.
minipage      minipage : Similar to \parbox, except it also makes this look like a page by
setting
\textwidth == \columnwidth == box width
changes footnotes by redefining:
\@mpfn == mpfootnote
\thempfn == \thempfootnote
\@footnotetext == \@mpfootnotetext
resets the following list environment parameters:
\@listdepth == \@mplistdepth
where \@mplistdepth is initialized to zero,
and executes \@minipagerestore to allow the document style to reset any
other parameters it desires. It sets @minipage true, and resets \everypar to set it
false. This switch keeps \addvspace from putting space at the top of a minipage.
Change added 24 May 89: \minipage sets @minipage globally; \endminipage
resets it false.
\rule      \rule[\langle raised \rangle]{\langle width \rangle}{\langle height \rangle} : Makes a \langle width \rangle * \langle height \rangle rule, raised
\langle raised \rangle.
\underline \underline{\langle text \rangle} : Makes an underlined hbox with \langle text \rangle in it.
\raisebox \raisebox{\langle distance \rangle}[\langle height \rangle][\langle depth \rangle]{\langle box \rangle} :
Raises \langle box \rangle up by \langle distance \rangle length (down if \langle distance \rangle negative). Makes TEX
think that the new box extends \langle height \rangle above the line and \langle depth \rangle below, for a
total vertical length of \langle height \rangle + \langle depth \rangle. Default values of \langle height \rangle & \langle depth \rangle =
actual height and depth of box in new position.
1 \langle *2kernel \rangle
2 \message{boxes,}

\makebox \makebox User level command just looks for optional [ or (.
3 \langle /2kernel \rangle
4 \langle latexrelease \rangle \IncludeInRelease{2015/01/01}%
5 \langle latexrelease \rangle { \makebox } { Make \makebox robust } %

```

```

6 <*2ekernel | latexrelease>
7 \DeclareRobustCommand\makebox{%
8   \leavevmode
9   \@ifnextchar(%)
10    \makepicbox
11    {\@ifnextchar[\@makebox\mbox}}}%
12 </2ekernel | latexrelease>
13 <latexrelease>\EndIncludeInRelease
14 <latexrelease>\IncludeInRelease{0000/00/00}%
15 <latexrelease>          {\makebox}{Make \makebox robust}%
16 <latexrelease>\def\makebox{%
17 <latexrelease>  \leavevmode
18 <latexrelease>  \@ifnextchar(%)
19 <latexrelease>    \makepicbox
20 <latexrelease>    {\@ifnextchar[\@makebox\mbox}}}%
21 <latexrelease>\EndIncludeInRelease
22 <*2ekernel>

```

`\mbox` The basic horizontal box command for L<sup>A</sup>T<sub>E</sub>X.

```
23 \long\def\mbox#1{\leavevmode\hbox{#1}}
```

`\@makebox` Look for a possible second optional argument (defaults to c).

```
24 \def\@makebox[#1]{%
25   \@ifnextchar [{\@imakebox[#1]}{\@imakebox[#1][c]}}
```

`\@begin@tempboxa` Helper macro for supporting `\height`, `\width` etc. Grab #1 into `\@tempboxa` and measure it.

```

26 \long\def\@begin@tempboxa#1#2{%
27   \begingroup
28   \setbox\@tempboxa#1{\color@begingroup#2\color@endgroup}%
29   \def\width{\wd\@tempboxa}%
30   \def\height{\ht\@tempboxa}%
31   \def\depth{\dp\@tempboxa}%
32   \let\totalheight\@ovri
33   \totalheight\height
34   \advance\totalheight\depth}

```

`\@end@tempboxa` End the group started by `\@begin@tempboxa`, so that the scope of `\height` only includes the ‘length’ argument to the user-command.

```
35 \let\@end@tempboxa\endgroup
```

`\bm@c` Set up spacing.

```

\bm@l 36 \def\bm@c{\hss\unhbox\@tempboxa\hss}
\bm@r 37 \def\bm@l{\unhbox\@tempboxa\hss}\let\bm@t\bm@l
\bm@s 38 \def\bm@r{\hss\unhbox\@tempboxa}\let\bm@b\bm@r
\bm@t 39 \def\bm@s{\unhbox\@tempboxa}
\bm@b

```

`\@imakebox` Internal form of `\makebox`.

```

40 \long\def\@imakebox[#1][#2]#3{%
41   \@begin@tempboxa\hbox{#3}%
42   \setlength\@tempdima{#1}%          support calc
43   \hb@xt@\@tempdima{\csname bm@#2\endcsname}%
44   \@end@tempboxa}

```

`\@makepicbox` Picture mode form of `\makebox`.  
45 `\def\@makepicbox(#1,#2){%`  
46 `\@ifnextchar[{ \@makepicbox(#1,#2)}{\@makepicbox(#1,#2) []}]}`

`\@imakepicbox` picture mode version  
47 `\long\def\@imakepicbox(#1,#2)[#3]#4{%`  
48 `\vbox to#2\unitlength`  
49 `{\let\mb@b\vss \let\mb@l\hss\let\mb@r\hss`  
50 `\let\mb@t\vss`  
51 `\@tfor\reserved@a :=#3\do{%`  
52 `\if s\reserved@a`  
53 `\let\mb@l\relax\let\mb@r\relax`  
54 `\else`  
55 `\expandafter\let\csname mb@\reserved@a\endcsname\relax`  
56 `\fi}%`  
57 `\mb@t`  
58 `\hb@xt@ #1\unitlength{\mb@l #4\mb@r}%`  
59 `\mb@b`

This kern ensures that a `b` option aligns on the bottom of the text rather than the baseline. this is the documented behaviour in the `LATEX`Book. The kern is removed in compatibility mode.

60 `\kern\z@}}`

`\set@color` This macro is initially a no-op, but the colour package will redefine it to insert a `\special`.

61 `\let\set@color\relax`

`\color@begingroup` These macros are initially a no-op, but the colour package will redefine them to be `\begingroup`, `\endgroup`, `\begingroup\set@color`,  
`\color@endgroup` `\hbox\bgroup\color@begingroup`, `\color@endgroup\egroup`. and *(set to main document colour)* respectively.

`\color@hbox` 62 `\let\color@begingroup\relax`  
`\color@vbox` 63 `\let\color@endgroup\relax`  
`\color@endbox` 64 `\let\color@setgroup\relax`  
65 `\let\normalcolor\relax`  
66 `\let\color@hbox\relax`  
67 `\let\color@vbox\relax`  
68 `\let\color@endbox\relax`

`\newsavebox` Allocate a new ‘savebox’.

69 `\def\newsavebox#1{\@ifdefinable{#1}{\newbox#1}}`

`\savebox` Save #1 in a box register.

70 `/2kernel)`  
71 `(latexrelease)\IncludeInRelease{2015/01/01}%`  
72 `(latexrelease) \savebox}{Make \savebox robust}%`  
73 `(*2kernel | latexrelease)`  
74 `\DeclareRobustCommand\savebox[1]{%`  
75 `\@ifnextchar(%)`  
76 `{\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}%`  
77 `/2kernel | latexrelease)`  
78 `(latexrelease)\EndIncludeInRelease`

```

79 \latexrelease\IncludeInRelease{0000/00/00}%
80 \latexrelease\def\savebox#1{\makebox[0pt]{\box#1}}%
81 \latexrelease\def\savebox#1{\makebox[0pt]{\box#1}}%
82 \latexrelease\def\savebox#1{\makebox[0pt]{\box#1}}%
83 \latexrelease\def\savepicbox#1{\makebox[0pt]{\box#1}}%
84 \latexrelease\EndIncludeInRelease
85 \*2kernel

\savebox Save #1 in a box register.
86 \long\def\savebox#1#2{\setbox#1\hbox{%
87   \color@setgroup#2\color@endgroup}}

\@savebox Look for second optional argument.
88 \def\@savebox#1[#2]{%
89   \ifnextchar [\@isavebox#1[#2]]{\@isavebox#1[#2][c]}}

\@isavebox
90 \long\def\@isavebox#1[#2][#3]#4{%
91   \savebox#1{\imakebox[#2][#3]{#4}}}

\@savepicbox Picture mode version of \savebox.
92 \def\@savepicbox#1(#2,#3){%
93   \ifnextchar [%]
94     {\@isavepicbox#1(#2,#3)}{\@isavepicbox#1(#2,#3) []}}

\@isavepicbox Picture mode version of \savebox.
95 \long\def\@isavepicbox#1(#2,#3)[#4]#5{%
96   \savebox#1{\@imakepicbox(#2,#3)[#4]{#5}}}

\lrbox \lrbox: the new environment form of \savebox. Use \aftergroup tricks to enable a
local assignment to be made to the box, in a way that it still has an effect outside
the \lrbox environment.
97 \def\lrbox#1{%
98   \edef\reserved@a{%
99     \endgroup
100     \setbox#1\hbox{%
101       \begingroup\aftergroup}%
102     \def\noexpand\@currentenv{\@currentenv}%
103     \def\noexpand\@currentline{\on@line}}%
104   \reserved@a
105   \endpefalse
106   \color@setgroup
107   \ignorespaces}

\endlrbox End the \lrbox environment.
108 \def\endlrbox{\unskip\color@endgroup}

\usebox unchanged
109 \def\usebox#1{\leavevmode\copy #1\relax}

\frame The following definition of \frame was written by Pavel Curtis (Extra space
removed 14 Jan 88) RmS 92/08/24: Replaced occurrence of \@halfwidth by
\@wholewidth

```

```

110 \long\def\frame#1{%
111   \leavevmode
112   \hbox{%
113     \hskip-\@wholewidth
114     \vbox{%
115       \vskip-\@wholewidth
116       \hrule \@height\@wholewidth
117       \hbox{%
118         \vrule\@width\@wholewidth
119         #1%
120         \vrule\@width\@wholewidth}%
121       \hrule \@height\@wholewidth
122       \vskip-\@wholewidth}%
123     \hskip-\@wholewidth}}

\fbboxrule user level parameters,
\fbboxsep 124 \newdimen\fbboxrule
125 \newdimen\fbboxsep

\fbbox Abbreviated framed box command.
126 \long\def\fbbox#1{%
127   \leavevmode
128   \setbox\@tempboxa\hbox{%
129     \color@begingroup
130     \kern\fbboxsep{#1}\kern\fbboxsep
131     \color@endgroup}%
132   \@framebox\relax}

\framebox Framed version of \makebox.
133 \ifx\kernel
134 \ifx\latexrelease\IncludeInRelease{2015/01/01}%
135 \ifx\latexrelease\framebox\Make\framebox robust}%
136 \ifx\kernel\ifx\latexrelease
137 \DeclareRobustCommand\framebox{%
138   \ifnextchar(
139     \framepicbox{\ifnextchar[\@framebox\fbbox}}%
140 \ifx\kernel\ifx\latexrelease
141 \ifx\latexrelease\EndIncludeInRelease
142 \ifx\latexrelease\IncludeInRelease{0000/00/00}%
143 \ifx\latexrelease\framebox\Make\framebox robust}%
144 \ifx\latexrelease\def\framebox{%
145 \ifx\latexrelease\ifnextchar(
146 \ifx\latexrelease\framepicbox{\ifnextchar[\@framebox\fbbox}}%
147 \ifx\latexrelease\EndIncludeInRelease
148 \ifx\kernel

\@framebox Deal with optional arguments.
149 \def\@framebox[#1]{%
150   \ifnextchar[
151     {\@framebox[#1]}%
152     {\@framebox[#1][c]}}

\@ifframebox The handling the optional arguments. In order to set the whole box, including
the frame to the specified dimension, we first determine that dimension from the
natural size of the text, #3. calculated width.

```

```

153 \long\def\@iframebox[#1][#2]#3{%
154   \leavevmode
155   \@begin@tempboxa\hbox{#3}%
156     \setlength\@tempdima{#1}%
157     \setbox\@tempboxa\hb@xt@\@tempdima
158       {\kern\fbboxsep\csname bm@#2\endcsname\kern\fbboxsep}%
159     \@frameb@x{\kern-\fbboxrule}%
160   \@end@tempboxa}

\@frameb@x Common part of \framebox and \fbbox. #1 is a negative kern in the \framebox
case so that the vertical rules do not add to the width of the box.
161 \def\@frameb@x#1{%
162   \@tempdima\fbboxrule
163   \advance\@tempdima\fbboxsep
164   \advance\@tempdima\dp\@tempboxa
165   \hbox{%
166     \lower\@tempdima\hbox{%
167       \vbox{%
168         \hrule\@height\fbboxrule
169         \hbox{%
170           \vrule\@width\fbboxrule
171           #1%
172           \vbox{%
173             \vskip\fbboxsep
174             \box\@tempboxa
175             \vskip\fbboxsep}%
176           #1%
177           \vrule\@width\fbboxrule}%
178         \hrule\@height\fbboxrule}%
179       }%
180     }%
181 }

\@framepicbox Picture mode version.
182 \def\@framepicbox(#1,#2){%
183   \@ifnextchar[{\@iframepicbox(#1,#2)}{\@iframepicbox(#1,#2) []}}

\@iframepicbox Picture mode version.
184 \long\def\@iframepicbox(#1,#2)[#3]#4{%
185   \frame{\@makepicbox(#1,#2)[#3]{#4}}

\parbox The main vertical-box command for LATEX.
186 \</2ekernel>
187 \<latexrelease>\IncludeInRelease{2015/01/01}%
188 \<latexrelease>          {\parbox}{Make \parbox robust}%
189 \<*2ekernel | latexrelease>
190 \DeclareRobustCommand\parbox{%
191   \@ifnextchar[%]
192     \@iparbox
193     {\@iiiiparbox c\relax[s]}}%
194 \</2ekernel | latexrelease>
195 \<latexrelease>\EndIncludeInRelease
196 \<latexrelease>\IncludeInRelease{0000/00/00}%
197 \<latexrelease>          {\parbox}{Make \parbox robust}%

```

```

198 \latexrelease\def\parbox{%
199 \latexrelease\@ifnextchar[%]
200 \latexrelease\@iparbox
201 \latexrelease\{\@iiiparbox c\relax[s]}\}%
202 \latexrelease\EndIncludeInRelease
203 \*2ekernel)

```

`\@iparbox` Optional argument handling.

```

204 \def\@iparbox[#1]{%
205 \@ifnextchar[%]
206 {\@iparbox{#1}}%
207 {\@iiiparbox{#1}\relax[s]}}

```

`\@iiiparbox` Optional argument handling.

```

208 \def\@iiiparbox#1[#2]{%
209 \@ifnextchar[%]
210 {\@iiiparbox{#1}{#2}}%
211 {\@iiiparbox{#1}{#2}[#1]}}

```

`\@iiiparbox` The internal version of `\parbox`.

```

\@parboxto 212 \let\@parboxto\@empty
213 \long\def\@iiiparbox#1#2[#3]#4#5{%
214 \leavevmode
215 \@pboxswfalse
216 \setlength\@tempdima{#4}%
217 \@begin@tempboxa\vbox{\hsize\@tempdima\@parboxrestore#5\@par}%
218 \ifx\relax#2\else
219 \setlength\@tempdimb{#2}%
220 \edef\@parboxto{to\the\@tempdimb}%
221 \fi
222 \if#1b\vbox
223 \else\if #1t\vtop
224 \else\ifmmode\vcenter
225 \else\@pboxswtrue $\vcenter
226 \fi\fi\fi
227 \@parboxto{\let\hss\vss\let\unhbox\unvbox
228 \csname bm@#3\endcsname}%
229 \if@pboxsw \m@th$\fi
230 \@end@tempboxa}

```

`\@arrayparboxrestore` Restore various paragraph parameters.

The rationale for allowing two normally global flags to be set locally here was stated originally by Donald Arsenau and extended by Chris Rowley. It is because these flags are only set globally to true by section commands, and these should never appear within boxes or, indeed, in any group; and they are only ever set globally to false when they are definitely true.

If anyone is unhappy with this argument then both flags should be treated as in `\set@nobreak`; otherwise this command will be redundant.

```

231 \def\@arrayparboxrestore{%
232 \let\if@nobreak\iffalse
233 \let\if@noskipsec\iffalse
234 \let\par\@par
235 \let\-\@dischyph

```

```

Redefined accents to allow changes in font encoding
236 \let\'@acci\let\'@accii\let\=@acciii
237 \parindent\z@ \parskip\z@skip
238 \everypar{}%
239 \linewidth\hsize
240 \@totalleftmargin\z@
241 \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
242 \parfillskip\@flushglue \lineskip\normallineskip
243 \baselineskip\normalbaselineskip
244 \sloppy}

\parboxrestore Restore various paragraph parameters, and also \\.
245 \def\@parboxrestore{\arrayparboxrestore\let\\\@normalcr}

\if@minipage Switch that is true at the start of a minipage.
246 \def\@minipagefalse{\global\let\if@minipage\iffalse}
247 \def\@minipagetrue {\global\let\if@minipage\iftrue}
248 \@minipagefalse

\minipage Essentially an environment form of \parbox.
249 \def\minipage{%
250   \@ifnextchar[%]
251     \@iminipage
252     {\@iiiminipage c\relax[s]}}

\iminipage Optional argument handling.
253 \def\iminipage[#1]{%
254   \@ifnextchar[%]
255     {\@iiiminipage{#1}}%
256     {\@iiiminipage{#1}\relax[s]}}

\@iiiminipage Optional argument handling.
257 \def\@iiiminipage#1[#2]{%
258   \@ifnextchar[%]
259     {\@iiiminipage{#1}{#2}}%
260     {\@iiiminipage{#1}{#2}[#1]}}

\@iiiminipage Internal form of minipage.
261 \def\@iiiminipage#1#2[#3]#4{%
262   \leavevmode
263   \@pboxswfalse
264   \setlength\@tempdima{#4}%
265   \def\@mpargs{#1}{#2}[#3]{#4}%
266   \setbox\@tempboxa\vbox\bgroup
267     \color@begingroup
268       \hsize\@tempdima
269       \textwidth\hsize \columnwidth\hsize
270       \parboxrestore
271       \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
272       \let\@footnotetext\mpfootnotetext
273       \let\@listdepth\@mplistdepth \@mplistdepth\z@
274       \minipagerestore
275       \setminipage}

```

```

\@minipagerestore Hook so that other styles can reset other commands in a minipage.
276 \let\@minipagerestore=\relax

\endminipage
277 \def\endminipage{%
278   \par
279   \unskip
280   \ifvoid\@mpfootins\else
281     \vskip\skip\@mpfootins
282     \normalcolor
283     \footnoterule
284     \unvbox\@mpfootins
285   \fi
286   \@minipagefalse %% added 24 May 89
287   \color@endgroup
288   \egroup
289   \expandafter\@iiparbox\@mpargs{\unvbox\@tempboxa}}

\@mplistdepth Versions of \@listdepth and \footins local to minipage.
\@mpfootins 290 \newcount\@mplistdepth
291 \newinsert\@mpfootins

\@mpfootnotetext Minipage version of \@footnotetext.
Final \strut added 27 Mar 89, on suggestion by Don Hosek
292 \long\def\@mpfootnotetext#1{%
293   \global\setbox\@mpfootins\vbox{%
294     \unvbox\@mpfootins
295     \reset@font\footnotesize
296     \hsize\columnwidth
297     \@parboxrestore
298     \protected@edef\@currentlabel
299       {\csname p@mpfootnote\endcsname\@thefnmark}%
300     \color@begingroup
301     \@makefntext{%
302       \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
303     \color@endgroup}}

304 \newif\if@pboxsw

\rule Draw a rule of the specified size.
305 (/2ekernel)
306 \if@latexrelease\IncludeInRelease{2015/01/01}%
307 \if@latexrelease{\rule}{\Make \rule robust}%
308 \if@*2ekernel\if@latexrelease
309 \DeclareRobustCommand\rule{\@ifnextchar[\@rule{\@rule[\z@]}}%
310 \if@*2ekernel\if@latexrelease
311 \if@latexrelease\EndIncludeInRelease
312 \if@latexrelease\IncludeInRelease{0000/00/00}%
313 \if@latexrelease{\rule}{\Make \rule robust}%
314 \if@latexrelease\def\rule{\@ifnextchar[\@rule{\@rule[\z@]}}%
315 \if@latexrelease\EndIncludeInRelease
316 \if@*2ekernel

```

```

\@rule Internal form of \rule.
317 \def\@rule[#1]#2#3{%
318 \leavevmode
319 \hbox{%
320 \setlength\@tempdima{#1}%
321 \setlength\@tempdimb{#2}%
322 \setlength\@tempdimc{#3}%
323 \advance\@tempdimc\@tempdima
324 \vrule\@width\@tempdimb\@height\@tempdimc\@depth-\@tempdima}}

\@@underline Saved primitive \underline.
325 \let\@@underline\underline

\underline LATEX version works outside math.
326 \def\underline#1{%
327 \relax
328 \ifmmode\@@underline{#1}%
329 \else $\@@underline{\hbox{#1}}\m@th$\relax\fi}

\raisebox Raise a box, and change its vertical dimensions.
330 \</2ekernel>
331 \<latexrelease>\IncludeInRelease{2015/01/01}%
332 \<latexrelease> \<raisebox>{\Make \raisebox robust}%
333 \<*2ekernel | latexrelease>
334 \DeclareRobustCommand\raisebox[1]{%
335 \leavevmode
336 \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1}[]}}
337 \</2ekernel | latexrelease>
338 \<latexrelease>\EndIncludeInRelease
339 \<latexrelease>\IncludeInRelease{0000/00/00}%
340 \<latexrelease> \<raisebox>{\Make \raisebox robust}%
341 \<latexrelease>\def\raisebox#1{%
342 \<latexrelease> \leavevmode
343 \<latexrelease> \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1}[]}}
344 \<latexrelease>\EndIncludeInRelease
345 \<*2ekernel>

\@rsbox Optional argument handling.
346 \def\@rsbox#1[#2]{%
347 \@ifnextchar[{\@irsbox{#1}[#2]}{\@irsbox{#1}[#2]}}

\@argsbox ...

\@irsbox Internal version of \raisebox (less than two optional args).
348 \long\def\@irsbox#1[#2]#3{%
349 \@begin@tempboxa\hbox{#3}%
350 \setlength\@tempdima{#1}%
351 \ifx\#2\\\else\setlength\@tempdimb{#2}\fi
352 \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
353 \ifx\#2\\\else\ht\@tempboxa\@tempdimb\fi
354 \box\@tempboxa
355 \@end@tempboxa}

```

`\@iirsbox` Internal version of `\raisebox` (two optional args).

```

356 \long\def\@iirsbox#1[#2][#3]#4{%
357   \@begin@tempboxa\hbox{#4}%
358   \setlength\@tempdima{#1}%
359   \setlength\@tempdimb{#2}%
360   \setlength\dimen@{#3}%
361   \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
362   \ht\@tempboxa\@tempdimb
363   \dp\@tempboxa\dimen@
364   \box\@tempboxa
365   \@end@tempboxa}

```

`\@finalstrut` This macro adds a special strut the *depth* of the box given as `#1`, and height and width 0pt. It is used for ensuring that the last line of a paragraph has the correct depth in ‘p’ columns of tables and in footnotes. In vertical mode nothing is done, as adding the strut (as done in 2.09) would start a new paragraph. It would be possible to inspect `\prevdepth` to check the depth of the just-completed paragraph, but we do not do that here. Actually we do even less now, skip the vmode test as it broke tabular ‘p’ columns. .

The `\nobreak` was added (1995/10/31) to allow hyphenation of the final word of the paragraph.

```

366 \def\@finalstrut#1{%
367   \unskip\ifhmode\nobreak\fi\vrule\@width\z@\@height\z@\@depth\dp#1}

```

## 55.1 Some low-level constructs

The following commands are basically inherited from plain T<sub>E</sub>X.

`\leftline` These macros place text on a full line either centred or left or right adjusted.

`\rightline`

```

368 \def\@@line{\hb@xt@\hsize}
369 \def\leftline#1{\@@line{#1\hss}}
\centerline 370 \def\rightline#1{\@@line{\hss#1}}
\@@line      371 \def\centerline#1{\@@line{\hss#1\hss}}

```

`\rlap` These macros place text to the left or right of the current reference point without

`\llap` taking up space.

```

372 \def\rlap#1{\hb@xt@\z@{#1\hss}}
373 \def\llap#1{\hb@xt@\z@{\hss#1}}

```

374 `/2ekernel`

## File C

# lftab.dtx

## 56 Tabbing, Tabular and Array Environments

This section deals with ‘Lining It Up in Columns’. First the `tabbing` environment is defined, and then in second part, `tabular` together with its variants, `tabular*` and `array`.

Note that the `tabular` defined here is essentially the original L<sup>A</sup>T<sub>E</sub>X 2.09 version, not the extended version described in *The L<sup>A</sup>T<sub>E</sub>X Companion*. Use the `array` package to obtain the extended version.

### 56.1 tabbing

`\dimen{\@firsttab + i}` = distance of tab stop `i` from left margin  
0 <= `i` <= 15 (?).

`\dimen\@firsttab` is initialized to `\@totalleftmargin`, so it starts at the prevailing left margin.

`\@maxtab` = number of highest defined tab register  
probably = `\@firsttab + 12`

`\@nxttabmar` = tab stop number of next line’s left margin

`\@curtabmar` = tab stop number of current line’s left margin

`\@curtab` = number of the current tab. At start of line,  
it equals `\@curtabmar`

`\@hightab` = largest tab number currently defined.

`\@tabpush` = depth of `\pushtab`’s

`\box\@curline` = contents of current line, excluding left margin  
skip, and excluding contents of current field

`\box\@curfield` = contents of current field

`@rjfield` = switch: T iff the last field of the line should  
be right-justified at the right margin.

`\tabbingsep` = distance left by the `\’` command between the  
current position and the field that is  
“left-shifted”.

#### UTILITY MACROS

`\@stopfield` : closes the current field

`\@addfield` : adds the current field to the current line.

`\@contfield` : continues the current field

`\@startfield` : begins the next field

`\@stopline` : closes the current line and outputs it

`\@startline` : starts the next line

`\@ifatmargin` : an `\if` that is true iff the current line.

has width zero

```

\@startline ==
BEGIN
  \@curtabmar :=G \@nxttabmar
  \@curtab :=G \@curtabmar
  \box\@curline :=G null
  \@startfield
  \strut
END

\@stopline ==
BEGIN
  \unskip
  \@stopfield
  if @rjfield = T
    then @rjfield :=G F
      \@tempdima := \@totalleftmargin + \linewidth
      \hb@xt@ \@tempdima{\@itemfudge
                          \hskip \dimen\@curtabmar
                          \box\@curline
                          \hfil
                          \box\@curfield}
    else \@addfield
      \hbox {\@itemfudge
            \hskip \dimen\@curtabmar
            \box\@curline}
    fi
  END

\@startfield ==
BEGIN
  \box\@curfield :=G \hbox {
  END

\@stopfield ==
BEGIN
  }
  END

\@contfield ==
BEGIN
  \box\@curfield :=G \hbox { \unhbox\@currfield %%} brace
matching
  END
\@addfield ==
BEGIN
  \box\@curline :=G \unbox\@curline * \unbox\@curfield
  END

```

```

\@ifatmargin ==
BEGIN
  if dim of box\@curline = 0pt then
  END

\tabbing ==
BEGIN
  \lineskip :=L 0pt
  \> == \@rtab
  \< == \@ltab
  \= == \@settab
  \+ == \@tabplus
  \- == \@tabminus
  \‘ == \@tabrj
  \’ == \@tablab
  \\\ == BEGIN \@stopline \@startline END
  \\[DIST] == BEGIN
    \@stopline \vskip DIST \@startline\ignorespaces
  END
  \\\* == BEGIN \@stopline \penalty 10000 \@startline END
  \\[DIST] == BEGIN \@stopline \penalty 10000 \vskip DIST
    \@startline\ignorespaces END
  \@hightab := \@nxttabmar :=G \@firsttab
  \@tabpush :=G 0
  \dimen\@firsttab := \@totalleftmargin
  @rjfield :=G F
  \trivlist \item\relax
  if @minipage = F then \vskip \parskip fi
  \box\@tabfbox = \rlap{\indent\the\everypar}
    % note: \the\everypar sets @inlabel :=G F
  \@itemfudge == BEGIN \box\@tabfbox END
  \@startline
  \ignorespaces
  END

\@endtabbing ==
BEGIN
  \@stopline
  if \@tabpush > 0 then error message: "unmatched \poptabs" fi
  \endtrivlist
  END

\@rtab ==
BEGIN
  \@stopfield
  \@addfield
  if \@curtab < \@hightab
    then \@curtab :=G \@curtab + 1
    else error message "Undefined Tab" fi

```

```

\@tempdima := \dimen\@curtab - \dimen\@curtabmar
              - width of box \@curline
\box\@curline :=G \hbox{\unhbox\@curline + \hskip\@tempdima}
\@startfield
END

\@settab ==
BEGIN
  \@stopfield
  \@addfield
  if \@curtab < \@maxtab
    then \@curtab :=G \@curtab+1
    else error message: "Too many tabs"      fi
  if \@curtab > \@hightab
    then \@hightab :=L \@curtab      fi
  \dimen\@curtab :=L \dimen\@curtabmar + width of \box\@curline
  \@startfield
END

\@ltab ==
BEGIN
  \@ifatmargin
  then if \@curtabmar > \@firsttab
    then \@curtab :=G \@curtab - 1
        \@curtabmar :=G \@curtabmar - 1
    else error message "Too many untab"      fi
  else error message "Left tab in middle of line"
  fi
END

\@tabplus ==
BEGIN
  if \@nxttabmar < \@hightab
    then \@nxttabmar :=G \@nxttabmar+1
    else error message "Undefined tab"
  fi
END

\@tabminus ==
BEGIN
  if \@nxttabmar > \@firsttab
    then \@nxttabmar :=G \@nxttabmar-1
    else error message "Too many untab"
  fi
END

\@tabrj ==
BEGIN \@stopfield
  \@addfield
  @rjfield :=G T

```

```

        \@startfield
    END

\@tablab ==
    BEGIN \@stopfield
        \box\@curline G:= \hbox{\box\@curline %% 'G' added 17 Jun 86
                                \hskip - width of \box\@curfield
                                \hskip -\tabbingsep
                                \box\@curfield
                                \hskip \tabbingsep }

        \@startfield
    END

\pushtabs ==
    BEGIN
        \@stopfield
        \@tabpush :=G \@tabpush + 1
        \begingroup
        \@contfield
    END

\poptabs ==
    BEGIN
        \@stopfield
        if \@tabpush > 0
            then \endgroup
                \@tabpush :=G \@tabpush - 1
            else error message: "Too many \poptabs"
        fi
        \@contfield
    END

```

\a The accents \‘ , \’ , and \= that have been redefined inside a tabbing environment can be called by typing \a‘ , \a’ , and \a=. The macro \a is defined in ltoutenc.dtx.

The ‘2ekernel’ code ensures that a \usepackage{autotabg} is essentially ignored if a ‘full’ format is being used that has picture mode already in the format.

```

1 <2ekernel>\expandafter\let\csname ver@autotabg.sty\endcsname\fmtversion

\@firsttab
\@maxtab 2 <*2ekernel>
3 \newdimen\@gtempa
4 \chardef\@firsttab=\the\allocationnumber
5 \newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa
6 \newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa
7 \newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa
8 \newdimen\@gtempa
9 \chardef\@maxtab=\the\allocationnumber
10 \dimen\@firsttab=0pt

```

```

\@nxttabmar
\@curtabmar 11 \newcount\@nxttabmar
\@curtab 12 \newcount\@curtabmar
\@hightab 13 \newcount\@curtab
\@tabpush 14 \newcount\@hightab
15 \newcount\@tabpush

\@curline
\@curfield 16 \newbox\@curline
\@tabfbox 17 \newbox\@curfield
18 \newbox\@tabfbox

\if@rjfield
19 \newif\if@rjfield

\@startline It is, in some sense, an error if the current margin tab setting is higher than
the value of \@hightab (which is a local variable). That this is allowed is a
fundamental design flaw which is not going to be corrected now.
20 \gdef\@startline{%
21     \ifnum \@nxttabmar >\@hightab
22         \@badtab
23         \global\@nxttabmar \@hightab
24     \fi
25     \global\@curtabmar \@nxttabmar
26     \global\@curtab \@curtabmar
27     \global\setbox\@curline \hbox {}%
28     \@startfield
29     \strut}

\@stopline
30 \gdef\@stopline{%
31     \unskip
32     \@stopfield
33     \if@rjfield
34         \global\@rjfieldfalse
35         \@tempdima\@totalleftmargin
36         \advance\@tempdima\linewidth
37         \hb@xt@\@tempdima{%
38             \@itemfudge\hskip\dimen\@curtabmar
39             \box\@curline
40             \hfil
41             \box\@curfield}%
42     \else
43         \@addfield
44         \hbox{\@itemfudge\hskip\dimen\@curtabmar\box\@curline}%
45     \fi}

\@startfield
46 \gdef\@startfield{%
47     \global\setbox\@curfield\hbox\bgroup\color@begingroup}

\@stopfield
48 \gdef\@stopfield{%
49     \color@endgroup\egroup}

```

```

\@contfield
50 \gdef\@contfield{%
51   \global\setbox\@curfield\hbox\bgroup\color@begingroup
52   \unhbox\@curfield}

\@addfield
53 \gdef\@addfield{\global\setbox\@curline\hbox{\unhbox
54   \@curline\unhbox\@curfield}}

\@ifatmargin
55 \gdef\@ifatmargin{\ifdim \wd\@curline =\z@}

\@tabcr
56 \gdef\@tabcr{\@stopline \@ifstar{\penalty \M \@xtabcr}\@xtabcr}

\@xtabcr
57 \gdef\@xtabcr{\@ifnextchar[\@itabcr{\@startline\ignorespaces}}

\@itabcr
58 \gdef\@itabcr[#1]{\vskip #1\@startline\ignorespaces}
59 \gdef\kill{\@stopfield\@startline\ignorespaces}

\tabbing We use \relax to prevent \item from scanning too far.
60 \gdef\tabbing{\lineskip \z@skip\let\>\@rtab\let\<\@ltab\let\=\@settab
61   \let\+\@tabplus\let\-\@tabminus\let\'\@tabrj\let\'\@tablab
62   \let\=\@tabcr
63   \@hightab\@firsttab
64   \global\@nxttabmar\@firsttab
65   \dimen\@firsttab\@totalleftmargin
66   \global\@tabpush\z@ \global\@rjfieldfalse
67   \trivlist \item\relax
68   \if@minipage\else\vskip\parskip\fi

69   \setbox\@tabfbox\hbox{%
70     \rlap{\hskip\@totalleftmargin\indent\the\everypar}}%
71   \def\@itemfudge{\box\@tabfbox}%
72   \@startline\ignorespaces}

\endtabbing
73 \gdef\endtabbing{%
74   \@stopline\ifnum\@tabpush >\z@ \@badpoptabs \fi\endtrivlist}

\@rtab Omitted \global added to \@rtab 17 Jun 86
75 \gdef\@rtab{\@stopfield\@addfield\ifnum \@curtab<\@hightab
76   \global\advance\@curtab \@one \else\@badtab\fi
77   \@tempdima\dimen\@curtab
78   \advance\@tempdima -\dimen\@curtabmar
79   \advance\@tempdima -\wd\@curline
80   \global\setbox\@curline\hbox{\unhbox\@curline\hskip\@tempdima}%
81   \@startfield\ignorespaces}

```

```

\@settab
82 \gdef\@settab{\@stopfield\@addfield
83   \ifnum \@curtab <\@maxtab
84     \ifnum\@curtab =\@hightab
85       \advance\@hightab \@ne
86     \fi
87     \global\advance\@curtab \@ne
88   \else
89     \latexerror{Tab overflow}\@ehd
90   \fi
91   \dimen\@curtab \dimen\@curtabmar
92   \advance\dimen\@curtab \wd\@curline
93   \@startfield
94   \ignorespaces}

\@ltab
95 \gdef\@ltab{\@ifatmargin\ifnum\@curtabmar >\@firsttab
96   \global\advance\@curtab \m@ne \global\advance\@curtabmar\m@ne\else
97   \@badtab\fi\else
98   \latexerror{string<\space in mid line}\@ehd\fi\ignorespaces}

\@tabplus
99 \gdef\@tabplus{%
100   \ifnum\@nxttabmar<\@hightab
101     \global\advance\@nxttabmar\@ne
102   \else
103     \@badtab
104   \fi
105   \ignorespaces}

\@tabminus
106 \gdef\@tabminus{%
107   \ifnum\@nxttabmar>\@firsttab
108     \global\advance\@nxttabmar\m@ne
109   \else
110     \@badtab
111   \fi
112   \ignorespaces}

\@tabrj
113 \gdef\@tabrj{%
114   \@stopfield\@addfield\global\@rjfieldtrue\@startfield\ignorespaces}

\@tablab \setbox\@curline made \global in \@tablab. 17 Jun 86
115 \gdef\@tablab{%
116   \@stopfield
117   \global\setbox\@curline\hbox{%
118     \box\@curline
119     \hskip-\wd\@curfield \hskip-\tabbingsep
120     \box\@curfield
121     \hskip\tabbingsep}%
122   \@startfield
123   \ignorespaces}

```

`\pushtabs`

```
124 \gdef\pushtabs{%
125   \@stopfield\@addfield\global\advance\@tabpush \@ne \begingroup
126     \@contfield}
```

`\poptabs` It is, in some sense, an error if, after the endgroup, the current tab setting is higher than the new value of `\@hightab` (which is a local variable). That this is allowed is a fundamental design flaw which is not going to be corrected now.

```
127 \gdef\poptabs{\@stopfield\@addfield
128   \ifnum \@tabpush >\z@
129     \endgroup
130     \global\advance\@tabpush \m@ne
131     \ifnum \@curtab >\@hightab
132       \global \@curtab \@hightab
133       \@badtab
134     \fi
135   \else
136     \@badpoptabs
137   \fi
138   \@contfield}
```

`\tabbingsep`

```
139 \newdimen\tabbingsep
```

## 56.2 array and tabular environments

ARRAY PARAMETERS:

`\arraycolsep`  
: half the width separating columns in an array environment

`\tabcolsep`  
: half the width separating columns in a tabular environment

`\arrayrulewidth`  
: width of rules

`\doublerulesep`  
: space between adjacent rules in array or tabular

`\arraystretch`  
: line spacing in array and tabular environments is done by placing a strut in every row of height and depth `\arraystretch` times the height and depth of the strut produced by an ordinary `\strut` command.

PREAMBLE:

The PREAMBLE argument of an array or tabular environment can contain the following:

`l,r,c` : indicate where entry is to be placed.  
`|` : for vertical rule  
`@{EXP}` : inserts the text EXP in every column.  
`\arraycolsep` or `\tabcolsep` spacing is suppressed.  
`*{N}{PRE}` : equivalent to writing N copies of PRE in the preamble.  
PRE may contain `*{N'}{EXP'}` expressions.  
`p{LEN}` : makes entry in parbox of width LEN.

SPECIAL ARRAY COMMANDS:

`\multicolumn{N}{FORMAT}{ITEM}` : replaces the next N column items by ITEM, formatted according to FORMAT.  
FORMAT should contain at most one l,r or c.  
If it contains none, then ITEM is ignored.

`\vline` : draws a vertical line the height of the current row. May appear in an array element entry.  
`\hline` : draws a horizontal line between rows. Must appear either before the first entry (to appear above the first row) or right after a `\\` command. If followed by another `\hline`, then adds a `\vskip` of `\doublerulesep`.

`\cline[i-j]` : draws horizontal lines between rows covering columns i through j, inclusive. Multiple commands may follow one another to provide lines covering several disjoint columns  
`\extracolsep{WIDTH}` : for use inside an @ in the preamble. Causes a WIDTH space to be added between columns for the rest of the columns. This is in addition to the ordinary intercolumn space.

```
\array ==
  BEGIN
    \@acol    == \@arrayacol
    \@classz  == \@arrayclassz
    \@classiv == \@arrayclassiv
    \\        == \@arraycr
    \@halignto == NULL
    \@tabarray
  END

\endarray{NAME} == BEGIN \crrc }} END
```

```
\tabular ==
  BEGIN
    \@halignto == NULL
    \@tabular
  END

\tabular*{WIDTH} ==
  BEGIN
    \@halignto == to WIDTH
    \@tabular
  END

\@tabular ==
  BEGIN
    \leavevmode
```

```

\hbox { $
  \@acol    == \@tabacol
  \@classz  == \@tabclassz
  \@classiv == \@tabclassiv
  \\\       == \@tabularcr
  \@tabarray
END

\endtabular == BEGIN \crrc}} $} END

\@tabarray == if next char = [ then \@array else \@array[c] fi

\@array[POS]{PREAMBLE} ==
BEGIN
  define \@arstrutbox to make \@arstrut produce strut of height
    and depth \arraystretch times the height and
    depth of a normal strut.
  \@mkpream{PREAMBLE}
  \@preamble == \halign \@halignto {\tabskip=0pt\@arstrut
                                eval{\@preamble}\tabskip = 0pt\cr %%}
  \@startpbox == \@@startpbox
  \@endpbox == \@@endpbox
  if POS = t then \vtop
    else if POS = b then \vbox
      else \vcenter
    fi
  fi
  {
    \par          ==L {} % changed 92/09/18
    \@sharp       == #
    \protect      == \relax
    \lineskip     :=L 0pt
    \baselineskip :=L 0pt
    \@preamble
  }
END

\@arraycr ==
BEGIN
  $                %% Prevents extra space at end of row's last entry.
  if next char = [
    then \@argarraycr
    else $ \cr      %% Needed to balance $
  fi
END

\@argarraycr[LENGTH] ==
BEGIN
  $                %% Needed to balance $ of \@arraycr
  if LENGTH > 0
    then \@tempdima := depth of \@arstrutbox + LENGTH
      \vrule height 0pt width 0pt depth \@tempdima
    fi
  fi
END

```

```

\cr
else \cr \noalign{\vskip LENGTH}
END

\@tabularcr and \@argtabularcr same as \@arraycr and
\@argarraycr
except without the extra $'s.

\extracolsep
140 \def\extracolsep#1{\tabskip #1\relax}

\array
141 \def\array{\let\@acol\@arrayacol \let\@classz\@arrayclassz
142 \let\@classiv\@arrayclassiv
143 \let\\\@arraycr\let\@halignto\@empty\@tabarray}

\endarray
\endtabular 144 \def\endarray{\crrcr\egroup\egroup}
\endtabular* 145 \def\endtabular{\crrcr\egroup\egroup $\egroup}
146 \expandafter \let \csname endtabular*\endcsname = \endtabular

\tabular
147 \def\tabular{\let\@halignto\@empty\@tabular}

\tabular* Note that the change to use \setlength slightly alters the timing of the expansion
and use of the length in #1 but this is very unlikely to have any practical effect.
148 \@namedef{tabular*}#1{%
149 \setlength\dimen@{#1}%
150 \edef\@halignto{to\the\dimen@}\@tabular}

\@tabular
151 \def\@tabular{\leavevmode \hbox \bgroup $\let\@acol\@tabacol
152 \let\@classz\@tabclassz
153 \let\@classiv\@tabclassiv \let\\\@tabularcr\@tabarray}

\@tabarray RmS 91/11/04 added \m@th.
154 \def\@tabarray{\m@th\@ifnextchar[\@array{\@array[c]}}

RmS 1993/11/03 changed \halign to \ialign and removed superfluous
\tabskip assignment

\@array
155 \def\@array[#1]#2{%
156 \if #1t\top \else \if#1b\bottom \else \vcenter \fi\fi
157 \bgroup

```

This next bit of code sets up the strut and then builds the halign and its preamble according to the specification in the second argument.

This code has been moved inside the box. A side effect of this has been to expose what was a buglet in the previous version: since the `\@arstrut` below is expanded and contains an `\ifmmode` then it could produce an unnecessary extra box in every row, thus wasting ‘lots of’ main memory.

```

158 \setbox\@arstrutbox\hbox{%
159   \vrule \@height\arraystretch\ht\strutbox
160         \@depth\arraystretch \dp\strutbox
161         \@width\z@}%
162 \mkpream{#2}%
163 \edef\@preamble{%
164   \ialign \noexpand\@halignto
165     \bgroup \@arstrut \@preamble \tabskip\z@skip \cr}%

```

That is the end of setting up the preamble; now we reset things before executing the `\halign` built-up in `\@preamble`. The restorations could be done by introducing an extra group, thus saving tokens.

```

166 \let\@startpbox\@startpbox \let\@endpbox\@endpbox
167 \let\@tabularnewline\@%
168   \let\@par\@empty
169   \let\@sharp##%
170   \set@typeset@protect
171   \lineskip\z@skip\baselineskip\z@skip

```

If the parsing of the preamble goes wrong there may be some characters left which  $\TeX$  then tries to typeset, i.e., we would be in horizontal mode. That would produce an endless loop because the `\halign` expects vertical mode thus issues a `\par` but that is a no-op at this point. So we better test this case issue some error message and make a crude recovery by ending that horizontal mode with force. A better fix would be to ensure that we never pick up more than a single character token (not done).

```

172   \ifhmode \@preamerr\z@ \@@par\fi
173   \@preamble}

```

`\@arraycr` Array version of `\@`.

```

174 \def\@arraycr{%
175   ${\ifnum0=}\fi\@ifstar\@xarraycr\@arraycr}

```

`\@arraycr`

```

176 \def\@xarraycr{\@ifnextchar[\@argarraycr{\ifnum0={\fi}${}\cr}}

```

`\@argarraycr`

```

177 \def\@argarraycr[#1]{%
178   \ifnum0={\fi}${}\ifdim #1>\z@ \@xargarraycr{#1}\else
179     \@yargarraycr{#1}\fi}

```

`\@tabularnewline` Tabular version of `\@`.

```

180 \let\@tabularnewline\relax

```

`\@tabularcr`

```

181 \def\@tabularcr{%
182   {\ifnum0=}\fi\@ifstar\@xtabularcr\@xtabularcr}

```

`\@xtabularcr`

```

183 \def\@xtabularcr{\@ifnextchar[\@argtabularcr{\ifnum0={\fi}\cr}}

```

`\@argtabularcr`

```

184 \def\@argtabularcr[#1]{%

```

```

185 \ifnum0='{ \fi}%
186 \ifdim #1>\z@
187 \unskip\@xargarraycr{#1}%
188 \else
189 \@yargarraycr{#1}%
190 \fi}

\@xargarraycr
191 \def\@xargarraycr#1{\@tempdima #1\advance\@tempdima \dp \@arstrutbox
192 \vrule \@height\z@ \@depth\@tempdima \@width\z@ \cr}

\@yargarraycr
193 \def\@yargarraycr#1{\cr\noalign{\vskip #1}}

\multicolumn \multicolumn{NUMBER}{FORMAT}{ITEM} ==
BEGIN
\multispan{NUMBER}
\begingroup
\@addamp == null
\@mkpream{FORMAT}
\@sharp == ITEM
\protect == \relax
\@startpbox == \@startpbox
\@endpbox == \@endpbox
\@arstrut
\@preamble
\endgroup
END

```

The command `\def\@addamp{}` was removed from `\multicolumn` on 6 Dec 86 because it caused embedded array environments not to work. I think that it was included originally to prevent an error message if the 2nd argument to the `\multicolumn` command had two column specifiers.

8 Feb 89 — `\hbox{}` added after `\@preamble` to correct bug that occurred if `\multicolumn` preceded `\\[D]` with  $D > 0$ , caused by `\\[]` command doing an `\unskip`, which removed `\tabcolsep` glue inserted by `\multicolumn`.

This has been made long so that, for example, a p-column can contain multiple paragraphs; maybe the arguments of @-expressions should also be able to contain multiple paragraphs.

```

194 \long\def\multicolumn#1#2#3{\multispan{#1}\begingroup
195 \mkpream{#2}%
196 \def\@sharp{#3}\set@typeset@protect
197 \let\@startpbox\@startpbox\let\@endpbox\@endpbox
198 \@arstrut \@preamble\hbox{\endgroup\ignorespaces}

```

Codes for classes and character numbers of array, tabular and multicolumn arguments.

Character	Class	Number
c	0	0
l	0	1

r	0	2
	1	-
@	2	-
p	3	-
{@-exp}	4	-
{p-arg}	5	-

`\@testpach \foo` : expands `\foo`, which should be an array parameter token, and sets `\@chclass` and `\@chnum` to its class and number. Uses `\@lastchclass` to distinguish 4 and 5

Preamble error codes

0: 'illegal character'  
 1: 'Missing @-exp'  
 2: 'Missing p-arg'

```
\@addamp ==
  BEGIN if \@firstamp = true then \@firstamp := false
        else &                                fi
  END
```

```
\@mkpream TOKENLIST ==
  BEGIN
    \@firstamp      := T
    \@lastchclass   := 6
    \@preamble      == null
    \@sharp         == \relax
    \@protect       == BEGIN \noexpand\protect\noexpand END
    \@startpbox     == \relax
    \@endpbox       == \relax
    \@expast{TOKENLIST}
    for \@nextchar := expand(\reserved@a)
      do \@testpach{\@nextchar}
        case of \@chclass
          0 -> \@classz
          1 -> \@classi
          ...
          5 -> \@classv
        end case
        \@lastchclass := \@chclass
      od
    case of \@lastchclass
      0 -> \hskip \arraycolsep                % lrc
      1 ->                                     % |
      2 -> \@preamerr1 % 'Missing @-exp'      % @
      3 -> \@preamerr2 % 'Missing p-arg'      % p
      4 ->                                     % @-exp
      5 -> \hskip \arraycolsep                % p-exp
    end case
```

```

END

\@arrayclassz ==
BEGIN
  \@preamble := \@preamble *
  case of \@lastchclass
    0 -> \hskip \arraycolsep \@addamp \hskip
\arraycolsep
    1 -> \@addamp \hskip \arraycolsep
    2 -> % impossible
    3 -> % impossible
    4 -> \@addamp
    5 -> \hskip \arraycolsep \@addamp \hskip
\arraycolsep
    6 -> \@addamp \hskip \arraycolsep
  end case
  * case of \@chnum
    0 -> \hfil$\relax\@sharp$\hfil
    1 -> $\relax\@sharp$\hfil
    2 -> \hfil$\relax\@sharp$
  end case
END

\@tabclassz == similar to \@arrayclassz

\@classi ==
BEGIN
  \@preamble := \@preamble *
  case of \@lastchclass
    0 -> \hskip \arraycolsep \@arrayrule
    1 -> \hskip \doublerulesep \@arrayrule
    2 -> % impossible
    3 -> % impossible
    4 -> \@arrayrule
    5 -> \hskip \arraycolsep \@arrayrule
    6 -> \@arrayrule
  end case
END

\@classii ==
BEGIN
  \@preamble := \@preamble *
  case of \@lastchclass
    0 ->
    1 -> \hskip .5\arrayrulewidth
    2 -> % impossible
    else ->
  end case
END

```

```

\@classiii ==
BEGIN
  \@preamble := \@preamble *
    case of \@lastchclass
      0 -> \hskip \arraycolsep \@addamp \hskip
\arraycolsep
      1 -> \@addamp \hskip \arraycolsep
      2 -> % impossible
      3 -> % impossible
      4 -> \@addamp
      5 -> \hskip \arraycolsep \@addamp \hskip
\arraycolsep
      6 -> \@addamp \hskip \arraycolsep
    end case
END

\@arrayclassiv ==
BEGIN \@preamble := \@preamble * $ \@nextchar$ END

\@tabclassiv == same as \@arrayclassv except without the $ ... $

\@classv ==
BEGIN
  \@preamble :=
    \@preamble * \@startpbox{\@nextchar}\ignorespaces\@sharp
    \@endpbox
END

\@expast{S}:
Sets \reserved@a := S with all instances of *{N}{STRING}
replaced by N copies of STRING, where N > 0. An *
appearing inside braces is ignored, but *-expressions
inside STRING are expanded, so nested *-expressions are
handled properly.

\@expast{S} == BEGIN \@expast S *0x\@@ END

\@expast S1 *{N}{S2} S3 \@@ ==
BEGIN
  \reserved@a := S1
  \@tempcnta := N
  if \@tempcnta > 0
    then while \@tempcnta > 0 do \reserved@a := \reserved@a S2
      \@tempcnta := \@tempcnta - 1 od
    \reserved@b == \@expast
    else \reserved@b == \@exnoop
  fi
  \expandafter \reserved@b \reserved@a S3 \@@
END

```

```

\@exnoop
199 \def\@exnoop #1\@{ }

\@expast
200 \def\@expast#1{\@expast #1*0x\@{ }

\@expast
201 \def\@expast#1*#2#3#4\@{ %
202   \edef\reserved@a{#1}%
203   \@tempcnta#2\relax
204   \ifnum\@tempcnta>\z@
205     \@whilenum\@tempcnta>\z@\do
206       {\edef\reserved@a{\reserved@a#3}\advance\@tempcnta \m@ne}%
207     \let\reserved@b\@expast
208   \else
209     \let\reserved@b\@exnoop
210   \fi
211   \expandafter\reserved@b\reserved@a #4\@{ }

\if@firstamp
\@addamp 212 \newif\if@firstamp
213 \def\@addamp{%
214   \if@firstamp
215     \@firstampfalse
216   \else
217     \edef\@preamble{\@preamble &}%
218   \fi}

\@arrayacol
\@tabacol 219 \def\@arrayacol{\edef\@preamble{\@preamble \hskip \arraycolsep}}
\@ampacol 220 \def\@tabacol{\edef\@preamble{\@preamble \hskip \tabcolsep}}
\@acolampacol 221 \def\@ampacol{\@addamp \@acol}
222 \def\@acolampacol{\@acol\@addamp\@acol}

\@mkpream
223 \def\@mkpream#1{\@firstamptrue\@lastchclass6
224   \let\@preamble\@empty
225   \let\protect\@unexpandable@protect
226   \let\@sharp\relax
227   \let\@startpbox\relax\let\@endpbox\relax
228   \@expast{#1}%
229   \expandafter\@tfor \expandafter
230     \@nextchar \expandafter:\expandafter=\reserved@a\do
231     {\@testpach\@nextchar
232       \ifcase \@chclass \@classz \or \@classi \or \@classii \or \@classiii
233       \or \@classiv \or \@classv \fi\@lastchclass\@chclass}%
234   \ifcase \@lastchclass \@acol
235     \or \or \@preamerr \@ne\or \@preamerr \tw@\or \or \@acol \fi}

\@arrayclassz
236 \def\@arrayclassz{\ifcase \@lastchclass \@acolampacol \or \@ampacol \or
237   \or \or \@addamp \or
238   \@acolampacol \or \@firstampfalse \@acol \fi}

```

```

239 \edef\@preamble{\@preamble
240 \ifcase \@chnum
241 \hfil$\relax\@sharp$\hfil \or $\relax\@sharp$\hfil
242 \or \hfil$\relax\@sharp$\fi}}

\@tabclassz RmS 91/08/14 inserted extra braces around entry for NFSS
243 \def\@tabclassz{%
244 \ifcase\@lastchclass
245 \@acolampacol
246 \or
247 \@ampacol
248 \or
249 \or
250 \or
251 \@addamp
252 \or
253 \@acolampacol
254 \or
255 \@firstampfalse\@acol
256 \fi
257 \edef\@preamble{%
258 \@preamble{%
259 \ifcase\@chnum
260 \hfil\ignorespaces\@sharp\unskip\hfil
261 \or
262 \hspace{1em}\ignorespaces\@sharp\unskip\hfil
263 \or
264 \hfil\hspace{1em}\ignorespaces\@sharp\unskip
265 \fi}}}

\@classi
266 \def\@classi{%
267 \ifcase\@lastchclass
268 \@acol\@arrayrule
269 \or
270 \@addtopreamble{\hspace{\doublerulesep}\@arrayrule
271 \or
272 \or
273 \or
274 \@arrayrule
275 \or
276 \@acol\@arrayrule
277 \or
278 \@arrayrule
279 \fi}

\@classii
280 \def\@classii{%
281 \ifcase\@lastchclass
282 \or
283 \@addtopreamble{\hspace{.5\arrayrulewidth}%
284 \fi}

```

```

\@classiii
285 \def\@classiii{\ifcase \@lastchclass \@acolampacol \or
286   \@addamp\@acol \or
287   \or \or \@addamp \or
288   \@acolampacol \or \@ampacol \fi}

\@tabclassiv
289 \def\@tabclassiv{\@addtopreamble\@nextchar}

\@arrayclassiv
290 \def\@arrayclassiv{\@addtopreamble{$\@nextchar$}}

\@classv
291 \def\@classv{\@addtopreamble{\@startpbox{\@nextchar}\ignorespaces
292 \@sharp\@endpbox}}

\@addtopreamble
293 \def\@addtopreamble#1{\edef\@preamble{\@preamble #1}}

\@chclass
\@lastchclass 294 \newcount\@chclass
\@chnum 295 \newcount\@lastchclass
296 \newcount\@chnum

\arraycolsep
\@tabcolsep 297 \newdimen\arraycolsep
\arrayrulewidth 298 \newdimen\@tabcolsep
\@doublerulesep 299 \newdimen\arrayrulewidth
300 \newdimen\@doublerulesep

\arraystretch
301 \def\arraystretch{1} % Default value.

\@arstrutbox
\@arstrut 302 \newbox\@arstrutbox
303 \def\@arstrut{%
304   \relax\ifmmode\copy\@arstrutbox\else\unhcopy\@arstrutbox\fi}

\@arrayrule
305 \def\@arrayrule{\@addtopreamble{\hskip -.5\arrayrulewidth
306   \vrule \@width \arrayrulewidth\hskip -.5\arrayrulewidth}}

\@testpatch
307 \def\@testpach#1{\@chclass \ifnum \@lastchclass=\tw@ 4 \else
308   \ifnum \@lastchclass=3 5 \else
309   \z@ \if #1c\@chnum \z@ \else
310     \if #1l\@chnum \@ne \else
311     \if #1r\@chnum \tw@ \else
312     \@chclass \if #1|\@ne \else
313     \if #1@\tw@ \else
314     \if #1p3 \else \z@ \@preamerr 0\fi
315   \fi \fi \fi \fi \fi \fi
316 \fi}

```

```

\hline
317 \def\hline{%
318   \noalign{\ifnum0='}\fi\hrule \@height \arrayrulewidth \futurelet
319   \reserved@a\@xhline}

\@xhline
320 \def\@xhline{\ifx\reserved@a\hline
321               \vskip\doublerulesep
Measure from the middle of the rules.
322               \vskip-\arrayrulewidth
323               \fi
324               \ifnum0='{ \fi}}

\vline
325 \def\vline{\vrule \@width \arrayrulewidth}

\cline The old LATEX 2.09 implementation of \cline used up quite a lot of memory and
\@cline two precious count registers. This new (1995/09/14) implementation does not use
any count registers. It is coded in a way that depends heavily on the definition of
\multispan so that command has been moved here from the file ltplain.dtx.
These counters are no longer declared.

\newcount\@cla
\newcount\@clb

326 \def\cline#1{\@cline#1\@nil}

327 \def\@cline#1-#2\@nil{%
328   \omit

Use the counter from \multispan.
329   \@multicnt#1%
330   \advance\@multispan\@m@ne
331   \ifnum\@multicnt=\@ne\@firstofone{&\omit}\fi
332   \@multicnt#2%
333   \advance\@multicnt-#1%
334   \advance\@multispan\@ne

The original had \unskip at this point, but how could a skip get here ???
335   \leaders\hrule\@height\arrayrulewidth\hfill
336   \cr

This is back spacing is fairly horrible, but it is what happened in the old version...
An alternative would be to make \cline look ahead for a following \cline as does
\hline. This would alter the spacing in existing documents so keep the old version
in the kernel. Perhaps a package should do this differently.
337   \noalign{\vskip-\arrayrulewidth}}

\mscount The \mscount counter is no longer declared, saving a csname and a register. It is
declared in compatibility mode.

\multispan Modify \multispan slightly from its plain TEX definition to allow more efficient
\@multispan code sharing with \multicolumn. Also share a count register with \multipt.
\sp@n 338 \def\multispan{\omit\@multispan}

```

```

339 \def\@multispan#1{%
340   \@multicnt#1\relax
341   \loop\ifnum\@multicnt>\@ne \sp@n\repeat}
342 \def\sp@n{\span\omit\advance\@multicnt\m@ne}

\@startpbox Helper macros for ‘p’ columns.
\@endpbox   \@startpbox{\width} text \egroup is essentially \parbox{\width}{\text}
            \@endpbox is essentially \unskip \strut \par \egroup\hfil (Changed 14
            Jan 89) (changed again 1994/05/13)
343 \def\@startpbox#1{\vtop\bgroup \setlength\hsize{#1}\@arrayparboxrestore}
344 \def\@endpbox{\@finalstrut\@arstrutbox\par\egroup\hfil}

            14 Jan 89: Def of \@endpbox changed from
            \def\@endpbox{\par\vskip\dp\@arstrutbox\egroup\hfil}
            so vertical spacing works out right if the last line of a ‘p’ entry has a descender.

\@@startpbox
\@@endpbox 345 \let\@@startpbox=\@startpbox
346 \let\@@endpbox=\@endpbox

347 \endkernel

```

# File D

## ltpictur.dtx

### 57 Picture Mode

Picture mode commands. In addition to the commands available in L<sup>A</sup>T<sub>E</sub>X2.09, This section adds the new `\qbezier` command for drawing curves.

`\qbezier` `[\langle N \rangle](\langle AX,AY \rangle)(\langle BX,BY \rangle)(\langle CX,CY \rangle)` plots a quadratic Bezier curve from  $(\langle AX,AY \rangle)$  to  $(\langle CX,CY \rangle)$ , with  $(\langle BX,BY \rangle)$  as the third Bezier point, using  $N + 1$  points equally spaced parametrically. If  $N = 0$  (the default value), then a sufficient number of points are used to draw a connected curve—except that at most `\qbeziermax` + 1 points are drawn. A “point” is a square of side `\@wholewidth`.

`\bezier` In addition, to be compatible with the old `bezier` package, a variant of this command, `\bezier`, is defined, in which the first argument is not optional.

<code>\unitlength</code>	= value of dimension argument
<code>\@wholewidth</code>	= current line width
<code>\@halfwidth</code>	= half of current line width
<code>\@linefnt</code>	= font for drawing lines
<code>\@circlefnt</code>	= font for drawing circles

`\linethickness{DIM}` : Sets the width of horizontal and vertical lines in a picture to DIM. Does not change width of slanted lines or circles. Width of all lines reset by `\thinlines` and `\thicklines`

```

\picture(XSIZE,YSIZE)(XORG,YORG)
  BEGIN
    \@picht :=L YSIZE * \unitlength
    box \@picbox :=
      \hb@xt@ XSIZE * \unitlength
      {\hskip -XORG * \unitlength
       \lower YORG * \unitlength
       \hbox{
         \ignorespaces      %% added 13 June 89
       }
      }
  END

\endpicture ==
  BEGIN
    } \hss }
    height of \@picbox := \@picht
    depth of \@picbox := 0
    \mbox{\box\@picbox}    %% change 26 Aug 91
  END

\put(X, Y){OBJ} ==
  BEGIN

```

```

\@killglue
\raise Y * \unitlength \hb@xt@ 0pt { \hskip X * \unitlength
                                OBJ \hss
}
\ignorespaces
END

```

```

\multiput(X,Y)(DELX,DELY){N}{OBJ} ==
BEGIN
\@killglue
\@multicnt := N
\@xdim := X * \unitlength
\@ydim := Y * \unitlength
while \@multicnt > 0
do \raise \@ydim \hb@xt@ 0pt { \hskip \@xdim
                                OBJ \hss }
\@multicnt := \@multicnt - 1
\@xdim := \@xdim + DELX * \unitlength
\@ydim := \@ydim + DELY * \unitlength
od
\ignorespaces
END

```

`\shortstack[POS]{TEXT}` : Makes a `\vbox` containing TEXT stacked as a one-column array, positioned l, r or c as indicated by POS.

The ‘2ekernel’ code ensures that a `\usepackage{autopict}` is essentially ignored if a ‘full’ format is being used that has picture mode already in the format.

```
1 (2ekernel)\expandafter\let\csname ver@autopict.sty\endcsname\fmtversion
```

```

\@wholewidth
\@halfwidth 2 (*2ekernel)
              3 \newdimen\@wholewidth
              4 \newdimen\@halfwidth

\unitlength
              5 \newdimen\unitlength \unitlength =1pt

```

```

\@picbox
\@picht 6 \newbox\@picbox
         7 \newdimen\@picht

```

`\picture` #1 should be white space.

```

\pictur@ #1 should be a ( (eating any white space before the bracket),
          8 \long\gdef\picture#1{\pictur@#1}
          9 \gdef\pictur@(#1){%
          10 \@ifnextchar({\@picture(#1)}{\@picture(#1)(0,0)}}

```

```

\@picture
11 \gdef\@picture(#1,#2)(#3,#4){%
12   \@picht#2\unitlength
13   \setbox\@picbox\hb@xt@#1\unitlength\bgroup
14     \hskip -#3\unitlength
15     \lower #4\unitlength\hbox\bgroup
16     \ignorespaces}

\endpicture
17 \gdef\endpicture{%
18   \egroup\hss\egroup
19   \ht\@picbox\@picht\dp\@picbox\z@
20   \mbox{\box\@picbox}}

In the definitions of \put and \multiput, \hskip was replaced by \kern just
in case arg #3 = "plus". (Bug detected by Don Knuth. changed 20 Jul 87).

21 \long\gdef\put(#1,#2)#3{%
22   \@killglue\raise#2\unitlength
23   \hb@xt@#3\z@{\kern#1\unitlength #3\hss}%
24   \ignorespaces}

\multiput #3 had better be a (.
25 \gdef\multiput(#1,#2)#3{%
26   \@xdim #1\unitlength
27   \@ydim #2\unitlength
28   \@multiput{ }

\multiput
29 \long\gdef\@multiput(#1,#2)#3#4{%
30   \@killglue\@multicnt #3\relax
31   \@whilenum \@multicnt >\z@\do
32     {\raise\@ydim\hb@xt@#4\z@{\kern\@xdim #4\hss}%
33     \advance\@multicnt\m@ne
34     \advance\@xdim#1\unitlength\advance\@ydim#2\unitlength}%
35   \ignorespaces}

\@killglue
36 \gdef\@killglue{\unskip\@whiledim \lastskip >\z@\do{\unskip}}

\thinlines
\thicklines
37 \gdef\thinlines{\let\@linefnt\tenln \let\@circlefnt\tencirc
38   \@wholewidth\fontdimen8\tenln \@halfwidth .5\@wholewidth}
39 \gdef\thicklines{\let\@linefnt\tenlnw \let\@circlefnt\tencircw
40   \@wholewidth\fontdimen8\tenlnw \@halfwidth .5\@wholewidth}

\linethickness
41 \gdef\linethickness#1{\@wholewidth #1\relax \@halfwidth .5\@wholewidth}

\ishortstack
42 \gdef\shortstack{\@ifnextchar[\@shortstack{\@shortstack[c]}}

```

```

\@ishortstack
43 \gdef\@shortstack[#1]{%
44   \leavevmode
45   \vbox\bgroup
46     \baselineskip-\p@\lineskip 3\p@
47     \let\mb@l\hss\let\mb@r\hss
48     \expandafter\let\csname mb@#1\endcsname\relax
49     \let\\ \@stackcr
50     \@ishortstack}

\@ishortstack
51 \gdef\@ishortstack#1{\ialign{\mb@l {##}\unskip\mb@r\cr #1\cr}\egroup}

\@stackcr
\@ixstackcr
52 \gdef\@stackcr{\@ifstar\@ixstackcr\@ixstackcr}
53 \gdef\@ixstackcr{\@ifnextchar[\@istackcr\cr\ignorespaces]}

\@istackcr
54 \gdef\@istackcr[#1]{\cr\noalign{\vskip #1}\ignorespaces}

\line(X,Y){LEN} ==
BEGIN
  \@xarg    := X
  \@yarg    := Y
  \@linelen := LEN * \unitlength
  if \@xarg = 0
    then \@vline
    else if \@yarg = 0
      then \@hline
      else \@sline
    if
  if
END

\@sline ==
BEGIN
  if \@xarg < 0
    then @negarg := T
      \@xarg := -\@xarg
      \@yyarg := -\@yarg
    else @negarg := F
      \@yyarg := \@yarg
  fi
  \@tempcnta := |\@yyarg|
  if \@tempcnta > 6
    then error: 'LATEX ERROR: Illegal \line or \vector argument.'
      \@tempcnta := 0
  fi
  \box\@linechar := \hbox{\@linefnt \@getlinechar(\@xarg,\@yyarg)}
}

```

```

if \@yarg > 0 then \@upordown = \raise
    \@clnht := 0
else \@upordown = \lower
    \@clnht := height of \box\@linechar
fi
\@clnwd := width of \box\@linechar
if @negarg
then \hskip - width of \box\@linechar
    \reserved@a == \hskip - 2* width of box \@linechar
else \reserved@a == \relax
fi
%% Put out integral number of line segments
while \@clnwd < \@linelen
do \upordown \@clnht \copy\@linechar
    \reserved@a
    \@clnht := \@clnht + ht of \box\@linechar
    \@clnwd := \@clnwd + width of \box\@linechar
od

%% Put out last segment
\@clnht := \@clnht - height of \box\@linechar
\@clnwd := \@clnwd - width of \box\@linechar
\@tempdima := \@linelen - \@clnwd
\@tempdimb := \@tempdima - width of \box\@linechar
if @negarg then \hskip -\@tempdimb
else \hskip \@tempdimb
fi
\@tempdima := 1000 * \@tempdima
\@tempcnta := \@tempdima / width of \box\@linechar
\@tempdima := (\@tempcnta * ht of \box\@linechar)/1000
\@clnht := \@clnht + \@tempdima
if \@linelen < width of box\@linechar
then \hskip width of box\@linechar
else \hbox{\@upordown \@clnht \copy\@linechar}
fi
END

\@hline ==
BEGIN
if \@xarg < 0 then \hskip -\@linelen \fi
\vrule height \@halfwidth depth \@halfwidth width \@linelen
if \@xarg < 0 then \hskip -\@linelen \fi
END

\@vline == if \@yarg < 0 \@downline else \@upline fi

\@getlinechar(X,Y) ==
BEGIN
\@tempcnta := 8*X - 9

```

```

        if Y > 0
        then \@tempcnta := \@tempcnta + Y
        else \@tempcnta := \@tempcnta - Y + 64
        fi
        \char\@tempcnta
    END

\vector(X,Y){LEN} ==
BEGIN
    \@xarg := X
    \@yarg := Y
    \@linelen := LEN * \unitlength
    if \@xarg = 0
    then \@vvector
    else if \@yarg = 0
    then \@hvector
    else \@svector
    fi
    if
    if
END

\@hvector ==
BEGIN
    \@hline
    {\@linefnt if \@xarg < 0 then \@getlarrow(1,0)
    else \@getrarrow(1,0)
    fi}
END

\@vvector == if \@yarg < 0 \@downvector else \@upvector fi

\@svector ==
BEGIN
    \@sline
    \@tempcnta := |\@yarg|
    if \@tempcnta < 5
    then \hskip - width of \box\@linechar
        \@upordown \@clnht \hbox
        {\@linefnt
        if @negarg then \@getlarrow(\@xarg,\@yyarg)
        else \@getrarrow(\@xarg,\@yyarg)
        fi }
    else error: 'LATEX ERROR: Illegal \line or \vector argument.'
    fi
END

\@getlarrow(X,Y) ==
BEGIN
    if Y = 0
    then \@tempcnta := '33

```

```

        else \@tempcnta := 16 * X - 9
          \@tempcntb := 2 * Y
          if \@tempcntb > 0
            then \@tempcnta := \@tempcnta + \@tempcntb
            else \@tempcnta := \@tempcnta - \@tempcntb + 64
          fi
        fi
      \char\@tempcnta
    END

\@getrarrow(X,Y) ==
BEGIN
  \@tempcntb := |Y|
  case of \@tempcntb
    0 : \@tempcnta := '55
    1 : if X < 3
        then \@tempcnta := 24*X - 6
        else if X = 3
            then \@tempcnta := 49
            else \@tempcnta := 58 fi
        fi
    2 : if X < 3
        then \@tempcnta := 24*X - 3
        else \@tempcnta := 51 % X must = 3
        fi
    3 : \@tempcnta := 16*X - 2
    4 : \@tempcnta := 16*X + 7
  endcase
  if Y < 0
    then \@tempcnta := \@tempcnta + 64
  fi
  \char\@tempcnta
END

\if@negarg
55 \newif\if@negarg

\line
56 \gdef\line(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
57 \@linelen #3\unitlength
58 \ifdim\@linelen<\z@\@badlinearg\else
59 \ifnum\@xarg =\z@ \@vline
60 \else \ifnum\@yarg =\z@ \@hline \else \@sline\fi
61 \fi
62 \fi}

\@sline
63 \gdef\@sline{%
64 \ifnum\@xarg<\z@ \@negargtrue \@xarg -\@xarg \@yyarg -\@yarg
65 \else \@negargfalse \@yyarg \@yarg \fi
66 \ifnum \@yyarg >\z@ \@tempcnta\@yyarg \else \@tempcnta -\@yyarg \fi

```

```

67 \ifnum \@tempcnta>6 \@badlinearg \@tempcnta \z@ \fi
68 \ifnum \@xarg>6 \@badlinearg \@xarg \ne \fi
69 \setbox \@linechar \hbox{\@linefnt\@getlinechar(\@xarg,\@yyarg)}%

```

If we have something like `\line(5,5){30}` the `\@linechar` will not contain a char and later on we will end in an infinite loop. So we check the width of the box and put in something as an emergency fix if necessary.

```

70 \ifdim \wd \@linechar=\z@
71   \setbox \@linechar \hbox{.}%
72   \@badlinearg
73 \fi
74 \ifnum \@yarg >\z@ \let \@upordown \raise \@clnht \z@
75   \else \let \@upordown \lower \@clnht \ht \@linechar \fi
76 \@clnwd \wd \@linechar
77 \if@negarg
78   \hskip -\wd \@linechar \def \reserved@a {\hskip -2\wd \@linechar}%
79 \else
80   \let \reserved@a \relax
81 \fi
82 \@whiledim \@clnwd <\@linelen \do
83   {\@upordown \@clnht \copy \@linechar
84     \reserved@a
85     \advance \@clnht \ht \@linechar
86     \advance \@clnwd \wd \@linechar}%
87 \advance \@clnht -\ht \@linechar
88 \advance \@clnwd -\wd \@linechar
89 \@tempdima \@linelen \advance \@tempdima -\@clnwd
90 \@tempdimb \@tempdima \advance \@tempdimb -\wd \@linechar
91 \if@negarg \hskip -\@tempdimb \else \hskip \@tempdimb \fi
92 \multiply \@tempdima \@m
93 \@tempcnta \@tempdima
94 \@tempdima \wd \@linechar \divide \@tempcnta \@tempdima
95 \@tempdima \ht \@linechar \multiply \@tempdima \@tempcnta
96 \divide \@tempdima \@m
97 \advance \@clnht \@tempdima
98 \ifdim \@linelen <\wd \@linechar
99   \hskip \wd \@linechar

```

Warn if line gets so short that it can't be printed. But don't warn if it is exactly zero since that was probably deliberate (e.g., to get a vector head only).

```

100 \ifdim \@linelen = \z@
101   \else
102     \@picture@warn
103   \fi
104 \else \@upordown \@clnht \copy \@linechar \fi}

```

`\@hline`

```

105 \gdef \@hline{\ifnum \@xarg <\z@ \hskip -\@linelen \fi
106 \vrule \@height \@halfwidth \@depth \@halfwidth \@width \@linelen
107 \ifnum \@xarg <\z@ \hskip -\@linelen \fi}

```

`\getlinechar`

```

108 \gdef \@getlinechar(#1,#2){\@tempcnta#1\relax \multiply \@tempcnta 8%
109   \advance \@tempcnta -9\ifnum #2>\z@ \advance \@tempcnta #2\relax \else

```

```

110 \advance\@tempcnta -#2\relax\advance\@tempcnta 64 \fi
111 \char\@tempcnta}

\vector
112 \gdef\vector(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
113 \@tempcnta \ifnum\@xarg<\z@ -\@xarg\else\@xarg\fi
114 \ifnum\@tempcnta<5\relax
115 \@linelen #3\unitlength
116 \ifdim\@linelen<\z@\@badlinearg\else
117 \ifnum\@xarg =\z@ \@vvector
118 \else \ifnum\@yarg =\z@ \@hvector \else \@svector\fi
119 \fi
120 \fi
121 \else\@badlinearg\fi}

\@hvector
122 \gdef\@hvector{\@hline\hb@xt@\z@\{\@linefnt
123 \ifnum \@xarg <\z@ \@getlarrow(1,0)\hss\else
124 \hss\@getrarrow(1,0)\fi}}

\@vvector
125 \gdef\@vvector{\ifnum \@yarg <\z@ \@downvector \else \@upvector \fi}

\@svector
126 \gdef\@svector{\@sline
127 \@tempcnta\@yarg \ifnum\@tempcnta <\z@ \@tempcnta -\@tempcnta\fi
128 \ifnum\@tempcnta <5%
129 \hskip -\wd\@linechar
130 \@upordown\@clnht \hbox{\@linefnt \if@negarg
131 \@getlarrow(\@xarg,\@yyarg)\else \@getrarrow(\@xarg,\@yyarg)\fi}%
132 \else\@badlinearg\fi}

\@getlarrow
133 \gdef\@getlarrow(#1,#2){\ifnum #2=\z@ \@tempcnta 27 % '33
134 \else
135 \@tempcnta #1\relax\multiply\@tempcnta \sixt@@n
136 \advance\@tempcnta -9 \@tempcntb #2\relax\multiply\@tempcntb \tw@
137 \ifnum \@tempcntb >\z@ \advance\@tempcnta \@tempcntb
138 \else\advance\@tempcnta -\@tempcntb\advance\@tempcnta 64
139 \fi\fi\char\@tempcnta}

\@getrarrow
140 \gdef\@getrarrow(#1,#2){\@tempcntb #2\relax
141 \ifnum\@tempcntb <\z@ \@tempcntb -\@tempcntb\relax\fi
142 \ifcase \@tempcntb\relax \@tempcnta 45 % '55
143 \or
144 \ifnum #1<\thr@@ \@tempcnta #1\relax\multiply\@tempcnta
145 24\advance\@tempcnta -6 \else \ifnum #1=\thr@@ \@tempcnta 49
146 \else\@tempcnta 58 \fi\fi\or
147 \ifnum #1<\thr@@ \@tempcnta=#1\relax\multiply\@tempcnta
148 24\advance\@tempcnta -\thr@@ \else \@tempcnta 51 \fi\or
149 \@tempcnta #1\relax\multiply\@tempcnta

```

```

150 \sixt@@n \advance\@tempcnta -\tw@ \else
151 \@tempcnta #1\relax\multiply\@tempcnta
152 \sixt@@n \advance\@tempcnta 7 \fi\ifnum #2<\z@ \advance\@tempcnta 64 \fi
153 \char\@tempcnta}

\@vline
154 \gdef\@vline{\ifnum \@yarg <\z@ \@downline \else \@upline\fi}

\@upline
155 \gdef\@upline{%
156   \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
157     \@height \@linelen \@depth \z@\hss}}

\@downline
158 \gdef\@downline{%
159   \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
160     \@height \z@ \@depth \@linelen \hss}}

\@upvector
161 \gdef\@upvector{\@upline\setbox\@tempboxa\hbox{\@linefnt\char 54}% '66
162   \raise \@linelen \hb@xt@\z@{\lower \ht\@tempboxa\box\@tempboxa\hss}}

\@downvector
163 \gdef\@downvector{\@downline\lower \@linelen
164   \hb@xt@\z@{\@linefnt\char 63 % '77
165     \hss}}

\dashbox{D}(X,Y) ==
  BEGIN
  leave vertical mode
  \hb@xt@ 0pt {
    \baselineskip := 0pt
    \lineskip := 0pt
    %% HORIZONTAL DASHES
    \@dashdim := X * \unitlength
    \@dashcnt := \@dashdim + 200 % to prevent roundoff error
    \@dashdim := D * \unitlength
    \@dashcnt := \@dashcnt / \@dashdim
    if \@dashcnt is odd
    then \@dashdim := 0pt
        \@dashcnt := (\@dashcnt + 1) / 2
    else \@dashdim := \@dashdim / 2
        \@dashcnt := \@dashcnt / 2 - 1
        \box\@dashbox := \hbox{\vrule height \@halfwidth
          depth \@halfwidth width \@dashdim}
        \put(0,0){\copy\@dashbox}
        \put(0,Y){\copy\@dashbox}
        \put(X,0){\hskip -\@dashdim\copy\@dashbox}
        \put(X,Y){\hskip -\@dashdim\box\@dashbox}
        \@dashdim := 3 * \@dashdim
    fi
  }

```

```

\box\@dashbox := \hbox{\vrule height \@halfwidth
                        depth \@halfwidth width D * \unitlength
                        \hskip D * \unitlength}

\@tempcnta := 0
\put(0,0){\hskip \@dashdim
          while \@tempcnta < \@dashcnt
            do \copy\@dashbox
              \@tempcnta := \@tempcnta + 1
            od
          }
\@tempcnta := 0
\put(0,Y){\hskip \@dashdim
          while \@tempcnta < \@dashcnt
            do \copy\@dashbox
              \@tempcnta := \@tempcnta + 1
            od
          }

%% vertical dashes
\@dashdim := Y * \unitlength
\@dashcnt := \@dashdim + 200 % to prevent roundoff error
\@dashdim := D * \unitlength
\@dashcnt := \@dashcnt / \@dashdim
if \@dashcnt is odd
  then \@dashdim := 0pt
       \@dashcnt := (\@dashcnt + 1) / 2
  else \@dashdim := \@dashdim / 2
       \@dashcnt := \@dashcnt / 2 - 1
       \box\@dashbox := \hbox{\hskip -\@halfwidth
                               \vrule width \@wholewidth
                               height \@dashdim }

       \put(0,0){\copy\@dashbox}
       \put(X,0){\copy\@dashbox}
       \put(0,Y){\lower\@dashdim\copy\@dashbox}
       \put(X,Y){\lower\@dashdim\copy\@dashbox}
       \@dashdim := 3 * \@dashdim
fi
\box\@dashbox := \hbox{\vrule width \@wholewidth
                        height D * \unitlength }

\@tempcnta := 0
\put(0,0){\hskip -\halfwidth
          \vbox{while \@tempcnta < \@dashcnt
                do \vskip D*\unitlength
                  \copy\@dashbox
                  \@tempcnta := \@tempcnta + 1
                od
                \vskip \@dashdim
              } }
\@tempcnta := 0
\put(X,0){\hskip -\halfwidth

```

```

        \vbox{while \@tempcnta < \@dashcnt
            do \vskip D*\unitlength
              \copy\@dashbox
              \@tempcnta := \@tempcnta + 1
            od
            \vskip \@dashdim
          }
        }
    } % END DASHES

\@imakepicbox(X,Y)
END

\dashbox
166 \gdef\dashbox#1(#2,#3){\leavevmode\hb@xt@\z@{\baselineskip \z@skip
167 \lineskip \z@skip
168 \@dashdim #2\unitlength
169 \@dashcnt \@dashdim \advance\@dashcnt 200
170 \@dashdim #1\unitlength\divide\@dashcnt \@dashdim
171 \ifodd\@dashcnt\@dashdim \z@
172 \advance\@dashcnt \@one \divide\@dashcnt \tw@
173 \else \divide\@dashdim \tw@ \divide\@dashcnt \tw@
174 \advance\@dashcnt \m@ne
175 \setbox\@dashbox \hbox{\vrule \@height \@halfwidth \@depth \@halfwidth
176 \@width \@dashdim}\put(0,0){\copy\@dashbox}%
177 \put(0,#3){\copy\@dashbox}%
178 \put(#2,0){\hskip-\@dashdim\copy\@dashbox}%
179 \put(#2,#3){\hskip-\@dashdim\box\@dashbox}%
180 \multiply\@dashdim \thr@@
181 \fi
182 \setbox\@dashbox \hbox{\vrule \@height \@halfwidth \@depth \@halfwidth
183 \@width #1\unitlength\hskip #1\unitlength}\@tempcnta\z@
184 \put(0,0){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
185 \do{\copy\@dashbox\advance\@tempcnta \@one }}\@tempcnta\z@
186 \put(0,#3){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
187 \do{\copy\@dashbox\advance\@tempcnta \@one }}%
188 \@dashdim #3\unitlength
189 \@dashcnt \@dashdim \advance\@dashcnt 200
190 \@dashdim #1\unitlength\divide\@dashcnt \@dashdim
191 \ifodd\@dashcnt \@dashdim \z@
192 \advance\@dashcnt \@one \divide\@dashcnt \tw@
193 \else
194 \divide\@dashdim \tw@ \divide\@dashcnt \tw@
195 \advance\@dashcnt \m@ne
196 \setbox\@dashbox\hbox{\hskip -\@halfwidth
197 \vrule \@width \@wholewidth
198 \@height \@dashdim}\put(0,0){\copy\@dashbox}%
199 \put(#2,0){\copy\@dashbox}%
200 \put(0,#3){\lower\@dashdim\copy\@dashbox}%
201 \put(#2,#3){\lower\@dashdim\copy\@dashbox}%
202 \multiply\@dashdim \thr@@
203 \fi
204 \setbox\@dashbox\hbox{\vrule \@width \@wholewidth

```

```

205 \@height #1\unitlength}\@tempcnta\z@
206 \put(0,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta <\@dashcnt
207 \do{\vskip #1\unitlength\copy\@dashbox\advance\@tempcnta \@ne }%
208 \vskip\@dashdim}}\@tempcnta\z@
209 \put(#2,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta<\@dashcnt
210 \do{\vskip #1\unitlength\copy\@dashbox\advance\@tempcnta \@ne }%
211 \vskip\@dashdim}}}\@makepicbox(#2,#3)}

```

## CIRCLES AND OVALS

### USER COMMANDS:

`\circle{D}` : Produces the circle with the diameter as close as possible to  $D * \text{\unitlength}$ . `\put(X,Y){\circle{D}}` puts the circle with its center at (X,Y).

`\oval(X,Y)` : Makes an oval as round as possible that fits in the rectangle of width  $X * \text{\unitlength}$  and height  $Y * \text{\unitlength}$ . The reference point is the center.

`\oval(X,Y)[POS]` : Save as `\oval(X,Y)` except it draws only the half or quadrant of the oval indicated by POS. E.G., `\oval(X,Y)[t]` draws just the top half and `\oval(X,Y)[br]` draws just the bottom right quadrant. In all cases, the reference point is the same as the unqualified `\oval(X,Y)` command.

`\@ovvert {DELTA1} {DELTA2}` : Makes a vbox containing either the left side or the right side of the oval being constructed. The baseline will coincide with the outside bottom edge of the oval; the left side of the box will coincide with the left edge of the vertical rule. The width of the box will be `\@tempdima`.

DELTA1 and DELTA2 are added to the character number in `\@tempcnta` to get the characters for the top and bottom quarter circle pieces.

`\@ovhorz` : Makes an hbox containing the straight rule for either the top or the bottom of the oval being constructed. The baseline will coincide with bottom edge of the rule; the left side of the box will coincide with the left side of the oval. The width of the box will be `\@ovxx`.

`\@getcirc {DIAM}` : Sets `\@tempcnta` to the character number of the top-right quarter circle with the largest diameter less than or equal to DIAM. Sets `\@tempboxa` to an hbox containing that character. Sets `\@tempdima` to `\wd \@tempboxa`, which is the distance from the circle's left outside edge to its right inside edge. (These characters are like those described in the

TeXbook, pp. 389-90.)

```

\@getcirc {DIAM} ==
BEGIN
  \@tempcnta      := integer coercion of (DIAM + 2pt)
                                     + 2pt added 1 Nov 88
  \@tempcnta      := \@tempcnta / integer coercion of 4pt
  if \@tempcnta > 10
    then \@tempcnta := 10 fi
  if \@tempcnta > 0
    then \@tempcnta := \@tempcnta-1
    else LaTeX Warning: Oval too small.
  fi
  \@tempcnta      := 4 * \@tempcnta
  \@tempboxa      := \hbox{\@circlefnt \char \@tempcnta}
  \@tempdima      := \wd \@tempboxa
END

\@put{X}{Y}{OBJ} ==
BEGIN
  \raise Y \hb@xt@ 0pt{\hskip X OBJ \hss}
END

\@oval(X,Y)[POS] ==
BEGIN
  \begingroup
  \boxmaxdepth := \maxdimen
  @ovt := @ovb := @ovl := @ovr := true
  for all E in POS
    do @ovE := false od
  \@ovxx      := X * \unitlength
  \@ovyy      := Y * \unitlength
  \@tempdimb := min(\@ovxx,\@ovyy)
  \@getcirc{\@tempdimb-2pt} %% "-2pt" added 7 Dec 89
  \@ovro      := \ht \@tempboxa
  \@ovri      := \dp \@tempboxa
  \@ovdx      := \@ovxx - \@tempdima
  \@ovdx      := \@ovdx/2
  \@ovdy      := \@ovyy - \@tempdima
  \@ovdy      := \@ovdy/2
  \@circlefnt
  \@tempboxa :=
    \hbox{
      if @ovr
        then \@ovvert{3}{2} \kern -\@tempdima
      fi
      if @ovl
        then \kern \@ovxx \@ovvert{0}{1} \kern
-\@tempdima
        \kern -\@ovxx

```

```

        fi
        if @ovt
            then \@ovhorz \@kern -\@ovxx
        fi
        if @ovb
            then \@raise \@ovyy \@ovhorz
        fi
    }
    \@ovdx := \@ovdx + \@ovro
    \@ovdy := \@ovdy + \@ovro
    \ht\@tempboxa := \dp\@tempboxa := 0
    \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}
\endgroup
END

\@ovvert {DELTA1} {DELTA2} ==
BEGIN
    \vbox to \@ovyy {
        if @ovb
            then \@tempcntb := \@tempcnta + DELTA1
                \@kern -\@ovro
                \hbox { \char \@tempcntb }
                \nointerlineskip
            else \@kern \@ovri \@kern \@ovdy
        fi
        \leaders \vrule width \@wholewidth \vfil
        \nointerlineskip
        if @ovt
            then \@tempcntb := \@tempcnta + DELTA2
                \hbox { \char \@tempcntb }
            else \@kern \@ovdy \@kern \@ovro
        fi
    }
END

\@ovhorz ==
BEGIN
    \hb@xt@ \@ovxx{
        \@kern \@ovro
        if @ovr
            then
            else \@kern \@ovdx
        fi
        \leaders \hrule height \@wholewidth \hfil
        if @ovl
            then
            else \@kern \@ovdx
        fi
        \@kern \@ovri
    }

```

```

END

\circle{DIAM} ==
BEGIN
  \begingroup
  \boxmaxdepth := maxdimen
  \@tempdimb := DIAM *\unitlength
  if \@tempdimb > 15.5pt
    then \@getcirc{\@tempdimb}
      \@ovro := \ht \@tempboxa
      \@tempboxa := \hbox{
        \@circlefnt
        \@tempcnta := \@tempcnta + 2
        \char \@tempcnta
        \@tempcnta := \@tempcnta - 1
        \char \@tempcnta
        \kern -2\@tempdima
        \@tempcnta := \@tempcnta + 2
        \raise \@tempdima \hbox { \char \@tempcnta }
        \raise \@tempdima \box\@tempboxa
      }
      \ht\@tempboxa := \dp\@tempboxa := 0
      \@put{-\@ovro}{-\@ovro}{\@tempboxa}
    else
      \@circ{\@tempdimb}{96}
  fi
\endgroup
END

```

```

\circle*{DIAM} == \@dot{DIAM} ==
\@circ{DIAM*\unitlength}{112}

```

```

\@circ{DIAM}{CHAR} ==
BEGIN
  \@tempcnta := integer coercion of (DIAM + .5pt)/1pt.
  if \@tempcnta > 15 then \@tempcnta := 15 fi
  if \@tempcnta > 1 then \@tempcnta := \@tempcnta - 1 fi
  \@tempcnta := \@tempcnta + CHAR
  \@circlefnt
  \char \@tempcnta
END

```

\if@ovt If producing the Top Bottom Left or Right of an oval.

\if@ovb 212 \newif\if@ovt

\if@ovl 213 \newif\if@ovb

\if@ovr 214 \newif\if@ovl

215 \newif\if@ovr

\@ovxx

\@ovyy 216 \newdimen\@ovxx

\@ovdx

\@ovdy

\@ovro File D: ltpictur.dtx Date: 2015/02/21 Version v1.1k

\@ovri

```

217 \newdimen\@ovyy
218 \newdimen\@ovdx
219 \newdimen\@ovdy
220 \newdimen\@ovro
221 \newdimen\@ovri

```

\advance\@tempdima 2pt\relax added 1 Nov 88 to fix bug in which size of drawn circle not monotonic function of argument of \circle, caused by different rounding for dimensions of large and small circles.

\@getcirc

```

222 \gdef\@getcirc#1{\@tempdima #1\relax \advance\@tempdima 2\p@
223   \@tempcnta\@tempdima
224   \@tempdima 4\p@ \divide\@tempcnta\@tempdima
225   \ifnum \@tempcnta >10\relax
226     \@picture@warn
227     \@tempcnta 10\relax
228   \fi
229   \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne
Warn if requirements for oval or circle can't be met.
230   \else \@picture@warn \fi
231   \multiply\@tempcnta 4\relax
232   \setbox \@tempboxa \hbox{\@circlefnt
233     \char \@tempcnta}\@tempdima \wd \@tempboxa}

```

\@picture@warn Generic warning for lines, vectors (used in \@sline) and oval or circle (used un \@getcirc) are not available at right size.

```

234 \def\@picture@warn{\@latex@warning{
235   \string\oval, \string\circle, or \string\line\space
236   size unavailable}}

```

\@put

```

237 \gdef\@put#1#2#3{\raise #2\hb@xt@\z@{\hskip #1#3\hss}}

```

\oval

```

238 \gdef\oval(#1,#2){\@ifnextchar[{\@oval(#1,#2)}{\@oval(#1,#2) []}}

```

\@oval

```

239 \gdef\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
240   \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue
241   \@tfor\reserved@a :=#3\do{\csname @ov\reserved@a false\endcsname}%
242   \@ovxx
243   #1\unitlength \@ovyy #2\unitlength
244   \@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx \@ovxx\else \@ovyy \fi
245   \advance \@tempdimb -2\p@
246   \@getcirc \@tempdimb
247   \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
248   \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
249   \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
250   \@circlefnt \setbox\@tempboxa
251   \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
252   \if@ovl \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx \fi
253   \if@ovt \@ovhorz \kern -\@ovxx \fi

```

```

254 \if@ovb \raise \@ovvy \@ovhorz \fi}\advance\@ovdx\@ovro
255 \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
256 \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
257 \endgroup}

\@ovvert
258 \gdef\@ovvert#1#2{\vbox to\@ovvy{%
259   \if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
260   \kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
261   \else \kern \@ovri \kern \@ovdy \fi
262   \leaders\vrule \@width \@wholewidth\vfil \nointerlineskip
263   \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
264   \hbox{\char \@tempcntb}%
265   \else \kern \@ovdy \kern \@ovro \fi}}

\@ovhorz
266 \gdef\@ovhorz{\hb@xt@\@ovxx{\kern \@ovro
267   \if@ovr \else \kern \@ovdx \fi
268   \leaders \hrule \@height \@wholewidth \hfil
269   \if@ovl \else \kern \@ovdx \fi
270   \kern \@ovri}}

\@circle
271 \gdef\@circle{\@inmatherr\circle\@ifstar\@dot\@circle}

\@circle
272 \gdef\@circle#1{%
273   \begingroup \boxmaxdepth \maxdimen \@tempdima #1\unitlength
274   \ifdim \@tempdima >15.5\p@ \@getcirc\@tempdima
275     \@ovro\ht\@tempboxa
276     \setbox\@tempboxa\hbox{\@circlefnt
277       \advance\@tempcnta\tw@ \char \@tempcnta
278       \advance\@tempcnta\m@ne \char \@tempcnta \kern -2\@tempdima
279       \advance\@tempcnta\tw@
280       \raise \@tempdima \hbox{\char\@tempcnta}\raise \@tempdima
281       \box\@tempboxa}\ht\@tempboxa\z@ \dp\@tempboxa\z@
282       \@put{-\@ovro}{-\@ovro}{\box\@tempboxa}%
283     \else \@circ\@tempdima{96}\fi\endgroup}

\@dot Internal form of \circle*.
284 \gdef\@dot#1{\@tempdima #1\unitlength \@circ\@tempdima{112}}

\@circ
285 \gdef\@circ#1#2{\@tempdima #1\relax \advance\@tempdima .5\p@
286   \@tempcnta\@tempdima \@tempdima \p@
287   \divide\@tempcnta\@tempdima
288   \ifnum\@tempcnta >15\relax \@tempcnta 15\relax \fi
289   \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne\fi
290   \advance\@tempcnta #2\relax
291   \@circlefnt \char\@tempcnta}

\@xarg Counters used for manipulating the ‘slope’ arguments.
\@yarg
292 \newcount\@xarg
\@yyarg
293 \newcount\@yarg
294 \newcount\@yyarg

```

`\@multicnt` Counter used in `\multiput`, and also `\multicolumn`.  
295 `\newcount\@multicnt`

`\@xdim` Length registers.  
`\@ydim` 296 `\newdimen\@xdim`  
297 `\newdimen\@ydim`

`\@linechar` Box for holding a line segment character, for sloping lines.  
298 `\newbox\@linechar`

`\@linelen` Length of the line currently being built.  
299 `\newdimen\@linelen`

`\@clnwd` Height and width of current line segment.  
`\@clnht` 300 `\newdimen\@clnwd`  
301 `\newdimen\@clnht`

`\@dashdim` `\dashbox` internal registers.  
`\@dashbox` 302 `\newdimen\@dashdim`  
`\@dashcnt` 303 `\newbox\@dashbox`  
304 `\newcount\@dashcnt`

Initialization: “`\thinlines`”  
305 `\let\@linefnt\tenln`  
306 `\let\@circlefnt\tencirc`  
307 `\@wholewidth\fontdimen8\tenln`  
308 `\@halfwidth .5\@wholewidth`

## 57.1 Curves

The new `\qbezier` command, based on the old `\bezier` defined in `bezier.sty`.

```
\qbezier[N] == \bezier{N}

\bezier{N}(AX,AY)(BX,BY)(CX,CY) ==
BEGIN
  IF N = 0
    THEN \@xdima := |BX - AX|
      \@xb := |CX - BX|
      \@xa := Max(\@xa, \@xb)
      \@ya := |BY - AY|
      \@yb := |CY - BY|
      \@ya := Max(\@ya, \@yb)
      @sc := Max(\@xa, \@ya)
      %% The coefficient .5 below is the degree of overlap of
      %% successive points, where 1 is no overlap and 0 is
      %% complete overlap. A coefficient of C multiplies
      %% the number of points plotted by 1/C.
      %%
      \@xa := .5 * \@halfwidth
      @sc := @sc / \@halfwidth
```

```

@sc := Max(@sc, qbeziermax)
ELSE @sc := N
@scp := @sc+1
\@xb := 2 * (BX - AX) * \unitlength
\@xa := ((CX-AX)*\unitlength - \@xb)/@sc
\@yb := 2 * (BY - AY) * \unitlength
\@ya := ((CY-AY)*\unitlength - \@yb)/@sc
\@pictdot := square rule of width \@wholewidth
\count@ := 0
WHILE \count@ < @scp
DO \@xdim := ((\count@*\@xa + @xb) / @sc) * \count@
\@ydim := ((\count@*\@ya + @yb) / @sc) * \count@
plot pt with relative coords (\@xdim,\@ydim)
\count@ := \count@+1
OD

```

`\qbeziermax` The maximum number of points to plot.

```
309 \gdef\qbeziermax{500}
```

In the code below, to save registers `\@a ...` are not used. Instead other registers are reused.

```

\newcounter{sc} -> \c@multicnt
\newcounter{scp} -> \@tempcnta
\newdimen\@xa -> \@ovxx
\newdimen\@xb -> \@ovdx
\newdimen\@ya -> \@ovyy
\newdimen\@yb -> \@ovdy
\newsavebox{\@pictdot} -> \@tempboxa

```

`\qbezier` Main user-level command to plot quadratic bezier curves. #2 should be (.

```
310 \newcommand\qbezier[2][0]{\bezier{#1}#2}
```

`\bezier` Form of `\bezier` compatible with 2.09 `bezier.sty`, but modified to ignore spaces between its arguments. #2 should be white space, and #4 should be (.

```
311 \gdef\bezier#1#2(#3)#4({\@bezier#1)(#3){}
```

`\@bezier`

```

312 \gdef\@bezier#1(#2,#3)(#4,#5)(#6,#7){%
313   \ifnum #1=\z@
314     \@ovxx #4\unitlength
315     \advance\@ovxx -#2\unitlength
316     \ifdim \@ovxx<\z@ \@ovxx -\@ovxx \fi
317     \@ovdx #6\unitlength
318     \advance\@ovdx -#4\unitlength
319     \ifdim \@ovdx<\z@ \@ovdx -\@ovdx \fi
320     \ifdim \@ovxx<\@ovdx \@ovxx \@ovdx \fi
321     \@ovyy #5\unitlength
322     \advance\@ovyy -#3\unitlength
323     \ifdim \@ovyy<\z@ \@ovyy -\@ovyy \fi
324     \@ovdy #7\unitlength

```

```

325         \advance\@ovdy -#5\unitlength
326         \ifdim \@ovdy<\z@ \@ovdy -\@ovdy \fi
327         \ifdim \@ovvy<\@ovdy \@ovvy \@ovdy \fi
328         \@multicnt
329         \ifdim \@ovxx>\@ovvy \@ovxx \else \@ovvy \fi
330         \@ovxx .5\@halfwidth \divide\@multicnt\@ovxx
331         \ifnum \qbeziermax<\@multicnt \@multicnt\qbeziermax\relax \fi
332     \else \@multicnt#1\relax \fi
333     \@tempcnta\@multicnt \advance\@tempcnta\@ne
334     \@ovdx #4\unitlength \advance\@ovdx -#2\unitlength
335     \multiply\@ovdx \tw@
336     \@ovxx #6\unitlength \advance\@ovxx -#2\unitlength
337     \advance\@ovxx -\@ovdx \divide\@ovxx\@multicnt
338     \@ovdy #5\unitlength \advance\@ovdy -#3\unitlength
339     \multiply\@ovdy \tw@
340     \@ovvy #7\unitlength \advance\@ovvy -#3\unitlength
341     \advance\@ovvy -\@ovdy \divide\@ovvy\@multicnt

342 \setbox\@tempboxa\hbox{%
343     \hskip -\@halfwidth
344     \vrule \@height\@halfwidth
345         \@depth \@halfwidth
346         \@width \@wholewidth}%
347 \put(#2,#3){%
348     \count@\z@
349     \@whilenum{\count@<\@tempcnta}\do
350     {\@xdim\count@\@ovxx
351         \advance\@xdim\@ovdx
352         \divide\@xdim\@multicnt
353         \multiply\@xdim\count@
354         \@ydim\count@\@ovvy
355         \advance\@ydim\@ovdy
356         \divide\@ydim\@multicnt
357         \multiply\@ydim\count@
358         \raise \@ydim
359         \hb@xt@\z@{\kern\@xdim
360             \unhcopy\@tempboxa\hss}%
361         \advance\count@\@ne}}
362 /2ekernel)

```

## File E

# ltthm.dtx

## 58 Theorem Environments

The user creates his own theorem-like environments with the command

```
\newtheorem{<name>}{<text>}[<counter>] or  
\newtheorem{<name>}[<oldname>]{<text>}
```

This defines the environment  $\langle name \rangle$  to be just as one would expect a theorem environment to be, except that it prints  $\langle text \rangle$  instead of “Theorem”.

If  $\langle oldname \rangle$  is given, then environments  $\langle name \rangle$  and  $\langle oldname \rangle$  use the same counter, so using a  $\langle name \rangle$  environment advances the number of the next  $\langle name \rangle$  environment, and vice-versa.

If  $\langle counter \rangle$  is given, then environment  $\langle name \rangle$  is numbered within  $\langle counter \rangle$ .

E.g., if  $\langle counter \rangle = \text{subsection}$ , then the first  $\langle name \rangle$  in subsection 7.2 is numbered  $\langle text \rangle$  7.2.1.

The way  $\langle name \rangle$  environments are numbered can be changed by redefining  $\backslash the\langle name \rangle$ .

### DOCUMENT STYLE PARAMETERS

$\backslash \thmcounter\{COUNTER\}$  : A command such that

```
\edef\theCOUNTER{\@thmcounter{COUNTER}}
```

defines  $\backslash theCOUNTER$  to produce a number for a theorem environment.

The default is:

```
BEGIN \noexpand\arabic{COUNTER} END
```

$\backslash \thmcountersep$  : A separator placed between a theorem number and the number of the counter within which it is numbered.

E.g., to make the third theorem of section 7.2 be numbered 7.2-3,  $\backslash \thmcountersep$  should be  $\backslash def$ 'ed to '-'. Its default is ' '.

$\backslash @begintheorem\{NAME\}\{NUMBER\}$  : A command that begins a theorem

environment for a 'theorem' named 'NAME NUMBER' – e.g.,  $\backslash @begintheorem\{Lemma\}\{3.7\}$  starts Lemma 3.7.

$\backslash @opargbegintheorem\{NAME\}\{NUMBER\}\{OPARG\}$  :

A command that begins a theorem

environment for a 'theorem' named 'NAME NUMBER' with optional

argument OPARG – e.g.,  $\backslash @begintheorem\{Lemma\}\{3.7\}\{Jones\}$  starts 'Lemma 3.7 (Jones):'.

$\backslash @endtheorem$  : A command that ends a theorem environment.

$\backslash newtheorem\{NAME\}\{TEXT\}[COUNTER] ==$

```

BEGIN
  if \NAME is definable
  then \@definecounter{NAME}
    if COUNTER present
    then \@newctr{NAME}[COUNTER] fi
    \theNAME == BEGIN \theCOUNTER \@thmcountersep
                        eval\@thmcounter{NAME}
  END
    else \theNAME == BEGIN eval\@thmcounter{NAME} END
    \NAME == \@thm{NAME}{TEXT}
    \endNAME == \@endtheorem
  else error
  fi
END

\newtheorem{NAME}[OLDNAME]{TEXT}==
BEGIN
  if counter OLDNAME nonexistent
  then ERROR
  else
    if \NAME is definable
    then BEGIN
      \theNAME == \theOLDNAME
      \NAME == \@thm{OLDNAME}{TEXT}
      \endNAME == \@endtheorem
      END
    else error
    fi
  fi
END

\@thm{NAME}{TEXT} ==
BEGIN
  \refstepcounter{NAME}
  if next char = [
    then \@ythm{NAME}{TEXT}
    else \@xthm{NAME}{TEXT}
  fi
END

\@xthm{NAME}{TEXT} ==
BEGIN
  \@begintheorem{TEXT}{\theNAME}
  \ignorespaces
END

\@ythm{NAME}{TEXT}[OPARG] ==
BEGIN
  \@opargbegintheorem{TEXT}{\theNAME}{OPARG}
  \ignorespaces

```

END

`\newtheorem` `\newtheorem` ought really be allowed only in the preamble Which would be good document style, and allow some main memory to be saved by declaring these commands to be `\@onlypreamble`. Unfortunately the L<sup>A</sup>T<sub>E</sub>X book indicates that `\newtheorem` may be used anywhere in the document...

```
1 \*2ekernel)
2 \def\newtheorem#1{%
3   \ifnextchar[{\@othm{#1}}{\@nthm{#1}}}
```

`\@nthm`

```
4 \def\@nthm#1#2{%
5   \ifnextchar[{\@xnthm{#1}{#2}}{\@ynthm{#1}{#2}}}
```

`\@xnthm` 92/09/18 RmS: Changed `\@addtoreset` to `\@newctr` to produce error message if counter #3 does not exist (to be consistent with behaviour of `\newcounter`)

```
6 \def\@xnthm#1#2[#3]{%
7   \expandafter\ifdefinable\csname #1\endcsname
8     {\@definecounter{#1}\@newctr{#1}[#3]%
9     \expandafter\xdef\csname the#1\endcsname{%
10      \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
11      \@thmcounter{#1}}}%
12   \global\@namedef{#1}{\@thm{#1}{#2}}%
13   \global\@namedef{end#1}{\@endtheorem}}}
```

`\@ynthm`

```
14 \def\@ynthm#1#2{%
15   \expandafter\ifdefinable\csname #1\endcsname
16     {\@definecounter{#1}%
17     \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}}%
18   \global\@namedef{#1}{\@thm{#1}{#2}}%
19   \global\@namedef{end#1}{\@endtheorem}}}
```

`\@othm`

```
20 \def\@othm#1[#2]#3{%
21   \@ifundefined{c@#2}{\@nocounterr{#2}}%
22   {\expandafter\ifdefinable\csname #1\endcsname
23     {\global\@namedef{the#1}{\@nameuse{the#2}}}%
24     \global\@namedef{#1}{\@thm{#2}{#3}}%
25     \global\@namedef{end#1}{\@endtheorem}}}
```

`\@thm`

```
26 \def\@thm#1#2{%
27   \refstepcounter{#1}%
28   \ifnextchar[{\@ythm{#1}{#2}}{\@xthm{#1}{#2}}}
```

`\@xthm`

`\@ythm`

```
29 \def\@xthm#1#2{%
30   \@begintheorem{#2}{\csname the#1\endcsname}\ignorespaces}
31 \def\@ythm#1#2[#3]{%
32   \@opargbegintheorem{#2}{\csname the#1\endcsname}{#3}\ignorespaces}
```

Default values

```

\@thmcounter
\@thmcountersep 33 \def\@thmcounter#1{\noexpand\arabic{#1}}
34 \def\@thmcountersep{.}

\@begintheorem Providing theorem defaults.
\@opargbegintheorem 35 \def\@begintheorem#1#2{\trivlist
\@endtheorem 36 \item[\hskip \labelsep{\bfseries #1\ #2}]{\itshape}
37 \def\@opargbegintheorem#1#2#3{\trivlist
38 \item[\hskip \labelsep{\bfseries #1\ #2\ (#3)}]{\itshape}
39 \def\@endtheorem{\endtrivlist}
40 \</2ekernel>

```

# File F

## ltsect.dtx

### 59 Sectioning Commands

This file defines the declarations such as `\author` which are used by `\maketitle`. `\maketitle` itself is defined by each class, not in the L<sup>A</sup>T<sub>E</sub>X kernel.

The second part of the file defines the generic commands used for defining sectioning commands such as `\chapter`. Again the actual document level commands are defined in the class files, in terms of these commands.

```
1 \*2ekernel)
2 \message{title,}
```

#### 59.1 The Title

```
\title The user defines the title and author by the declarations \title{<name>},
\author \author{<name>}
\date Similarly the date is declared with \date{<date>}.
\thanks Inside these, the \thanks{<footnote text>} command may be used to make
\and acknowledgements, notice of address, etc. in a footnote. If there are multiple
\maketitle authors, they have to be separated with the \and command.
And finally, the \maketitle command produces the actual title, using the
information previously saved with the other commands.

\title \title for use in \maketitle. If not given \maketitle will produce an error
\@title message.
3 \def\title#1{\gdef\@title{#1}}
4 \def\@title{\@latex@error{No \noexpand\title given}\@ehc}

\author \author for use in \maketitle. If not given \maketitle will produce a warning
\@author message.
5 \def\author#1{\gdef\@author{#1}}
6 \def\@author{\@latex@warning@no@line{No \noexpand\author given}}

\date \date for use in \maketitle. If not given \maketitle will produce \today as the
\@date default.
7 \def\date#1{\gdef\@date{#1}}
8 \gdef\@date{\today}

\thanks
9 \def\thanks#1{\footnotemark
10 \protected@xdef\@thanks{\@thanks
11 \protect\footnotetext[\the\c@footnote]{#1}}%
12 }

\@thanks
13 \let\@thanks\@empty
```

```

\and
14 \def\and{%                % \begin{tabular}
15   \end{tabular}%
16   \hskip 1em \@plus.17fil%
17   \begin{tabular}[t]{c}}%  % \end{tabular}

18 \message{sectioning,}

```

## 59.2 Sectioning

```

\@secpenalty
19 \newcount\@secpenalty
20 \@secpenalty = -300

\if@noskipsec  Way back in 1991 (08/26) FMi & RmS set the \@noskipsec switch to true for the
\@noskipsectrue preamble and to false in \document. This was done to trap lists and related text
in the preamble but it does not catch everything.
21 \newif\if@noskipsec \@noskipsectrue

\@startsection The \@startsection{<name>}{<level>}{<indent>}{<before skip>}{
{<after skip>}{<style>}* [<altheading> ]{<heading>} command is the mother of all
the user level sectioning commands. The part after the *, including the * is
optional.

```

**name:** e.g., 'subsection'

**level:** a number, denoting depth of section – e.g., chapter=1, section = 2, etc.

**indent:** Indentation of heading from left margin

**before skip:** Absolute value = skip to leave above the heading. If negative, then paragraph indent of text following heading is suppressed.

**after skip:** if positive, then skip to leave below heading, else negative of skip to leave to right of run-in heading.

**style:** Commands to set style. Since June 1996 release the *last* command in this argument may be a command such as `\MakeUppercase` or `\fbox` that takes an argument. The section heading will be supplied as the argument to this command. So setting #6 to, say, `\bfseries\MakeUppercase` would produce bold, uppercase headings.

If ‘\*’ is missing, then increment the counter. If it is present, then there should be no [*altheading*] argument. The command uses the counter ‘secnumdepth’. It contains a pointer to the highest section level that is to be numbered.

**Warning:** The `\@startsection` command should be at the same or higher grouping level as the text that follows it. For example, you should *not* do something like

```

\def\foo{ \begingroup ...
          \paragraph{...}
          \endgroup}

```

```

Pseudocode for the \@startsection command
\@startsection
{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE} ==
BEGIN
  IF @noskipsec = T THEN \leavevmode FI
                                % true if previous section had no body.

  \par
  \@tempskipa := BEFORESKIP
  @afterindent := T
  IF \@tempskipa < 0 THEN \@tempskipa := -\@tempskipa
                                @afterindent := F
  FI
  IF @nobreak = true
    THEN \everypar == null
    ELSE \addpenalty{\@secpenalty}
         \addvspace{\@tempskipa}
  FI
  IF * next
    THEN \@ssect{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}
    ELSE \@dblarg{\@sect
                  {NAME}{LEVEL}{INDENT}
                  {BEFORESKIP}{AFTERSKIP}{STYLE}}
  FI
END

22 \def\@startsection#1#2#3#4#5#6{%
23   \if@noskipsec \leavevmode \fi
24   \par
25   \@tempskipa #4\relax
26   \@afterindenttrue
27   \ifdim \@tempskipa <\z@
28     \@tempskipa -\@tempskipa \@afterindentfalse
29   \fi
30   \if@nobreak
31     \everypar{}%
32   \else
33     \addpenalty\@secpenalty\addvspace\@tempskipa
34   \fi
35   \@ifstar
36     {\@ssect{#3}{#4}{#5}{#6}}%
37     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}

```

\@sect Pseudocode for the \@sect command

```

\@sect{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}[ARG1]{ARG2}
==
BEGIN
  IF LEVEL > \c@secnumdepth
    THEN \@svsec :=L null
    ELSE \refstepcounter{NAME}
         \@svsec :=L BEGIN \@seccntformat{#1}\relax END

```

```

FI
IF AFTERSKIP > 0
  THEN \beginngroup
        STYLE
        \@hangfrom{\hskip INDENT\@svsec}
        {\interlinepenalty 10000 ARG2\par}
      \endgroup
      \NAMEmark{ARG1}
      \addcontentsline{toc}{NAME}
      { IF LEVEL > \c@secnumdepth
        ELSE \protect\numberline{\theNAME} FI
        ARG1 }
  ELSE \@svsechd == BEGIN STYLE
        \hskip INDENT\@svsec
        ARG2
        \NAMEmark{ARG1}
        \addcontentsline{toc}{NAME}
        { IF LEVEL > \c@secnumdepth
          ELSE

\protect\numberline{\theNAME}

        FI
        ARG1 }

      END

    FI
    \@xsect{AFTERSKIP}
  END

38 \def\@sect#1#2#3#4#5#6[#7]#8{%
39   \ifnum #2>\c@secnumdepth
40     \let\@svsec\empty
41   \else
42     \refstepcounter{#1}%

```

Since \@seccntformat might end with an improper \hskip which is scanning forward for plus or minus we end the definition of \@svsec with \relax as a precaution.

```

43   \protected@edef\@svsec{\@seccntformat{#1}\relax}%
44   \fi
45   \@tempskipa #5\relax
46   \ifdim \@tempskipa>\z@
47     \beginngroup

```

This { used to be after the argument to \@hangfrom but was moved here to allow commands such as \MakeUppercase to be used at the end of #6.

```

48     #6{%
49       \@hangfrom{\hskip #3\relax\@svsec}%
50       \interlinepenalty \@M #8\@par}%
51   \endgroup
52   \csname #1mark\endcsname{#7}%
53   \addcontentsline{toc}{#1}{%
54     \ifnum #2>\c@secnumdepth \else
55       \protect\numberline{\csname the#1\endcsname}%

```

```

56      \fi
57      #7}%
58  \else
\relax added 2 May 90
59      \def\@svsechd{%
60          #6{\hskip #3\relax
61              \@svsec #8}%
62          \csname #1mark\endcsname{#7}%
63          \addcontentsline{toc}{#1}{%
64              \ifnum #2>\c@secnumdepth \else
65                  \protect\numberline{\csname the#1\endcsname}%
66              \fi
67              #7}}%
68  \fi
69  \@xsect{#5}}

\@xsect Pseudocode for the \@xsect command
\@xsect{AFTERSKIP} ==
BEGIN
  IF AFTERSKIP > 0
    THEN \par \nobreak
         \vskip AFTERSKIP
         \@afterheading
    ELSE @nobreak :=G F
         @noskipsec :=G T
         \everypar{ IF @noskipsec = T
                     THEN @noskipsec :=G F
                        \clubpenalty :=G 10000
                        \hskip -\parindent
                        \begingroup
                        \@svsechd
                        \endgroup
                        \unskip
                        \hskip -AFTERSKIP \relax
                        %% relax added 14 Jan 91
                     ELSE \clubpenalty :=G \@clubpenalty
                        \everypar := NULL
                     FI
                 }
    FI

  FI

END

70 \def\@xsect#1{%
71   \@tempskipa #1\relax
72   \ifdim \@tempskipa>\z@

Why not combine \@sect and \@xsect and save doing the same test twice? It is
not possible to change this now as these have become hooks!
This \par seems unnecessary.

73   \par \nobreak
74   \vskip \@tempskipa

```

```

75   \@afterheading
76   \else

77   \@nobreakfalse
78   \global\@noskipsecttrue
79   \everypar{%
80     \if@noskipsec
81       \global\@noskipsecfalse
82       {\setbox\z@\lastbox}%
83       \clubpenalty\@M
84       \begingroup \@svsechd \endgroup
85       \unskip
86       \@tempskipa #1\relax
87       \hskip -\@tempskipa
88     \else
89       \clubpenalty \@clubpenalty
90       \everypar{}}%
91   \fi}%
92   \fi
93   \ignorespaces}

```

`\@secntformat` This command formats the section number including the space following it.

```

94 \def\@secntformat#1{\csname the#1\endcsname\quad}

```

Pseudocode for the `\@ssect` command

```

\@ssect{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}{ARG} ==
BEGIN
  IF AFTERSKIP > 0
    THEN \begingroup
          STYLE
          \@hangfrom{\hskip INDENT}{\interlinepenalty 10000
ARG\par}
          \endgroup
    ELSE \@svsechd == BEGIN STYLE
                          \hskip INDENT
                          ARG
                        END
  FI
  \@xsect{AFTERSKIP}
END

```

Pseudocode for the `\@afterheading` command

```

\@afterheading ==
BEGIN
  @nobreak :=G true
  \everypar := BEGIN IF @nobreak = T
                      THEN @nobreak :=G false
                        \clubpenalty :=G 10000
                        IF @afterindent = F
                          THEN remove \lastbox
                        FI
                      ELSE \clubpenalty :=G \@clubpenalty
                        \everypar := NULL

```

```

                                FI
                                END
                                END

\@ssect
95 \def\@ssect#1#2#3#4#5{%
96   \@tempskipa #3\relax
97   \ifdim \@tempskipa>\z@
98     \begingroup
This { used to be after the argument to \@hangfrom but was moved here to allow
commands such as \MakeUppercase to be used at the end of #4.
99     #4{%
100       \@hangfrom{\hskip #1}%
101       \interlinepenalty \@M #5\@par}%
102     \endgroup
103   \else
104     \def\@svsechd{#4{\hskip #1\relax #5}}%
105     \fi
106   \@xsect{#3}}

\if@afterindent
\@afterindenttrue 107 \newif\if@afterindent \@afterindenttrue

\@afterheading This hook is used in setting up custom-built headings in classes.dtx.
108 \def\@afterheading{%
109   \@nobreaktrue
110   \everypar{%
111     \if@nobreak
112       \@nobreakfalse
113       \clubpenalty \@M
114       \if@afterindent \else
115         {\setbox\z@\lastbox}%
116       \fi
117     \else
118       \clubpenalty \@clubpenalty
119       \everypar{}%
120     \fi}}

\@hangfrom \text : Puts text in a box, and makes a hanging indentation of
the following material up to the first \par. Should be used in vertical mode.
121 \def\@hangfrom#1{\setbox\@tempboxa\hbox{#1}}%
122   \hangindent \wd\@tempboxa\noindent\box\@tempboxa}

\c@secnumdepth
\c@tocdepth 123 \newcount\c@secnumdepth
124 \newcount\c@tocdepth

\secdef \secdef{\unstarcmds}{\unstarcmds}{\starcmds}
When defining a \chapter or \section command without using \@startsection,
you can use \secdef as follows:
1. \def\chapter{ ... \secdef \starcmd \unstarcmd }

```

2. `\def\starcmd{#1}#2{...}` % Command to define `\chapter[...]{...}`

3. `\def\unstarcmd{#1}{...}` % Command to define `\chapter*{...}`

```
125 \def\secdef#1#2{\@ifstar{#2}{\@dblarg{#1}}}
```

### 59.2.1 Initializations

```
\sectionmark
\subsectionmark 126 \let\sectionmark\@gobble
\subsubsectionmark 127 \let\subsectionmark\@gobble
\paragraphmark 128 \let\subsubsectionmark\@gobble
\subparagraphmark 129 \let\paragraphmark\@gobble
130 \let\subparagraphmark\@gobble

131 \message{contents,}
```

## 59.3 Table of Contents etc.

### 59.3.1 Convention

`\tf@{foo}` = file number for output for table foo. The file is opened only if `@filesw = true`.

### 59.3.2 Commands

A `\l@{type}{<entry>}{<page>}` Macro needs to be defined by document style for making an entry of type `<type>` in a table of contents, etc. E.g., the document style should define `\l@chapter`, `\l@section`, etc.

**Note:** When the `\protect` command is used in the `<entry>` or `<text>` of one of the commands below, it causes the following control sequence to be written on the file without being expanded. The sequence will be expanded when the table of contents entry is processed.

**Surprise:** Inside an `\addcontentsline` or `\addtocontents` command argument, the commands: `\index`, `\glossary`, and `\label` are no-ops. This could cause a problem if the user puts an `\index` or `\label` into one of the commands he writes, or into the optional ‘short version’ argument of a `\section` or `\caption` command.

`\@starttoc` The `\@starttoc{<ext>}` command is used to define the commands: `\tableofcontents`, `\listoffigures`, etc.

For example: `\@starttoc{lof}` is used in `\listoffigures`. This command reads the `.<ext>` file and sets up to write the new `.<ext>` file.

```
\@starttoc{EXT} ==
BEGIN
  \begingroup
  \makeatletter
  read file \jobname.EXT
  IF @filesw = true
    THEN open \jobname.EXT as file \tf@EXT
  FI
  @nobreak :=G FALSE %% added 24 May 89
```

```

\endgroup
END

```

```

132 \def\starttoc#1{%
133   \begingroup
134   \makeatletter
135   \@input{\jobname.#1}%
136   \if@filesw
137     \expandafter\newwrite\csname tf@#1\endcsname
138     \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
139   \fi
140   \@nobreakfalse
141 \endgroup}

```

**\addcontentsline** The `\addcontentsline{<table>}{<type>}{<entry>}` command allows the user to add his/her own entry to a table of contents, etc. The command adds the entry `\contentsline{<type>}{<entry>}{<page>}` to the `.<table>` file.

This macro is implemented as an application of `\addtocontents`. Note that `\thepage` is not expandable during `\protected@write` therefore one gets the page number at the time of the `\shipout`.

```

142 \def\addcontentsline#1#2#3{%
143   \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}}}

```

**\addtocontents** The `\addtocontents{<table>}{<text>}` command adds `<text>` to the `.<table>` file, with no page number.

```

144 \long\def\addtocontents#1#2{%
145   \protected@write\@auxout
146     {\let\label@gobble \let\index@gobble \let\glossary@gobble}%
147     {\string\@writefile{#1}{#2}}}

```

**\contentsline** The `\contentsline{<type>}{<entry>}{<page>}` macro produces a `<type>` entry in a table of contents, etc. It will appear in the `.toc` or other file. For example, The entry for subsection 1.4.3 in the table of contents for example, might be produced by:

```

\contentsline{subsection}
  {\makebox{30pt}[r]{1.4.3} Gnats and Gnus}{22}

```

The `\protect` command causes command sequences to be written without expanding them.

```

148 \def\contentsline#1{\csname l@#1\endcsname}

```

`\@dottedtocline{<level>}{<indent>}{<numwidth> }{<title>}{<page>}`: Macro to produce a table of contents line with the following parameters:

**level** If `<level>` > `\c@tocdepth`, then no line produced.

**indent** Total indentation from the left margin.

**numwidth** Width of box for number if the `<title>` has a `\numberline` command. As of 25 Jan 1988, this is also the amount of extra indentation added to second and later lines of a multiple line entry.

**title** Contents of entry.

**page** Page number.

Uses the following parameters, which must be set by the document style. They should be defined with `\def`'s.

**pnumwidth** Width of box in which page number is set.

**tocrmarg** Right margin indentation for all but last line of multiple-line entries.

**dotsep** Separation between dots, in mu units. Should be `\def`'d to a number like 2 or 1.7

`\@dottedtocline`

```
149 \def\@dottedtocline#1#2#3#4#5{%
150   \ifnum #1>\c@tocdepth \else
151     \vskip \z@ \@plus.2\p@
152     {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
153      \parindent #2\relax\@afterindenttrue
154      \interlinepenalty\@M
155      \leavevmode
156      \@tempdima #3\relax
157      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
158      {#4}\nobreak
159      \leaders\hbox{$\m@th
```

If a document uses fonts other than computer modern, the use of a dot from math can be very disturbing despite the fact that this might be the only place in a document that then uses computer modern. Therefore we surround the dot with an `\hbox` to escape to the surrounding text font.

```
160       \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
161       mu$}\hfill
162       \nobreak
163       \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
164       \par}%
165 \fi}
```

**Note:** `\nobreak`'s added 7 Jan 86 to prevent bad line break that left the page number dangling by itself at left edge of a new line.

Changed 25 Jan 88 to use `\leftskip` instead of `\hangindent` so leaders of multiple-line contents entries would line up properly.

`\numberline` `\numberline{<number>}`: For use in a `\contentsline` command. It puts `<number>` flushleft in a box of width `\@tempdima` (Before 25 Jan 88 change, it also added `\@tempdima` to the hanging indentation.)

```
166 \def\numberline#1{\hb@xt@\@tempdima{#1\hfil}}
167 </2ekernel>
```

## File G

# ltfloat.dtx

## 60 Floats

The different types of floats are identified by a  $\langle type \rangle$  name, which is the name of the counter for that kind of float. For example, figures are of type ‘figure’ and tables are of type ‘table’. Each  $\langle type \rangle$  has associated a positive  $\langle type\ number \rangle$ , which is a power of two. E.g., figures might have type number 1, tables type number 2, programs type number 4, etc.

The locations where a float can go are specified by a  $\langle placement\ specifier \rangle$ , which is a list of the possible locations, each denoted by a letter as follows:

h : here	— at the current location in the text.
t : top	— at the top of a text page.
b : bottom	— at the bottom of a text page.
p : page	— on a separate float page

In addition, in conjunction with these, you can use ‘!’ which means that the current values of the float positioning parameters are ignored for this float. (Has no effect on ‘p’, float page positioning.) For example, ‘pht’ specifies that the float can appear in any of three locations: page, here or top.

### 60.1 Floating Environments

```
1 \*2ekernel
2 \message{floats,}
```

Where floats may appear on a page, and how many may appear there are specified by the following float placement parameters. The numbers are named like counters so the user can set them with the ordinary counter-setting commands.

<code>\c@topnumber</code>	: Number of floats allowed at the top of a column.
<code>\topfraction</code>	: Fraction of column that can be devoted to floats.
<code>\c@dbltopnumber, \dbltopfraction</code>	: Same as above, but for double-column floats.
<code>\c@bottomnumber, \bottomfraction</code>	: Same as above for bottom of page.
<code>\c@totalnumber</code>	: Number of floats allowed in a single column, including in-text floats.
<code>\textfraction</code>	: Minimum fraction of column that must contain text.
<code>\floatpagefraction</code>	: Minimum fraction of page that must be taken up by float page.
<code>\dblfloatpagefraction</code>	: Same as above, for double-column floats.

The document style must define the following.

`\fps@TYPE` : The default placement specifier for floats of type TYPE.

`\ftype@TYPE` : The type number for floats of type TYPE.

`\ext@TYPE` : The file extension indicating the file on which the contents list for float type TYPE is stored.  
For example, `\ext@figure = 'lof'`.

`\fnum@TYPE` : A macro to generate the figure number for a caption.  
For example, `\fnum@TYPE == Figure \thefigure`.

`\@makecaption{NUM}{TEXT}` :  
A macro to make a caption, with NUM the value produced by `\fnum@...` and TEXT the text of the caption. It can assume it's in a `\parbox` of the appropriate width.

`\@float{TYPE}[PLACEMENT]` : This macro begins a float environment for a  
single-column float of type TYPE with PLACEMENT as the placement specifier. The default value of PLACEMENT is defined by `\fps@TYPE`. The environment is ended by `\end@float`.  
E.g., `\figure == \@float{figure}, \endfigure == \end@float`.

`\@float{TYPE}[PLACEMENT] ==`  
`BEGIN`  
    if hmode then `\@bsphack`  
        `\@floatpenalty := -10002`  
    else `\@floatpenalty := -10003`  
    fi  
    `\@captype ==L TYPE`  
    `\@dblflset`  
    `\@fps ==L PLACEMENT`  
    `\@onelevel@sanitize \@fps`  
    add default PLACEMENT if at most ! in PLACEMENT ==  
`\@fpsadddefault`  
    if inner  
        then LaTeX Error: 'Not in outer paragraph mode.'  
        `\@floatpenalty := 0`  
    else if `\@freelist` nonempty  
        then `\@currbox :=L head of \@freelist`  
            `\@freelist :=G tail of \@freelist`  
            `\count\@currbox :=G 32*\ftype@TYPE +`  
  bits determined by  
PLACEMENT  
        else `\@floatpenalty := 0`  
            LaTeX Error: 'Too many unprocessed floats'  
    fi

```

fi
\@currbox :=G \color@vbox
\normalcolor
\ vbox{
%% 15 Dec 87 -
%% removed \boxmaxdepth :=L 0pt
%% that made box 0 depth because it screwed
%% things up. Instead, added \vskip0pt at
end

\hsize = \columnwidth
\@parboxrestore
\@floatboxreset

END

```

```

\caption ==
BEGIN
\refstepcounter{\@capttype}
\@dblarg{\caption{\@capttype}}
END

```

In following definition, `\par` moved from after `\addcontentsline` to before `\addcontentsline` because the `\write` could cause an extra blank line to be added to the paragraph above the caption. (Change made 12 Jun 87)

```

\caption{TYPE}[STEXT]{TEXT} ==
BEGIN
\par

\addcontentsline{\ext@TYPE}{TYPE}{\numberline{\theTYPE}{STEXT}}
\begingroup
\@parboxrestore
\@normalsize
\@makecaption{\fnum@TYPE}{TEXT}
\par
\endgroup
END

```

`\@dblfloat{TYPE}[PLACEMENT]` : Macro to begin a float environment for

a double-column float of type TYPE with PLACEMENT as the placement

specifier. The default value of PLACEMENT is 'tp'

The environment is ended by `\end@dblfloat`.

E.g., `\figure* == \@dblfloat{figure}`,

`\endfigure* == \end@dblfloat`.

```

\@dblfloat{TYPE}[PLACEMENT] ==

```

Identical to `\@float{TYPE}[PLACEMENT]` except `\hsize` and `\linewidth` are set to `\textwidth`.

`\@floatpenalty`

3 `\newcount\@floatpenalty`

`\caption` This is set to be an error message outside a float since no captype is defined there; this may need to be changed by some classes.

```

4 \def\caption{%
5   \ifx\@captype\@undefined
6     \latexerror{\noexpand\caption outside float}\@ehd
7     \expandafter\@gobble
8   \else
9     \refstepcounter\@captype
10    \expandafter\@firstofone
11  \fi
12  {\@dblarg{\@caption\@captype}}%
13 }
```

`\@caption`

```

14 \long\def\@caption#1[#2]#3{%
15   \par
16   \addcontentsline{\csname ext@#1\endcsname}{#1}%
17   {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
18   \begingroup
```

The paragraph setting parameters are normalised at this point, however `\@parboxrestore` resets `\everypar` which is not correct in this context so `\@setminipage` is called if needed.

The float mechanism, like minipage, sets the flag `@minipage` true before executing the user-supplied text. Many L<sup>A</sup>T<sub>E</sub>X constructs test for this flag and do not add vertical space when it is true. The intention is that this emulates T<sub>E</sub>X's 'top of page' behaviour. The flag must be set false at the start of the first paragraph. This is achieved by a redefinition of `\everypar`, but the call to `\@parboxrestore` removes that redefinition, so it is re-inserted if needed. If the flag is already false then the `\caption` was not the first entry in the float, and so some other paragraph has already activated the special `\everypar`. In this case no further action is needed.

```

19   \@parboxrestore
20   \if@minipage
21     \@setminipage
22   \fi
23   \normalsize
24   \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
25 \endgroup
```

`\@float`

`\@dblflset`

```

26 \def\@float#1{%
27   \@ifnextchar[%
28     {\xfloat{#1}}%
29     {\edef\reserved@a{\noexpand\xfloat{#1}[\csname fps@#1\endcsname]}}%
30     \reserved@a}}
```

`\@dblfloat`

```
31 \def\@dblfloat{%
32   \if@twocolumn\let\reserved@a\@dbflt\else\let\reserved@a\@float\fi
33   \reserved@a}
```

`\fps@dbl` Note that all double floats have default fps ‘tp’.

`\@setfps` This sets the fps, dealing with error conditions by adding the default.

`\@xfloat` The first part of this sets the count register that stores all the information about the type and fps of the float.

We assume here that the default specifiers already contain no active characters.

It may be better to store the defaults as numbers, rather than symbol strings.

```
34 \</2kernel>
35 \<latexrelease>\IncludeInRelease{2015/01/01}%
36 \<latexrelease>          {\@xfloat}{Check float options}%
37 \<*2kernel|latexrelease>
38 \def\@xfloat #1[#2]{%
39   \nodocument
40   \def \@captype {#1}%
41   \def \@fps {#2}%
42   \@onelevel@sanitize \@fps
43   \def \reserved@b {}%
44   \ifx \reserved@b \@fps
45     \@fpsadddefault
46   \else
47     \ifx \@fps \@empty
48       \@fpsadddefault
49     \fi
50   \fi
51   \ifhmode
52     \@bsphack
53     \@floatpenalty -\@Mii
54   \else
55     \@floatpenalty-\@Miii
56   \fi
57   \ifinner
58     \@parmoderr\@floatpenalty\z@
59   \else
60     \@next\@currbox\@freelist
61     {%
62       \@tempcnta \sixt@@n
63       \expandafter \@tfor \expandafter \reserved@a
64       \expandafter :\expandafter =\@fps
65       \do
```

Start of changes, use a nested if structure, ending in an error.

```
66       {%
67         \if \reserved@a h%
68           \ifodd \@tempcnta
69             \else
70               \advance \@tempcnta \@ne
71             \fi
72         \else\if \reserved@a t%
```

```

73         \@setfpsbit \tw@
74         \else\if \reserved@a b%
75             \@setfpsbit 4%
76         \else\if \reserved@a p%
77             \@setfpsbit 8%
78         \else\if \reserved@a !%
79             \ifnum \@tempcnta>15
80                 \advance\@tempcnta -\sixt@@n\relax
81             \fi
82         \else
83             \@latex@error{Unknown float option ‘\reserved@a’}%
84             {Option ‘\reserved@a’ ignored and ‘p’ used.}%
85             \@setfpsbit 8%
86         \fi\fi\fi\fi\fi
87     }%

```

End of changes

```

88     \@tempcntb \csname ftype@\@capytype \endcsname
89     \multiply \@tempcntb \@xxxii
90     \advance \@tempcnta \@tempcntb
91     \global \count\@currbox \@tempcnta
92 }%
93 \fltovf
94 \fi

```

The remainder sets up the box in which the float is typeset, and the typesetting environment to be used. It is essential to have the extra box to avoid the unwanted space that would otherwise often be put at the top of the float.

It ends with a hook; not sure how useful this is but it is needed at present to deal with double-column floats.

```

95 \global \setbox\@currbox
96 \color@vbox
97 \normalcolor
98 \vbox \bgroup
99 \hsize\columnwidth
100 \@parboxrestore
101 \floatboxreset
102 }%
103 </2ekernel | latexrelease>
104 <latexrelease>\EndIncludeInRelease
105 <latexrelease>\IncludeInRelease{0000/00/00}%
106 <latexrelease>          {\@xfloat}{Check float options}%
107 <latexrelease>\def\@xfloat #1[#2]{%
108 <latexrelease> \@nodocument
109 <latexrelease> \def \@capytype {#1}%
110 <latexrelease> \def \@fps {#2}%
111 <latexrelease> \@onelevel@sanitize \@fps
112 <latexrelease> \def \reserved@b {!}%
113 <latexrelease> \ifx \reserved@b \@fps
114 <latexrelease>     \fpsadddefault
115 <latexrelease> \else
116 <latexrelease>     \ifx \@fps \@empty
117 <latexrelease>         \fpsadddefault
118 <latexrelease>     \fi
119 <latexrelease> \fi

```

```

120 \latexrelease\ifhmode
121 \latexrelease\@bsphack
122 \latexrelease\@floatpenalty -\@Mii
123 \latexrelease\else
124 \latexrelease\@floatpenalty-\@Miii
125 \latexrelease\fi
126 \latexrelease\ifinner
127 \latexrelease\@parmoderr\@floatpenalty\z@
128 \latexrelease\else
129 \latexrelease\@next\@currbox\@freelist
130 \latexrelease{%
131 \latexrelease\@tempcnta \sixt@n
132 \latexrelease\expandafter \tfor \expandafter \reserved@a
133 \latexrelease\expandafter :\expandafter =\@fps
134 \latexrelease\do
135 \latexrelease{%
136 \latexrelease\if \reserved@a h%
137 \latexrelease\ifodd \@tempcnta
138 \latexrelease\else
139 \latexrelease\advance \@tempcnta \@ne
140 \latexrelease\fi
141 \latexrelease\fi
142 \latexrelease\if \reserved@a t%
143 \latexrelease\@setfpsbit \tw@
144 \latexrelease\fi
145 \latexrelease\if \reserved@a b%
146 \latexrelease\@setfpsbit 4%
147 \latexrelease\fi
148 \latexrelease\if \reserved@a p%
149 \latexrelease\@setfpsbit 8%
150 \latexrelease\fi
151 \latexrelease\if \reserved@a !%
152 \latexrelease\ifnum \@tempcnta>15
153 \latexrelease\advance\@tempcnta -\sixt@n\relax
154 \latexrelease\fi
155 \latexrelease\fi
156 \latexrelease}%
157 \latexrelease\@tempcntb \csname ftype@\@capttype \endcsname
158 \latexrelease\multiply \@tempcntb \@xxxii
159 \latexrelease\advance \@tempcnta \@tempcntb
160 \latexrelease\global \count\@currbox \@tempcnta
161 \latexrelease}%
162 \latexrelease\@fltovf
163 \latexrelease\fi
164 \latexrelease\global \setbox\@currbox
165 \latexrelease\color@vbox
166 \latexrelease\normalcolor
167 \latexrelease\vbox \bgroup
168 \latexrelease\hsize\columnwidth
169 \latexrelease\@parboxrestore
170 \latexrelease\@floatboxreset
171 \latexrelease}%
172 \latexrelease\EndIncludeInRelease
173 \*2kernel)

```

`\@floatboxreset` The rationale for allowing these normally global flags to be set locally here, via `\@parboxrestore`, was stated originally by Donald Arseneau and extended by Chris Rowley. It is because these flags are only set globally to true by section commands, and these should never appear within marginals or floats or, indeed, in any group; and they are only ever set globally to false when they are definitely true.

If anyone is unhappy with this argument then both flags should be treated as in `\set@nobreak`; otherwise this command will be redundant.

```
174 \def \@floatboxreset {%
175     \reset@font
176     \normalsize
177     \@setminipage
178 }
```

`\@setnobreak`

```
179 \def \@setnobreak{%
180     \if@nobreak
181         \let\outer@nobreak\@nobreaktrue
182         \@nobreakfalse
183     \fi
184 }
```

`\@setminipage`

```
185 \def \@setminipage{%
186     \@minipagetrue
187     \everypar{\@minipagefalse\everypar{}}%
188 }
```

`\end@float`

```
189 \def\end@float{%
190     \@endfloatbox
191     \ifnum\@floatpenalty <\z@

192         \@largefloatcheck
193         \@cons\@currlist\@currbox
194         \ifnum\@floatpenalty <-\@Mii
195             \penalty -\@Miv
```

Saving and restoring `\prevdepth` added 26 May 87 to prevent extra vertical space when used in vertical mode.

```
196         \@tempdima\prevdepth
197         \vbox{}%
198         \prevdepth\@tempdima
199         \penalty\@floatpenalty

200     \else
201         \vadjust{\penalty -\@Miv \vbox{}\penalty\@floatpenalty}\@Esphack
202     \fi
203 \fi
204 }
```

`\end@dblfloat`

```
205 </2ekernel>
206 <latexrelease>\IncludeInRelease{2015/01/01}%
207 <latexrelease>          {\end@dblfloat}{float order in 2-column}%
208 <*2ekernel | latexrelease>
209 \def\end@dblfloat{%
210   \if@twocolumn
211     \@endfloatbox
212     \ifnum\@floatpenalty <\z@
213       \@largefloatcheck
214       \global\dp\@currbox1sp %
215       \@cons\@currlist\@currbox
216       \ifnum\@floatpenalty <-\@Mii
217         \penalty -\@Miv
218         \@tempdima\prevdepth
219         \vbox{}%
220         \prevdepth\@tempdima
221         \penalty\@floatpenalty
222       \else
223         \vadjust{\penalty -\@Miv \vbox{}}\penalty\@floatpenalty\@Esphack
224       \fi
225     \fi
226   \else
227     \end@float
228   \fi
229 }%
230 </2ekernel | latexrelease>
231 <latexrelease>\EndIncludeInRelease
232 <latexrelease>\IncludeInRelease{0000/00/00}%
233 <latexrelease>          {\end@dblfloat}{float order in 2-column}%
234 <latexrelease>\def\end@dblfloat{%
235 <latexrelease>\if@twocolumn
236 <latexrelease>  \@endfloatbox
237 <latexrelease>  \ifnum\@floatpenalty <\z@
238   We make sure that we never exceed \textheight, otherwise float will never get
239   typeset (91/03/15 FMi).
240   \@largefloatcheck
241   \@cons\@dbldeferlist\@currbox
242   \fi
243   RmS 92/03/18 changed \@esphack to \@Esphack.
244   \ifnum \@floatpenalty =-\@Mii \@Esphack\fi
245   \else
246     \end@float
247   \fi
248   \fi
249 }%
250 <latexrelease>\EndIncludeInRelease
251 <*2ekernel>
```

`\@endfloatbox` This macro is not intended to be a hook; it is designed to help maintain the integrity of this code, which is used twice and, as can be seen, is subject to frequent changes.

```

248 \def \@endfloatbox{%
249     \par\vskip\z@skip      %% \par\vskip\z@ added 15 Dec 87

250     \@minipagefalse
251     \outer@nobreak
252     \egroup                %% end of vbox
253     \color@endbox
254 }
255 %
256 % \begin{macro}{\outer@nobreak}
257 % \changes{v1.0h}{1994/05/20}{Macro added: default is to do nothing.}
258 % \begin{macrocode}
259 \let\outer@nobreak\@empty

```

`\@largefloatcheck` This calculates by how much a float is oversize for the page and prints this in a warning message.

```

260 \def \@largefloatcheck{%
261     \ifdim \ht\@currbox>\textheight
262         \tempdima -\textheight
263         \advance \tempdima \ht\@currbox

264         \@latex@warning {Float too large for page by \the\tempdima}%
265         \ht\@currbox \textheight
266     \fi
267 }

```

`\@dbflt`

```

\@xdblfloat 268 \def\@dbflt#1{\@ifnextchar[{\@xdblfloat{#1}}{\@xdblfloat{#1}[tp]}}
269 \def\@xdblfloat#1[#2]{%
270     \@xfloat{#1}[#2]\hsize\textwidth\linewidth\textwidth}

```

Moved to ltoutput 93/12/16

```

271 %\newcount\c@topnumber
272 %\newcount\c@dbltopnumber
273 %\newcount\c@bottomnumber
274 %\newcount\c@totalnumber

```

`\@dblfloatplacement` An analysis of `\@floatplacement`:  
This should be called whenever `\@colht` has been set.

```

275 \def\@floatplacement{\global\@topnum\c@topnumber
276     % Textpage bit, global:
277     \global\@toproom \topfraction\@colht
278     \global\@botnum \c@bottomnumber
279     \global\@botroom \bottomfraction\@colht
280     \global\@colnum \c@totalnumber
281     % Floatpage bit, local:
282     \fpmmin \floatpagefraction\@colht}
283 (/2kernel)

```

`\@dblfloatplacement` This should be called only within a group. Now changed to provide extra checks in `\@addtodblcol`, needed when processing a BANG float.

```
284 <latexrelease>\IncludeInRelease{2015/01/01}%
285 <latexrelease>          {\@dblfloatplacement}{float order in 2-column}%
286 <*2kernel | latexrelease>
```

When making two column float area, look for floats with 1sp depth.

```
287 \def\@dblfloatplacement{\global\@dbltopnum\c@dbltopnumber
288   \global\@dbltoproom \dbltopfraction\@colht
289   \@textmin \@colht
290   \advance \@textmin -\@dbltoproom
291   \@fpmin \dblfloatpagefraction\textheight
292   \@fptop \@dblfpptop
293   \@fpsep \@dblfpsep
294   \f@pbot \@dblfpbot
```

`\f@depth` is used in `\@testwrongwidth` to look for either column or dbl-column floats. A value of 1sp signals the latter. Because of this setting here, `\@dblfloatplacement` needs to be called inside a group which is a questionable design.

```
295   \def\f@depth{1sp}}%
296 </2kernel | latexrelease>
297 <latexrelease>\EndIncludeInRelease
298 <latexrelease>\IncludeInRelease{0000/00/00}%
299 <latexrelease>          {\@dblfloatplacement}{float order in 2-column}%
300 <latexrelease>\def \@dblfloatplacement {%
```

Textpage bit: global, but need not be.

```
301 <latexrelease> \global \@dbltopnum \c@dbltopnumber
302 <latexrelease> \global \@dbltoproom \dbltopfraction\@colht
```

This new bit uses `\@textmin` to locally store the amount of extra room in the column.

```
303 <latexrelease> \@textmin \@colht
304 <latexrelease> \advance \@textmin -\@dbltoproom
```

Floatpage bit: must be local.

```
305 <latexrelease> \@fpmin \dblfloatpagefraction\textheight
306 <latexrelease> \@fptop \@dblfpptop
307 <latexrelease> \@fpsep \@dblfpsep
308 <latexrelease> \f@pbot \@dblfpbot
309 <latexrelease>}%
310 <latexrelease>\EndIncludeInRelease
311 <*2kernel>
```

## MARGINAL NOTES:

Marginal notes use the same mechanism as floats to communicate with the `\output` routine. Marginal notes are distinguished from floats by having a negative placement specification. The command `\marginpar [LTEXT]{RTEXT}` generates a marginal note in a parbox, using LTEXT if it's on the left and RTEXT if it's on the right. (Default is RTEXT = LTEXT.) It uses the following parameters.

`\marginparwidth` : Width of marginal notes.  
`\marginparsep` : Distance between marginal note and text.  
the page layout to determine how to move the marginal  
note into the margin. E.g., `\@leftmargin skip ==`  
`\hskip -\marginparwidth \hskip -\marginparsep .`  
`\marginparpush` : Minimum vertical separation between `\marginpar`'s

Marginal notes are normally put on the outside of the page  
if `@mparswitch = true`, and on the right if `@mparswitch = false`.  
The command `\reversemarginpar` reverses the side where they  
are put. `\normalmarginpar` undoes `\reversemarginpar`.  
These commands have no effect for two-column output.

SURPRISE: if two marginal notes appear on the same line of  
text, then the second one could appear on the next page, in  
a funny position.

```

\marginpar [LTEXT]{RTEXT} ==
BEGIN
  if hmode then \bsphack
    \floatpenalty := -10002
  else \floatpenalty := -10003
  fi
  if inner
    then LaTeX Error: 'Not in outer paragraph mode.'
    \floatpenalty := 0
  else if \@freelist has two elements:
    then get \@marbox, \@currbox from \@freelist
    \count\@marbox :=G -1
    else \floatpenalty := 0
    LaTeX Error: 'Too many unprocessed floats'
    \@currbox, \@marbox := \@tempboxa    %%use \def
  fi
  fi
  if optional argument
    then %% \xmpar ==
      \@savemarbox\@marbox{LTEXT}
      \@savemarbox\@currbox{RTEXT}
    else %% \ympar ==
      \@savemarbox\@marbox{RTEXT}
      \box\@currbox :=G \box\@marbox
    fi
  \xympar
END

\reversemarginpar == BEGIN \@parbottom :=G 0
                      @reversemargin :=G true
                      END

```

```

\normalmarginpar == BEGIN \@mparbottom :=G 0
                  @reversemargin :=G false
                  END

```

\marginpar

```

312 \def\marginpar{%
313   \ifhmode
314     \bsphack
315     \@floatpenalty -\@Mii
316   \else
317     \@floatpenalty-\@Miii
318   \fi
319   \ifinner
320     \@parmoderr
321     \@floatpenalty\z@
322   \else
323     \@next\@currbox\@freelist{}\{}%
324     \@next\@marbox\@freelist{\global\count\@marbox\m@ne}%
325     {\@floatpenalty\z@
326       \@fltovf\def\@currbox{\@tempboxa}\def\@marbox{\@tempboxa}}%
327   \fi
328   \@ifnextchar [\@xmpar\@ympar}

```

\@xmpar

```

329 \long\def\@xmpar[#1]#2{%
330   \@savemarbox\@marbox{#1}%
331   \@savemarbox\@currbox{#2}%
332   \@xympar}

```

\@ympar

```

333 \long\def\@ympar#1{%
334   \@savemarbox\@marbox{#1}%
335   \global\setbox\@currbox\copy\@marbox
336   \@xympar}

```

\@savemarbox

```

337 \long\def \@savemarbox #1#2{%
338   \global\setbox #1%
339     \color@vbox
340     \vtop{%
341       \hsize\marginparwidth
342       \@parboxrestore
343       \@marginparreset
344       #2%
345       \@minipagefalse
346       \outer@nobreak
347     }%
348   \color@endbox
349 }

```

\@marginparreset The rationale for allowing these normally global flags to be set locally here, via \@parboxrestore was stated originally by Donald Arsenau and extended by Chris

Rowley. It is because these flags are only set globally to true by section commands, and these should never appear within marginals or floats or, indeed, in any group; and they are only ever set globally to false when they are definitely true.

If anyone is unhappy with this argument then both flags should be treated as in `\set@nobreak`; otherwise this command will be redundant.

```
350 \def \@marginparreset {%
351     \reset@font
352     \normalsize
353 %     \let\if@nobreak\iffalse
354 %     \let\if@noskipsec\iffalse
355 %     \setnobreak
356     \@setminipage
357 }
```

`\@xympar`

Setting the box here is done only because the code uses `\end@float`; it will be empty and gets discarded.

```
358 \def \@xympar{%
359     \ifnum\@floatpenalty <\z@\@cons\@currlist\@marbox\fi
360     \setbox\@tempboxa
361     \color@vbox
362     \vbox \bgroup
363     \end@float
364     \@ignorefalse
365     \@esphack
366 }
```

`\reversemarginpar`

`\normalmarginpar`

```
367 \def\reversemarginpar{\global\@mparbottom\z@ \@reversemargintrue}
368 \def\normalmarginpar{\global\@mparbottom\z@ \@reversemarginfalse}

369 \message{footnotes,}
```

## 60.2 Footnotes

`\footnote{NOTE}` : User command to insert a footnote.

`\footnote[NUM]{NOTE}`: User command to insert a footnote numbered *NUM*, where *NUM* is a number – 1, 2, etc. For example, if footnotes are numbered \*, \*\*, etc. within pages, then `\footnote[2]{...}` produces footnote '\*\*'. This command does not step the footnote counter.

`\footnotemark[NUM]` : Command to produce just the footnote mark in the text, but no footnote. With no argument, it steps the footnote counter before generating the mark.

`\footnotetext[NUM]{TEXT}` : Command to produce the footnote but no mark. `\footnote` is equivalent to

`\footnotemark \footnotetext .`

As in PLAIN, footnotes use `\insert\footins`, and the following parameters:

`\footnotesize` : Size-changing command for footnotes.

`\footnotesep` : The height of a strut placed at the beginning of every footnote.

`\skip\footins` : Space between main text and footnotes. The rule separating footnotes from text occurs in this space. This space lies above the strut of height `\footnotesep` which is at the beginning of the first footnote.

`\footnoterule` : Macro to draw the rule separating footnotes from text. It is executed right after a `\vspace` of `\skip\footins`. It should take zero vertical space—i.e., it should to a negative skip to compensate for any positive space it occupies. (See PLAIN.TEX.)

`\interfootnotelinepenalty` : Interline penalty for footnotes.

`\thefootnote` : In usual LaTeX style, produces the footnote number. If footnotes are to be numbered within pages, then the document style file must include an `\@addtoreset` command to cause the footnote counter to be reset when the page counter is stepped. This is not a good idea, though, because the counter will not always be reset in time to ensure that the first footnote on a page is footnote number one.

`\@thefnmark` : Holds the current footnote's mark—e.g., `\dag` or `'1'` or `'a'`.

`\@mpfnnumber` : A macro that generates the numbers for `\footnote` and `\footnotemark` commands. It == `\thefootnote` outside a minipage environment, but can be changed inside to generate numbers for `\footnote`'s.

`\@makefnmark` : A macro to generate the footnote marker from `\@thefnmark`. The default definition was `\hbox{$^\@thefnmark$}`.

This is now replaced by  
`\textsuperscript{\@thefnmark}`

`\@makefntext{NOTE}` :  
Must produce the actual footnote, using `\@thefnmark` as the mark

of the footnote and NOTE as the text. It is called when effectively inside a `\parbox`, with `\hsize = \columnwidth`.

For example, it might be as simple as

```
$^{\@thefnmark}$ NOTE
```

In a minipage environment, `\footnote` and `\footnotetext` are redefined so that

- (a) they use the counter `mpfootnote`
- (b) the footnotes they produce go at the bottom of the minipage.

The switch is accomplished by letting `\@mpfn == footnote` or `mpfootnote` and `\thempfn == \thefootnote` or `\thempfootnote`, and by redefining `\@footnotetext` to be `\@mpfootnotetext` in the minipage.

```
\footnote{NOTE} ==
BEGIN
  \stepcounter{\@mpfn}
  begingroup
    \protect == \noexpand
    \@thefnmark :=G eval (\thempfn)
  endgroup
  \@footnotemark
  \@footnotetext{NOTE}
END

\footnote[NUM]{NOTE} ==
BEGIN
  begingroup
    \protect == \noexpand
    counter \@mpfn :=L NUM
    \@thefnmark :=G eval (\thempfn)
  endgroup
  \@footnotemark
  \@footnotetext{NOTE}
END

\footnotemark ==
BEGIN \stepcounter{footnote}
  begingroup
    \protect == \noexpand
    \@thefnmark :=G eval(\thefootnote)
  endgroup
  \@footnotemark
END

\footnotemark[NUM] ==
BEGIN
  begingroup
    footnote counter :=L NUM
    \protect == \noexpand
    \@thefnmark :=G eval(\thefootnote)
```

```

        endgroup
        \@footnotemark
    END

\@footnotemark ==
    BEGIN
        \leavevmode
        IF hmode THEN \@x@sf := \the\spacefactor FI
        \@makefnmark          % put number in main text
        IF hmode THEN \spacefactor := \@x@sf FI
    END

\footnotetext      ==
    BEGIN begingroup \protect == \noexpand
        \@thefnmark :=G eval (\thempfn)
    endgroup
    \@footnotetext
    END

\footnotetext[NUM] ==
    BEGIN begingroup counter \@mpfn :=L NUM
        \protect == \noexpand
        \@thefnmark :=G eval (\thempfn)
    endgroup
    \@footnotetext
    END

\footins  LATEX does use the same insert for footnotes as PLAIN.
370 \newinsert\footins
    LATEX leaves these initializations for the \footins insert.
371 \skip\footins=\bigskipamount % space added when footnote is present
372 \count\footins=1000 % footnote magnification factor (1 to 1)
373 \dimen\footins=8in % maximum footnotes per page

\footnoterule  LATEX keeps PLAIN TEX's \footnoterule as the default.
374 \def\footnoterule{\kern-3\p@
375   \hrule \@width 2in \kern 2.6\p@} % the \hrule is .4pt high

\thefootnote
376 \@definecounter{footnote}
377 \def\thefootnote{\@arabic\c@footnote}

\thempfootnote  The default display for the footnote counter in minipages is to use italic letters.
                We use \itshape not \textit as the latter would add an italic correction.
378 \@definecounter{mpfootnote}
379 \def\thempfootnote{{\itshape\@alph\c@mpfootnote}}

\@makefnmark  Default definition.
380 %\def\@makefnmark{\hbox{$^{\@thefnmark}\m@th$}}
381 \def\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}
```

`\textsuperscript` This command provides superscript characters in the current text font. It's implementation might change!!!

```

382 \DeclareRobustCommand*\textsuperscript[1]{%
383   \@textsuperscript{\selectfont#1}}

```

`\@textsuperscript` This command should not be used directly, but may be used to define other commands `\textsuperscript`, `\@makefnmark`. #1 should always start with a font selection command, to activate the font size switch.

```

384 \def\@textsuperscript#1{%
385   {\m@th\ensuremath{^{\mbox{\fontsize\sf@size\z@#1}}}}

```

`\textsubscript`

```

386 (/2ekernel)
387 \latexrelease\IncludeInRelease{2015/01/01}%
388 \latexrelease           {\textsubscript}{\textsubscript}%
389 (*2ekernel | latexrelease)

```

`\DeclareRobustCommand*\textsubscript[1]{%`

```

390   \@textsubscript{\selectfont#1}}%
391

```

`\@textsubscript`

```

392 \def\@textsubscript#1{%
393   {\m@th\ensuremath{_{\mbox{\fontsize\sf@size\z@#1}}}}%

```

`/2ekernel | latexrelease`

```

395 \latexrelease\EndIncludeInRelease
396 \latexrelease\IncludeInRelease{0000/00/00}%
397 \latexrelease           {\textsubscript}{\textsubscript}%
398 \latexrelease\let\textsubscript\@undefined
399 \latexrelease\let\@textsubscript\@undefined
400 \latexrelease\EndIncludeInRelease
401 (*2ekernel)

```

`\def\@textsubscript#1{%`

```

402   {\m@th\ensuremath{_{\mbox{\fontsize\sf@size\z@#1}}}}
403

```

`\footnotesep`

```

404 \newdimen\footnotesep

```

`\footnote`

```

405 \def\footnote{\@ifnextchar[\@xfootnote{\stepcounter\@mpfn
406   \protected@xdef\@thefnmark{\thempfn}%
407   \@footnotemark\@footnotetext}}

```

`\@xfootnote`

```

408 \def\@xfootnote[#1]{%
409   \begingroup
410     \csname c@\@mpfn\endcsname #1\relax
411     \unrestored@protected@xdef\@thefnmark{\thempfn}%
412   \endgroup
413   \@footnotemark\@footnotetext}

```

```

\@footnotetext
414 \long\def\@footnotetext#1{\insert\footins{%
415   \reset@font\footnotesize
416   \interlinepenalty\interfootnotelinepenalty
417   \splittopskip\footnotesep
418   \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
419   \hsize\columnwidth \@parboxrestore
420   \protected@edef\@currentlabel{%
421     \csname p@footnote\endcsname\@thefnmark
422   }%
423   \color@begingroup
424     \@makefnmark{%
425       \rule{z@}{\footnotesep}\ignorespaces#1\@finalstrut\strutbox}%
426   \color@endgroup}}%

\footnotemark
427 \def\footnotemark{%
428   \@ifnextchar[\@xfootnotemark
429     {\stepcounter{footnote}%
430     \protected@xdef\@thefnmark{\thefootnote}%
431     \@footnotemark}}

\@xfootnotemark
432 \def\@xfootnotemark[#1]{%
433   \begingroup
434     \c@footnote #1\relax
435     \unrestored@protected@xdef\@thefnmark{\thefootnote}%
436   \endgroup
437   \@footnotemark}

\@footnotemark
438 \def\@footnotemark{%
439   \leavevmode
440   \ifhmode\edef\x@sf{\the\spacefactor}\nobreak\fi
441   \@makefnmark
442   \ifhmode\spacefactor\x@sf\fi
443   \relax}

\footnotetext
444 \def\footnotetext{%
445   \@ifnextchar [\@xfootnotenext
446     {\protected@xdef\@thefnmark{\thempfn}%
447     \@footnotetext}}

\@xfootnotenext
448 \def\@xfootnotenext[#1]{%
449   \begingroup
450     \csname c@\mpfn\endcsname #1\relax
451     \unrestored@protected@xdef\@thefnmark{\thempfn}%
452   \endgroup
453   \@footnotetext}

```

```

\thempfn
\@mpfn 454 \def\@mpfn{footnote}
        455 \def\thempfn{\thefootnote}
        456 \endkernel

```

# File H

## ltidxglo.dtx

### 61 Index and Glossary Generation

Index and Glossary commands.

```

\makeindex      A preamble command to turn on indexing.
\makeglossary   A preamble command to turn on making glossary entries.
  \index        Make an index entry for #1.
  \glossary     Make a glossary entry for #1.

\makeindex ==
  BEGIN
    \index == BEGIN \@bsphack
              \begingroup
              \protect{X} == \string X\space
              %% added 3 Feb 87 for \index

commands

              %% in \footnotes
              re-\catcode special characters
              to 'other'
              \@wrindex

  END

  \@wrindex{ITEM} ==
    BEGIN
      write of {\indexentry{ITEM}{page number}}
    \endgroup
    \@esphack
  END

INITIALIZATION:

\index == BEGIN \@bsphack
          \begingroup
          re-\catcode special characters (in case '%' there)
          \@index

  END

  \@index{ITEM} == BEGIN \endgroup \@esphack END

Changes made 14 Apr 89 to write \glossaryentry's instead of
\indexentry's on the .glo file.

1 (*2kernel)
2 \message{index,}

\makeindex

3 \def\makeindex{%
4   \newwrite\@indexfile

```

```

5 \immediate\openout\@indexfile=\jobname.idx
6 \def\index{\@bsphack\beginngroup
7     \@sanitize
8     \@wrindex}\typeout
9     {Writing index file \jobname.idx}%

```

Opening the write channel should be done only once since on some OS multiple opens are forbidden and in any case it is useless. So we turn this into a no-op after use.

```

10 \let\makeindex\@empty
11 }
12 \@onlypreamble\makeindex

```

\@wrindex

```

13 \def\@wrindex#1{%
14     \protected@write\@indexfile{%
15         {\string\indexentry{#1}{\thepage}}%
16     \endgroup
17     \@esphack}

```

\index

```

18 \def\index{\@bsphack\beginngroup \@sanitize\@index}

```

\@index

```

19 \def\@index#1{\endgroup\@esphack}

```

\makeglossary

```

20 \def\makeglossary{%
21     \newwrite\@glossaryfile
22     \immediate\openout\@glossaryfile=\jobname.glo
23     \def\glossary{\@bsphack\beginngroup
24         \@sanitize
25         \@wrglossary}\typeout
26         {Writing glossary file \jobname.glo }%

```

Opening the write channel should be done only once since on some OS multiple opens are forbidden and in any case it is useless. So we turn this into a no-op after use.

```

27     \let\makeglossary\@empty
28 }
29 \@onlypreamble\makeglossary

```

\@wrglossary

```

30 \def\@wrglossary#1{%
31     \protected@write\@glossaryfile{%
32         {\string\glossaryentry{#1}{\thepage}}%
33     \endgroup
34     \@esphack}

```

\glossary

```

35 \def\glossary{\@bsphack\beginngroup\@sanitize\@index}
36 \endkernel)

```

# File I

## ltbibl.dtx

### 62 Bibliography Generation

A bibliography is created by the `thebibliography` environment, which generates a title such as “References”, and a list of entries. The `BIBTEX` program will create a file containing such an environment, which will be read in by the `\bibliography` command. With `BIBTEX`, the following commands will be used.

<code>\bibliography</code>	<code>\bibliography{⟨file1,file2, ...,filen⟩}</code> : specifies the bibdata files. Writes a <code>\bibdata</code> entry on the <code>.aux</code> file and tries to read in <code>mainfile.bbl</code> .
<code>\bibliographystyle</code>	<code>\bibliographystyle{⟨style⟩}</code> : Writes a <code>\bibstyle</code> entry on the <code>.aux</code> file.
<code>thebibliography</code>	The <code>thebibliography</code> environment is a list environment. To save the use of an extra counter, it should use <code>enumiv</code> as the item counter. Instead of using <code>\item</code> , items in the bibliography are produced by the following commands: <code>\bibitem{⟨name⟩}</code> : Produces a numbered entry cited as <code>⟨name⟩</code> . <code>\bibitem[⟨label⟩]{⟨name⟩}</code> : Produces an entry labeled by <code>⟨Label⟩</code> and cited by <code>⟨name⟩</code> .

The former is used for bibliographies with citations like [1], [2], etc.; the latter is used for citations like [Knuth82].

The document class must define the `thebibliography` environment. This environment has a single argument, which is the widest bibliography label— e.g., if the [Knuth67] is the widest entry, then this argument will be Knuth67. The `\thebibliography` command must begin a list environment, which the `\endthebibliography` command ends.

<code>\cite</code>	Entries are cited by the command <code>\cite{⟨name⟩}</code> .
<code>\nocite</code>	<code>\nocite{⟨citations⟩}</code> puts information on the <code>.aux</code> file that causes <code>BIBTEX</code> to include the <code>{⟨citations⟩}</code> list in the bibliography, but puts nothing in the text. <code>\nocite{*}</code> is special: it tells <code>BIBTEX</code> to put the whole of a collection of references into the bibliography.

```
1 (*2ekernel)
2 \message{bibliography,}
```

#### PARAMETERS

<code>\@cite</code>	: A macro such that <code>\@cite{LABEL1,LABEL2}{NOTE}</code> produces the output for a <code>\cite[NOTE]{FOO1,FOO2}</code>
command,	where entry <code>FOOi</code> is defined by <code>\bibitem[LABELi]{FOOi}</code> . The switch <code>@tempswa</code> is true if the optional <code>NOTE</code>
argument	is present. The default definition is : <pre>\@cite{LABELS}{NOTE} ==   BEGIN [LABELS     IF @tempswa = T THEN , NOTE FI   ]   END</pre>

`\@biblabel` : A macro to produce the label in the bibliography entry. For `\bibitem[LABEL]{NAME}`, the label is generated by `\@biblabel{LABEL}`. It has the default definition `\@biblabel{LABEL} -> [LABEL]`.

#### CONVENTION

`\b@F00` : The name or number of the reference created by `\cite{FOO}`  
 E.g., if `\cite{FOO} -> [17]` , then `\b@F00 -> 17`.

```

\@bibitem
3 \def\bibitem{\@ifnextchar[\@lbibitem\@bibitem}

\@lbibitem
4 \def\@lbibitem[#1]#2{\item[\@biblabel{#1}\hfill]\if@filesw
5     {\let\protect\noexpand
6       \immediate
7       \write\@auxout{\string\bibcite{#2}{#1}}}\fi\ignorespaces}

\@bibitem
8 \def\@bibitem#1{\item\if@filesw \immediate\write\@auxout
9     {\string\bibcite{#1}{\the\value{\@listctr}}}\fi\ignorespaces}

\bibcite
10 \def\bibcite{\@newl@bel b}

\citation
11 \let\citation\@gobble

\cite
12 \DeclareRobustCommand\cite{%
13   \@ifnextchar [{\@tempswattrue\@citex}{\@tempswafalse\@citex[]}]

\@citex \penalty\@m added to definition of \@citex to allow a line break after the ‘,’ in
citations like [Jones80,Smith77] (Added 23 Oct 86)
      space added after the ‘,’ (21 Nov 87)
14 \def\@citex[#1]#2{\leavevmode
15   \let\@citea\@empty
16   \@cite{\@for\@citeb:=#2\do
17     {\@citea\def\@citea{,\penalty\@m\ }%
18     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
19     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}}\fi

```

Using `\hbox` instead of `\mbox` is fine because of the `\leavevmode` above. In fact the use of a box around the citation contents is more than questionable in my view (FMi), but within 2e I have to keep that for compatibility reasons as it would probably change too many existing documents. Its main reason is to avoid hyphenation of labels such as [FOOB89] into [FOO- B89] so in certain styles it makes sense; but, for example, in author year citations it becomes more than questionable.

So Chris added yet another hook here, as suggested by, at least, Donald Arsena. Note that this one is inside the first argument of the `\@cite` hook. This decouples the top-level typesetting of the citation from the details of the other business conducted here. All this really needs a complete rethink to get the right modularity.

```

20     \@ifundefined{b@\@citeb}{\hbox{\reset@font\bfseries ?}}%
21     \G@refundefinedtrue
22     \@latex@warning
23     {Citation ‘\@citeb’ on page \thepage \space undefined}}%
24     {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}

```

`\bibdata`

```

\bibstyle 25 \let\bibdata=\@gobble
          26 \let\bibstyle=\@gobble

```

`\bibliography`

```

27 \def\bibliography#1{%
28   \if@filesw
29     \immediate\write\@auxout{\string\bibdata{#1}}%
30   \fi
31   \@input{\jobname.bbl}}

```

`\bibliographystyle`

```

32 \def\bibliographystyle#1{%
33   \ifx\@begindocumenthook\@undefined\else
34     \expandafter\AtBeginDocument
35   \fi
36   {\if@filesw
37     \immediate\write\@auxout{\string\bibstyle{#1}}%
38     \fi}}

```

`\nocite` (Added 14 Jun 85)

This puts information on the `.aux` file that causes BibTeX to include the citation list in the bibliography, but puts nothing in the text.

RmS 93/08/06: Made loop for `\nocite` like that for `\@citex`, to get rid of leading spaces.

```

39 \def\nocite#1{\@bsphack

```

With the implementation designed already in L<sup>A</sup>T<sub>E</sub>X 2.09 the `\nocite` command will not work before `\begin{document}` since it tries to write to the `.aux` file which is not open before that point. As a result the “reference” will appear on the terminal and nothing else will happen.

This would be easy to fix, but then a document using the fix will silently fail on an older release of L<sup>A</sup>T<sub>E</sub>X, missing all citations done with `\nocite`. Thus we do only generate an error message and leave the fix for a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> successor.

```

40   \ifx\@onlypreamble\document

```

Since we are after `\begin{document}` we can do the citations:

```

41     \@for\@citeb:=#1\do{%
42       \edef\@citeb{\expandafter\@firstofone\@citeb}%
43       \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
44       \@ifundefined{b@\@citeb}{\G@refundefinedtrue
45         \@latex@warning{Citation ‘\@citeb’ undefined}}{}%
46     \else

```

But before `\begin{document}` we raise an error message:

```
47 \latex@error{Cannot be used in preamble}\@eha
```

Without the compatibility problems we could fix the problem as follows:

```
48 % \AtBeginDocument{\nocite{#1}}
```

```
49 \fi
```

```
50 \esphack}
```

Since `\nocite{*}` should not produce a warning about undefined citation keys (see PR 557), we need to set the control sequence ‘`\b@*`’ to something other than `\relax`. As a result `\cite{*}` will not warn either (but that never worked with  $\text{\LaTeX}$  in the first place).

```
51 \expandafter\let\csname b@*\endcsname\@empty
```

## 62.1 Default definitions

This hook determines the ‘relative formatting’ of the two logical parts of a citation with comment.

`\@cite`

```
52 \def\@cite#1#2{[{#1\if@tempswa , #2\fi}]}
```

`\@cite@ofmt`

This is, in general, a command that appears to have one argument whose value is, in the kernel, a single cs whose name is the expansion of `b@*\@citeb`; the expansion of this cs will typically be some hmode material that produces the detailed typeset form of just the citations themselves.

```
53 \let\@cite@ofmt\hbox
```

`\@biblabel`

```
54 \def\@biblabel#1{[#1]}
```

```
55 \</2ekernel>
```

## File J

# ltpage.dtx

## 63 Page styles and related commands

### 63.1 Page Style Commands

`\pagestyle{<style>}` : sets the page style of the current and succeeding pages to *style*

`\thispagestyle{<style>}` : sets the page style of the current page only to *style*.

To define a page style *style*, you must define `\ps@style` to set the page style parameters.

### 63.2 How a page style makes running heads and feet

The `\ps@...` command defines the macros `\@oddhead`, `\@oddfoot`, `\@evenhead`, and `\@evenfoot` to define the running heads and feet. (See output routine.) To make headings determined by the sectioning commands, the page style defines the commands `\chaptermark`, `\sectionmark`, etc., where `\chaptermark{<text>}` is called by `\chapter` to set a mark. The `\...mark` commands and the `\...head` macros are defined with the help of the following macros.

(All the `\...mark` commands should be initialized to no-ops.)

### 63.3 marking conventions

L<sup>A</sup>T<sub>E</sub>X extends T<sub>E</sub>X's `\mark` facility by producing two kinds of marks a 'left' and a 'right' mark, using the following commands:

`\markboth{<left>}{<right>}` : Adds both marks.

`\markright{<right>}` : Adds a 'right' mark.

`\leftmark` : Used in the output routine, gets the current 'left' mark. Works like T<sub>E</sub>X's `\botmark`.

`\rightmark` : Used in the output routine, gets the current 'right' mark. Works like T<sub>E</sub>X's `\firstmark`. The marking commands work reasonably well for right marks 'numbered within' left marks—e.g., the left mark is changed by a `\chapter` command and the right mark is changed by a `\section` command. However, it does produce somewhat anomalous results if 2 `\markboth`'s occur on the same page.

Commands like `\tableofcontents` that should set the marks in some page styles use a `\mkboth` command, which is `\let` by the `pagestyle` command (`\ps@...`) to `\markboth` for setting the heading or to `\@gobbletwo` to do nothing.

```
1 (*2ekernel)
```

`\pagestyle` User command to set the page style for this and following pages.

```
2 \def\pagestyle#1{%
3   \@ifundefined{ps@#1}%
4     \undefinedpagestyle
5     {\@nameuse{ps@#1}}}
```

`\thispagestyle` User command to set the page style for this page only.

```

6 \def\thispagestyle#1{%
7   \ifundefined{ps@#1}%
8     \undefinedpagestyle
9     {\global\@specialpagetrue\gdef\@specialstyle{#1}}}
```

`\ps@empty` The empty page style: No head or foot line.

```

10 \def\ps@empty{%
11   \let\@mkboth\@gobbletwo\let\@oddhead\@empty\let\@oddfoot\@empty
12   \let\@evenhead\@empty\let\@evenfoot\@empty}
```

`\ps@plain` The plain page style: No head, centred page number in foot.

```

13 \def\ps@plain{\let\@mkboth\@gobbletwo
14   \let\@oddhead\@empty\def\@oddfoot{\reset@font\hfil\thepage
15     \hfil}\let\@evenhead\@empty\let\@evenfoot\@oddfoot}
```

`\@leftmark` We implement `\@leftmark` and `\@rightmark` in terms of already defined commands to save token space. We can't get rid of them since they are sometimes used in applications.

```

16 \let\@leftmark\@firstoftwo
17 \let\@rightmark\@secondoftwo
```

`\markboth` User commands for setting L<sup>A</sup>T<sub>E</sub>X marks.

`\markright` Test for `\@nobreak` added 15 Apr 86 in `\markboth` and `\markright` letting `\label` and `\index` to `\relax` added 22 Feb 86 so these commands can appear in sectioning command arguments RmS 91/06/21 Same for `\glossary`

```

18 \def\markboth#1#2{%
19   \begingroup
20   \let\label\relax \let\index\relax \let\glossary\relax
21   \unrestored@protected@xdef\@themark {{#1}{#2}}%
22   \@temptokena \expandafter{\@themark}%
23   \mark{\the\@temptokena}%
24   \endgroup
25   \if@nobreak\ifvmode\nobreak\fi\fi}
26 \def\markright#1{%
27   \begingroup
28   \let\label\relax \let\index\relax \let\glossary\relax
```

Protection is handled inside `\@markright`.

```

29   \expandafter\@markright\@themark {#1}%
30   \@temptokena \expandafter{\@themark}%
31   \mark{\the\@temptokena}%
32   \endgroup
33   \if@nobreak\ifvmode\nobreak\fi\fi}
```

`\@markright`

```

\leftmark 34 \def\@markright#1#2#3{\@temptokena {#1}%
\rightmark 35   \unrestored@protected@xdef\@themark{{\the\@temptokena}{#3}}}
```

```

36 \def\leftmark{\expandafter\@leftmark\botmark\@empty\@empty}
37 \def\rightmark{\expandafter\@rightmark\firstmark\@empty\@empty}
```

`\@themark` Initialise L<sup>A</sup>T<sub>E</sub>X's marks without setting a T<sub>E</sub>X mark *<whatsit>*.

```

38 \def\@themark{{}{}}
```

`\mark` Test versions of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> initialised T<sub>E</sub>X's `\mark` system at this point, but this was removed before the first release.

```
\AtBeginDocument{\mark{\}{}}
```

`\raggedbottom` `\raggedbottom` typesets pages with no vertical stretch, so they have their natural height instead of all being exactly the same height. (Uses a space of .0001fil to avoid interfering with the 1fil space of `\newpage`.)

```
39 \def\raggedbottom{%
40   \def\@textbottom{\vskip \z@ \@plus.0001fil}\let\@texttop\relax}
```

`\flushbottom` `\flushbottom`: Inverse of `\raggedbottom` — makes all pages the same height.

```
41 \def\flushbottom{%
42   \let\@textbottom\relax \let\@texttop\relax}
```

`\sloppy` `\sloppy` will never (well, hardly ever) produce overfull boxes, but may produce underfull ones. (14 June 85)

```
43 \def\sloppy{%
44   \tolerance 9999%
45   \emergencystretch 3em%
46   \hfuzz .5\p@
47   \vfuzz\hfuzz}
```

`sloppypar` A `sloppypar` environment is equivalent to `{\par \sloppy ... \par}`.

```
48 \def\sloppypar{\par\sloppy}
49 \def\endsloppypar{\par}
```

`\fussy` Resets T<sub>E</sub>X's parameters to their normal finicky values.

```
50 \def\fussy{%
51   \emergencystretch\z@
52   \tolerance 200%
53   \hfuzz .1\p@
54   \vfuzz\hfuzz}
```

`\overfullrule` L<sup>A</sup>T<sub>E</sub>X default is no overfull box rule. Changed by document class option.

```
55 \overfullrule 0pt

56 </2ekernel>
```

# File K

## ltoutput.dtx

### 64 Output Routine

#### 64.1 Floats

The ‘2ekernel’ code ensures that a `\usepackage{autoout1}` is essentially ignored if a ‘full’ format is being used that has the autoload file mode already in the format.

```

1 <defx>\begingroup
2 <defx>\makeatletter
3 <defx>\nfss@catcodes
4 <2ekernel>\expandafter\let\csname ver@autoout1.sty\endcsname\fmtversion
5 <*2ekernel>
6 \message{output,}

*****
*                               *
*                               *
*****

```

#### PAGE LAYOUT PARAMETERS

```

\topmargin      : Extra space added to top of page.
@twoside        : boolean.  T if two-sided printing
\oddsidemargin  : IF @twoside = T
                  THEN extra space added to left of odd-numbered
                  pages.
                  ELSE extra space added to left of all pages.
\evensidemargin : IF @twoside = T
                  THEN extra space added to left of
even-numbered
                  pages.
\headheight     : height of head
\headsep        : separation between head and text
\footskip       : distance separation between baseline of last
                  line of text and baseline of foot.
                  Note difference between \footSKIP and \headSEP.
\textheight     : height of text on page, excluding head and foot
\textwidth      : width of printing on page
\columnsep      : IF @twocolumn = T
                  THEN width of space between columns
\columnseprule  : IF @twocolumn = T
                  THEN width of rule between columns (0 if none).
\columnwidth    : IF @twocolumn = T
                  THEN (\textwidth - \columnsep)/2
                  ELSE \textwidth
                  It is set by the \twocolumn and

```

`\onecolumn` commands.

`\@textbottom` : Command executed at bottom of vbox holding text of page (including figures). The `\raggedbottom` command almost `\let`'s this to `\vfil` (actually sets it to `\vskip \z@ plus.0001fil`). Should have depth 0pt.

`\@texttop` : Command executed at top of vbox holding text of page (including figures). Used by letter style; can also be used to produce centered pages. Let to `\relax` by `\raggedbottom` and `\flushbottom`.

Page layout must initialize `\@colht` and `\@colroom` to `\textheight`.

#### PAGE STYLE PARAMETERS:

`\floatsep` : Space left between floats.

`\textfloatsep` : Space between last top float or first bottom float and the text.

`\topfigrule` : Command to place rule (or whatever) between floats at top of page and text. Executed in inner vertical mode right before the `\textfloatsep` skip separating the floats from the text. Must occupy zero vertical space. (See `\footnoterule`.)

`\botfigrule` : Same as `\topfigrule`, but put after the `\textfloatsep` skip separating text from the floats at bottom of page.

`\intextsep` : Space left on top and bottom of an in-text float.

`\dblfloatsep` : Space between double-column floats.

`\dbltextfloatsep` : Space between top double-column floats and text.

`\dblfigrule` : Similar to `\topfigrule`, but for double-column floats.

`\@fptop` : Glue to go at top of float column – must be 0pt + stretch

`\@fpsep` : Glue to go between floats in a float column.

`\@fpbot` : Glue to go at bottom of float column – must be 0pt + stretch

`\@dblfpptop`, `\@dblfpsep`, `\@dblfpbot` : Analogous for double-column float page in two-column format.

FOOTNOTES: As in PLAIN, footnotes use `\insert\footins`.

#### PAGE LAYOUT SWITCHES AND MACROS

`@twocolumn` : Boolean. T if two columns per page globally.

## PAGE STYLE MACROS AND SWITCHES

```

\@oddhead      : IF @twoside = T
                  THEN macro to generate head of
odd-numbered
                  pages.
                  ELSE macro to generate head of all pages.
\@evenhead     : IF @twoside = T
                  THEN macro to generate head of
even-numbered
                  pages.
\@oddfoot      : IF @twoside = T
                  THEN macro to generate foot of
odd-numbered
                  pages.
                  ELSE macro to generate foot of all pages.
\@evenfoot     : IF @twoside = T
                  THEN macro to generate foot of
even-numbered
                  pages.
@specialpage   : boolean. T if current page is to have a special
                  format.
\@specialstyle : If its value is foo then
                  IF @specialpage = T
                  THEN the command \ps@foo is executed to
                  temporarily reset the page style parameters
                  before composing the current page.
                  This command should execute only \def's
and
                  \edef's, making only local definitions.

```

## FLOAT PLACEMENT PARAMETERS

The following parameters are set by the macro `\@floatplacement`.  
When `\@floatplacement` is called,

`\@colht` is the height of the page or column being built. I.e.:

- \* For single-column page it equals `\textheight`.
- \* For double-column page it equals `\textheight - height`  
of double-column floats on page.

Note that some are set globally and some locally:

```

\@topnum :=G Maximum number of floats allowed on the top of a
column.
\@toproom :=G Maximum amount of top of column devoted to floats—
excluding \textfloatsep separation below the floats
and \floatsep separation between them. For
two-column output, should be computed as a function
of \@colht.
\@botnum, \@botroom
: Analogous to above.

```

`\@colnum` :=G Maximum number of floats allowed in a column, including in-text floats.

`\@textmin` :=L Minimum amount of text (excluding footnotes) that must appear on a text page.

%% 27 Sep 85 : made local to  
%% `\@addtocurcol` and `\@addtonextcol`

It is now also used locally in processing double floats.

`\@fpmin` :=L Minimum height of floats in a float column.

The macro `\@dblfloatplacement` sets the following parameters.

`\@dbltopnum` :=G Maximum number of double-column floats allowed at the top of a two-column page.

`\@dbltoproom` :=G Maximum height of double-column floats allowed at top of two-column page.

`\@fpmin` :=L Minimum height of floats in a float column.

It should also perform the following local assignments where necessary – i.e., where the new value differs from the old one:

`\@fptop` :=L `\@dblfpptop`  
`\@fpsep` :=L `\@dblfpsep`  
`\@fpbot` :=L `\@dblfpbot`

## OUTPUT ROUTINE VARIABLES

`\@colht` : The total height of the current column. In single column style, it equals `\textheight`. In two-column style, it is `\textheight` minus the height of the double-column floats on the current page. MUST BE INITIALIZED TO `\textheight`.

`\@colroom` : The height available in the current column for text and footnotes. It equals `\@colht` minus the height of all floats committed to the top and bottom of the current column.

`\@textfloatsheight` : The total height of in-text floats on the current page.

`\@footins` : Footnote insertion number.

`\@maxdepth` : Saved value of TeX's `\maxdepth`. Must be set when any routine sets `\maxdepth`.

## CALLING THE OUTPUT ROUTINE

---

The output routine is called either by TeX's normal page-breaking mechanism, or by a macro putting a penalty  $<$  or  $= -10000$  in the output list. In the latter case, the penalty indicates why the output

routine was called, using the following code.

penalty	reason
-10000	<code>\pagebreak</code> <code>\newpage</code>
-10001	<code>\clearpage</code> ( <code>\penalty -10000 \vbox{} \penalty -10001</code> )
-10002	float insertion, called from horizontal mode
-10003	float insertion, called from vertical mode.
-10004	float insertion.

Note: A float or marginpar puts the following sequence in the output list:

- (i) a penalty of -10004,
- (ii) a null `\vbox`
- (iii) a penalty of -10002 or -10003.

This solves two special problems:

1. If the float comes right after a `\newpage` or `\clearpage`, then the first penalty is ignored, but the second one invokes the output routine.
2. If there is a split footnote on the page, the second 'page' puts out the rest of the footnote.

## THE OUTPUT ROUTINE

### FUNCTIONS USED IN THE OUTPUT ROUTINE:

`\@outputpage` : Produces an output page with the contents of box `\@outputbox` as the text part.

Also sets `\@colht :=G \textheight`.

The page style is determined as follows.

IF `@thispagestyle = true`

THEN use `\thispagestyle` style

ELSE use ordinary page style.

`\@tryfcolumn\FLIST` : Tries to form a float column composed of floats from `\FLIST` (if nonempty) with the following parameters:

`\@colht` : height of box

`\@fpmin` : minimum height of floats in the box

`\@fpsep` : interfloat space

`\@fptop` : glue at top of box

`\@fpbot` : glue at bottom of box.

If it succeeds, then it does the following:

\* `\@outputbox :=L` the composed float box.

\* `@fcolmade :=G true`

\* `\FLIST :=G \FLIST` - floats put in box

\* `\@freelist :=G \@freelist +` floats put in box

If it fails, then:

\* `@fcolmade :=G false`

NOTE: BIT MUST BE A SINGLE TOKEN!

`\@makefcolumn \FLIST` : Same as `\@tryfcolumn` except that it fails to make a float column only if `\FLIST` is empty. Otherwise, it makes a float column containing at least the first box in `\FLIST`, disregarding `\@fpmin`.

`\@startcolumn` :  
 Calls `\@tryfcolumn\@deferlist`. If `\@tryfcolumn` returns with (globally set) `@fcolmade = false`, then:

- \* Globally sets `\@toplist` and `\@botlist` to floats from `\@deferlist` to go at top and bottom of column, deleting them from `\@deferlist`. It does this using `\@colht` as the total height, the page style parameters `\@floatsep` and `\@textfloatsep`, and the float placement parameters `\@topnum`, `\@toproom`, `\@botnum`, `\@botroom`, `\@colnum` and `\textfraction`.
- \* Globally sets `\@colroom` to `\@colht` minus the height of the added floats.

`\@startdblcolumn` :  
 Calls `\@tryfcolumn\@dbldeferlist{8}`. If `\@tryfcolumn` returns with (globally set) `@fcolmade = false`, then:

- \* Globally sets `\@dbltoplist` to floats from `\@dbldeferlist` to go at top and bottom of column, deleting them from `\@dbldeferlist`. It does this using `\textheight` as the total height, and the parameters `\@dblfloatsep`, etc.
- \* Globally sets `\@colht` to `\textheight` minus the height of the added floats.

`\@combinefloats` : Combines the text from box `\@outputbox` with the floats from `\@toplist` and `\@botlist`, putting the new box in `\@outputbox`. It uses `\floatsep` and `\textfloatsep` for the appropriate separations. It puts the elements of `\TOPLIST` and `\BOTLIST` onto `\@freelist`, and makes those lists null.

`\@makecol` : Makes the contents of `\box255` plus the accumulated footnotes, plus the floats in `\@toplist` and `\@botlist`, into a single column of height `\@colht` (unless the page height has been locally changed), which it puts into box `\@outputbox`. It puts boxes in `\@midlist` back onto `\@freelist` and restores `\maxdepth`.

`\@opcol` : Outputs a column whose text is in box `\@outputbox`. If `@twocolumn = false`, then it calls `\@outputpage`, sets `\@colht := G \textheight`, and calls `\@floatplacement`.

If @twocolumn = true, then:

If @firstcolumn = true, then it puts box \@outputbox into \@leftcolumn and sets @firstcolumn :=G false.

If @firstcolumn = false, then it puts out the current two-column page, any possible two-column float pages, and determines \@dbltoplist for the next page.

## USER COMMANDS THAT CALL OR AFFECT THE OUTPUT ROUTINE

---

```
\newpage == BEGIN \par\vfil\penalty -10000 END
```

```
\clearpage == BEGIN \newpage
                  \write -1{}      % Part of hack to make sure no
                  \vbox{}          % \write's get lost.
                  \penalty -10001
                END
```

```
\cleardoublepage == BEGIN \clearpage
                        if @twoside = true and c@page is even
                        then \hbox{} \newpage fi
                END
```

`\twocolumn[BOX]` : starts a new page, changing to twocolumn setting and puts BOX in a parbox of width `\textwidth` across the top. Useful for full-width titles for double-column pages.

SURPRISE: The stretch from `\@dbltextfloatsep` will be inserted between the BOX and the top of the two columns.

## FLOAT-HANDLING MECHANISMS

---

The float environment obtains an insertion number B from the `\@freelist` (see below for a description of list manipulation), puts the float into box B and sets `\count B` to a FLOAT SPECIFIER. For a normal (not double-column) float, it then causes a page break in one of the following two ways:

- In outer hmode: `\adjust{\penalty -10002}`
- In vmode : `\penalty -10003`.

For a double-column float, it puts B onto the `\@dbldeferlist`.

The float specifier has two components:

- \* A PLACEMENT SPECIFICATION, describing where the float may be placed.
- \* A TYPE, which is a power of two—e.g., figures might be

type 1 floats, tables type 2 floats, programs type 4 floats, etc.  
The float specifier is encoded as follows, where bit 0 is the least significant bit.

Bit	Meaning
0	1 iff the float may go where it appears in the text.
1	1 iff the float may go on the top of a page.
2	1 iff the float may go on the bottom of a page.
3	1 iff the float may go on a float page.
4	1 unless the PLACEMENT includes a !
5	1 iff a type 1 float
6	1 iff a type 2 float
etc.	

A negative float specifier is used to indicate a marginal note.

## MACROS AND DATA STRUCTURES FOR PROCESSING FLOATS

A FLOAT LIST consisting of the floats in boxes `\boxa ... \boxN` has the form:

```
\@elt \boxa ... \@elt \boxN
```

where `\boxI` is defined by

```
\newinsert\boxI
```

Normally, `\@elt` is `\let` to `\relax`. A test can be performed on the entire float list by locally `\def`'ing `\@elt` appropriately and executing the list.

This is a lot more efficient than looping through the list.

The following macros are used for manipulating float lists.

```
\@next \CS \LIST {NONEMPTY}{EMPTY} == %% NOTE: ASSUME
\@elt = \relax
  BEGIN  assume that \LIST == \@elt \B1 ... \@elt \Bn
        if n = 0
          then EMPTY
        else \CS      :=L \B1
              \LIST :=G \@elt \B2 ... \@elt \Bn
              NONEMPTY
        fi
  END
```

`\@bitor\NUM\LIST` : Globally sets switch `@test` to the disjunction for all I of bit `log2 \NUM` of the float specifiers of all the floats in `\LIST`.

I.e., `@test` is set to true iff there is at least one float in `\LIST` having bit `log2 \NUM` of its float specifier equal to 1.

Note:  $\log_2 [(\backslash\text{count I})/32]$  is the bit number corresponding to the type of float I. To see if there is any float in `\LIST` having the same type as float I, you run `\@bitor` with

$$\backslash\text{NUM} = [(\backslash\text{count I})/32] * 32.$$

```
\@bitor\NUM\LIST ==
BEGIN
  @test :=G false
  { \@elt \CTR == if \NUM <> 0 then
                    if \count\CTR / \NUM is odd
                    then @test := true          fi fi
    \LIST
  }
END
```

`\@cons\LIST\NUM` : Globally sets `\LIST := \LIST * \@elt \NUM`

```
\@cons\LIST\NUM ==
BEGIN { \@elt == \relax
        \LIST :=G \LIST \@elt \NUM
      }
```

#### BOX LISTS FOR FLOAT-PLACEMENT ALGORITHMS

```
\@freelist      : List of empty boxes for placing new floats.
\@toplist       : List of floats to go at top of current column.
\@midlist       : List of floats in middle of current column.
\@botlist       : List of floats to go at bottom of current column.
\@deferlist     : List of floats to go after current column.
\@dbltoplist    : List of double-col. floats to go at top of current
                  page.
\@dbldeferlist  : List of double-column floats to go on subsequent
                  pages.
```

#### FLOAT-PLACEMENT ALGORITHMS

`\@addtobot` : Tries to put insert `\@currbox` on `\@botlist`.

Called only when:

- \* `\ht BOX < \@colroom`
- \* type of `\@currbox` not on `\@deferlist`
- \* `\@colnum > 0`
- \* `@insert = false`

If it succeeds, then:

- \* sets `@insert true`
- \* decrements `\@botroom` by `\ht BOX`
- \* decrements `\@botnum` and `\@colnum` by 1

```

* decrements \@colroom by \ht BOX + either
\floatsep
    or \textfloatsep, as appropriate.
* sets \maxdepth to 0pt

\@addtotoporbot : Tries to put insert \@currbox on \@toplist or
                  \@botlist.
                  Called only under same conditions as \@addtobot.
                  If it succeeds, then:
                    * sets @insert true
                    * decrements \@toproom or \@botroom by \ht
BOX
                    * decrements \@colnum and either \@topnum or
                      \@botnum by 1
                    * decrements \@colroom by \ht BOX +
\floatsep
    or \textfloatsep, as appropriate.

\@addtocurcol : Tries to add \@currbox to current column, setting
                @insert true if it succeeds, false otherwise.
                It will add \@currbox to top only if bit 0 of
                \count \@currbox is 0, and to the bottom only if
                bit 0 = 0 or an earlier float of the same type is
                put on the bottom.
                If the float is put in the text, then
                \penalty\interlinepenalty is put
                right after the float, before the following \vskip,
                and \outputpenalty :=L 0.

\@addtonextcol : Tries to add \@currbox to the next column, setting
                @insert true if it succeeds, false otherwise.

\@addtodblcol : Tries to add \@currbox to the next double-column page,
                adding it to \@dbltoplist if it succeeds and
                \@dbldeferlist if it fails.

\@addmarginpar ==
BEGIN
  if \@currlist nonempty
    then remove \@marbox from \@currlist
      add \@marbox and \@currbox to \@freelist
      %% NOTE: \@currbox = left box
    else LaTeX error: ? %% shouldn't happen
  fi
  \@tempcnta := 1      %% 1 = right, -1 = left
  if @twocolumn = true
    then if @firstcolumn = true
      then \@tempcnta := -1
    fi

```

```

else if @mparswitch = true
  then if count0 odd
    else \@tempcnta := -1
    fi
  fi
  if @reversemargin = true
    then \@tempcnta := -\@tempcnta
    fi
  fi
if \@tempcnta < 0 then \box\@marbox :=G \box\@currbox
fi
\@tempdima :=L maximum(\@mparbottom - \@pageht
                        + ht of \@marbox, 0)
if \@tempdima > 0 then LaTeX warning: 'marginpar moved' fi
\@mparbottom :=G \@pageht + \@tempdima + depth of \@marbox
                + \marginparpush
\@tempdima :=L \@tempdima - ht of \@marbox
\box\@marbox :=G \box\@currbox
                    \vbox { \vskip \@tempdima
                            \box\@marbox
                            }
height of \@marbox :=G depth of \@marbox :=G 0
\kern -\@pagedp
\nointerlineskip
\hbox{ if @tempcnta > 0 then \hskip \columnwidth
                    \hskip \marginparsep
                    else \hskip -\marginparsep
                    \hskip -\marginparwidth
                    fi
        \box\@marbox \hss
    }
\nobreak
\nointerlineskip
\hbox{\vrule height 0 width 0 depth \@pagedp}
END

```

Floats and marginpars add a lot of dead cycles.

```

7 \maxdeadcycles = 100

8 \let\@elt\relax

9 \def\@next#1#2#3#4{\ifx#2\@empty #4\else
10 \expandafter\@xnext #2\@#1#2#3\fi}

11 \def\@xnext \@elt #1#2\@#3#4{\def#3{#1}\gdef#4{#2}}

\changes{v1.1v}{1996/07/26}{put \cs{global} into definition}

12 \def\@testfalse{\global\let@if@test\iffalse}
13 \def\@testtrue {\global\let@if@test\iftrue}
14 \@testfalse

```

\changes{v1.1v}{1996/07/26}{remove \cs{global} before \cs{@test...}}

```
15 \def\@bitor#1#2{\@testfalse {\let\@elt\@xbitor
16   \@tempcnta #1\relax #2}}
```

RmS 91/11/22: Added test for |\count#1 = 0|.  
Suggested by Chris Rowley.

\changes{v1.1v}{1996/07/26}{remove \cs{global} before \cs{@test...}}

```
17 \def\@xbitor #1{\@tempcntb \count#1
18   \ifnum \@tempcnta =\z@
19     \else
20       \divide\@tempcntb\@tempcnta
21       \ifodd\@tempcntb \@testtrue\fi
22     \fi}
```

DEFINITION OF FLOAT BOXES:

\changes{v1.3a}{2015/09/205}  
{extended \cs{@freelist}}

```
23 \</2kernel>
24 \<latexrelease>\IncludeInRelease{2015/10/01}%
25 \<latexrelease>          {\bx@ZZ}{Extended float list}%
26 \<*2kernel | latexrelease>
27 \let\@elt\newinsert
28 \<*2kernel>
29 \def\@freelist{%
30   \@elt\bx@A\@elt\bx@B\@elt\bx@C\@elt\bx@D\@elt\bx@E
31   \@elt\bx@F\@elt\bx@G\@elt\bx@H\@elt\bx@I\@elt\bx@J
32   \@elt\bx@K\@elt\bx@L\@elt\bx@M\@elt\bx@N
33   \@elt\bx@O\@elt\bx@P\@elt\bx@Q\@elt\bx@R}
34 \@freelist
35 \</2kernel>
36 \ifx\numexpr\@undefined\else
37 \def\reserved@a{%
38   \@elt\bx@S\@elt\bx@T\@elt\bx@U\@elt\bx@V
39   \@elt\bx@W\@elt\bx@X\@elt\bx@Y\@elt\bx@Z
40   \@elt\bx@AA\@elt\bx@BB\@elt\bx@CC\@elt\bx@DD\@elt\bx@EE
41   \@elt\bx@FF\@elt\bx@GG\@elt\bx@HH\@elt\bx@II\@elt\bx@JJ
42   \@elt\bx@KK\@elt\bx@LL\@elt\bx@MM\@elt\bx@NN
43   \@elt\bx@OO\@elt\bx@PP\@elt\bx@QQ\@elt\bx@RR
44   \@elt\bx@SS\@elt\bx@TT\@elt\bx@UU\@elt\bx@VV
45   \@elt\bx@WW\@elt\bx@XX\@elt\bx@YY\@elt\bx@ZZ}
46 \reserved@a
47 \def\@elt{\noexpand\@elt\noexpand}
48 \edef\@freelist{\@freelist\reserved@a}
49 \fi
50 \let\reserved@a\relax
51 \let\@elt\relax
52 \</2kernel | latexrelease>
53 \<latexrelease>\EndIncludeInRelease
54 \<latexrelease>\IncludeInRelease{0000/00/00}%
55 \<latexrelease>          {\bx@ZZ}{Extended float list}%
56 \<latexrelease>\def\@freelist{%
```

```

57 \latexrelease \elt\bx@A\elt\bx@B\elt\bx@C\elt\bx@D\elt\bx@E
58 \latexrelease \elt\bx@F\elt\bx@G\elt\bx@H\elt\bx@I\elt\bx@J
59 \latexrelease \elt\bx@K\elt\bx@L\elt\bx@M\elt\bx@N
60 \latexrelease \elt\bx@O\elt\bx@P\elt\bx@Q\elt\bx@R}
61 \latexrelease \insecunt=234
62 \latexrelease\EndIncludeInRelease
63 {*2kernel}

64 \gdef\@toplist{}
65 \gdef\@botlist{}
66 \gdef\@midlist{}
67 \gdef\@currlist{}
68 \gdef\@deferlist{}
69 \gdef\@dblistoplist{}
70 % \begin{macrocode}
71 % \changes{v1.2m}{2015/03/12}
72 % {initialise \cs{@dbldeferlist} again}
73 % The new algorithm stores page wide floats together with column floats
74 % in a single |\@deferlist| list. We keep |\@dbldeferlist|
75 % initialised as empty so that packages that are testing for
76 % deferred floats can use the same code for old or new float
77 % handling.

```

```

\gdef\@dbldeferlist{}
\end{macrocode}

```

## PAGE LAYOUT PARAMETERS

```

78 \newdimen\topmargin
79 \newdimen\oddsidemargin
80 \newdimen\evensidemargin
81 \let\@themargin=\oddsidemargin
82 \newdimen\headheight
83 \newdimen\headsep
84 \newdimen\footskip
85 \newdimen\textheight
86 \newdimen\textwidth
87 \newdimen\columnwidth
88 \newdimen\columnsep
89 \newdimen\columnseprule
90 \newdimen\marginparwidth
91 \newdimen\marginparsep
92 \newdimen\marginparpush

```

\AtBeginDvi We use a box register in which to put stuff that must appear before anything else in the .dvi file.

The stuff in the box should not add any typeset material to the page when it is unboxed.

```

93 \newbox\@begindvibox
94 \def \AtBeginDvi #1{%
95   \global \setbox \@begindvibox
96   \vbox{\unvbox \@begindvibox #1}%
97 }

```

```

\@maxdepth This is not the right place to set this; it needs to be set in a class/style file when
\maxdepth is set.
    Also, many settings to \maxdepth should be to \@maxdepth, probably?
98 \newdimen\@maxdepth
99 \@maxdepth = \maxdepth

\paperheight New \paper... registers.
\paperwidth 100 \newdimen\paperheight
101 \newdimen\paperwidth

\if@insert Local switches first:
\if@fcolmade 102 \newif \if@insert
\if@specialpage These should definitely be global:
\if@firstcolumn 103 \newif \if@fcolmade
\if@twocolumn 104 \newif \if@specialpage \@specialpagefalse
\if@twoside
\if@reversemarginpar These should be global but are not always set globally in other files.
\if@mparswitch 105 \newif \if@firstcolumn \@firstcolumntrue
\col@number 106 \newif \if@twocolumn \@twocolumnfalse

Not sure about these: two questions. Should things which must apply to a whole
document be local or global (they probably should be ‘preamble only’ commands)?
Are these three such things?
107 \newif \if@twoside \@twosidefalse
108 \newif \if@reversemargin \@reversemarginfalse
109 \newif \if@mparswitch \@mparswitchfalse

This counter has been imported from ‘multicol’.
110 \newcount \col@number
111 \col@number \@ne

INTERNAL REGISTERS

112 \newcount\@topnum
113 \newdimen\@toproom
114 \newcount\@dbltopnum
115 \newdimen\@dbltoproom
116 \newcount\@botnum
117 \newdimen\@botroom
118 \newcount\@colnum
119 \newdimen\@textmin
120 \newdimen\@fpmin
121 \newdimen\@colht
122 \newdimen\@colroom
123 \newdimen\@pageht
124 \newdimen\@pagedp
125 \newdimen\@mparbottom \@mparbottom\z@
126 \newcount\@currtype
127 \newbox\@outputbox
128 \newbox\@leftcolumn
129 \newbox\@holdpg

130 \def\@thehead{\@oddhead} % initialization
131 \def\@thefoot{\@oddfoot}

```

`\clearpage` The tests at the beginning are an experimental attempt to avoid a completely empty page after a `\twocolumn[...]`. This prevents the text from the argument vanishing into a float box, never to be seen again. We hope that it does not produce wrong formatting in other cases.

```

132 \def\clearpage{%
133   \ifvmode
134     \ifnum \@dbltopnum =\m@ne
135       \ifdim \pagetotal <\topskip
136         \hbox{}%
137       \fi
138     \fi
139   \fi
140   \newpage
141   \write\m@ne{}%
142   \vbox{}%
143   \penalty -\@Mi
144 }
```

`\cleardoublepage`

```

145 \def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
146   \hbox{}\newpage\if@twocolumn\hbox{}\newpage\fi\fi\fi}
147 \endkernel
```

`\onecolumn`

```

148 (*2ekernel | fltrace)
149 \def\onecolumn{%
150   \clearpage
151   \global\columnwidth\textwidth
152   \global\hsize\columnwidth
153   \global\linewidth\columnwidth
154   \global\@twocolumnfalse
155   \col@number \@one
156   \@floatplacement}
```

`\newpage` The two checks at the beginning ensure that an item label or run-in section title immediately before a `\newpage` get printed on the correct page, the one before the page break.

All three tests are largely to make error processing more robust; that is why they all reset the flags explicitly, even when it would appear that this would be done by a `\leavevmode`.

```

157 \def \newpage {%
158   \if@noskipsec
159     \ifx \@nodocument\relax
160       \leavevmode
161       \global \@noskipsecfalse
162     \fi
163   \fi
164   \if@inlabel
165     \leavevmode
166     \global \@inlabelfalse
167   \fi
168   \if@nobreak \@nobreakfalse \everypar{}\fi
169   \par
```

```

170 \vfil
171 \penalty -\@M}

\@emptycol It may be better to use an invisible rule rather than an empty box here.
172 \def \@emptycol {\vbox{}}\penalty -\@M}

\twocolumn There are several bug fixes to the two-column stuff here.
\@topnewpage
173 \def \twocolumn {%
174 \clearpage
175 \global\columnwidth\textwidth
176 \global\advance\columnwidth-\columnsep
177 \global\divide\columnwidth\tw@
178 \global\hsize\columnwidth
179 \global\linewidth\columnwidth
180 \global\@twocolumntrue
181 \global\@firstcolumntrue
182 \col@number \tw@

There is no reason to put a \@dblfloatplacement here since \@topnewpage ig-
nores these settings. The \@floatplacement is needed in case this comes after
some changes.
183 \@ifnextchar [\@topnewpage\@floatplacement
184 }

Note that here, getting a box from the freelist can assume success since this
comes just after a \clearpage.
185 \long\def \@topnewpage [#1]{%
186 \@nodocument
187 \@next\@currbox\@freelist{ }{%
188 \global \setbox\@currbox
189 \color@vbox
190 \normalcolor
191 \vbox {%
192 \hsize\textwidth
193 \@parboxrestore
194 \col@number \@ne
195 #1%
196 \vskip -\dbltextfloatsep
197 }%
198 \color@endbox

Added size test and warning message; perhaps we should use an error message.
199 \ifdim \ht\@currbox>\textheight
200 \ht\@currbox \textheight
201 \fi

This next line is not essential but it is more robust to make this value non-zero,
in case of weird errors.

This next bit is what is needed from \@addtodblcol, plus some extra checks
for error trapping.
202 \global \count\@currbox \tw@
203 \@tempdima -\ht\@currbox
204 \advance \@tempdima -\dbltextfloatsep
205 \global \advance \colht \@tempdima
206 \ifx \@dbltoplist \@empty

```

```

207 \else
208 \latexerr{Float(s) lost}\@ehb
209 \let \@dbltoplist \@empty
210 \fi
211 \@cons \@dbltoplist \@currbox

```

This setting of \@dbltopnum is used only to change the typesetting in \@combinedblfloats.

```

212 \global \@dbltopnum \m@ne
213 (*trace)
214 \fl@trace{dbltopnum set to -1 (= \the \@dbltopnum) (topnewpage)}%
215 /trace)

```

At points such as this we need to check that there is still a minimal amount of room left on the page; this uses an arbitrary small value at present; but note that this value is larger than that used when checking that page is too full of normal floats.

If there is little room left we just force a page-break, OK? This involves producing two empty columns. The second empty column may be produced by \output, in which case an extra, misleading, warning will be generated, OK? (This happens only when there is too little room left on the page for any float.) Otherwise (i.e. if the size is such that it is allowed as a normal float) the extra \@emptycol will be invoked in the second column by the conditional code guarded by the \if@firstcolumn test.

I now think that the cut-off point here should be 3\baselineskip, but we make it a bit less so that 3 lines of text will be allowed, OK?

Since this happens only when there is nothing on the page but the ‘top-box’, the empty box should not cause any problem other than some overfull box messages, which is not entirely misleading.

Here we need two page-ends since both columns need to be empty.

```

216 \ifdim \@colht<2.5\baselineskip
217 \latex@warning@no@line {Optional argument of \noexpand\twocolumn
218 too tall on page \thepage}%
219 \@emptycol
220 \if@firstcolumn
221 \else
222 \@emptycol
223 \fi
224 \else
225 \global \vsize \@colht
226 \global \@colroom \@colht
227 \@floatplacement
228 \fi
229 }

```

\output This needs some small adjustments. We cannot guarantee that the float mechanism will interact correctly with this stuff, but that mechanism does not always work properly with footnotes already.

RmS 91/09/29:

added reset of \par to the output routine. This avoids problems when the output routine is called within a list where \par may be a no-op.

```

230 \output {%
231 \let \par \@@par

```

```

232 \ifnum \outputpenalty<-\@M
233   \@specialoutput
234 \else
235   \@makecol
236   \@opcol

```

Moved to \@opcol: \@floatplacement.

```

237   \@startcolumn

```

This loop could be replaced by an \expandafter tail recursion in \@startcolumn.

```

238   \@whilesw \if@fcolmade \fi
239   {%
240 (*trace)
241   \fl@trace{PAGE: float \if@twocolumn column \else page \fi
242               completed}%
243 (/trace)
244   \@opcol\@startcolumn}%
245 \fi
246 \ifnum \outputpenalty>-\@Miv

```

At points such as this we need to check that there is still a minimal amount of room left on the page; this uses an arbitrary small value at present. If there is little room left we just force a page-break, OK?

This bit is essential only if a float has just been processed so maybe it should be moved; but this is the natural place at which to set the vsize and a test would need to be done anyway. A check has been added to ensure that there really has been a change in the value of \@colroom.

Since this happens only when there is nothing on the page but floats, the empty box should not cause any problem other than some overfull box messages, which is not entirely misleading.

The twocolumn case does not need any extra code here since this is the \output itself; in the second column there will still not be enough room left so \@emptycol will be executed again when the OR is called by the page builder when it gets to the penalty inserted by the first execution. (The page-builder is never invoked whilst the OR is being executed since it builds a inner vlist; thus any conditional code for the two-column case within \output may not get executed with the correct value of \if@firstcolumn.

```

247 \ifdim \@colroom<1.5\baselineskip
248 \ifdim \@colroom<\textheight
249   \@latex@warning@no@line {Text page \thepage\space
250                           contains only floats}%
251   \@emptycol
252 %   \if@twocolumn
253 %   \if@firstcolumn
254 %   \else
255 %   \@emptycol
256 %   \fi
257 %   \fi
258 \else
259   \global \vsize \@colroom
260 \fi
261 \else
262   \global \vsize \@colroom
263 \fi

```

```

264 \else
265 \global \vsize \maxdimen
266 \fi
267 }

```

CHANGES TO \@specialoutput:

\* \penalty\z@ changed to \penalty\interlinepenalty so \samepage works properly with figure and table environments.  
(Changed 23 Oct 86)

\* Definition of \@specialoutput changed 26 Feb 88 so \@pageht and \@pagedp aren't changed for a marginal note.  
(Change suggested by Chris Rowley.)

```

268 \gdef\@specialoutput{%
269 \ifnum \outputpenalty>-\@Mii
270 \doclearpage
271 \else
272 \ifnum \outputpenalty<-\@Miii
273 \ifnum \outputpenalty<-\@MM \deadcycles \z@ \fi
274 \global \setbox\@holdpg \vbox {\unvbox\@cclv}%
275 \else

```

Note that \boxmaxdepth should not be set here since we wish to record the natural depth of the holdpg box.

This is changed so as to not lose anything, such as writes and marks, which may get into box 255 and should be returned to the list. This should only happen when the first penalty in the mechanism is discarded and therefore \@holdpg should always be void in this case. This can happen because a penalty is discarded whenever there is no box on the list.

It was just: \setbox\@tempboxa \box \@cclv.

The last box which is removed is the box put there by the double-penalty mechanism. The \unskip then removes the \topskip which is put there since the box is the first on the page.

```

276 \global \setbox\@holdpg \vbox{%
277 \unvbox\@holdpg
278 \unvbox\@cclv

```

We must now remove the box added by the float mechanism and the \topskip glue therefore added above it by T<sub>E</sub>X.

```

279 \setbox\@tempboxa \lastbox
280 \unskip
281 }%

```

These two are needed as separate dimensions only by \@addmarginpar; for other purposes we put the whole size into \@pageht (see below).

```

282 \@pagedp \dp\@holdpg
283 \@pageht \ht\@holdpg
284 \unvbox \@holdpg
285 \@next\@currbox\@currlist{%
286 \ifnum \count\@currbox>\z@

```

Putting the whole size into \@pageht (see above).

```

287 \advance \@pageht \@pagedp
288 \ifvoid\footins \else

```

```

289         \advance \@pageht \ht\footins
290         \advance \@pageht \skip\footins
291         \advance \@pageht \dp\footins
292     \fi
293     \ifvbox \@kludgeins

```

We want to make the adjustment due to this insert only if the non-star form is used. The \*-form will probably not work with floats, but maybe it still could make some adjustment here even so?

```

294         \ifdim \wd\@kludgeins=\z@
295         \advance \@pageht \ht\@kludgeins
296     (*trace)
297         \fl@trace {Extra size added: \the \ht\@kludgeins}%
298     (/trace)
299     \fi
300 \fi

```

This version puts the inserts back just before the additional material; it could be moved earlier, before unboxing the page-so-far. Neither is guaranteed not to put things on the wrong page. This version is similar to the original version.

```

301     \@reinserts
302     \@addtocurcol
303 \else
304     \@reinserts
305     \@addmarginpar
306 \fi
307 }\@latexbug

```

A 2e change: use `\addpenalty` instead of `\penalty` here. Some penalty is needed to create a potential break-point immediately after the reinserts (or the marginal). Otherwise there can be no possibility to break here and this can cause the reinserts or the marginal to appear on the next page (which is often incorrect). However, if the nobreak flag is true, a `\nobreak` must be correct.

```

308     \ifnum \outputpenalty<\z@
309     \if@nobreak
310     \nobreak
311     \else
312     \addpenalty \interlinepenalty
313     \fi
314 \fi
315 \fi
316 \fi
317 }
318 (/2ekernel | fltrace)

```

`\@testwrongwidth` Test if the float box has the wrong width when trying to place it into some area.  
`\f@depth` (Actually the test is for a conventional depth setting rather than for the width of the float. For that reason the box depth was explicitly tailored when the float was created).

```

319 (latexrelease)\IncludeInRelease{2015/01/01}%
320 (latexrelease)          {\@testwrongwidth}{float order in 2-column}%
321 (*2ekernel | latexrelease | fltrace)

322 \def\@testwrongwidth #1{%
323     \ifdim\dp#1=\f@depth

```

```

324 (*trace)
325   \fl@trace{\string#1
326           \ifdim\f@depth=\z@ single \else double \fi
327           column float -- ok}%
328 (/trace)
329 \else
330   \global\@testtrue
331 (*trace)
332   \fl@trace{\string#1
333           \ifdim\f@depth=\z@ double \else single \fi
334           column float -- wrong}%
335 (/trace)
336 \fi}%

```

Normally looking for single column floats, which have zero depth.

```

337 \let\f@depth\z@
338 (/2ekernel | latexrelease | fltrace)
339 (latexrelease)\EndIncludeInRelease
340 (latexrelease)\IncludeInRelease{0000/00/00}%
341 (latexrelease)           {\@testwrongwidth}{float order in 2-column}%
342 (latexrelease)\let\@testwrongwidth\@undefined
343 (latexrelease)\let\f@depth\@undefined
344 (latexrelease)\EndIncludeInRelease

```

`\@doclearpage` This is a very much an emergency action, just dumping everything: footnotes first then floats. A more sophisticated version is needed; but even more urgent is a bug-free version (see, for example, pr/3528).

Also, it puts any left-over non-boxes (writes, specials, etc.) back after any float pages created: this is a very bad bug since, for example, a kludge insert will be in quite the wrong place and, worse, be irremovable and uncancellable.

All the remaining changes are replacing the double column defer list or inserting the extra test `\@testwrongwidth{<box>}` at suitable places. That is at places where a box is taken off the deferlist.

```

345 (latexrelease)\IncludeInRelease{2015/01/01}{\@doclearpage}%
346 (latexrelease)           {float order in 2-column}%
347 (*2ekernel | latexrelease)
348 \def \@doclearpage {%
349   \ifvoid\footins
350     \ifvbox\@kludgeins
351       {\setbox \@tempboxa \box \@kludgeins}%
352 (*trace)
353       \fl@trace {kludgeins box made void}%
354 (/trace)
355     \fi
356     \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
357     \setbox\@tempboxa\box\@cclv
358     \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
359     \global \let \@toplist \@empty
360     \global \let \@botlist \@empty
361     \global \@colroom \@colht
362     \ifx \@currlist\@empty
363       \else

```

```

364      \@latexerr{Float(s) lost}\@ehb
365      \global \let \@currlist \@empty
366      \fi
367      \@makefcolumn\@deferlist
368      \@whiles\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}%
369      \if@twocolumn
370      \if@firstcolumn
371      \xdef\@deferlist{\@dbltoplist\@deferlist}%
372      \global \let \@dbltoplist \@empty
373      \global \@colht \textheight
374      \begingroup
375      \dblfloatplacement
376      \@makefcolumn\@deferlist
377      \@whiles\if@fcolmade \fi{\@outputpage
378      \@makefcolumn\@deferlist}%
379      \endgroup
380      \else
381      \vbox{}\clearpage
382      \fi
383      \fi

```

the next line is needed to avoid losing floats in certain circumstances a single call to the original `\doclearpage` will now no longer output all floats.

```

384      \ifx\@deferlist\@empty \else\clearpage \fi
385      \else
386      \setbox\@cclv\vbox{\box\@cclv\vfll}%
387      \@makecol\@opcol
388      \clearpage
389      \fi
390 }%
391 \</2ekernel | latexrelease>
392 \<latexrelease>\EndIncludeInRelease
393 \<latexrelease>\IncludeInRelease{0000/00/00}{\@doclearpage}%
394 \<latexrelease> \float order in 2-column}%
395 \<latexrelease>\def \@doclearpage {%
396 \<latexrelease> \ifvoid\footins

```

We empty any left over kludge insert box here; this is a temporary fix. It should perhaps be applied to one page of cleared floats, but who cares? The whole of this stuff needs completely redoing for many such reasons.

```

397 \<latexrelease> \ifvbox\@kludgeins
398 \<latexrelease> {\setbox \@tempboxa \box \@kludgeins}%
399 \<*trace>
400 \<latexrelease> \fl@trace {kludgeins box made void}%
401 \</trace>
402 \<latexrelease> \fi
403 \<latexrelease> \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
404 \<latexrelease> \setbox\@tempboxa\box\@cclv
405 \<latexrelease> \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
406 \<latexrelease> \global \let \@toplist \@empty
407 \<latexrelease> \global \let \@botlist \@empty
408 \<latexrelease> \global \@colroom \@colht

```

```

409 \latexrelease> \ifx \@currlist\@empty
410 \latexrelease> \else
411 \latexrelease> \latexerr{Float(s) lost}\@ehb

412 \latexrelease> \global \let \@currlist \@empty
413 \latexrelease> \fi
414 \latexrelease> \makefcolumn\@deferlist
415 \latexrelease> \whiles\if@fcolmade \fi
416 \latexrelease> {\@opcol\@makefcolumn\@deferlist}%
417 \latexrelease> \if@twocolumn
418 \latexrelease> \if@firstcolumn
419 \latexrelease> \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%

420 \latexrelease> \global \let \@dbltoplist \@empty
421 \latexrelease> \global \@colht \textheight
422 \latexrelease> \begingroup
423 \latexrelease> \dblfloatplacement
424 \latexrelease> \makefcolumn\@dbldeferlist
425 \latexrelease> \whiles\if@fcolmade \fi
426 \latexrelease> {\@outputpage\@makefcolumn\@dbldeferlist}%
427 \latexrelease> \endgroup
428 \latexrelease> \else
429 \latexrelease> \vbox{}\clearpage
430 \latexrelease> \fi
431 \latexrelease> \fi
432 \latexrelease> \else
433 \latexrelease> \setbox\@cclv\vbox{\box\@cclv\vfil}%
434 \latexrelease> \makecol\@opcol
435 \latexrelease> \clearpage
436 \latexrelease> \fi
437 \latexrelease> }%
438 \latexrelease> \EndIncludeInRelease

```

\@opcol Several changes in detail here.

```

439 \*2kernel | fltrace)
440 \def \@opcol {%
441 \if@twocolumn
442 \outputdblcol
443 \else
444 \outputpage
445 \*trace)
446 \fl@trace{PAGE: one column (float? see above) page completed}%
447 \*trace)

```

Not needed since it comes after \@outputpage:

```

448 % \global\@colht\textheight
449 \fi

```

These do not need to be done every time \@opcol is used: they should be grouped together since they all need to be done at the end of the non-special output routine, or at the end of a clearpage one.

```

450 \global \@mparbottom \z@ \global \@textfloatsheight \z@
451 \@floatplacement
452 }
453 \*2kernel | fltrace)

```

`\@makecol` We must rewrite this macro to allow for variations in page-makeup required by changes in page-length.

This uses a different macro if a special-length column is being produced.

```

454 (*2kernel)
455 \gdef \@makecol {%
456   \ifvoid\footins
457     \setbox\@outputbox \box\@cclv
458   \else
459     \setbox\@outputbox \vbox {%

```

This `\boxmaxdepth` setting is to ensure that deep footnotes do not overwrite the footer (on account of the negative skip added later): it should use `\@maxdepth` otherwise the change is pointless when there are footnotes.

But see also its use when combining floats.

```

460   \boxmaxdepth \@maxdepth

461 %   \@tempdima\dp\@cclv
462   \unvbox \@cclv
463 %   \vskip-\@tempdima
464   \vskip \skip\footins

465   \color@begingroup
466   \normalcolor
467   \footnoterule
468   \unvbox \footins
469   \color@endgroup
470 }%
471 \fi

```

The h floats have now been finally committed to this page so we can reset their list. The top and bottom floats are then added to the page.

```

472 \let\@elt\relax
473 \xdef\@freelist{\@freelist\@midlist}%

474 \global \let \@midlist \@empty
475 \@combinefloats

```

The variations start here in case `\enlargethispage` has been used.

```

476 \ifvbox\@kludgeins
477   \@makespecialcolbox
478 \else

```

This extra reboxing is only needed to add the `\@texttop` and `\@textbottom` but this could be done earlier, when the floats are added.

The `\boxmaxdepth` resetting here will have no effect unless `\@textbottom` ends with a box or rule. So is this (or possibly `\@maxdepth`) the correct value?

The `\vskip -\dimen@` ensures that the visible depth of the box does not affect the placement of anything on the page. Thus very deep pages will overprint the footer; but these should have been prevented by suitable settings of the maxdepths at appropriate times.

If `\@textbottom` ends with a box or rule of non-zero depth then this skip adjustment should be done again after it.

I think that the final boxing of the main text page could have a common ending which may make it simpler to see what is going on.

This needs further investigation, especially in the ‘special case’.

Also, the `\boxmaxdepth` setting here affects what happens within `\@texttop` and `\@textbottom`, should it? Is it needed at all?

RmS 91/10/22: Replaced `\dimen128` by `\dimen@`.

```

479 \setbox\@outputbox \vbox to\@colht {%
480 % \boxmaxdepth \maxdepth %??
481 \@texttop
482 \dimen@ \dp\@outputbox
483 \unvbox \@outputbox
484 \vskip -\dimen@
485 \@textbottom
486 }%
487 \fi
488 \global \maxdepth \@maxdepth
489 }

```

`\@reinserts` This is the code which reinserts the inserts. It puts them all in one place; this can make some of them come out on the wrong page. It has been put into a separate macro to expedite experimentation.

```

490 \gdef \@reinserts{%
491 \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
492 \ifvbox\@kludgeins\insert\@kludgeins
493 {\unvbox\@kludgeins}\fi
494 }
495 </2kernel>

```

`\@makespecialcolbox` This implements certain variations in page-makeup.

```

496 (*2kernel | fltrace)
497 \gdef \@makespecialcolbox {%
498 (*trace)
499 \fl@trace{Kludgeins ht \the\ht\@kludgeins\space
500 dp \the\dp\@kludgeins\space
501 wd \the\wd\@kludgeins}%
502 </trace>

```

First we find the natural height of the column.

See above for discussion of what is happening here.

This needs further investigation, especially in this ‘special case’.

```

503 \setbox\@outputbox \vbox {%
504 \@texttop
505 \dimen@ \dp\@outputbox
506 \unvbox\@outputbox
507 \vskip-\dimen@
508 }%
509 \@tempdima \@colht
510 \ifdim \wd\@kludgeins>\z@

```

Note that in this case (the \*-version), the height of the `\@kludgeins` box is not used since its value is somewhat arbitrary: it need only be big enough to ensure that the page-break is not taken prematurely.

Here we calculate how much vertical space needs to be added in order to enable the column to fit into a box of size `\@colht` using the best information we have about the amount of shrink available (another thing which is known internally about a box, but cannot be accessed at the  $\TeX$  level!).

This needs T<sub>E</sub>X3 otherwise `\pageshrink` is zero anyway; it may not be exactly the figure we wish as it is the total available from the all the material collected before the page-break decision is made. It will, we think, always be an overestimate of the actual shrink in the box; therefore this should always force the shortest possible column with the possibility of an overfull box.

This should work for both the flush- and ragged-bottom setting since it makes the contents no smaller than the size (`\@colht`) of the box into which they are put.

There should perhaps be an upper limit, of 0pt?, on the extra space added to force shrinking.

See above for a discussion of the `\boxmaxdepth` setting here.

```

511     \advance \@tempdima -\ht\@outputbox
512     \advance \@tempdima \pageshrink
513 (*trace)
514     \fl@trace {Natural ht of col: \the \ht\@outputbox}%
515     \fl@trace {\string \@colht: \the \@colht}%
516     \fl@trace {Pageshrink added: \the \pageshrink}%
517     \fl@trace {Hence, space added: \the \@tempdima}%
518 (/trace)
519     \setbox\@outputbox \vbox to \@colht {%
520 %         \boxmaxdepth \maxdepth
521         \unvbox\@outputbox
522         \vskip \@tempdima
523         \@textbottom
524     }%
```

For the unstarred version, the final size of the page is precisely specified. Therefore, at least for the flush-bottom case, we need to ensure that, visually, it has this size exactly.

Thus we calculate this size and set the material in a box of this size, which is then put into a box of size `\@colht` with `\vss` at the bottom.

```

525     \else
526     \advance \@tempdima -\ht\@kludgeins
527 (*trace)
528     \fl@trace {Natural ht of col: \the \ht\@outputbox}%
529     \fl@trace {\string \@colht: \the \@colht}%
530     \fl@trace {Extra size added: -\the \ht \@kludgeins}%
531     \fl@trace {Hence, height of inner box: \the \@tempdima}%
532     \fl@trace {Max? pageshrink available: \the \pageshrink}%
533 (/trace)
```

This type of final packaging could be done always; this may simplify all of this page-makeup.

It is not necessary to set `\boxmaxdepth` here since the `\@outputbox` ends with glue.

```

534     \setbox \@outputbox \vbox to \@colht {%
535         \vbox to \@tempdima {%
536             \unvbox\@outputbox
537             \@textbottom}%
538         \vss}%
539     \fi
```

Finally we need to explicitly make the insert box void.

```

540   {\setbox \@tempboxa \box \@kludgeins}%
541 (*trace)
542   \fl@trace {kludgeins box made void}%
543 
```

```

544 }
545 
```

```

545 
```

```

\@texttop These do nothing as a default.
\@textbottom
```

```

546 (*2ekernel)
547 \let \@texttop \relax
548 \let \@textbottom \relax
```

```

\@resetactivechars RmS 93/09/06: added hook to protect against certain active characters in the
\@activechar@info output routine. Default checks are for active space and end-of-line.
```

```

549 \def\@activechar@info #1{%
550   \@latex@info@no@line {Active #1 character found while
551                         output routine is active
552                         \MessageBreak
553                         This may be a bug in a package file
554                         you are using}%
555 }
```

Do not put any spaces in this next bit!

```

556 \begingroup
557 \obeylines\obeyspaces%
558 \catcode'\'\active%
559 \gdef\@resetactivechars{%
560 \def~M{\@activechar@info{EOL}\space}%
561 \def {\@activechar@info{space}\space}%
562 \let'\active@math@prime}%
563 \endgroup
```

```

\@outputpage The \color@hbox hooks here are used to avoid putting just a colour special into
\@shipoutsetup an otherwise empty box (in a header or footer). These boxes are often set to be
\@writesetup completely empty and so adding a special produces a very underfull box message.
```

There has been extensive tidying up of the old code here; including the removal of a level of grouping.

The setting of `\protect` immediately before the `\shipout` is needed so that protected commands within `\writes` are handled correctly.

Within `shipout's vbox` it is reset to its default value, `\relax`.

Resetting it to its default value after the `shipout` has been completed (and the contents of the `writes` have been expanded) must be done by use of `\aftergroup`. This is because it must have the value `\relax` before macros coming from other uses of `\aftergroup` within this box are expanded.

Putting this into the `\aftergroup` token list does not affect the definition used in expanding the `\writes` because the `aftergroup` token list is only constructed when popping the save-stack, it is not expanded until after the `shipout` is completed.

Question: should things from an `\aftergroup` within the shipped out box be executed in the environment set up for the `writes`, or after it finishes?

A lot of this code has been in-lined tp prevent mis-use of internal commands as hooks.

```

564 \def\@outputpage{%
565 \begingroup          % the \endgroup is put in by \aftergroup

```

Now all the set-up stuff has been in-lined for Frank.

First the stuff for the writes.

From here ... was in the command \@writesetup.

```

566 \let \protect \noexpand

```

RmS 93/08/19: Redefined accents to allow changes in font encoding; but exactly why was this needed?

The \catcode'\ = 10 was removed as it was considered useless (presumably because nothing gets tokenised during shipout).

This was put in as some error produced active spaces in a mark, I think.

Why was the hyphen reset?

```

567 \@resetactivechars

```

If a page break happens between the start of a list and its first item the @newlist will be true and this will mess up any list that is used in the header or footer of the page. So we have to reset that flag.

```

568 \global\let\@@if@newlist\if@newlist

```

```

569 \global\@newlistfalse

```

This next hook replaces the following:

```

\let\-\@dischyph
\let'\@acci\let'\@accii\let\=\@acciii
\let\\@normalcr
\let\par\@@par %% 15 Sep 87 (this was once inside the box)

```

and it does more than they did; in particular it sets:

```

\parindent\z@
\parskip\z@skip
\everypar{}%
\leftskip\z@skip
\rightskip\z@skip
\parfillskip\@flushglue
\lineskip\normallineskip
\baselineskip\normalbaselineskip
\sloppy

```

```

570 \@parboxrestore

```

... to here was in the command \@writesetup.

```

571 \shipout \vbox{%

```

```

572 \set@typeset@protect

```

```

573 \aftergroup \endgroup

```

```

574 \aftergroup \set@typeset@protect

```

```

575                                     % correct? or just restore by ending

```

```

576                                     % the group?

```

This first bit has been moved inside the shipped out box.

Now the setup inside the shipped out box; this should contain all the stuff that could only affect typesetting; other stuff may need to be reset for the writes also.

From here ... was in the command \@shipoutsetup.

```

577 \if@specialpage
578   \global\@specialpagefalse\@nameuse{ps@\@specialstyle}%
579 \fi
580 \if@twoside
581   \ifodd\count\z@ \let\@thehead\@oddhead \let\@thefoot\@oddfoot
582     \let\@themargin\oddsidemargin
583   \else \let\@thehead\@evenhead
584     \let\@thefoot\@evenfoot \let\@themargin\evensidemargin
585   \fi
586 \fi

```

The rest was always inside the box.

RmS 91/08/15: aded this line:

```

587 \reset@font

```

RmS 93/08/06 Added \lineskiplimit=0pt to guard against it being nonzero:  
e.g. by \offinterlineskip being in effect.

There are probably lots of other things that may need resetting.

```

588 \normalsize

```

Reset the space factors.

```

589 \normalsfcodes

```

Reset these here (previously reset separately for head and foot)

```

590 \let\label\@gobble
591 \let\index\@gobble
592 \let\glossary\@gobble
593 \baselineskip\z@skip \lineskip\z@skip \lineskiplimit\z@

```

... to here was in the command \@shipoutsetup.

```

594 \@beginndvi
595 \vskip \topmargin
596 \moveright\@themargin \vbox {%
597   \setbox\@tempboxa \vbox to\headheight{%
598     \vfil
599     \color@hbox
600     \normalcolor
601     \hb@xt@\textwidth{\@thehead}%
602     \color@endbox
603   }%
604   \dp\@tempboxa \z@
605   \box\@tempboxa
606   \vskip \headsep
607   \box\@outputbox
608   \baselineskip \footskip
609   \color@hbox
610   \normalcolor
611   \hb@xt@\textwidth{\@thefoot}%
612   \color@endbox
613 }%
614 }%

```

\endgroup now inserted by \aftergroup

Restore \if@newlist

```

615 \global\let\if@newlist\@if@newlist

```

```

616 \global \colht \textheight
617 \stepcounter{page}%

```

It is now clear that this does something useful, thanks to Piet van Oostrum. It is needed because a float page is made without using TeX's page-builder; thus the output routine is never called so the marks are not updated.

```

618 \let\firstmark\botmark
619 }

```

`\@beginndvi` This unboxes stuff that must appear before anything else in the .dvi file, then returns that box register to the free list and cancels itself.

The stuff in the box should not add any typeset material to the page.

```

620 \def \@beginndvi{%
621   \unvbox \@beginndvibox
622   \global\let \@beginndvi \@empty
623 }

```

`\@combinefloats` The `\boxmaxdepth` setting here was not made local to a box so was dangerous. It is needed only within the box made by `\@cflt` (and not normally even there), so `\@cflb` it has been moved there; this also agrees with the original pseudocode.

```

624 \def \@combinefloats {%
625   \boxmaxdepth \maxdepth
626   \ifx \@toplist\@empty \else \@cflt \fi
627   \ifx \@botlist\@empty \else \@cflb \fi
628 }

629 \def \@cflt{%
630   \let \@elt \@comflelt
631   \setbox\@tempboxa \vbox{}%
632   \@toplist
633   \setbox\@outputbox \vbox{%
634     \boxmaxdepth \maxdepth
635     \unvbox\@tempboxa
636     \vskip -\floatsep
637     \topfigrule
638     \vskip \textfloatsep
639     \unvbox\@outputbox
640   }%
641   \let\@elt\relax
642   \xdef\@freelist{\@freelist\@toplist}%
643   \global\let\@toplist\@empty
644 }

645 \def \@cflb {%
646   \let\@elt\@comflelt
647   \setbox\@tempboxa \vbox{}%
648   \@botlist
649   \setbox\@outputbox \vbox{%
650     \unvbox\@outputbox
651     \vskip \textfloatsep
652     \botfigrule
653     \unvbox\@tempboxa
654     \vskip -\floatsep
655   }%

```

```

656 \let\@elt\relax
657 \xdef\@freelist{\@freelist\@botlist}%
658 \global \let \@botlist\@empty
659 }

\@comflelt
\@comdblfelet 660 \def\@comflelt#1{\setbox\@tempboxa
\@combinedblfloats 661 \vbox{\unvbox\@tempboxa\box #1\vskip\floatsep}}
662 \def\@comdblfelet#1{\setbox\@tempboxa
663 \vbox{\unvbox\@tempboxa\box #1\vskip\dblfloatsep}}
664 \def \@combinedblfloats{%
665 \ifx \@dbltoplist \@empty
666 \else
667 \setbox\@tempboxa \vbox{}%
668 \let \@elt \@comdblfelet
669 \@dbltoplist
670 \let \@elt \relax
671 \xdef \@freelist {\@freelist\@dbltoplist}%
672 \global\let \@dbltoplist \@empty
673 \setbox\@outputbox \vbox to\textheight

```

The setting of `\boxmaxdepth` here has no effect since the `\@outputbox` should already have depth zero. Even so, it would have no effect on the layout of the page.

```

674 {\boxmaxdepth\maxdepth %% probably not needed, CAR
675 \unvbox\@tempboxa\vskip-\dblfloatsep

```

Here we need different typesetting if the top float comes from `\@topnewpage`.

```

676 \ifnum \@dbltopnum>\m@ne
677 \dblfigrule
678 \fi
679 \vskip \dbltextfloatsep
680 \box\@outputbox
681 }%
682 \fi
683 }
684 \if2ekernel\

```

`\@startcolumn` We could combine (most of) these two into `\@startcol <list>`. Note that `\@startdblcolumn` `\@xstartcol` was only used once (i.e. in `\@startcolumn`); it has therefore been removed. This is not quite as efficient but it now has the same structure as `\@startdblcolumn`.

The empty-list test has been moved to `\@tryfcolumn`.

```

685 (*2ekernel | fltrace)
686 \def \@startcolumn {%
687 \global \@colroom \@colht
688 \@tryfcolumn \@deferlist
689 \if@fcolmade
690 (*trace)
691 \fl@trace{PAGE: float \if@twocolumn column \else page \fi
692 completed}%
693 \if2ekernel\
694 \else

```

```

695 \begingroup
696 \let \reserved@b \@deferlist
697 \global \let \@deferlist \@empty
698 \let \@elt \@scolelt
699 \reserved@b
700 \endgroup
701 \fi
702 }

```

This one does not need to set \@colht.

```

703 </2ekernel | fltrace>
704 <latexrelease | fltrace>\IncludeInRelease{2015/01/01}%
705 <latexrelease | fltrace> {\@startdblcolumn}{float order in 2-column}%
706 <*2ekernel | latexrelease | fltrace>
707 \def \@startdblcolumn {%
708 \@tryfcolumn \@deferlist
709 \if@fcolmade
710 <fltrace> \fl@trace{PAGE: double float page completed}%
711 \else
712 \begingroup
713 \let \reserved@b \@deferlist
714 \global \let \@deferlist \@empty
715 \let \@elt \@sdblcolelt
716 \reserved@b
717 \endgroup
718 \fi
719 }%
720 </2ekernel | latexrelease | fltrace>
721 <latexrelease | fltrace>\EndIncludeInRelease
722 <latexrelease | fltrace>\IncludeInRelease{0000/00/00}%
723 <latexrelease | fltrace> {\@startdblcolumn}{float order in 2-column}%
724 <latexrelease | fltrace>\def \@startdblcolumn {%

```

Not needed since this always comes after \@outputpage:

```

725 <latexrelease | fltrace>% \global \@colht \textheight
726 <latexrelease | fltrace> \@tryfcolumn \@dbldeferlist
727 <latexrelease | fltrace> \if@fcolmade
728 <*trace>
729 <latexrelease | fltrace> \fl@trace{PAGE: double float page completed}%
730 </trace>
731 <latexrelease | fltrace> \else
732 <latexrelease | fltrace> \begingroup
733 <latexrelease | fltrace> \let \reserved@b \@dbldeferlist
734 <latexrelease | fltrace> \global \let \@dbldeferlist \@empty
735 <latexrelease | fltrace> \let \@elt \@sdblcolelt
736 <latexrelease | fltrace> \reserved@b
737 <latexrelease | fltrace> \endgroup
738 <latexrelease | fltrace> \fi
739 <latexrelease | fltrace>}%
740 <latexrelease | fltrace>\EndIncludeInRelease
741 <*2ekernel | fltrace>

```

\@tryfcolumn Now tests if its list is empty before any further exertion.

```

742 \def \@tryfccolumn #1{%
743   \global \@fcolmadefalse
744   \ifx #1\@empty
745   \else
746   (*trace)
747     \fl@trace{PAGE: try float \if@twocolumn column/page\else page\fi
748       ---\string #1}%
749     \fl@trace{----- \string #1: #1}%
750   (/trace)

751   \xdef\@trylist{#1}%
752   \global \let \@failedlist \@empty
753   \begingroup
754     \let \@elt \@xtryfc \@trylist
755   \endgroup
756   \if@fcolmade
757     \@vtryfc #1%
758   \fi
759 \fi
760 }
761 (/2ekernel | fltrace)

762 (*2ekernel)

\@scolelt
763 \def\@scolelt#1{\def\@currbox{#1}\@addtonextcol}

\@sdblcolelt
764 \def\@sdblcolelt#1{\def\@currbox{#1}\@addtodblcol}

\@vtryfc
765 \def\@vtryfc #1{%
766   \global\setbox\@outputbox\vbox{}%
767   \let\@elt\@wtryfc
768   \@flsucceed
769   \global\setbox\@outputbox \vbox to\@colht{%
770     \vskip \@fptop
771     \vskip -\@fpsep
772     \unvbox \@outputbox
773     \vskip \@fpbot}%
774   \let\@elt\relax
775   \xdef #1{\@failedlist\@flfail}%
776   \xdef\@freelist{\@freelist\@flsucceed}}

\@wtryfc
777 \def\@wtryfc #1{%
778   \global\setbox\@outputbox\vbox{%
779     \unvbox\@outputbox
780     \vskip\@fpsep
781     \box #1}}

\@xtryfc
782 (/2ekernel)
783 (latexrelease)\IncludeInRelease{2015/01/01}{\@xtryfc}%

```

```

784 <latexrelease> {float order in 2-column}%
785 <*2ekernel | latexrelease>
786 \def\@xtryfc #1{%
787   \@next\reserved@a\@trylist{}\}%
788   \@currtype \count #1%
789   \divide\@currtype\@xxxii
790   \multiply\@currtype\@xxxii
791   \@bitor \@currtype \@failedlist
792   \@testfp #1%
793   \@testwrongwidth #1%
794   \ifdim \ht #1>\@colht
795     \@testtrue
796   \fi
797   \if@test
798     \@cons\@failedlist #1%
799   \else
800     \@ytryfc #1%
801   \fi}%
802 </2ekernel | latexrelease>
803 <latexrelease>\EndIncludeInRelease
804 <latexrelease>\IncludeInRelease{0000/00/00}\{\@xtryfc}%
805 <latexrelease> {float order in 2-column}%
806 <latexrelease>\def\@xtryfc #1{%
807   \@next\reserved@a\@trylist{}\}%
808 <latexrelease> \@currtype \count #1%
809 <latexrelease> \divide\@currtype\@xxxii
810 <latexrelease> \multiply\@currtype\@xxxii
811 <latexrelease> \@bitor \@currtype \@failedlist
812 <latexrelease> \@testfp #1%
813 <latexrelease> \ifdim \ht #1>\@colht
814 <latexrelease> \@testtrue
815 <latexrelease> \fi
816 <latexrelease> \if@test
817 <latexrelease> \@cons\@failedlist #1%
818 <latexrelease> \else
819 <latexrelease> \@ytryfc #1%
820 <latexrelease> \fi}%
821 <latexrelease>\EndIncludeInRelease
822 <*2ekernel>

\@ytryfc
823 \def\@ytryfc #1{%
824   \begingroup
825   \gdef\@flsucceed{\@elt #1}%
826   \global\let\@flfail\@empty
827   \@tempdima\ht #1%
828   \let\@elt\@ztryfc
829   \@trylist
830   \ifdim \@tempdima >\@fpmin
831     \global\@fcolmadetrue
832   \else
833     \@cons\@failedlist #1%
834   \fi

```

```

835 \endgroup
836 \if@fcolmade
837 \let\@elt\@gobble
838 \fi}

\@ztryfc
839 (/2ekernel)
840 (latexrelease)\IncludeInRelease{2015/01/01}{@ztryfc}%
841 (latexrelease) {float order in 2-column}%
842 (*2ekernel | latexrelease)
843 \def\@ztryfc #1{%
844 \@tempcnta\count #1%
845 \divide\@tempcnta\@xxxii
846 \multiply\@tempcnta\@xxxii
847 \@bitor \@tempcnta {\@failedlist \@flfail}%
848 \@testfp #1%
    not in fixfloats?
849 \@testwrongwidth #1%

850 \@tempdimb\@tempdima
851 \advance\@tempdimb\ht #1%
852 \advance\@tempdimb\@fpsep
853 \ifdim \@tempdimb >\@colht
854 \@testtrue
855 \fi
856 \if@test
857 \@cons\@flfail #1%
858 \else
859 \@cons\@flsucceed #1%
860 \@tempdima\@tempdimb
861 \fi}%
862 (/2ekernel | latexrelease)
863 (latexrelease)\EndIncludeInRelease
864 (latexrelease)\IncludeInRelease{0000/00/00}{@ztryfc}%
865 (latexrelease) {float order in 2-column}%
866 (latexrelease)\def\@ztryfc #1{%
867 (latexrelease) \@tempcnta \count#1%
868 (latexrelease) \divide\@tempcnta\@xxxii
869 (latexrelease) \multiply\@tempcnta\@xxxii
870 (latexrelease) \@bitor \@tempcnta {\@failedlist \@flfail}%
871 (latexrelease) \@testfp #1%
872 (latexrelease) \@tempdimb\@tempdima
873 (latexrelease) \advance\@tempdimb \ht#1%
874 (latexrelease) \advance\@tempdimb\@fpsep
875 (latexrelease) \ifdim \@tempdimb >\@colht
876 (latexrelease) \@testtrue
877 (latexrelease) \fi
878 (latexrelease) \if@test
879 (latexrelease) \@cons\@flfail #1%
880 (latexrelease) \else
881 (latexrelease) \@cons\@flsucceed #1%
882 (latexrelease) \@tempdima\@tempdimb
883 (latexrelease) \fi}%
884 (latexrelease)\EndIncludeInRelease

```

The major changes for float suppression and the changes to the float mechanism to make it conform to the documentation are in these next macros.

`\@addtobot` Lots of changes.

```

885 (*2ekernel | fltrace)
886 \def \@addtobot {%
887 (*trace)
888   \fl@trace{***Start addtobot}%
889 (/trace)
890   \@getfpsbit 4\relax
891 (*trace)
892   \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi bot:
893                                           \the \@fpstype}%
894 (/trace)
895   \ifodd \@tempcnta
896     \@flsetnum \@botnum
897     \ifnum \@botnum>\z@
898       \@tempswafalse
899       \@flcheckspace \@botroom \@botlist
900       \if@tempswa

```

This next line means that this page is produced with box 255 having depth zero, rather than the normal maxdepth: is this needed, useful?

```

901     \global \maxdepth \z@
902     \@flupdates \@botnum \@botroom \@botlist
903 (*trace)
904     \fl@trace{colroom (after-bot) = \the \@colroom}%
905     \fl@trace{colnum (after-bot) = \the \@colnum}%
906     \fl@trace{botnum (after-bot) = \the \@botnum}%
907     \fl@trace{***Success: bot}%
908 (/trace)
909     \@inserttrue
910   \fi
911 (*trace)
912   \else
913     \fl@trace{Fail: botnum = \the \@botnum:
914                                     fpstype \the \@fpstype=ORD?}%
915     \ifnum \@fpstype<\sixt@n
916       \fl@trace{ERROR: !b float not successful (addtobot)}%
917     \fi
918 (/trace)
919   \fi
920 \fi
921 }

```

`\@addtotoporbot` Lots of changes.

```

922 \def \@addtotoporbot {%
923 (*trace)
924   \fl@trace{***Start addtotoporbot}%
925 (/trace)
926   \@getfpsbit \tw@
927 (*trace)
928   \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi top:
929                                           \the \@fpstype}%

```

```

930 </trace>
931   \ifodd \@tempcnta
932     \@flsetnum \@topnum
933     \ifnum \@topnum>\z@
934       \@tempwafalse
935       \@flcheckspace \@toproom \@toplist
936       \if@tempswa
937         \@bitor\@currtype{\@midlist\@botlist}%
938 (*trace)
939         \fl@trace{(mid+bot)list: \@midlist, \@botlist:
940                   (addtotoporbot-before)}%
941 </trace>
942   \if@test
943 (*trace)
944   \fl@trace{type already on list: mid or bot---sent to addtobot}%
945 </trace>
946   \else
947     \@flupdates \@topnum \@toproom \@toplist
948 (*trace)
949     \fl@trace{colroom (after-top) = \the \@colroom}%
950     \fl@trace{colnum (after-top) = \the \@colnum}%
951     \fl@trace{topnum (after-top) = \the \@topnum}%
952     \fl@trace{***Success: top}%
953 </trace>
954     \@inserttrue
955   \fi
956 \fi
957 (*trace)
958   \else
959     \fl@trace{Fail: topnum = \the \@topnum: fpstype
960               \the \@fpstype=ORD?}%
961     \ifnum \@fpstype<\sixt@@n
962       \fl@trace{ERROR: !t float not successful (addtotoporbot)}%
963     \fi
964 </trace>
965   \fi
966 \fi
967 \if@insert
968   \else
969 (*trace)
970   \fl@trace{sent to addtobot (addtotoporbot)}%
971 </trace>
972   \@addtobot
973 \fi
974 }
975 </2ekernel | fltrace>

```

\@addtocurcol Lots of changes.

```

976 <latexrelease | fltrace | flafter>\IncludeInRelease{2015/01/01}%
977 <latexrelease | fltrace | flafter> {\@addtocurcol}{float order in 2-column}%
978 (*2ekernel | latexrelease | fltrace | flafter)
979 \def \@addtocurcol {%
980 (*trace)
981   \fl@trace{***Start addtocurcol}%

```

```

982 </trace>
983   \@insertfalse
984   \@setfloattypecounts
985   \ifnum \@fpstype=8
986 < *trace>
987   \fl@trace{fpstype !p only (addtocurcol): \the \@fpstype = 8?}%
988 </trace>
989   \else
990   \ifnum \@fpstype=24
991 < *trace>
992   \fl@trace{fpstype p only (addtocurcol): \the \@fpstype = 24?}%
993 </trace>
994   \else
995   \@flsettextmin

```

This is a new adjustment which is quite a major change in functionality; but it implements the documentation. Note that \@reqcolroom will include the whole of the page-so-far, and hence includes \@textfloatsheight of floats, so before comparing it with \@textmin, we add this to \@textmin also.

```

996 < *trace>
997   \fl@trace{textfloatsheight (before) = \the \@textfloatsheight}%
998 </trace>
999   \advance \@textmin \@textfloatsheight
1000   \@reqcolroom \@pageht

```

This line must be removed since \@specialoutput changed.

```

1001 %       \advance \@reqcolroom \@pagedp
1002 < *trace>
1003   \fl@trace{textmin + textfloatsheight: \the \@textmin}%
1004   \fl@trace{page-so-far: \the \@reqcolroom}%
1005 </trace>
1006   \ifdim \@textmin>\@reqcolroom
1007   \@reqcolroom \@textmin
1008 < *trace>
1009   \fl@trace{ORD? textmin being used}%
1010 </trace>
1011   \fi
1012   \advance \@reqcolroom \ht\@currbox
1013 < *trace>
1014   \fl@trace{float size = \the \ht \@currbox (addtocurcol)}%
1015   \fl@trace{colroom = \the \@colroom (addtocurcol)}%
1016   \fl@trace{reqcolroom = \the \@reqcolroom (addtocurcol)}%
1017 </trace>
1018   \ifdim \@colroom>\@reqcolroom
1019   \@flsetnum \@colnum
1020   \ifnum \@colnum>\z@
1021   \@bitor\@currtype\@deferlist

```

We need to defer the float also if its width doesn't fit.

```

1022   \@testwrongwidth\@currbox
1023 < *trace>
1024   \fl@trace{deferlist: \@deferlist: (addtocurcol-before)}%
1025 </trace>
1026   \if@test

```

```

1027 (*trace)
1028         \fl@trace{type already on list: defer (addtocurcol)}%
1029 (/trace)
1030     \else
1031         \@bitor\@currtype\@botlist
1032 (*trace)
1033         \fl@trace{botlist: \@botlist: (addtocurcol-before)}%
1034 (/trace)
1035     \if@test
1036 (*trace)
1037         \fl@trace{type already on list: bot---sent to addtobot}%
1038 (/trace)
1039         \@addtobot
1040     \else
1041 (*trace)
1042         \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi
1043             here: \the \@fpstype}%
1044 (/trace)
1045         \ifodd \count\@currbox
1046             \advance \@reqcolroom \intextsep
1047             \ifdim \@colroom>\@reqcolroom
1048                 \global \advance \@colnum \m@ne
1049                 \global \advance \@textfloatsheight \ht\@currbox

```

This may sometimes give an overestimate.

```

1050         \global \advance \@textfloatsheight 2\intextsep
1051         \@cons \@midlist \@currbox
1052 (*trace)
1053         \fl@trace{***Success: here}%
1054         \fl@trace{textfloatsheight (after-here) =
1055             \the \@textfloatsheight}%
1056         \fl@trace{colnum (after-here) = \the \@colnum}%
1057 (/trace)

```

CHANGE TO \@addtocurcol:

\penalty\z@ changed to \penalty\interlinepenalty so \samepage works properly with figure and table environments. (Changed 23 Oct 86)

There is also an \addpenalty\interlinepenalty above.

Since in 2e \samepage is no longer supported, these could be removed.

Although it is best to use \addvspace in case two h floats come together, this makes other spacing more difficult to adjust; whereas if a user specifies two h floats together then they can more easily get the spacing correct by ad hoc commands.

It is necessary to adjust for the addition of \parskip here in case the float is added between paragraphs (i.e. when in vertical mode).

If the nobreak switch is true we need to reset it and clear \everypar since the float may not reset the flag and cannot reset the \everypar globally.

Typesetting starts here (we are in vertical mode).

```

1058         \if@nobreak
1059             \nobreak
1060             \@nobreakfalse
1061             \everypar{}%
1062         \else
1063             \addpenalty \interlinepenalty
1064         \fi

```

```

1065          \vskip \intextsep
1066          \box\@currbox
1067          \penalty\interlinepenalty
1068          \vskip\intextsep
1069          \ifnum\outputpenalty <-\@Mii \vskip -\parskip\fi

Typesetting ends here.

1070          \outputpenalty \z@
1071          \@inserttrue
1072 (*trace)
1073          \else
1074          \fl@trace{Fail---no room at 2nd test of colroom
1075                      (addtocurcol \string\intextsep)}%
1076 (/trace)
1077          \fi
1078          \fi
1079          \if@insert
1080          \else

Next set of docstrip guards are a bit weird, essentially \@addtotoporbot ends
up inside the kernel and the fltrace package and \@addtobot shows up in the
flafter package. Guess that could have been done a bit more obvious :-)

1081 (*2ekernel | fltrace | latexrelease)
1082 (*trace)
1083          \fl@trace{not here: sent to addtotoporbot}%
1084 (/trace)
1085          \@addtotoporbot
1086 (/2ekernel | fltrace | latexrelease)
1087 (*!2ekernel&!fltrace&!latexrelease)
1088 (*trace)
1089          \fl@trace{not here: sent to addtobot}%
1090 (/trace)
1091          \@addtobot
1092 (/!2ekernel&!fltrace&!latexrelease)
1093          \fi
1094          \fi
1095          \fi
1096 (*trace)
1097          \else
1098          \fl@trace{Fail: colnum = \the \@colnum:
1099                      fpstype \the \@fpstype=ORD?}%
1100          \ifnum \@fpstype<\sist@n
1101          \fl@trace{ERROR: BANG float not successful (addtocurcol)}%
1102          \fi
1103 (/trace)
1104          \fi
1105 (*trace)
1106          \else
1107          \fl@trace{Fail---no room: fl box ht: \the \ht \@currbox
1108                      (addtocurcol)}%
1109 (/trace)
1110          \fi
1111          \fi
1112          \fi
1113          \if@insert

```

```

1114 \else
1115 \@resetfhps
1116 <*trace>
1117 \fl@trace{put on deferlist (addtocurcol)}%
1118 </trace>
1119 \@cons\@deferlist\@currbox
1120 <*trace>
1121 \fl@trace{deferlist: \@deferlist: (addtocurcol-after)}%
1122 </trace>
1123 \fi
1124 }%
1125 </2ekernel | latexrelease | fltrace | flafter>
1126 <latexrelease | fltrace | flafter>\EndIncludeInRelease
1127 <latexrelease | fltrace | flafter>\IncludeInRelease{0000/00/00}%
1128 <latexrelease | fltrace | flafter> {\@addtocurcol}{float order in 2-column}%
1129 <latexrelease | fltrace | flafter>\def \@addtocurcol {%
1130 <*trace>
1131 <latexrelease | fltrace | flafter> \fl@trace{***Start addtocurcol}%
1132 </trace>
1133 <latexrelease | fltrace | flafter> \@insertfalse
1134 <latexrelease | fltrace | flafter> \@setfloattypescounts
1135 <latexrelease | fltrace | flafter> \ifnum \@fpstype=8
1136 <*trace>
1137 <latexrelease | fltrace | flafter> \fl@trace{fpstype !p only (addtocurcol):
1138 <latexrelease | fltrace | flafter> \the \@fpstype = 8?}%
1139 </trace>
1140 <latexrelease | fltrace | flafter> \else
1141 <latexrelease | fltrace | flafter> \ifnum \@fpstype=24
1142 <*trace>
1143 <latexrelease | fltrace | flafter> \fl@trace{fpstype p only (addtocurcol):
1144 <latexrelease | fltrace | flafter> \the \@fpstype = 24?}%
1145 </trace>
1146 <latexrelease | fltrace | flafter> \else
1147 <latexrelease | fltrace | flafter> \@flsettextmin

This is a new adjustment which is quite a major change in functionality; but it
implements the documentation. Note that \@reqcolroom will include the whole
of the page-so-far, and hence includes \@textfloatsheight of floats, so before
comparing it with \@textmin, we add this to \@textmin also.

1148 <*trace>
1149 <latexrelease | fltrace | flafter> \fl@trace{textfloatsheight (before) =
1150 <latexrelease | fltrace | flafter> \the \@textfloatsheight}%
1151 </trace>
1152 <latexrelease | fltrace | flafter> \advance \@textmin \@textfloatsheight
1153 <latexrelease | fltrace | flafter> \@reqcolroom \@pageht

This line must be removed since \@specialoutput changed.

1154 % \advance \@reqcolroom \@pagedp
1155 <*trace>
1156 <latexrelease | fltrace | flafter> \fl@trace{textmin + textfloatsheight:
1157 <latexrelease | fltrace | flafter> \the \@textmin}%
1158 <latexrelease | fltrace | flafter> \fl@trace{page-so-far: \the \@reqcolroom}%
1159 <latexrelease | fltrace | flafter>
1160 </trace>
1161 <latexrelease | fltrace | flafter> \ifdim \@textmin>\@reqcolroom

```

```

1162 <latexrelease | fltrace | flafter> \@reqcolroom \@textmin
1163 <*trace>
1164 <latexrelease | fltrace | flafter> \fl@trace{ORD? textmin being used}%
1165 </trace>
1166 <latexrelease | fltrace | flafter> \fi
1167 <latexrelease | fltrace | flafter> \advance \@reqcolroom \ht\@currbox
1168 <*trace>
1169 <latexrelease | fltrace | flafter> \fl@trace{float size =
1170 <latexrelease | fltrace | flafter> \the \ht \@currbox (addtocurcol)}}%
1171 <latexrelease | fltrace | flafter> \fl@trace{colroom =
1172 <latexrelease | fltrace | flafter> \the \@colroom (addtocurcol)}}%
1173 <latexrelease | fltrace | flafter> \fl@trace{reqcolroom =
1174 <latexrelease | fltrace | flafter> \the \@reqcolroom (addtocurcol)}}%
1175 </trace>
1176 <latexrelease | fltrace | flafter> \ifdim \@colroom>\@reqcolroom
1177 <latexrelease | fltrace | flafter> \@flsetnum \@colnum
1178 <latexrelease | fltrace | flafter> \ifnum \@colnum>\z@
1179 <latexrelease | fltrace | flafter> \@bitor\@currtype\@deferlist
1180 <*trace>
1181 <latexrelease | fltrace | flafter> \fl@trace{deferlist:
1182 <latexrelease | fltrace | flafter> \@deferlist: (addtocurcol-before)}}%
1183 </trace>
1184 <latexrelease | fltrace | flafter> \if@test
1185 <*trace>
1186 <latexrelease | fltrace | flafter> \fl@trace{type already on list:
1187 <latexrelease | fltrace | flafter> defer (addtocurcol)}}%
1188 </trace>
1189 <latexrelease | fltrace | flafter> \else
1190 <latexrelease | fltrace | flafter> \@bitor\@currtype\@botlist
1191 <*trace>
1192 <latexrelease | fltrace | flafter> \fl@trace{botlist: \@botlist:
1193 <latexrelease | fltrace | flafter> (addtocurcol-before)}}%
1194 </trace>
1195 <latexrelease | fltrace | flafter> \if@test
1196 <*trace>
1197 <latexrelease | fltrace | flafter> \fl@trace{type already on list:
1198 <latexrelease | fltrace | flafter> bot---sent to addtobot}}%
1199 </trace>
1200 <latexrelease | fltrace | flafter> \@addtobot
1201 <latexrelease | fltrace | flafter> \else
1202 <*trace>
1203 <latexrelease | fltrace | flafter> \fl@trace{fpstype
1204 <latexrelease | fltrace | flafter> \ifodd \@tempcnta OK \else not \fi
1205 <latexrelease | fltrace | flafter> here: \the \@fpstype}}%
1206 </trace>
1207 <latexrelease | fltrace | flafter> \ifodd \count\@currbox
1208 <latexrelease | fltrace | flafter> \advance \@reqcolroom \intextsep
1209 <latexrelease | fltrace | flafter> \ifdim \@colroom>\@reqcolroom
1210 <latexrelease | fltrace | flafter> \global \advance \@colnum \m@ne
1211 <latexrelease | fltrace | flafter> \global \advance
1212 <latexrelease | fltrace | flafter> \@textfloatsheight\ht\@currbox
This may sometimes give an overestimate.
1213 <latexrelease | fltrace | flafter> \global \advance
1214 <latexrelease | fltrace | flafter> \@textfloatsheight 2\intextsep

```

```

1215 <latexrelease | fltrace | flafter>          \@cons \@midlist \@currbox
1216 < *trace>
1217 <latexrelease | fltrace | flafter>          \fl@trace{***Success: here}%
1218 <latexrelease | fltrace | flafter>          \fl@trace{textfloatsheight
1219 <latexrelease | fltrace | flafter>              (after-here) =
1220 <latexrelease | fltrace | flafter>              \the \@textfloatsheight}%
1221 <latexrelease | fltrace | flafter>          \fl@trace{colnum (after-here) =
1222 <latexrelease | fltrace | flafter>              \the \@colnum}%
1223 </trace>

CHANGE TO \@addtocurcol:
\penalty\z@ changed to \penalty\interlinepenalty so \samepage works
properly with figure and table environments. (Changed 23 Oct 86)
There is also an \addpenalty\interlinepenalty above.
Since in 2e \samepage is no longer supported, these could be removed.
Although it is best to use \addvspace in case two h floats come together, this
makes other spacing more difficult to adjust; whereas if a user specifies two h floats
together then they can more easily get the spacing correct by ad hoc commands.
It is necessary to adjust for the addition of \parskip here in case the float is
added between paragraphs (i.e. when in vertical mode).
If the nobreak switch is true we need to reset it and clear \everypar since the
float may not reset the flag and cannot reset the \everypar globally.
Typesetting starts here (we are in vertical mode).
1224 <latexrelease | fltrace | flafter>          \if@nobreak
1225 <latexrelease | fltrace | flafter>          \nobreak
1226 <latexrelease | fltrace | flafter>          \@nobreakfalse
1227 <latexrelease | fltrace | flafter>          \everypar{}%
1228 <latexrelease | fltrace | flafter>          \else
1229 <latexrelease | fltrace | flafter>          \addpenalty\interlinepenalty
1230 <latexrelease | fltrace | flafter>          \fi
1231 <latexrelease | fltrace | flafter>          \vskip \intextsep
1232 <latexrelease | fltrace | flafter>          \box\@currbox
1233 <latexrelease | fltrace | flafter>          \penalty\interlinepenalty
1234 <latexrelease | fltrace | flafter>          \vskip\intextsep
1235 <latexrelease | fltrace | flafter>          \ifnum\outputpenalty
1236 <latexrelease | fltrace | flafter>              <-\@Mii \vskip
1237 <latexrelease | fltrace | flafter>          -\parskip\fi

Typesetting ends here.
1238 <latexrelease | fltrace | flafter>          \outputpenalty \z@
1239 <latexrelease | fltrace | flafter>          \@inserttrue
1240 < *trace>
1241 <latexrelease | fltrace | flafter>          \else
1242 <latexrelease | fltrace | flafter>          \fl@trace{Fail---no room at 2nd test of colroom
1243 <latexrelease | fltrace | flafter>              (addtocorcol \string\intextsep)}%
1244 </trace>
1245 <latexrelease | fltrace | flafter>          \fi
1246 <latexrelease | fltrace | flafter>          \fi
1247 <latexrelease | fltrace | flafter>          \if@insert
1248 <latexrelease | fltrace | flafter>          \else

```

Next set of docstrip guards are a bit weird, essentially \@addtotoporbot ends up inside the kernel and the fltrace package and \@addtotoporbot shows up in the flafter package. Guess that could have been done a bit more obvious :-)

```

1249 (*2ekernel | fltrace)
1250 (*trace)
1251 \latexrelease | fltrace | flafter) \fl@trace{not here: sent to addtotoporbot}%
1252 \traced)
1253 \latexrelease | fltrace | flafter) \@addtotoporbot
1254 \traced)
1255 (*!2ekernel&!autoload&!fltrace)
1256 (*trace)
1257 \latexrelease | fltrace | flafter) \fl@trace{not here: sent to addtobot}%
1258 \traced)
1259 \latexrelease | fltrace | flafter) \@addtobot
1260 \traced)
1261 \latexrelease | fltrace | flafter) \fi
1262 \latexrelease | fltrace | flafter) \fi
1263 \latexrelease | fltrace | flafter) \fi
1264 (*trace)
1265 \latexrelease | fltrace | flafter) \else
1266 \latexrelease | fltrace | flafter) \fl@trace{Fail: colnum = \the \@colnum:
1267 \latexrelease | fltrace | flafter) fpstype \the \@fpstype=ORD?}%
1268 \latexrelease | fltrace | flafter) \ifnum \@fpstype<\sist@on
1269 \latexrelease | fltrace | flafter) \fl@trace{ERROR: BANG float not successful
1270 \latexrelease | fltrace | flafter) (addtocurcol)}}%
1271 \latexrelease | fltrace | flafter) \fi
1272 \traced)
1273 \latexrelease | fltrace | flafter) \fi
1274 (*trace)
1275 \latexrelease | fltrace | flafter) \else
1276 \latexrelease | fltrace | flafter) \fl@trace{Fail---no room: fl box ht:
1277 \latexrelease | fltrace | flafter) \the \ht \@currbox (addtocurcol)}}%
1278 \traced)
1279 \latexrelease | fltrace | flafter) \fi
1280 \latexrelease | fltrace | flafter) \fi
1281 \latexrelease | fltrace | flafter) \fi
1282 \latexrelease | fltrace | flafter) \if@insert
1283 \latexrelease | fltrace | flafter) \else
1284 \latexrelease | fltrace | flafter) \@resetfps
1285 (*trace)
1286 \latexrelease | fltrace | flafter) \fl@trace{put on deferlist (addtocurcol)}}%
1287 \traced)
1288 \latexrelease | fltrace | flafter) \@cons\@deferlist\@currbox
1289 (*trace)
1290 \latexrelease | fltrace | flafter) \fl@trace{deferlist: \@deferlist:
1291 \latexrelease | fltrace | flafter) (addtocurcol-after)}}%
1292 \traced)
1293 \latexrelease | fltrace | flafter) \fi
1294 \latexrelease | fltrace | flafter) }%
1295 \latexrelease | fltrace | flafter)\EndIncludeInRelease

```

\@addtonextcol Lots of changes.

```

1296 \latexrelease | fltrace)\IncludeInRelease{2015/01/01}
1297 \latexrelease | fltrace) {\@addtonextcol}{float order in 2-column}%
1298 (*2ekernel | fltrace)
1299 \def\@addtonextcol{%
1300 \begin{group

```

```

1301 (*trace)
1302   \fl@trace{***Start addtonextcol}%
1303 (/trace)
1304   \@insertfalse
1305   \@setfloattypecounts
1306   \ifnum \@fpstype=8
1307 (*trace)
1308   \fl@trace{fpstype not curcol: \the \@fpstype = 8?}%
1309 (/trace)
1310   \else
1311   \ifnum \@fpstype=24
1312 (*trace)
1313   \fl@trace{fpstype not curcol: \the \@fpstype = 24?}%
1314 (/trace)
1315   \else
1316   \@flsettextmin
1317 (*trace)
1318   \fl@trace{text-so-far: Opt (top of col)}%
1319 (/trace)
1320   \@reqcolroom \ht\@currbox
1321 (*trace)
1322   \fl@trace{float size: \the \@reqcolroom (addtonextcol)}%
1323 (/trace)
1324   \advance \@reqcolroom \@textmin
1325 (*trace)
1326   \fl@trace{colroom = \the \@colroom (addtonextcol)}%
1327   \fl@trace{reqcolroom = \the \@reqcolroom (addtonextcol)}%
1328 (/trace)
1329   \ifdim \@colroom>\@reqcolroom
1330   \@flsetnum \@colnum
1331   \ifnum \@colnum>\z@
1332   \@bitor\@currtype\@deferlist
1333 (*trace)
1334   \fl@trace{deferlist: \@deferlist: (addtonextcol-before)}%
1335 (/trace)
1336   \@testwrongwidth\@currbox
1337   \if@test
1338 (*trace)
1339   \fl@trace{type already on list: defer (addtonextcol)}%
1340 (/trace)
1341   \else
1342 (*trace)
1343   \fl@trace{sent to addtotoporbot (addtonextcol)}%
1344 (/trace)
1345   \@addtotoporbot
1346   \fi
1347   \fi
1348 (*trace)
1349   \else
1350   \fl@trace{Fail---no room: fl box ht: \the \ht \@currbox
1351   (addtonextcol)}%
1352 (/trace)
1353   \fi

```

```

1354     \fi
1355     \fi
1356     \if@insert
1357     \else
1358     (*trace)
1359     \fl@trace{put back on deferlist (addtonextcol)}%
1360     /trace)
1361     \cons\@deferlist\@currbox
1362     (*trace)
1363     \fl@trace{deferlist: \@deferlist: (addtonextcol-after)}%
1364     /trace)
1365     \fi
1366     (*trace)
1367     \fl@trace{End of addtonextcol -- locally counts:}%
1368     \fl@trace{col: \the\@colnum. top: \the \@topnum. bot: \the \@botnum.}%
1369     /trace)
1370     \endgroup
1371     (*trace)
1372     \fl@trace{End of addtonextcol -- globally counts:}%
1373     \fl@trace{col: \the\@colnum. top: \the \@topnum. bot: \the \@botnum.}%
1374     /trace)
1375 }%
1376 /2ekernel | fltrace)
1377 (latexrelease | fltrace)\EndIncludeInRelease
1378 (latexrelease | fltrace)\IncludeInRelease{0000/00/00}%
1379 (latexrelease | fltrace) {\@addfloatcol}{float order in 2-column}%
1380 (latexrelease | fltrace)\def\@addtonextcol{%
1381 (latexrelease | fltrace) \begin{group
1382 (*trace)
1383 (latexrelease | fltrace) \fl@trace{***Start addtonextcol}%
1384 /trace)
1385 (latexrelease | fltrace) \@insertfalse
1386 (latexrelease | fltrace) \@setfloattypecounts
1387 (latexrelease | fltrace) \ifnum \@fpstype=8
1388 (*trace)
1389 (latexrelease | fltrace) \fl@trace{fpstype not curcol:
1390 (latexrelease | fltrace) \the \@fpstype = 8?}%
1391 /trace)
1392 (latexrelease | fltrace) \else
1393 (latexrelease | fltrace) \ifnum \@fpstype=24
1394 (*trace)
1395 (latexrelease | fltrace) \fl@trace{fpstype not curcol:
1396 (latexrelease | fltrace) \the \@fpstype = 24?}%
1397 /trace)
1398 (latexrelease | fltrace) \else
1399 (latexrelease | fltrace) \@flsettextmin
1400 (*trace)
1401 (latexrelease | fltrace) \fl@trace{text-so-far: Opt (top of col)}%
1402 /trace)
1403 (latexrelease | fltrace) \@reqcolroom \ht\@currbox
1404 (*trace)
1405 (latexrelease | fltrace) \fl@trace{float size:
1406 (latexrelease | fltrace) \the \@reqcolroom (addtonextcol)}%
1407 (latexrelease | fltrace)

```

```

1408 </trace>
1409 <latexrelease | fltrace> \advance \@reqcolroom \@textmin
1410 <*trace>
1411 <latexrelease | fltrace> \fl@trace{colroom =
1412 <latexrelease | fltrace> \the \@colroom (addtonextcol)}%
1413 <latexrelease | fltrace> \fl@trace{reqcolroom =
1414 <latexrelease | fltrace> \the \@reqcolroom (addtonextcol)}%
1415 </trace>
1416 <latexrelease | fltrace> \ifdim \@colroom>\@reqcolroom
1417 <latexrelease | fltrace> \@flsetnum \@colnum
1418 <latexrelease | fltrace> \ifnum \@colnum>\z@
1419 <latexrelease | fltrace> \@bitor \@currtype \@deferlist
1420 <*trace>
1421 <latexrelease | fltrace> \fl@trace{deferlist: \@deferlist:
1422 <latexrelease | fltrace> (addtonextcol-before)}%
1423 </trace>
1424 <latexrelease | fltrace> \if@test
1425 <*trace>
1426 <latexrelease | fltrace> \fl@trace{type already on list:
1427 <latexrelease | fltrace> defer (addtonextcol)}%
1428 </trace>
1429 <latexrelease | fltrace> \else
1430 <*trace>
1431 <latexrelease | fltrace> \fl@trace{sent to addtotoporbot
1432 <latexrelease | fltrace> (addtonextcol)}%
1433 </trace>
1434 <latexrelease | fltrace> \@addtotoporbot
1435 <latexrelease | fltrace> \fi
1436 <latexrelease | fltrace> \fi
1437 <*trace>
1438 <latexrelease | fltrace> \else
1439 <latexrelease | fltrace> \fl@trace{Fail---no room: fl box ht:
1440 <latexrelease | fltrace> \the \ht \@currbox (addtonextcol)}%
1441 </trace>
1442 <latexrelease | fltrace> \fi
1443 <latexrelease | fltrace> \fi
1444 <latexrelease | fltrace> \fi
1445 <latexrelease | fltrace> \if@insert
1446 <latexrelease | fltrace> \else
1447 <*trace>
1448 <latexrelease | fltrace> \fl@trace{put back on deferlist
1449 <latexrelease | fltrace> (addtonextcol)}%
1450 </trace>
1451 <latexrelease | fltrace> \@cons \@deferlist \@currbox
1452 <*trace>
1453 <latexrelease | fltrace> \fl@trace{deferlist: \@deferlist:
1454 <latexrelease | fltrace> (addtonextcol-after)}%
1455 </trace>
1456 <latexrelease | fltrace> \fi
1457 <*trace>
1458 <latexrelease | fltrace> \fl@trace{End of addtonextcol --
1459 <latexrelease | fltrace> locally counts:}%
1460 <latexrelease | fltrace> \fl@trace{col: \the \@colnum.
1461 <latexrelease | fltrace> top: \the \@topnum. bot: \the \@botnum.}%

```

```

1462 </trace>
1463 <latexrelease | fltrace> \endgroup
1464 <*trace>
1465 <latexrelease | fltrace> \fl@trace{End of addtonextcol --
1466 <latexrelease | fltrace> globally counts:}%
1467 <latexrelease | fltrace> \fl@trace{col: \the \@colnum.
1468 <latexrelease | fltrace> top: \the \@topnum. bot: \the \@botnum.}%
1469 </trace>
1470 <latexrelease | fltrace>}%
1471 <latexrelease | fltrace>\EndIncludeInRelease

```

\@addtodblcol Lots of changes.

```

1472 <latexrelease | fltrace>\IncludeInRelease{2015/01/01}%
1473 <latexrelease | fltrace> {\@addtodblcol}{float order in 2-column}%
1474 <*2ekernel | latexrelease | fltrace>
1475 \def\@addtodblcol{%
1476 \begingroup
1477 <*trace>
1478 \fl@trace{***Start addtodblcol}%
1479 </trace>
1480 \@insertfalse
1481 \@setfloattypecounts
1482 \@getfpsbit \tw@
1483 <*trace>
1484 \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi dbltop:
1485 \the \@fpstype}%
1486 </trace>
1487 \ifodd\@tempcnta
1488 \@flsetnum \@dbltopnum
1489 \ifnum \@dbltopnum>\z@
1490 \@tempswafalse
1491 \ifdim \@dbltoproom>\ht\@currbox
1492 \@tempswatrue
1493 <*trace>
1494 \fl@trace{Space OK: \@dbltoproom =
1495 \the \@dbltoproom > \the \ht \@currbox
1496 (dbltoproom)}%
1497 </trace>
1498 \else
1499 <*trace>
1500 \fl@trace{fpstype: \the \@fpstype (addtodblcol)}%
1501 </trace>
1502 \ifnum \@fpstype<\sist@@n
1503 <*trace>
1504 \fl@trace{BANG float ignoring \@dbltoproom}%
1505 \fl@trace{\@spaces \@dbltoproom = \the \@dbltoproom.
1506 Ht float: \the \ht \@currbox-BANG}%
1507 </trace>

```

Need to check that there is room on the page, using the local value of \@textmin to make the necessary adjustment to \@dbltoproom.

```

1508 \advance \@dbltoproom \@textmin
1509 <*trace>
1510 \fl@trace{Local value of texmin: \the\@textmin}%

```

```

1511      \fl@trace{\@spaces space on page = \the \@dbltoproom.
1512              Ht float: \the \ht \@currbox-BANG}%
1513    </trace>
1514      \ifdim \@dbltoproom>\ht\@currbox
1515        \@tempwattrue
1516    (*trace)
1517      \fl@trace{Space OK BANG: space on page =
1518              \the \@dbltoproom > \the \ht \@currbox}%
1519      \else
1520        \fl@trace{fpstype: \the \@fpstype}%
1521        \fl@trace{Fail---no room dbltoproom-BANG?:}%
1522        \fl@trace{\@spaces space on page = \the \@dbltoproom.
1523              Ht float: \the \ht \@currbox}%
1524    </trace>
1525      \fi
1526      \advance \@dbltoproom -\@textmin
1527    (*trace)
1528      \else
1529        \fl@trace{fpstype: \the \@fpstype}%
1530        \fl@trace{Fail---no room dbltoproom-ORD?:}%
1531        \fl@trace{\@spaces \@dbltoproom = \the \@dbltoproom.
1532              Ht float: \the \ht \@currbox}%
1533    </trace>
1534      \fi
1535      \fi
1536      \if@tempwa
1537        \@bitor \@currtype \@deferlist
1538    (*trace)
1539      \fl@trace{(dbl)deferlist: \@deferlist: (before)}%
1540    </trace>
1541      not in fixfloats?
1542      \@testwrongwidth\@currbox
1543    \if@test
1544      (*trace)
1545        \fl@trace{type already on list: (dbl)defer}%
1546    </trace>
1547      \else
1548        \@tempdima -\ht\@currbox
1549        \advance\@tempdima
1550        -\ifx \@dbltoplist\@empty \dbltextfloatsep \else
1551          \dblfloatsep \fi
1552        \global \advance \@dbltoproom \@tempdima
1553        \global \advance \@colht \@tempdima
1554        \global \advance \@dbltopnum \m@ne
1555        \@cons \@dbltoplist \@currbox
1556    (*trace)
1557      \fl@trace{dbltopnum (after) = \the \@dbltopnum}%
1558      \fl@trace{***Success: dbltop}%
1559    </trace>
1560      \@inserttrue
1561    \fi
1562  (*trace)

```

```

1563     \else
1564         \fl@trace{Fail: dbltopnum = \the \@dbltopnum: fpstype
1565                                     \the \@fpstype=ORD?}%
1566         \ifnum \@fpstype<\sist@n
1567             \fl@trace{ERROR: !t float not successful (addtodblcol)}%
1568         \fi
1569     \trace>
1570     \fi
1571     \fi
1572     \if@insert
1573     \else
1574     \*trace>
1575         \fl@trace{put on deferlist}%
1576     \trace>
1577         \@cons\@deferlist\@currbox
1578     \*trace>
1579         \fl@trace{(dbl)deferlist: \@deferlist: (after)}%
1580     \trace>
1581     \fi
1582     \*trace>
1583         \fl@trace{End of addtodblcol -- locally count:}%
1584         \fl@trace{dbltop: \the \@dbltopnum.}%
1585     \trace>
1586     \endgroup
1587     \*trace>
1588         \fl@trace{End of addtodblcol -- globally count:}%
1589         \fl@trace{dbltop: \the \@dbltopnum.}%
1590     \trace>
1591 }%
1592 \kernel | latexrelease | fltrace>
1593 \latexrelease | fltrace>\EndIncludeInRelease
1594 \latexrelease | fltrace>\IncludeInRelease{0000/00/00}%
1595 \latexrelease | fltrace> {\@addtodblcol}{float order in 2-column}%
1596 \latexrelease | fltrace>\def\@addtodblcol{%
1597 \latexrelease | fltrace> \begingroup
1598 \*trace>
1599 \latexrelease | fltrace> \fl@trace{***Start addtodblcol}%
1600 \trace>
1601 \latexrelease | fltrace> \insertfalse
1602 \latexrelease | fltrace> \@setfloattyperecounts
1603 \latexrelease | fltrace> \@getfpsbit \tw@
1604 \*trace>
1605 \latexrelease | fltrace> \fl@trace{fpstype \ifodd \@tempcnta OK
1606 \latexrelease | fltrace> \else not \fi dbltop: \the \@fpstype}%
1607 \trace>
1608 \latexrelease | fltrace> \ifodd\@tempcnta
1609 \latexrelease | fltrace> \flsetnum \@dbltopnum
1610 \latexrelease | fltrace> \ifnum \@dbltopnum>\z@
1611 \latexrelease | fltrace> \@tempwafalse
1612 \latexrelease | fltrace> \ifdim \@dbltoproom>\ht\@currbox
1613 \latexrelease | fltrace> \@tempwattrue
1614 \*trace>
1615 \latexrelease | fltrace> \fl@trace{Space OK: \@dbltoproom =
1616 \latexrelease | fltrace> \the \@dbltoproom > \the \ht \@currbox

```

```

1617 \latexrelease | fltrace> (dbltoproom)}%
1618 \tracetrace>
1619 \latexrelease | fltrace> \else
1620 (*tracetrace>
1621 \latexrelease | fltrace> \fl@trace{fpstype: \the \@fpstype (addtodblcol)}%
1622 \tracetrace>
1623 \latexrelease | fltrace> \ifnum \@fpstype<\sist@n
1624 (*tracetrace>
1625 \latexrelease | fltrace> \fl@trace{BANG float ignoring \@dbltoproom}%
1626 \latexrelease | fltrace> \fl@trace{\@spaces \@dbltoproom =
1627 \latexrelease | fltrace> \the \@dbltoproom.
1628 \latexrelease | fltrace> Ht float: \the \ht \@currbox-BANG}%
1629 \tracetrace>

```

Need to check that there is room on the page, using the local value of \@textmin to make the necessary adjustment to \@dbltoproom.

```

1630 \latexrelease | fltrace> \advance \@dbltoproom \@textmin
1631 (*tracetrace>
1632 \latexrelease | fltrace> \fl@trace{Local value of texmin: \the\@textmin}%
1633 \latexrelease | fltrace> \fl@trace{\@spaces space on page =
1634 \latexrelease | fltrace> \the \@dbltoproom.
1635 \latexrelease | fltrace> Ht float: \the \ht \@currbox-BANG}%
1636 \tracetrace>
1637 \latexrelease | fltrace> \ifdim \@dbltoproom>\ht\@currbox
1638 \latexrelease | fltrace> \@tempwattrue
1639 (*tracetrace>
1640 \latexrelease | fltrace> \fl@trace{Space OK BANG: space on page =
1641 \latexrelease | fltrace> \the\@dbltoproom > \the\ht\@currbox}%
1642 \latexrelease | fltrace> \else
1643 \latexrelease | fltrace> \fl@trace{fpstype: \the \@fpstype}%
1644 \latexrelease | fltrace> \fl@trace{Fail---no room dbltoproom-BANG?:}%
1645 \latexrelease | fltrace> \fl@trace{\@spaces space on page =
1646 \latexrelease | fltrace> \the \@dbltoproom.
1647 \latexrelease | fltrace> Ht float: \the \ht \@currbox}%
1648 \tracetrace>
1649 \latexrelease | fltrace> \fi
1650 \latexrelease | fltrace> \advance \@dbltoproom -\@textmin
1651 (*tracetrace>
1652 \latexrelease | fltrace> \else
1653 \latexrelease | fltrace> \fl@trace{fpstype: \the \@fpstype}%
1654 \latexrelease | fltrace> \fl@trace{Fail---no room dbltoproom-ORD?:}%
1655 \latexrelease | fltrace> \fl@trace{\@spaces \@dbltoproom =
1656 \latexrelease | fltrace> \the \@dbltoproom.
1657 \latexrelease | fltrace> Ht float: \the \ht \@currbox}%
1658 \tracetrace>
1659 \latexrelease | fltrace> \fi
1660 \latexrelease | fltrace> \fi
1661 \latexrelease | fltrace> \if@tempswa
1662 \latexrelease | fltrace> \@bitor \@currtype \@dbldeferlist
1663 (*tracetrace>
1664 \latexrelease | fltrace> \fl@trace{dbldeferlist:
1665 \latexrelease | fltrace> \@dbldeferlist: (before)}%
1666 \tracetrace>
1667 \latexrelease | fltrace> \if@test

```

```

1668 (*trace)
1669 \latexrelease|fltrace) \fl@trace{type already on list: dbldefer}%
1670 /trace)
1671 \latexrelease|fltrace) \else
1672 \latexrelease|fltrace) \@tempdima -\ht\@currbox
1673 \latexrelease|fltrace) \advance\@tempdima
1674 \latexrelease|fltrace) -\ifx \@dbltoplist\@empty
1675 \latexrelease|fltrace) \dbltextfloatsep
1676 \latexrelease|fltrace) \else \dblfloatsep \fi
1677 \latexrelease|fltrace) \global \advance \@dbltoproom \@tempdima
1678 \latexrelease|fltrace) \global \advance \@colht \@tempdima
1679 \latexrelease|fltrace) \global \advance \@dbltopnum \m@ne
1680 \latexrelease|fltrace) \@cons \@dbltoplist \@currbox
1681 (*trace)
1682 \latexrelease|fltrace) \fl@trace{dbltopnum (after) =
1683 \latexrelease|fltrace) \the \@dbltopnum}%
1684 \latexrelease|fltrace) \fl@trace{***Success: dbltop}%
1685 /trace)
1686 \latexrelease|fltrace) \@inserttrue
1687 \latexrelease|fltrace) \fi
1688 \latexrelease|fltrace) \fi
1689 (*trace)
1690 \latexrelease|fltrace) \else
1691 \latexrelease|fltrace) \fl@trace{Fail: dbltopnum = \the \@dbltopnum:
1692 \latexrelease|fltrace) fpstype \the \@fpstype=ORD?}%
1693 \latexrelease|fltrace) \ifnum \@fpstype<\sist@n
1694 \latexrelease|fltrace) \fl@trace{ERROR: !t float not successful
1695 \latexrelease|fltrace) (addtodblcol)}%
1696 \latexrelease|fltrace) \fi
1697 /trace)
1698 \latexrelease|fltrace) \fi
1699 \latexrelease|fltrace) \fi
1700 \latexrelease|fltrace) \if@insert
1701 \latexrelease|fltrace) \else
1702 (*trace)
1703 \latexrelease|fltrace) \fl@trace{put on dbldeferlist}%
1704 /trace)
1705 \latexrelease|fltrace) \@cons\@dbldeferlist\@currbox
1706 (*trace)
1707 \latexrelease|fltrace) \fl@trace{dbldeferlist: \@dbldeferlist: (after)}%
1708 /trace)
1709 \latexrelease|fltrace) \fi
1710 (*trace)
1711 \latexrelease|fltrace) \fl@trace{End of addtodblcol -- locally count:}%
1712 \latexrelease|fltrace) \fl@trace{ dbltop: \the \@dbltopnum.}%
1713 /trace)
1714 \latexrelease|fltrace) \endgroup
1715 (*trace)
1716 \latexrelease|fltrace) \fl@trace{End of addtodblcol -- globally count:}%
1717 \latexrelease|fltrace) \fl@trace{dbltop: \the \@dbltopnum.}%
1718 /trace)
1719 \latexrelease|fltrace)}%
1720 \latexrelease|fltrace)\EndIncludeInRelease

```

\@addmarginpar

```

1721 (*2kernel)
1722 \def\@addmarginpar{\@next\@marbox\@currlist{\@cons\@freelist\@marbox
1723   \@cons\@freelist\@currbox}\@latexbug\@tempcnta\@ne
1724   \if@twocolumn
1725     \if@firstcolumn \@tempcnta\m@ne \fi
1726   \else
1727     \if@mparswitch
1728       \ifodd\c@page \else\@tempcnta\m@ne \fi
1729     \fi
1730     \if@reversemargin \@tempcnta -\@tempcnta \fi
1731   \fi
1732   \ifnum\@tempcnta <\z@ \global\setbox\@marbox\box\@currbox \fi
1733   \@tempdima\@mparbottom
1734   \advance\@tempdima -\@pageht
1735   \advance\@tempdima\ht\@marbox
1736   \ifdim\@tempdima >\z@
1737     \@latex@warning@no@line {Marginpar on page \thepage\space moved}%
1738   \else
1739     \@tempdima\z@
1740   \fi
1741   \global\@mparbottom\@pageht
1742   \global\advance\@mparbottom\@tempdima
1743   \global\advance\@mparbottom\dp\@marbox
1744   \global\advance\@mparbottom\marginparpush
1745   \advance\@tempdima -\ht\@marbox

```

Putting box movement inside the ‘marbox’:

```

1746   \global\setbox \@marbox
1747     \vbox {\vskip \@tempdima
1748       \box \@marbox}%
1749   \global \ht\@marbox \z@
1750   \global \dp\@marbox \z@

```

Sticking (rather than gluing:-) the ‘marbox’ to the line above, changed vskip to kern:

```

1751   \kern -\@pagedp
1752   \nointerlineskip
1753   \hb@xt@\columnwidth
1754     {\ifnum \@tempcnta >\z@
1755       \hskip\columnwidth \hskip\marginparsep
1756     \else
1757       \hskip -\marginparsep \hskip -\marginparwidth
1758     \fi
1759     \box\@marbox \hss}%

```

For this reason the following code can vanish:

```

\ nobreak          %% No longer needed. CAR92/12
\ vskip -\@tempdima %% No longer needed. CAR92/12

1760   \nointerlineskip
1761   \hbox{\vrule \@height\z@ \@width\z@ \@depth\@pagedp}}

```

### 64.1.1 Kludgeins

This part of the file is part of the implementation of the following two new commands for L<sup>A</sup>T<sub>E</sub>X2e.

`\enlargethispage{<dim>}`

Adds <dim> to the height of the current column only. On the printed page the bottom of this column is extended downwards by exactly <dim> without having any effect on the placement of the footer; this may result in an overprinting.

`\enlargethispage*{<dim>}`

Similar to `\enlargethispage` but it tries to squeeze the column to be printed in as small a space as possible, ie it uses any shrinkability in the column. If the column was not explicitly broken (e.g. with `\pagebreak`) this may result in an overfull box message but except for this it will come out as expected (if you know what to expect).

The star form of this command is dedicated to Leslie Lamport, the other we need for ourselves (FMi, CAR).

These commands may well have unwanted effects if used soon before a `\clearpage`: please give keep them clear of such places.

`\@kludgeins` The insert which makes T<sub>E</sub>X do a lot of the necessary work. All we need to put into it is the amount by which the pagegoal should be changed.

```
1762 \newinsert \@kludgeins
1763 \global\dimen\@kludgeins \maxdimen
1764 \global\count\@kludgeins 1000
```

`\enlargethispage` The user command.

```
\enlargethispage* 1765 \gdef \enlargethispage {%
1766     \ifstar
1767     {%
1768     (*trace)
1769         \fl@trace{Enlarging page height * }%
1770     </trace>
1771         \@enlargepage{\hbox{\kern\p@}}}%
1772     {%
1773     (*trace)
1774         \fl@trace{Enlarging page height exactly---}%
1775     </trace>
1776         \@enlargepage\@empty}%
1777 }
```

`\@enlargepage` This actually inserts the insert, after checking for extreme values of the change.

```
1778 \gdef\@enlargepage#1#2{%
1779     (*trace)
1780     \fl@trace{\@spaces\@spaces by #2}%
1781     </trace>
1782     \@tempkipa#2\relax
1783     \ifdim \@tempkipa>.5\maxdimen
```

```

1784     \@latexerr{Suggested\space extra\space height\space
1785               (\the\@tempskipa)\space dangerously\space
1786               large}\@eha
1787   \else
1788     \ifdim \vsize<.5\maxdimen
1789 (*trace)
1790     \fl@trace {Kludgeins added--pagegoal before: \the\pagegoal}%
1791 (/trace)
1792     \@bsphack
1793     \insert\@kludgeins{#1\vskip-\@tempskipa}%
1794     \@esphack
    This next bit is for tracing only:
1795 (*trace)
1796     \ifvmode \par
1797     \fl@trace {Kludgeins added--pagegoal after: \the\pagegoal}%
1798     \fi
1799 (/trace)
1800   \else
1801     \@latexerr{Page\space height\space already\space
1802               too\space large}\@eha
1803   \fi
1804 \fi
1805 }
1806 (/2ekernel)

```

### 64.1.2 Float control

This part implements controllable floats and other changes to the float mechanism.

It provides, at the document level, the following command for inclusion in L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>.

`\suppressfloats`

This suppresses all further floats on the current page.

With an optional argument it suppresses only floats only in certain positions on the current page.

[t] suppresses only floats at the top of the page [b] suppresses only floats at the bottom of the page

It also enables the use of an extra specifier, !, in the location optional argument of a float. If this is present then, just for this particular float, whenever it is processed by the float mechanism the followinghg are ignored:

- all restrictions on the number of floats which can appear;
- all explicit restrictions on the amount of space which should (not) be occupied by floats and/or text.

The mechanism will still attempt to ensure that pages are not overfull.

These specifiers override, for the single float, the suppression commands described above.

In its current form, it also supplies a reasonably exhaustive, and somewhat baroque, means of tracing some aspects of the float mechanism.

More tracing.

```

\fl@trace Set-up tracing for floats independent of other tracing as it produces mega-output.
\tracefloatsoff Default is no tracing.
\tracefloats 1807 \*fltrace
\fl@traceval 1808 \def \fl@tracemessage #1{\let\@elt\@empty\typeout{LaTeX2e: #1}}
\tracefloatvals 1809 \def \tracefloats{\let \fl@trace \fl@tracemessage}
\fl@tracemessage 1810 \def \tracefloatsoff {\let \fl@trace \@gobble}
1811 \tracefloatsoff
1812 \def \fl@traceval #1{\fl@trace{\string #1 = \the #1}}
1813 \IncludeInRelease{2015/01/01}{\tracefloatvals}%
1814 {trace float vals}%
1815 \def \tracefloatvals{%

```

As `\@dblfloatplacement` sets `\f@depth` it needs to be run inside a group, otherwise the float placement will test for the wrong value.<sup>8</sup>

```

1816 \begingroup
1817 \@dblfloatplacement
1818 \@floatplacement
1819 \fl@trace{***Float placement parameters:}%
1820 \fl@traceval\@colnum
1821 \fl@traceval\@colroom
1822 \fl@traceval\@topnum
1823 \fl@traceval\@toproom
1824 \fl@traceval\@botnum
1825 \fl@traceval\@botroom
1826 \fl@traceval\@fpmin
1827 \fl@trace{\string\textfraction = \textfraction}%
1828 \fl@traceval\@dbltopnum
1829 \fl@traceval\@dbltoproom
1830 \fl@trace{\string\textfraction = \textfraction}%
1831 \fl@trace{toplist: \@toplist}%
1832 \fl@trace{botlist: \@botlist}%
1833 \fl@trace{midlist: \@midlist}%
1834 \fl@trace{deferlist: \@deferlist}%
1835 \fl@trace{dbltoplist: \@dbltoplist}%
1836 %Fmi \fl@trace{dbldeferlist: \@dbldeferlist}%
1837 \endgroup
1838 }
1839 \EndIncludeInRelease
1840 \IncludeInRelease{0000/00/00}{\tracefloatvals}%
1841 {trace float vals}%
1842 \def \tracefloatvals{%
1843 \begingroup
1844 \@dblfloatplacement
1845 \@floatplacement
1846 \fl@trace{***Float placement parameters:}%
1847 \fl@traceval\@colnum
1848 \fl@traceval\@colroom
1849 \fl@traceval\@topnum

```

---

<sup>8</sup>This is a somewhat questionable design.

```

1850 \fl@traceval\@toproom
1851 \fl@traceval\@botnum
1852 \fl@traceval\@botroom
1853 \fl@traceval\@fpmin
1854 \fl@trace{\string\textfraction = \textfraction}%
1855 \fl@traceval\@dbltopnum
1856 \fl@traceval\@dbltoproom
1857 \fl@trace{\string\textfraction = \textfraction}%
1858 \fl@trace{toplist: \@toplist}%
1859 \fl@trace{botlist: \@botlist}%
1860 \fl@trace{midlist: \@midlist}%
1861 \fl@trace{deferlist: \@deferlist}%
1862 \fl@trace{dbltoplist: \@dbltoplist}%
1863 % next line only in old releases
1864 \fl@trace{dbldeferlist: \@dbldeferlist}%
1865 \endgroup
1866 }
1867 \EndIncludeInRelease

```

We need to make sure that `fltrace` comes before `flafter` to make the tracing work.

```

1868 \@ifpackageloaded{flafter}
1869 {\PackageWarningNoLine
1870   {fltrace}{Load 'fltrace' before 'flafter'\MessageBreak
1871             Attempting to recover by reloading 'flafter'}}%

```

Hide the fact that `flafter` was already loaded and then request it anew.

```

1872 \expandafter\let\csname ver@flafter.sty\endcsname\relax
1873 \def\reserved@a#1{%
1874   \expandafter\let\csname\string#1+flafter+IIR\endcsname\relax}%
1875 \reserved@a\@addtocurcol
1876 \reserved@a\@addtonextcol
1877 \RequirePackage{flafter}}{}
1878 </fltrace>

```

As the code for `flafter` will contain tracing calls so that it works in conjunction with `fltrace` we need to provide a dummy definition for `\fl@trace` in that package.

```

1879 (*flafter)
1880 \providecommand\fl@trace[1]{}
1881 </flafter>

```

`\suppressfloats` Float suppression commands: these set the relevant counter globally to zero. Thus `\@flstop` they are overridden for a particular float by an `!` specifier.

```

1882 (*2kernel)
1883 \def \suppressfloats {%
1884   \ifnextchar [%
1885     \@flstop
1886     {\global \@colnum \z@}%
1887 }

```

Maybe this should be a loop over `#1`?

```

1888 \def \@flstop [#1]{%
1889   \if t#1%
1890     \global \@topnum \z@

```

```

1891 \fi
1892 \if b#1%
1893 \global \@botnum \z@
1894 \fi
1895 }

```

Manipulation of float placement and type; both their strings and the corresponding count registers.

`\@fpstype` First a new count register to go with `\@currtype`.  
`\@reqcolroom` Then a new skip register, for information needed to remove the `\@maxsep` conservatism: it is possible that this could use a temporary register.  
`\@textfloatsheight` Finally a dimension register to hold the total height of in-text floats on the current page. This is needed to implement a major change in the functionality of `\@addtocurcol` which is, nevertheless, a bug fix. It is not local and therefore cannot be a temporary register.

```

1896 \newcount \@fpstype
1897 \newdimen \@reqcolroom
1898 \newdimen \@textfloatsheight
1899 \endkernel

```

`\@fpsadddefault` Adds the default placement to what is already there.  
Should not need to change this, but could do it as follows:

```

\def \@fpsadddefault {%
  \temptokena \expandafter\expandafter\expandafter
    {\csname fps@\@capytype \endcsname}%
  \edef \reserved@a {\the\temptokena}%
  \onelevel@sanitize \reserved@a
  \edef \@fps {\@fps\reserved@a}%
}

```

```

1900 \endkernel\fltrace)
1901 \def \@fpsadddefault {%
1902 \trace)
1903 \fl@trace{fps changed from: \@fps}%
1904 \trace)
1905 \edef \@fps {\@fps\csname fps@\@capytype \endcsname}%
1906 \latex@warning {%
1907 No positions in optional float specifier.\MessageBreak
1908 Default added (so using '\@fps')}%
1909 }

```

`\@setfloattypescounts` Sets counters `\@fpstype` and `\@currtype`.  
BANG == bit4 of `\count\@currbox` = 0.

```

1910 \def \@setfloattypescounts {%
1911 \@currtype \count\@currbox
1912 \@fpstype \count\@currbox
1913 \divide\@currtype\@xxxii \multiply\@currtype\@xxxii
1914 \advance \@fpstype -\@currtype
1915 \trace)
1916 \fl@trace{(mod 32) fpstype: \the \@fpstype}%
1917 \fl@trace{(mult of 32) currtype: \the \@currtype}%

```

```

1918 % Tracing only: but some should be changed into real errors/warnings?
1919 \ifnum \@fpstype<\sist@n
1920   \ifnum \@fpstype=\z@
1921     \fl@trace{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 0?}%
1922   \fi
1923   \ifnum \@fpstype=\@ne
1924     \fl@trace{WARNING: only h, fpstype = \the \@fpstype = 1?}%
1925   \fi
1926   \fl@trace{BANG float}%
1927 \else
1928   \ifnum \@fpstype=\sist@n
1929     \fl@trace{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 16?}%
1930   \fi
1931   \ifnum \@fpstype=17
1932     \fl@trace{WARNING: only h, fpstype = \the \@fpstype = 17?}%
1933   \fi
1934   \fl@trace{ORD float}%
1935 \fi
1936 /trace>
1937 }
1938 /2ekernel | fltrace)

```

Macros for getting, testing and setting bits of the fps.

`\@getfpsbit` Sets `\@tempcnta` to required bit of `\count\@currbox`.

```

1939 (*2ekernel)
1940 \def \@getfpsbit {%
1941   \@boxfpsbit \@currbox
1942 }

```

`\@boxfpsbit` Used above.

```

1943 \def \@boxfpsbit #1#2{%
1944   \@tempcnta \count#1
1945   \divide \@tempcnta #2\relax
1946 }

```

`\@testfp` New definition of the float page test.

```

1947 \def \@testfp #1{%
1948   \@boxfpsbit #18\relax % Really ‘#1 8’ for human readers!
1949   \ifodd \@tempcnta
1950   \else
1951     \@testtrue
1952   \fi
1953 }

```

`\@setfpsbit` Sets required bit of `\@tempcnta` (to 1).

```

1954 \def \@setfpsbit #1{%
1955   \@tempcntb \@tempcnta
1956   \divide \@tempcntb #1\relax
1957   \ifodd \@tempcntb
1958   \else
1959     \advance \@tempcnta #1\relax
1960   \fi

```

```

1961 }
1962 </2ekernel>

```

**\@resetfhps** Globally adds t as a possible location for an h or !h only placement: this must be done using the count.

Although it will leave \@fpstype set to 17 even if it was originally 1, this does not matter since it is the last thing in \@addtocurcol.

```

1963 (*2ekernel | fltrace)
1964 \def \@resetfhps {%
1965   \let\reserved@a\@empty
1966   \ifnum \@fpstype=\@ne
1967     \def \reserved@a {!}%
1968     \@fpstype 17
1969   \fi
1970   \ifnum \@fpstype=17
1971     \global \advance \count\@currbox \tw@
1972     \@latex@warning@no@line {%
1973       '\reserved@a h' float specifier changed to '\reserved@a ht'}%
1974   (*trace)
1975     \fl@trace{%
1976       't' added to '\reserved@a h'- new Count: \the \count\@currbox}%
1977   </trace>
1978   \fi
1979 }

```

Special stuff for BANG floats.

**\@flsetnum** Ignores any zero float counter value in case BANG.

It uses a local assignment to the normally global counter: a bit naughty, perhaps?

These assignments are safe so long as the counter involved is only consulted once (i.e. only for the 'bang float') with the changed value. This is the case within \@addtocurcol because it is used only once within a call of the output routine (which forms a group).

For \@addtonextcol this is achieved by putting a group around its code; this is needed because it is called (by \@startcolumn) for each float which was on the deferlist. Almost identical considerations pertain to \@addtodblcol. There may be more efficient ways to handle this, but the group seems to be the simplest.

```

1980 \def \@flsetnum #1{%
1981   (*trace)
1982     \fl@trace{fpstype: \the \@fpstype (flsetnum \string#1)}%
1983   </trace>
1984   \ifnum \@fpstype<\sist@n
1985     \ifnum #1=\z@
1986   (*trace)
1987     \fl@trace{BANG float resetting \string#1 to 1}%
1988   </trace>
1989     #1\@ne
1990   \fi
1991   \fi
1992   (*trace)
1993     \fl@trace{#1 (before) = \the #1}%
1994   </trace>

```

1995 }

\@flsettextmin This ignores \textfraction space restriction in case BANG.

```

1996 \def \@flsettextmin {%
1997 \*trace
1998   \fl@trace{fpstype: \the \@fpstype (flsettextmin)}%
1999 \*trace
2000   \ifnum \@fpstype<\sixt@@n
2001 \*trace
2002     \fl@trace{BANG ignoring textmin}%
2003 \*trace
2004     \@textmin \z@
2005   \else
2006     \@textmin \textfraction\@colht
2007 \*trace
2008     \fl@trace{ORD textmin = \the \@textmin}%
2009 \*trace
2010   \fi
2011 }
```

\@flcheckspace This ignores space restriction in case BANG; this is still slightly conservative since it does not allow for the fact that, if there is no text in the column then \textfloatsep is not needed. Sets @tempswa true if there is room for \@currbox.

```

2012 \def \@flcheckspace #1#2{%
2013   \advance \@reqcolroom
2014   \ifx #2\@empty \textfloatsep \else \floatsep \fi
2015 \*trace
2016   \fl@trace{colroom = \the \@colroom
2017                                     (flcheckspace \string#1 \string#2)}%
2018   \fl@trace{reqcolroom = \the \@reqcolroom
2019                                     (flcheckspace \string#1 \string#2)}%
2020 \*trace
2021   \ifdim \@colroom>\@reqcolroom
2022     \ifdim #1>\ht\@currbox
2023       \@tempwattrue
2024 \*trace
2025     \fl@trace{Space OK: #1 = \the #1 > \the \ht \@currbox
2026                                     (flcheckspace \string#1 \string#2)}%
2027 \*trace
2028   \else
2029 \*trace
2030     \fl@trace{fpstype: \the \@fpstype
2031                                     (flcheckspace \string#1 \string#2)}%
2032 \*trace
2033     \ifnum \@fpstype<\sixt@@n
2034 \*trace
2035       \fl@trace{BANG float ignoring #1
2036                                     (flcheckspace \string#1 \string#2):}%
2037       \fl@trace{\@spaces #1 = \the #1. Ht float: \the \ht \@currbox
2038                                     BANG}%
2039 \*trace
2040     \@tempwattrue
2041 \*trace
```

```

2042     \else
2043     \fl@trace{Fail---no room (flcheckspace \string#1 \string#2)
2044             (fpstype \the \@fpstype=ORD?):}%
2045     \fl@trace{\@spaces #1 = \the #1. Ht float: \the \ht \@currbox
2046             ORD?:}%
2047 \end{trace}
2048 \fi
2049 \fi
2050 \end{trace}
2051 \else
2052     \fl@trace{Fail---no room at 2nd test of colroom
2053             (flcheckspace \string#1 \string#2):}%
2054 \end{trace}
2055 \fi
2056 }
2057 \end{2ekernel} \end{fltrace}

```

`\@flupdates` This updates everything when a float is placed.

```

2058 \end{2ekernel}
2059 \def \@flupdates #1#2#3{%
2060     \global \advance #1\m@ne
2061     \global \advance \@colnum \m@ne
2062     \@tempdima -\ht\@currbox
2063     \advance \@tempdima
2064     -\ifx #3\@empty \textfloatsep \else \floatsep \fi
2065     \global \advance #2\@tempdima
2066     \global \advance \@colroom \@tempdima
2067     \@cons #3\@currbox
2068 }
2069 \end{2ekernel}

```

Interesting facts about float mechanisms past and present, together with a summary of various features, some unresolved:

1. The value `\textfraction` does not affect the processing of doublecol floats: this seems sensible, but should be documented.
2. `\twocolumn` floatplacement was wrong: `dbl` not needed, `ord` needed.
3. `\floatplacement` was not called after `\startdblcol` or `\topnewpage`. This has been changed; it is clearly a bug fix.
4. The use `\@topnewpage` when `\dblfigrule` is non-trivial produced a rule in the wrong place. This has been fixed by not using `\dblfigrule` when processing the ‘float’ from `\@topnewpage`.
5. If the specifier was just `h` and the float could not be put here, it went on the deferlist and stayed there until a clearpage. It now gets changed to a ‘th’: this is only an error-recovery action, putting just `h` or `!h` should be deprecated.
6. `\@dblmaxsep` was ‘the maximum of `\dblfloatsep` and `\dbltextfloatsep`’. But it was never used! Now gone completely, like `\@maxsep`.

7. After an h float is put on a page, it was counted as text when applying the `\textfraction` test; this is possibly too big a change although it is a bug fix?
8. Two consecutive h floats are separated by twice `\intertextsep`: this could be changed to one by use of `\addvspace`, OK? Note that it would also mean that less space is put in if an h float immediately follows other spaces. This is also possibly too big a change, at least for compatibility mode? Or it may be simply wrong! It has not been changed.
9. Now `\@addtocurcol` checks first for just p fps. I think that this is an increase in efficiency, but maybe the coding should be made even more efficient.
10. `\@tryfcolumn` now tests if the list is empty first, otherwise lots of wasted time! Thus this test has been removed from `\@startcolumn`. As Frank pointed out, this makes `\@startcolumn` less efficient. But it is now the same as `\@startdblcolumn`: I can see no reason why they should be different, but which is best?
11. Why is `\@colroom` set in `\@doclearpage`?
12. Footnotes. Check what `\clearpage` does when footnotes are left over. Footnotes are not put on float pages and, also, `\@addtonextcol` ignores the existence of held-over footnotes in deciding what floats can go on the page. Not changed.
13. `\clearpage` can still lose non-boxes, at least when floats are involved. It also moves some to the ‘wrong page’, but this may be a coding problem.
14. The `!` option makes it necessary to check in `\output` that there is enough room left on the page after adding a float. (This would have been necessary anyway if anyone set `\@textmin` too close to zero! A similar danger existed also if the text in a `\twocolumn[text]` entity gets too large.) The current implementation of this also makes the normal case a little less efficient, OK? Not enough room means, at present, less than `\baselineskip`, with a warning: is this OK? Should it be made generic (another parameter)?
15. There are four possibilities for supporting this:  
`\twocolumn[\maketitle more text]`  
 One is to change `\maketitle` slightly to allow this. Another is to change `\@topnewpage` so that more than one `\twocolumn[]` command is allowed; in this case `\maketitle\twocolumn[more text]` will work. The former is more robust from the user’s viewpoint, but makes the code for `\maketitle` rather ad hoc (maybe it is already?). Another is to misuse the global `twocolumn` flag locally within `\@topnewpage`. Yet another is to move the column count register from the multicol package into the kernel. This has been done.
16. Where should the reinserts be put to maximise the probability that footnotes come out on the correct page? Or should we go for as much compatibility as possible (but see next item)?

17. Should we continue to support (as much as possible) `\samepage`? Some of its intended functionality is now advertised as being provided by `\enlargethispage`. Use of either is likely to result in wrongly placed footnotes, marginals, etc. Which should have priority: obeying the pagination instructions, or correct placement of notes/marginalia?
18. Is the adjustment of space to cause shrinking in the kludge-\* case correct? Should it be limited to 0pt?
19. Is the setting of `\boxmaxdepth` in `makecol` and friends needed? It only has any effect if `\@textbottom` ends with a box or rule, in which case the `vskip` to allow for its depth should also be added. If it is kept, it should probably be the last thing in the box. It has now been removed.  
  
It would perhaps be better to document that `\@textbottom` and `\@texttop` must have natural height 0pt.
20. I cannot see why the `vskip` adjustment for the depth is needed if `boxmaxdepth` is used to ensure that there is never a too deep box.
21. The value of `\boxmaxdepth` should be explicitly set whenever necessary: it is too risky to assume that it has any particular value. Care is needed in deciding what to set it to.  
  
It is interesting to note that the value of `\boxmaxdepth` is unique in being read before the local settings for the box group are reset; all other parameter settings which affect the box construction use their values outside the box group.
22. Should `\@maxdepth` store the setting of `\maxdepth` from `lplain`? Or should we provide a proper interface to class files for setting these?

An analysis of various other macros.

`\@opcol` should do `\@floatplacement`, but where? Right at the end, since it always occurs at the start of a column.

```
\def\@opcol{%
% Why is this done first?
\global \@mparbottom \z@
\if@twocolumn
\@outputdblcol
\else
\@outputpage
% This is not needed since it is done at the end of
% |\@outputpage|:
\global \@colht \textheight
\fi}
```

Only tracing has been added to these.

```
2070 (*2ekernel | fltrace)
2071 \def\@makefcolumn #1{%
2072   \begingroup
2073     \@fpmin \z@
2074     \let \@testfp \@gobble
```

```

2075 \@tryfcolumn #1%
2076 \endgroup
2077 (*trace)
2078 \if@fcolmade
2079 \fl@trace{PAGE: in \string\clearpage
2080 \if@twocolumn ---twocolumn\fi---}%
2081 \fl@trace{----- float column/page completed from \string#1}%
2082 \fi
2083 /trace)
2084 }

```

This will line up the last baselines in the two columns provided they are constructed in the normal way: i.e. ending in a skip of minus the original depth, with `\@textbottom` adding nothing.

Thus again it is essential for `\@textbottom` to have depth 0pt.

```

2085 /2kernel | fltrace)
2086 (latexrelease | fltrace)\IncludeInRelease{2015/01/01}%
2087 (latexrelease | fltrace) {\@outputdblcol}{2 column marks}%
2088 (*2kernel | fltrace | latexrelease)

```

This is just a change to the single command `\@outputdblcol` so that it saves mark information for the first column and restores it in the second column.

```

2089 \def\@outputdblcol{%
2090 \if@firstcolumn
2091 \global\@firstcolumnfalse

```

Save the left column

```

2092 \global\setbox\@leftcolumn\copy\@outputbox
2093 (fltrace) \fl@trace{PAGE: first column boxed}%

```

Remember the marks from the first column

```

2094 \splitmaxdepth\maxdimen
2095 \vbadness\maxdimen

```

In case of `\enlargethispage` we will have infinite negative glue at the bottom of the page (coming from `\vss`) and that will earn us an error message if we `\vsplit` to get at the marks. So we need to remove the last glue (if any) at the end of `\@outputbox` as we are only interested in marks that change doesn't matter.

```

2096 \setbox\@outputbox\vbox{\unvbox\@outputbox\unskip}%
2097 \setbox\@outputbox\vsplit\@outputbox to\maxdimen

```

One minor difference from the current `fixmarks` package, pass the marks through a token register to stop any `#` tokens causing an error in a `\def`.

```

2098 \toks@\expandafter{\topmark}%
2099 \xdef\@firstcoltopmark{\the\toks@}%
2100 \toks@\expandafter{\splitfirstmark}%
2101 \xdef\@firstcolfirstmark{\the\toks@}%

```

This test does not work if truly empty marks have been inserted, but  $\text{\LaTeX}$  marks should always have (at least) two brace groups. (Except before the first mark is used, when the marks are empty, but that is OK here.)

```

2102 \ifx\@firstcolfirstmark\@empty
2103 \global\let\@setmarks\relax
2104 \else
2105 \gdef\@setmarks{%
2106 \let\firstmark\@firstcolfirstmark

```

```

2107      \let\topmark\@firstcoltopmark}%
2108      \fi
      End of change
2109      \else
2110      \global\@firstcolumntrue
2111      \setbox\@outputbox\vbox{%
2112      \hb@xt@\textwidth{%
2113      \hb@xt@\columnwidth{\box\@leftcolumn \hss}%
2114      \hfil
      The color of the \vrule should be \normalcolor as to not inherit the color from
      the column.
2115      {\normalcolor\vrule \@width\columnseprule}%
2116      \hfil
2117      \hb@xt@\columnwidth{\box\@outputbox \hss}}}%
2118      \if@trace \fl@trace{PAGE: second column also boxed}%
2119      \@combinedblfloats
      Override current first and top with those of first column if necessary
2120      \@setmarks
      End of change
2121      \@outputpage
2122      \if@trace \fl@trace{PAGE: two column page completed}%
2123      \begingroup
2124      \@dblfloatplacement
2125      \@startdblcolumn
2126      \@whilesw\if@fcolmade \fi{\@outputpage
2127      \if@trace \fl@trace{PAGE: double float page completed}%
2128      \@startdblcolumn}%
2129      \endgroup
2130      \fi}%
2131      \if@latexrelease \if@trace \EndIncludeInRelease
2132      \if@latexrelease \if@trace \IncludeInRelease{0000/00/00}%
2133      \if@latexrelease \if@trace {\@outputdblcol}{2 column marks}%
2134      \if@latexrelease \if@trace \def\@outputdblcol{%
2135      \if@firstcolumn
2136      \if@latexrelease \if@trace \global \@firstcolumnfalse
2137      \if@latexrelease \if@trace \global \setbox\@leftcolumn \box\@outputbox
2138      \if@trace \fl@trace{PAGE: first column boxed}%
2139      \if@latexrelease \if@trace \fl@trace{PAGE: first column boxed}%
2140      \if@trace \fl@trace{PAGE: first column boxed}%
2141      \if@latexrelease \if@trace \else
2142      \if@latexrelease \if@trace \global \@firstcolumntrue
2143      \if@latexrelease \if@trace \setbox\@outputbox \vbox {%
2144      \if@latexrelease \if@trace \hb@xt@\textwidth {%
2145      \if@latexrelease \if@trace \hb@xt@\columnwidth {%
2146      \if@latexrelease \if@trace \box\@leftcolumn \hss}%
2147      \if@latexrelease \if@trace \hfil
2148      \if@latexrelease \if@trace {\normalcolor\vrule
2149      \if@latexrelease \if@trace \@width\columnseprule}%
2150      \if@latexrelease \if@trace \hfil
2151      \if@latexrelease \if@trace \hb@xt@\columnwidth {%
2152      \if@latexrelease \if@trace \box\@outputbox \hss}%

```

```

2153 <latexrelease | fltrace>                                     }%
2154 <latexrelease | fltrace>                                     }%
2155 <*trace>
2156 <latexrelease | fltrace>      \fl@trace{PAGE: second column also boxed}%
2157 </trace>
2158 <latexrelease | fltrace>      \@combinedblfloats
2159 <latexrelease | fltrace>      \@outputpage
2160 <*trace>
2161 <latexrelease | fltrace>      \fl@trace{PAGE: two column page completed}%
2162 </trace>
2163 <latexrelease | fltrace>      \begingroup
2164 <latexrelease | fltrace>      \@dblfloatplacement
2165 <latexrelease | fltrace>      \@startdblcolumn

```

This loop could be replaced by an `\expandafter` tail recursion in `\@startdblcolumn`.

```

2166 <latexrelease | fltrace>      \@whilesw@if@fcolmade \fi
2167 <latexrelease | fltrace>      {\@outputpage
2168 <*trace>
2169 <latexrelease | fltrace>      \fl@trace{PAGE: double float page completed}%
2170 </trace>
2171 <latexrelease | fltrace>      \@startdblcolumn}%
2172 <latexrelease | fltrace>      \endgroup
2173 <latexrelease | fltrace>      \fi
2174 <latexrelease | fltrace>}%
2175 <latexrelease | fltrace>\EndIncludeInRelease
2176 </2ekernel | fltrace | latexrelease>

```

### 64.1.3 Float placement parameters

The main purpose of this section is to ensure that all the float-placement parameters which need to be set in a class file or package have been declared. It also describes their use and sets values for them which are reasonable for typical documents using US letter or A4 sized paper.

#### Limits for the placement of floating objects

<code>\c@topnumber</code>	This counter holds the maximum number of floats that can appear at the top of a text page or column.
<pre> 2177 &lt;*2ekernel&gt; 2178 \newcount\c@topnumber 2179 \setcounter{topnumber}{2} </pre>	
<code>\topfraction</code>	This macro holds the maximum proportion (as a decimal number) of a text page or column that can be occupied by floats at the top.
<pre> 2180 \newcommand\topfraction{.7} </pre>	
<code>\c@bottomnumber</code>	This counter holds the maximum number of floats that can appear at the bottom of a text page or column.
<pre> 2181 \newcount\c@bottomnumber 2182 \setcounter{bottomnumber}{1} </pre>	

`\bottomfraction` This macro holds the maximum proportion (as a decimal number) of a text page or column that can be occupied by floats at the bottom.  
2183 `\newcommand\bottomfraction{.3}`

`\c@totalnumber` This counter holds the maximum number of floats that can appear on any text page or column.  
2184 `\newcount\c@totalnumber`  
2185 `\setcounter{totalnumber}{3}`

`\textfraction` This macro holds the minimum proportion (as a decimal number) of a text page or column that must be occupied by text.  
2186 `\newcommand\textfraction{.2}`

`\floatpagefraction` This macro holds the minimum proportion (as a decimal number) of a page or column that must be occupied by floating objects before a ‘float page’ is produced.  
2187 `\newcommand\floatpagefraction{.5}`

`\c@dbltopnumber` This counter holds the maximum number of double-column floats that can appear on the top of a two-column text page.  
2188 `\newcount\c@dbltopnumber`  
2189 `\setcounter{dbltopnumber}{2}`

`\dbltopfraction` This macro holds the maximum proportion (as a decimal number) of a two-column text page that can be occupied by double-column floats at the top.  
2190 `\newcommand\dbltopfraction{.7}`

`\dblfloatpagefraction` This macro holds the minimum proportion (as a decimal number) of a page that must be occupied by double-column floating objects before a ‘double-column float page’ is produced.  
2191 `\newcommand\dblfloatpagefraction{.5}`

### Floats on a text page

`\floatsep` When a floating object is placed on a page with text, these parameters control the separation between the float and the other objects on the page. These parameters  
`\textfloatsep` are used for both one-column mode and single-column floats in two-column mode.  
`\intextsep` They are all rubber lengths.

`\floatsep` is the space between adjacent floats that are placed at the top or bottom of the text page or column.

`\textfloatsep` is the space between the main text and floats at the top or bottom of the page or column.

`\intextsep` is the space between in-text floats and the text.

2192 `\newskip\floatsep`

2193 `\newskip\textfloatsep`

2194 `\newskip\intextsep`

2195 `\setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}`

2196 `\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}`

2197 `\setlength\intextsep {12\p@ \@plus 2\p@ \@minus 2\p@}`

`\dblfloatsep` `\dbltextfloatsep` When double-column floats (floating objects that span the whole `\textwidth`) are placed at the top of a text page in two-column mode, the separation between the float and the text is controlled by `\dblfloatsep` and `\dbltextfloatsep`. They are rubber lengths.

`\dblfloatsep` is the space between adjacent double-column floats placed at the top of the text page.

`\dbltextfloatsep` is the space between the main text and double-column floats at the top of the page.

```
2198 \newskip\dblfloatsep
2199 \newskip\dbltextfloatsep
2200 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
2201 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

### Floats on their own page or column

`\@fptop` `\@fpsep` `\@fpbot` When floating objects are placed on a separate page or column, called a ‘float page’, the layout of the page is controlled by these parameters, which are rubber lengths.

At the top of the page `\@fptop` is inserted; typically this supplies some stretchable whitespace. At the bottom of the page `\@fpbot` is inserted. Between adjacent floats `\@fpsep` is inserted.

These parameters are used for all floating objects on a ‘float page’ in one-column mode, and for single-column floats in two-column mode.

Note that at least one of the two parameters `\@fptop` and `\@fpbot` should contain a `plus ...fil` so as to fill the remaining empty space.

```
2202 \newskip\@fptop
2203 \newskip\@fpsep
2204 \newskip\@fpbot
2205 \setlength\@fptop{0\p@ \@plus 1fil}
2206 \setlength\@fpsep{8\p@ \@plus 2fil}
2207 \setlength\@fpbot{0\p@ \@plus 1fil}
```

`\@dblfpptop` Double-column ‘float pages’ in two-column mode use similar parameters.

```
\@dblfpsep 2208 \newskip\@dblfpptop
\@dblfpbot 2209 \newskip\@dblfpsep
2210 \newskip\@dblfpbot
2211 \setlength\@dblfpptop{0\p@ \@plus 1fil}
2212 \setlength\@dblfpsep{8\p@ \@plus 2fil}
2213 \setlength\@dblfpbot{0\p@ \@plus 1fil}
```

`\topfigrule` `\botfigrule` The macros can be used to put in rules between floats and text; whatever they insert should be vertical mode material which takes up zero space.

```
\dblfigrule 2214 \let\topfigrule=\relax
2215 \let\botfigrule=\relax
2216 \let\dblfigrule=\relax
2217 </2kernel>
```

# File L

## ltclass.dtx

### 65 Introduction

This file implements the following declarations, which replace `\documentstyle` in  $\text{\LaTeX 2}_{\epsilon}$  documents.

Note that old documents containing `\documentstyle` will be run using a compatibility option—thus keeping everyone happy, we hope!

The overall idea is that there are two types of ‘style files’: ‘class files’ which define elements and provide a default formatting for them; and ‘packages’ which provide extra functionality. One difference between  $\text{\LaTeX 2}_{\epsilon}$  and  $\text{\LaTeX 2.09}$  is that  $\text{\LaTeX 2}_{\epsilon}$  packages may have options. Note that options to classes/packages may be implemented such that they input files, but these file names are not necessarily directly related to the option name.

### 66 User interface

`\documentclass[<main-option-list>]{<class>}[<version>]`

There must be exactly one such declaration, and it must come first. The *<main-option-list>* is a list of options which can modify the formatting of elements which are defined in the *<class>* file as well as in all following `\usepackage` declarations (see below). The *<version>* is a version number, beginning with a date in the format YYYY/MM/DD. If an older version of the class is found, a warning is issued.

`\documentstyle[<main-option-list>]{<class>}[<version>]`

The `\documentstyle` declaration is kept in order to maintain upward compatibility with  $\text{\LaTeX 2.09}$  documents. It is similar to `\documentclass`, but it causes all options in *<main-option-list>* that the *<class>* does not use to be passed to `\RequirePackage` after the options have been processed. This maintains compatibility with the 2.09 behaviour. Also a flag is set to indicate that the document is to be processed in  $\text{\LaTeX 2.09}$  compatibility mode. As far as most packages are concerned, this only affects the warnings and errors  $\text{\LaTeX}$  generates. This flag does affect the definition of font commands, and `\sloppy`.

`\usepackage[<package-option-list>]{<package-list>}[<version>]`

There can be any number of these declarations. All packages in *<package-list>* are called with the same options.

Each *<package>* file defines new elements (or modifies those defined in the *<class>*), and thus extends the range of documents which can be processed. The *<package-option-list>* is a list of options which can modify the formatting of elements defined in the *<package>* file. The *<version>* is a version number, beginning with a date in the format YYYY/MM/DD. If an older version of the package is found, a warning is issued.

Each package is loaded only once. If the same package is requested more than once, nothing happens, unless the package has been requested with options that were not given the first time it was loaded, in which case an error is produced.

As well as processing the options given in the  $\langle package-option-list \rangle$ , each package processes the  $\langle main-option-list \rangle$ . This means that options that affect all of the packages can be given globally, rather than repeated for every package.

Note that class files have the extension `.cls`, packages have the extension `.sty`.

`filecontents`

The environment `filecontents` is intended for passing the contents of packages, options, or other files along with a document in a single file. It has one argument, which is the name of the file to create. If that file already exists (maybe only in the current directory if the OS supports a notion of a ‘current directory’ or ‘default directory’) then nothing happens (except for an information message) and the body of the environment is bypassed. Otherwise, the body of the environment is written verbatim to the file name given as the first argument, together with some comments about how it was produced.

The environment is allowed only before `\documentclass` to ensure that all packages or options necessary for this particular run are present when needed. The begin and end tags should each be on a line by itself. There is also a star-form; this does not write extra comments into the file.

## 66.1 Option processing

When the options are processed, they are divided into two types: *local* and *global*:

- For a class, the options in the `\documentclass` command are local.
- For a package, the options in the `\usepackage` command are local, and the options in the `\documentclass` command are global.

The options for `\documentclass` and `\usepackage` are processed in the following way:

1. The local and global options that have been declared (using `\DeclareOption` as described below) are processed first.

In the case of `\ProcessOptions`, they are processed in the order that they were declared in the class or package.

In the case of `\ProcessOptions*`, they are processed in the order that they appear in the option-lists. First the global options, and then the local ones.

2. Any remaining local options are dealt with using the default option (declared using the `\DeclareOption*` declaration described below). For document classes, this usually does nothing, but records the option on a list of unused options. For packages, this usually produces an error.

Finally, when `\begin{document}` is reached, if there are any global options which have not been used by either the class or any package, the system will produce a warning.

## 67 Class and Package interface

### 67.1 Class name and version

`\ProvidesClass`

A class can identify itself with the `\ProvidesClass{\langle name \rangle}[\langle version \rangle]` command. The  $\langle version \rangle$  should begin with a date in the format YYYY/MM/DD.

## 67.2 Package name and version

`\ProvidesPackage` A package can identify itself with the `\ProvidesPackage{⟨name⟩}[⟨version⟩]` command. The `⟨version⟩` should begin with a date in the format YYYY/MM/DD.

## 67.3 Requiring other packages

`\RequirePackage` Packages or classes can load other packages using `\RequirePackage[⟨options⟩]{⟨name⟩}[⟨version⟩]`. If the package has already been loaded, then nothing happens unless the requested options are not a subset of the options with which it was loaded, in which case an error is called.

`\LoadClass` Similar to `\RequirePackage`, but for classes, may not be used in package files.

`\PassOptionsToPackage` Packages can pass options to other packages using:

`\PassOptionsToPackage{⟨options⟩}{⟨package⟩}`.

`\PassOptionsToClass` This adds the `⟨options⟩` to the options list of any future `\RequirePackage` or `\usepackage` command. For example:

```
\PassOptionsToPackage{foo,bar}{fred}
\RequirePackage[baz]{fred}
```

is the same as:

```
\RequirePackage[foo,bar,baz]{fred}
```

`\LoadClassWithOptions` `\LoadClassWithOptions{⟨name⟩}[⟨version⟩]:`

This is similar to `\LoadClass`, but it always calls class `⟨name⟩` with exactly the same option list that is being used by the current class, rather than an option explicitly supplied or passed on by `\PassOptionsToClass`.

`\RequirePackageWithOptions` `\RequirePackageWithOptions` is the analogous command for packages.

This is mainly intended to allow one class to simply build on another, for example:

```
\LoadClassWithOptions{article}
```

This should be contrasted with the slightly different construction

```
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions
\LoadClass{article}
```

As used here, the effects are more or less the same, but the version using `\LoadClassWithOptions` is slightly quicker (and less to type). If, however, the class declares options of its own then the two constructions are different; compare, for example:

```
\DeclareOption{landscape}{...}
\ProcessOptions
\LoadClassWithOptions{article}
```

with:

```
\DeclareOption{landscape}{...}
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions
\LoadClass{article}
```

In the first case, the `article` class will be called with option `landscape` precisely when the current class is called with this option; but in the second example it will not as in that case `article` is only passed options by the default option handler, which is not used for `landscape` as that option is explicitly declared.

<code>\@ifpackageloaded</code> <code>\@ifclassloaded</code> <code>\@ifpackagelater</code>  <code>\@ifclasslater</code> <code>\@ifpackagewith</code> <code>\@ifclasswith</code>	<p>To find out if a package has already been loaded, use <code>\@ifpackageloaded{&lt;package&gt;}{&lt;true&gt;}{&lt;false&gt;}</code>.</p> <p>To find out if a package has already been loaded with a version equal to or more recent than <code>&lt;version&gt;</code>, use <code>\@ifpackagelater{&lt;package&gt;}{&lt;version&gt;}{&lt;true&gt;}{&lt;false&gt;}</code>.</p> <p>To find out if a package has already been loaded with at least the options <code>&lt;options&gt;</code>, use <code>\@ifpackagewith{&lt;package&gt;}{&lt;options&gt;}{&lt;true&gt;}{&lt;false&gt;}</code>.</p> <p>There exists one package that can't be tested with the above commands: the <code>fontenc</code> package pretends that it was never loaded to allow for repeated reloading with different options (see <code>ltoutenc.dtx</code> for details).</p>
--	--

## 67.4 Declaring new options

Options for classes and packages are built using the same macros.

<code>\DeclareOption</code> <code>\DeclareOption*</code>	<p>To define a builtin option, use <code>\DeclareOption{&lt;name&gt;}{&lt;code&gt;}</code>.</p> <p>To define the default action to perform for local options which have not been declared, use <code>\DeclareOption*{&lt;code&gt;}</code>.</p>
---	--

*Note:* there should be no use of

`\RequirePackage`, `\DeclareOption`, `\DeclareOption*` or `\ProcessOptions` inside `\DeclareOption` or `\DeclareOption*`.

Possible uses for `\DeclareOption*` include:

`\DeclareOption*{}`

Do nothing. Silently accept unknown options. (This suppresses the usual warnings.)

`\DeclareOption*{\@unknownoptionerror}`

Complain about unknown local options. (The initial setting for package files.)

`\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{<pkg-name>}`

Handle the the current option by passing it on to the package `<pkg-name>`, which will presumably be loaded via `\RequirePackage` later in the file. This is useful for building ‘extension’ packages, that perhaps handle a couple of new options, but then pass everything else on to an existing package.

`\DeclareOption*{\InputIfFileExists{xx-\CurrentOption.yyy}%  
 {}%  
 {\OptionNotUsed}}`

Handle the option `foo` by loading the file `xx-foo.yyy` if it exists, otherwise do nothing, but declare that the option was not used. Actually the `\OptionNotUsed` declaration is only needed if this is being used in class files, but does no harm in package files.

## 67.5 Safe Input Macros

<code>\InputIfFileExists</code>  <code>\IfFileExists</code>	<p><code>\InputIfFileExists{&lt;file&gt;}{&lt;then&gt;}{&lt;else&gt;}</code>          Inputs <code>&lt;file&gt;</code> if it exists. Immediately before the input, <code>&lt;then&gt;</code> is executed. Otherwise <code>&lt;else&gt;</code> is executed.</p> <p>As above, but does not input the file.          One thing you might like to put in the <code>&lt;else&gt;</code> clause is</p>
---	--

<code>\@missingfileerror</code>	This starts an interactive request for a filename, supplying default extensions. Just hitting return causes the whole input to be skipped and entering <code>x</code> quits the current run.
<code>\input</code>	This has been redefined from the L <sup>A</sup> T <sub>E</sub> X2.09 definition, in terms of the new commands <code>\InputIfFileExists</code> and <code>\@missingfileerror</code> .
<code>\listfiles</code>	Giving this declaration in the preamble causes a list of all files input via the ‘safe input’ commands to be listed at the end. Any strings specified in the optional argument to <code>\ProvidesPackage</code> are listed alongside the file name. So files in standard (and other non-standard) distributions can put informative strings in this argument.

## 68 Implementation

	1 <code>\*2kernel</code>
<code>\if@compatibility</code>	The flag for compatibility mode. 2 <code>\newif\if@compatibility</code>
<code>\@documentclasshook</code>	The hook called after the first <code>\documentclass</code> command. By default this checks to see if <code>\@normalsize</code> is undefined, and if so, sets it to <code>\normalsize</code> . 3 <code>\def\@documentclasshook{%</code> 4 <code>  \ifx\@normalsize\undefined</code> 5 <code>    \let\@normalsize\normalsize</code> 6 <code>  \fi</code> 7 <code>}</code>
<code>\@declaredoptions</code>	This list is automatically built by <code>\DeclareOption</code> . It is the list of options (separated by commas) declared in the class or package file and it defines the order in which the the corresponding <code>\ds@<i>option</i></code> commands are executed. All local <code>\@option</code> s which are not declared will be processed in the order defined by the optional argument of <code>\documentclass</code> or <code>\usepackage</code> . 8 <code>\let\@declaredoptions\@empty</code>
<code>\@classoptionslist</code>	List of options of the main class. 9 <code>\let\@classoptionslist\relax</code> 10 <code>\@onlypreamble\@classoptionslist</code>
<code>\@unusedoptionlist</code>	List of options of the main class that haven’t been declared or loaded as class option files. 11 <code>\let\@unusedoptionlist\@empty</code> 12 <code>\@onlypreamble\@unusedoptionlist</code>
<code>\CurrentOption</code>	Name of current package or option. 13 <code>\let\CurrentOption\@empty</code>
<code>\@currname</code>	Name of current package or option. 14 <code>\let\@currname\@empty</code>
<code>\@currentt</code>	The current file extension. 15 <code>\global\let\@currentt=\@empty</code>

```

\@clsextension The two possible values of \@currentt.
\@pkgextension 16 \def\@clsextension{cls}
               17 \def\@pkgextension{sty}
               18 \@onlypreamble\@clsextension
               19 \@onlypreamble\@pkgextension

\@pushfilename Commands to push and pop the file name and extension.
\@popfilename  #1 current name.
\@currnamestack #2 current extension.
               #3 current catcode of @.
               #4 Rest of the stack.
               20 \def\@pushfilename{%
               21   \xdef\@currnamestack{%
               22     {\@currname}%
               23     {\@currentt}%
               24     {\the\catcode'\@}%
               25     \@currnamestack}}
               26 \@onlypreamble\@pushfilename
               27 \def\@popfilename{\expandafter\@p@pfilename\@currnamestack\@nil}
               28 \@onlypreamble\@popfilename
               29 \def\@p@pfilename#1#2#3#4\@nil{%
               30   \gdef\@currname{#1}%
               31   \gdef\@currentt{#2}%
               32   \catcode'\@#3\relax
               33   \gdef\@currnamestack{#4}}
               34 \@onlypreamble\@p@pfilename
               35 \gdef\@currnamestack{}
               36 \@onlypreamble\@currnamestack

\@optionlist Returns the option list of the file.
               37 \def\@optionlist#1{%
               38   \@ifundefined{opt@#1}\@empty{\csname opt@#1\endcsname}}
               39 \@onlypreamble\@optionlist

\@ifpackageloaded \@ifpackageloaded{<name>} Checks to see whether a file has been loaded.
\@ifclassloaded  40 \def\@ifpackageloaded{\@ifl@aded\@pkgextension}
                  41 \def\@ifclassloaded{\@ifl@aded\@clsextension}
                  42 \@onlypreamble\@ifpackageloaded
                  43 \@onlypreamble\@ifclassloaded
                  44 \def\@ifl@aded#1#2{%
                  45   \expandafter\ifx\csname ver@#2.#1\endcsname\relax
                  46     \expandafter\@secondoftwo
                  47   \else
                  48     \expandafter\@firstoftwo
                  49   \fi}
                  50 \@onlypreamble\@ifl@aded

\@ifpackagelater \@ifpackagelater{<name>}{YYYY/MM/DD} Checks that the package loaded is
\@ifclasslater   more recent than the given date.
                  51 \def\@ifpackagelater{\@ifl@ter\@pkgextension}
                  52 \def\@ifclasslater{\@ifl@ter\@clsextension}
                  53 \@onlypreamble\@ifpackagelater
                  54 \@onlypreamble\@ifclasslater

```

```

55 \def\@ifl@ter#1#2{%
56   \expandafter\@ifl@t@r
57   \csname ver@#2.#1\endcsname}
58 \onlypreamble\@ifl@ter

This internal macro is also used in \NeedsTeXFormat.

59 \def\@ifl@t@r#1#2{%
60   \ifnum\expandafter\@parse@version#1//00\@nil<%
61     \expandafter\@parse@version#2//00\@nil
62     \expandafter\@secondoftwo
63   \else
64     \expandafter\@firstoftwo
65   \fi}
66 \onlypreamble\@ifl@t@r

67 \def\@parse@version#1/#2/#3#4#5\@nil{#1#2#3#4 }
68 \onlypreamble\@parse@version

```

`\@ifpackagewith` `\@ifclasswith` `\@ifpackagewith{<name>}{<option-list>}` Checks that `<option-list>` is a subset of the options `with` which `<name>` was loaded.

```

69 \def\@ifpackagewith{\@ifoptions\@pkgextension}
70 \def\@ifclasswith{\@ifoptions\@clsextension}
71 \onlypreamble\@ifpackagewith
72 \onlypreamble\@ifclasswith

73 \def\@ifoptions#1#2{%
74   \@expandtwoargs\@ifpti@ns{\@optionlist{#2.#1}}
75 \onlypreamble\@ifoptions

```

Probably shouldn't use `\CurrentOption` here... (changed to `\reserved@b`.)

```

76 \def\@ifpti@ns#1#2{%
77   \let\reserved@a\@firstoftwo
78   \@for\reserved@b:=#2\do{%
79     \ifx\reserved@b\@empty
80     \else
81       \expandafter\in@\expandafter{\expandafter,\reserved@b},{, #1,}%
82       \ifin@
83       \else
84         \let\reserved@a\@secondoftwo
85       \fi
86     \fi
87   }%
88   \reserved@a}
89 \onlypreamble\@ifpti@ns

```

`\ProvidesPackage` Checks that the current filename is correct, and defines `\ver@filename`.

```

90 \def\ProvidesPackage#1{%
91   \xdef\@gtempa{#1}%
92   \ifx\@gtempa\@currname\else
93     \@latex@warning@no@line{You have requested
94       \@cls@pkg\space'\@currname',\MessageBreak
95       but the \@cls@pkg\space provides '#1'}%
96   \fi
97   \@ifnextchar[\@pr@videpackage{\@pr@videpackage[]}]%
98 \onlypreamble\ProvidesPackage

```

```

99 \def\@pr@videpackage[#1]{%
100 \expandafter\xdef\csname ver@\currname.\@current\endcsname{#1}%
101 \ifx\@current\@clsextension
102 \typeout{Document Class: \@gtempa\space#1}%
103 \else
104 \wlog{Package: \@gtempa\space#1}%
105 \fi}
106 \@onlypreamble\@pr@videpackage

\ProvidesClass Like \ProvidesPackage, but for classes.
107 \let\ProvidesClass\ProvidesPackage
108 \@onlypreamble\ProvidesClass

\ProvidesFile Like \ProvidesPackage, but for arbitrary files. Do not apply \@onlypreamble to
these, as we may want to label files input during the document.

\@providesfile
109 \def\ProvidesFile#1{%
110 \begingroup
111 \catcode'\ 10 %
112 \ifnum \endlinechar<256 %
113 \ifnum \endlinechar>\m@ne
114 \catcode\endlinechar 10 %
115 \fi
116 \fi
117 \@makeother\/%
118 \@makeother\&%

119 \kernel@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

During intex a special version of \@providesfile is used. The real definition
is installed right at the end, in ltfinal.dtx.

\def\@providesfile#1[#2]{%
\wlog{File: #1 #2}%
\expandafter\xdef\csname ver@#1\endcsname{#2}%
\endgroup}
\end{macrocode}

\PassOptionsToPackage If the package has been loaded, we check that it was first loaded with the options.
\PassOptionsToClass Otherwise we add the option list to that of the package.
120 \def\@pass@options#1#2#3{%
121 \expandafter\xdef\csname opt@#3.#1\endcsname{%
122 \@ifundefined{opt@#3.#1}\@empty
123 {\csname opt@#3.#1\endcsname,}%
124 \zap@space#2 \@empty}}
125 \@onlypreamble\@pass@options

126 \def\PassOptionsToPackage{\@pass@options\@pkgextension}
127 \def\PassOptionsToClass{\@pass@options\@clsextension}
128 \@onlypreamble\PassOptionsToPackage
129 \@onlypreamble\PassOptionsToClass

\DeclareOption Adds an option as a \ds@ command, or the default \default@ds command.
\DeclareOption* 130 \def\DeclareOption{%

```

```

131 \let\@fileswith@pti@ns\@badrequireerror
132 \@ifstar\@defdefault@ds\@declareoption}
133 \long\def\@declareoption#1#2{%
134   \xdef\@declaredoptions{\@declaredoptions,#1}%
135   \toks@{#2}%
136   \expandafter\edef\csname ds@#1\endcsname{\the\toks@}}
137 \long\def\@defdefault@ds#1{%
138   \toks@{#1}%
139   \edef\default@ds{\the\toks@}}
140 \@onlypreamble\DeclareOption
141 \@onlypreamble\@declareoption
142 \@onlypreamble\@defdefault@ds

```

**\OptionNotUsed** If we are in a class file, add `\CurrentOption` to the list of unused options. Otherwise, in a package file do nothing.

```

143 \def\OptionNotUsed{%
144   \ifx\@current\@clsextension
145     \xdef\@unusedoptionlist{%
146       \ifx\@unusedoptionlist\empty\else\@unusedoptionlist,\fi
147       \CurrentOption}%
148   \fi}
149 \@onlypreamble\OptionNotUsed

```

**\default@ds** The default default option code. Set by `\@onefilewithoptions` to either `\OptionNotUsed` for classes, or `\@unknownoptionerror` for packages. This may be reset in either case with `\DeclareOption*`.

```

150 % \let\default@ds\OptionNotUsed

```

**\ProcessOptions** `\ProcessOptions` calls `\ds@option` for each known package option, then calls `\default@ds` for each option on the local options list. Finally resets all the declared options to `\relax`. The empty option does nothing, this has to be reset on the off chance it's set to `\relax` if an empty element gets into the `\@declaredoptions` list.

The star form is similar but executes options given in the order specified in the document, not the order they are declared in the file. In the case of packages, global options are executed before local ones.

```

151 \def\ProcessOptions{%
152   \let\ds@\empty
153   \edef\@curroptions{\@ptionlist{\@currname.\@current}}%
154   \@ifstar\@xprocessoptions\@processoptions}
155 \@onlypreamble\ProcessOptions

156 \def\@processoptions{%
157   \@for\CurrentOption:=\@declaredoptions\do{%
158     \ifx\CurrentOption\empty\else
159       \@expandtwoargs\in@{,\CurrentOption,}%{
160         ,\ifx\@current\@clsextension\else\@classoptionslist,\fi
161         \@curroptions,}%
162       \ifin@
163         \@use@option
164         \expandafter\let\csname ds@\CurrentOption\endcsname\empty
165       \fi
166     \fi}%

```

```

167 \@process@ptions}
168 \@onlypreamble\@process@ptions

169 \def\@xprocess@ptions{%
170   \ifx\@currentx\@clsextension\else
171     \@for\CurrentOption:=\@classoptionslist\do{%
172       \ifx\CurrentOption\@empty\else
173         \@expandtwoargs\in@{,\CurrentOption,}{,\@declaredoptions,}%
174         \ifin@
175           \@use@option
176           \expandafter\let\csname ds@\CurrentOption\endcsname\@empty
177         \fi
178       \fi}%
179   \fi
180 \@process@ptions}
181 \@onlypreamble\@xprocess@ptions

```

The common part of `\ProcessOptions` and `\ProcessOptions*`.

```

182 \def\@process@ptions{%
183   \@for\CurrentOption:=\@curroptions\do{%
184     \@ifundefined{ds@\CurrentOption}%
185       {\@use@option
186        \default@ds}%

```

There should not be any non-empty definition of `\CurrentOption` at this point, as all the declared options were executed earlier. This is for compatibility with 2.09 styles which use `\def\ds@...` directly, and so have options which do not appear in `\@declaredoptions`.

```

187     \@use@option}%

```

Clear all the definitions for option code. First set all the declared options to `\relax`, then reset the ‘default’ and ‘empty’ options. and the list of declared options.

```

188   \@for\CurrentOption:=\@declaredoptions\do{%
189     \expandafter\let\csname ds@\CurrentOption\endcsname\relax}%

190   \let\CurrentOption\@empty
191   \let\@fileswith@ptions\@fileswith@ptions
192   \AtEndOfPackage{\let\@unprocessedoptions\relax}}
193 \@onlypreamble\@process@ptions

```

`\@options` `\@options` is a synonym for `\ProcessOptions*` for upward compatibility with L<sup>A</sup>T<sub>E</sub>X 2.09 style files.

```

194 \def\@options{\ProcessOptions*}
195 \@onlypreamble\@options

```

`\@use@option` Execute the code for the current option.

```

196 \def\@use@option{%
197   \@expandtwoargs\@removeelement\CurrentOption
198   \@unusedoptionlist\@unusedoptionlist
199   \csname ds@\CurrentOption\endcsname}
200 \@onlypreamble\@use@option

```

`\ExecuteOptions` `\ExecuteOptions{<option-list>}` executes the code declared for each option.

```

201 \def\ExecuteOptions#1{%

```

```

202 \def\reserved@a##1\@nil{%
203   \@for\CurrentOption:=#1\do{\csname ds@\CurrentOption\endcsname}%
204   \edef\CurrentOption{##1}}%
205 \expandafter\reserved@a\CurrentOption\@nil}
206 \@onlypreamble\ExecuteOptions

```

The top-level commands, which just set some parameters then call the internal command, \@fileswithoptions.

**\documentclass** The main new-style class declaration.

```

207 \def\documentclass{%
208   \let\documentclass\@twoclasseserror
209   \if@compatibility\else\let\usepackage\RequirePackage\fi
210   \@fileswithoptions\@clsextension}
211 \@onlypreamble\documentclass

```

**\documentstyle** 2.09 style class ‘style’ declaration.

```

212 \def\documentstyle{%
213   \makeatletter\input{latex209.def}\makeatother
214   \documentclass}
215 \@onlypreamble\documentstyle

```

**\RequirePackage** Load package if not already loaded.

```

216 \def\RequirePackage{%
217   \@fileswithoptions\@pkgextension}
218 \@onlypreamble\RequirePackage

```

**\LoadClass** Load class.

```

219 \def\LoadClass{%
220   \ifx\@current\@pkgextension
221     \@latex@error
222     {\noexpand\LoadClass in package file}%
223     {You may only use \noexpand\LoadClass in a class file.}%
224   \fi
225   \@fileswithoptions\@clsextension}
226 \@onlypreamble\LoadClass

```

**\@loadwithoptions** Pass the current option list on to a class or package. #1 is \@cls-or-pkgextension, #2 is \RequirePackage or \LoadClass, #3 is the class or package to be loaded.

```

227 \def\@loadwithoptions#1#2#3{%
228   \expandafter\let\csname opt@#3.#1\expandafter\endcsname
229   \csname opt@\@currname.\@current\endcsname
230   #2{#3}}
231 \@onlypreamble\@loadwithoptions

```

**\LoadClassWithOptions** Load class ‘#1’ with the current option list.

```

232 \def\LoadClassWithOptions{%
233   \@loadwithoptions\@clsextension\LoadClass}
234 \@onlypreamble\LoadClassWithOptions

```

**\RequirePackageWithOptions** Load package ‘#1’ with the current option list.

```

235 \def\RequirePackageWithOptions{%
236   \AtEndOfPackage{\let\@unprocessedoptions\relax}%
237   \@loadwithoptions\@pkgextension\RequirePackage}
238 \@onlypreamble\RequirePackageWithOptions

```

`\usepackage` To begin with, `\usepackage` produces an error. This is reset by `\documentclass`.

```

239 \def\usepackage#1{%
240   \@latex@error
241     {\noexpand \usepackage before \string\documentclass}%
242     {\noexpand \usepackage may only appear in the document
243       preamble, i.e.,\MessageBreak
244       between \noexpand\documentclass and
245       \string\begin{document}.}%
246   \@gobble}
247 \@onlypreamble\usepackage

```

`\NeedsTeXFormat` Check that the document is running on the correct system.

```

248 \def\NeedsTeXFormat#1{%
249   \def\reserved@a{#1}%
250   \ifx\reserved@a\fmtname
251     \expandafter\@needsformat
252   \else
253     \@latex@error{This file needs format ‘\reserved@a’%
254       \MessageBreak but this is ‘\fmtname’}%
255     The current input file will not be processed
256     further,\MessageBreak
257     because it was written for some other flavor of
258     TeX.\MessageBreak\@ehd}%

```

If the file is not meant to be processed by L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> we stop inputting it, but we do not end the run. We just end inputting the current file.

```

259   \endinput \fi}
260 \@onlypreamble\NeedsTeXFormat

261 \def\@needsformat{%
262   \@ifnextchar[%]
263     \@needsformat
264   {}}
265 \@onlypreamble\@needsformat

266 \def\@needsformat[#1]{%
267   \ifl@t@r\fmtversion{#1}{}%
268   {\@latex@warning@no@line
269     {You have requested release ‘#1’ of LaTeX,\MessageBreak
270     but only release ‘\fmtversion’ is available}}%
271 \@onlypreamble\@needsformat

```

`\zap@space` `\zap@space foo<space>\@empty` removes all spaces from `foo` that are not protected by `{ }` groups.

```

272 \def\zap@space#1 #2{%
273   #1%
274   \ifx#2\@empty\else\expandafter\zap@space\fi
275   #2}

```

`\@fileswithoptions` The common part of `\documentclass` and `\usepackage`.

```

276 \def\@fileswithoptions#1{%
277   \@ifnextchar[%]
278     {\@fileswithoptions#1}%

```

```

279     {\@fileswith@ptions#1[]}}
280 \@onlypreamble\@fileswithoptions

281 \def\@fileswith@ptions#1[#2]#3{%
282   \ifnextchar[%]
283     {\@fileswith@ptions#1[{#2}]#3}%
284     {\@fileswith@ptions#1[{#2}]#3[]}}
285 \@onlypreamble\@fileswith@ptions

```

Then we do some work.

First of all, we define the global variables. Then we look to see if the file has already been loaded. If it has, we check that it was first loaded with at least the current options. If it has not, we add the current options to the package options, set the default version to be 0000/00/00, and load the file if we can find it. Then we check the version number.

Finally, we restore the old file name, reset the default option, and we set the catcode of @.

For classes, we can immediately process the file. For other types, #2 could be a comma separated list, so loop through, processing each one separately.

```

286 \def\@fileswith@ptions#1[#2]#3[#4]{%
287   \ifx#1\@clsextension
288     \ifx\@classoptionslist\relax
289       \xdef\@classoptionslist{\zap@space#2 \@empty}%
290       \def\reserved@a{%
291         \@onefilewithoptions#3[{#2}][{#4}]#1%
292         \@documentclasshook}%
293       \else
294         \def\reserved@a{%
295           \@onefilewithoptions#3[{#2}][{#4}]#1%
296         \fi
297       \else

```

build up a list of calls to \@onefilewithoptions (one for each package) without thrashing the parameter stack.

```

298   \def\reserved@b##1,{%
299     \ifx\@nil##1\relax\else
300       \ifx\relax##1\relax\else
301         \noexpand\@onefilewithoptions##1[{#2}][{#4}]%
302         \noexpand\@pkgextension
303       \fi
304       \expandafter\reserved@b
305     \fi}%
306   \edef\reserved@a{\zap@space#3 \@empty}%
307   \edef\reserved@a{\expandafter\reserved@b\reserved@a,\@nil,}%
308   \fi
309   \reserved@a}
310 \@onlypreamble\@fileswith@ptions

```

Have the main argument as #1, so we only need one \@expandafter above.

```

311 \def\@onefilewithoptions#1[#2][#3]#4{%
312   \@pushfilename
313   \xdef\@currname{#1}%
314   \global\let\@current#4%
315   \expandafter\let\csname\@currname.\@current-h@@k\endcsname\@empty
316   \let\CurrentOption\@empty

```

```

317 \resetoptions
318 \makeatletter

Grab everything in a macro, so the parameter stack is popped before any processing begins.

319 \def\reserved@a{%
320   \ifl@aded\@currentx{#1}%
321   {\@ifoptions\@currentx{#1}{#2}{}}%
322   {\@latexerror
323     {Option clash for \cls@pkg\space #1}%
324     {The package #1 has already been loaded
325       with options:\MessageBreak
326       \space\space[\@optionlist{#1.\@currentx}]\MessageBreak
327       There has now been an attempt to load it
328       with options:\MessageBreak
329       \space\space[#2]\MessageBreak
330       Adding the global options:\MessageBreak
331       \space\space
332       \@optionlist{#1.\@currentx},#2\MessageBreak
333       to your \noexpand\documentclass declaration may fix this.%
334       \MessageBreak
335       Try typing \space <return> \space to proceed.}}}%
336   {\@passoptions\@currentx{#2}{#1}%

337   \global\expandafter
338   \let\csname ver@\@currname.\@currentx\endcsname\@empty
339   \InputIfFileExists
340     {\@currname.\@currentx}%
341     {}%
342     {\@missingfileerror\@currname\@currentx}%

\@unprocessedoptions will generate an error for each specified option in a package unless a \ProcessOptions has appeared in the package file.

343   \let\@unprocessedoptions\@unprocessedoptions
344   \csname\@currname.\@currentx-h@k\endcsname
345   \expandafter\let\csname\@currname.\@currentx-h@k\endcsname
346     \@undefined
347   \@unprocessedoptions}

348   \@ifl@ter\@currentx{#1}{#3}{}%
349   {\@latexwarning@no@line
350     {You have requested,\on@line,
351       version\MessageBreak
352       ‘#3’ of \cls@pkg\space #1,\MessageBreak
353       but only version\MessageBreak
354       ‘\csname ver@#1.\@currentx\endcsname’\MessageBreak
355       is available}}}%

356   \ifx\@currentx\clsextension\let\LoadClass\@twoloadclasserror\fi
357   \popfilename
358   \resetoptions}%
359 \reserved@a}
360 \onlypreamble\@onefilewithoptions

```

`\@fileswith@ptions` Save the definition (for error checking).

```

361 \let\@fileswith@pti@ns\@fileswith@pti@ns
362 \@onlypreamble\@fileswith@pti@ns

\@reset@ptions Reset the default option, and clear lists of declared options.
363 \def\@reset@ptions{%
364   \global\ifx\@currentx\@clsextension
365     \let\default@ds\OptionNotUsed
366   \else
367     \let\default@ds\@unknownoptionerror
368   \fi
369   \global\let\ds@\@empty
370   \global\let\@declaredoptions\@empty}
371 \@onlypreamble\@reset@ptions

```

## 68.1 Hooks

Allow code do be saved to be executed at specific later times.

Save things in macros, I considered using toks registers, (and `\addto@hook` from the NFSS code, that would require stacking the contents in the case of required packages, so just generate a new macro for each package.

```

\@begindocumenthook Stuff to appear at the beginning or end of the document.
\@enddocumenthook
372 \ifx\@begindocumenthook\@undefined
373   \let\@begindocumenthook\@empty
374 \fi
375 \let\@enddocumenthook\@empty

\g@addto@macro Globally add to the end of a macro.
376 \long\def\g@addto@macro#1#2{%
377   \begingroup
378     \toks@\expandafter{#1#2}%
379     \xdef#1{\the\toks@}%
380   \endgroup}

\AtEndOfPackage The access functions.
\AtEndOfClass
\AtBeginDocument
\AtEndDocument
381 \def\AtEndOfPackage{%
382   \expandafter\g@addto@macro\csname\@currname.\@currentx-h@@k\endcsname}
383 \let\AtEndOfClass\AtEndOfPackage
384 \@onlypreamble\AtEndOfPackage
385 \@onlypreamble\AtEndOfClass

386 \def\AtBeginDocument{\g@addto@macro\@begindocumenthook}
387 \def\AtEndDocument{\g@addto@macro\@enddocumenthook}
388 \@onlypreamble\AtBeginDocument

\@cls@pkg The current file type.
389 \def\@cls@pkg{%
390   \ifx\@currentx\@clsextension
391     document class%
392   \else
393     package%
394   \fi}
395 \@onlypreamble\@cls@pkg

```

```

\@unknownoptionerror Bad option.
396 \def\@unknownoptionerror{%
397   \@latex@error
398   {Unknown option ‘\CurrentOption’ for \cls@pkg\space‘\@currname’}%
399   {The option ‘\CurrentOption’ was not declared in
400     \cls@pkg\space‘\@currname’, perhaps you\MessageBreak
401     misspelled its name.
402     Try typing \space <return>
403     \space to proceed.}}
404 \onlypreamble\@unknownoptionerror

\@@unprocessedoptions Declare an error for each option, unless a \ProcessOptions occurred.
405 \def\@@unprocessedoptions{%
406   \ifx\@current\@pkgextension
407     \edef\@curoptions{\@optionlist{\@currname.\@current}}%
408     \@for\CurrentOption:=\@curoptions\do{%
409       \ifx\CurrentOption\@empty\else\@unknownoptionerror\fi}%
410   \fi}
411 \onlypreamble\@@unprocessedoptions
412 \onlypreamble\@@unprocessedoptions

\@badrequireerror \RequirePackage or \LoadClass occurs in the options section.
413 \def\@badrequireerror#1[#2]#3[#4]{%
414   \@latex@error
415   {\noexpand\RequirePackage or \noexpand\LoadClass
416     in Options Section}%
417   {The \cls@pkg\space ‘\@currname’ is defective.\MessageBreak
418     It attempts to load ‘#3’ in the options section, i.e.,\MessageBreak
419     between \noexpand\DeclareOption and \string\ProcessOptions.}}
420 \onlypreamble\@badrequireerror

\@twoloadclasserror Two \LoadClass in a class.
421 \def\@twoloadclasserror{%
422   \@latex@error
423   {Two \noexpand\LoadClass commands}%
424   {You may only use one \noexpand\LoadClass in a class file}}
425 \onlypreamble\@twoloadclasserror

\@twoclasseserror Two \documentclass or \documentstyle.
426 \def\@twoclasseserror#1#{%
427   \@latex@error
428   {Two \noexpand\documentclass or \noexpand\documentstyle commands}%
429   {The document may only declare one class.}\@gobble}
430 \onlypreamble\@twoclasseserror

```

## 68.2 Providing shipment

```

\two@digits Prefix a number less than 10 with ‘0’.
431 \def\two@digits#1{\ifnum#1<10 0\fi\number#1}

\filecontents This environment implements inline files. The star-form does not write extra
\endfilecontents comments into the file.

```

```

432 \begingroup%
433 \catcode'\*=11 %
434 \catcode'\^~M\active%
435 \catcode'\^~L\active\let^~L\relax%
436 \catcode'\^~I\active%

437 \gdef\filecontents{\@tempswatrue\filec@ntents}%
438 \gdef\filecontents*{\@tempswafalse\filec@ntents}%

439 \gdef\filec@ntents#1{%
440   \openin\@inputcheck#1 %
441   \ifeof\@inputcheck%
442     \@latex@warning@no@line%
443       {Writing file '\@currdir#1'}%

444   \chardef\reserved@c15 %
445   \ch@ck7\reserved@c\write%
446   \immediate\openout\reserved@c#1\relax%
447   \else%

448   \closein\@inputcheck%
449   \@latex@warning@no@line%
450     {File '#1' already exists on the system.\MessageBreak%
451       Not generating it from this source}%
452   \let\write\@gobbletwo%
453   \let\closeout\@gobble%
454   \fi%
455   \if@tempswa%

456   \immediate\write\reserved@c{%
457     \@percentchar\@percentchar\space%
458     \expandafter\@gobble\string\LaTeX2e file '#1'^~J%
459     \@percentchar\@percentchar\space generated by the %
460     '\@currenvir' \expandafter\@gobblefour\string\newenvironment^~J%
461     \@percentchar\@percentchar\space from source '\jobname' on %
462     \number\year/\two@digits\month/\two@digits\day.^~J%
463     \@percentchar\@percentchar}%
464   \fi%
465   \let\do\@makeoother\dospecials%

466   \edef\E{\@backslashchar end\string{\@currenvir\string}}%
467   \edef\reserved@b{%
468     \def\noexpand\reserved@b%
469       #####1\E####2\E####3\relax}%
470   \reserved@b{%
471     \ifx\relax##3\relax%

There was no \end{filecontents}

472     \immediate\write\reserved@c{##1}%
473     \else%

There was a \end{filecontents}, so stop this time.

474     \edef^~M{\noexpand\end{\@currenvir}}%
475     \ifx\relax##1\relax%
476     \else%

```

Text before the `\end`, write it with a warning.

```

477      \@latex@warning{Writing text ‘##1’ before %
478      \string\end{\@currenvir}\MessageBreak as last line of #1}%
479      \immediate\write\reserved@c{##1}%
480      \fi%
481      \ifx\relax##2\relax%
482      \else%
```

Text after the `\end`, ignore it with a warning.

```

483      \@latex@warning{%
484      Ignoring text ‘##2’ after \string\end{\@currenvir}}%
485      \fi%
486      \fi%
487      ^^M}%
```

```

488 \catcode'\^^L\active%
489 \let\L\@undefined%
490 \def^^L{\@ifundefined L^^J^^J^^J}%
491 \catcode'\^^I\active%
492 \let\I\@undefined%
493 \def^^I{\@ifundefined I\space\space}%
494 \catcode'\^^M\active%
495 \edef^^M##1^^M{%
496 \noexpand\reserved@b##1\E\E\relax}}%
497 \endgroup%

498 \begingroup
499 \catcode'\=\catcode'\%
500 \catcode'\%=12
501 \catcode'\*=11
502 \gdef\@percentchar{%}
503 \gdef\endfilecontents{|
504 \immediate\closeout\reserved@c
505 \def\T##1##2##3{|
506 \ifx##1\@undefined\else
507 \latex@warning@no@line{##2 has been converted to Blank ##3e}|
508 \fi}|
509 \T\L{Form Feed}{Lin}|
510 \T\I{Tab}{Spac}|
511 \immediate\write\@unused{}}
512 \global\let\endfilecontents*\endfilecontents
513 \@onlypreamble\filecontents
514 \@onlypreamble\endfilecontents
515 \@onlypreamble\filecontents*
516 \@onlypreamble\endfilecontents*
517 \endgroup
518 \@onlypreamble\filecontents
```

519 `/2kernel`)

## 69 After Preamble

Finally we declare a package that allows all the commands declared above to be `\@onlypreamble` to be used after `\begin{document}`.

```

520 <*afterpreamble>
521 \NeedsTeXFormat{LaTeX2e}
522 \ProvidesPackage{pkgindoc}
523         [1994/10/20 v1.1 Package Interface in Document (DPC)]
524 \def\reserved@a#1\do\@classoptionslist#2\do\filecontents#3\relax{%
525     \gdef\@preamblecmds{#1#3}}
526 \expandafter\reserved@a\@preamblecmds\relax
527 </afterpreamble>

```

## File M

# lthyphen.dtx

This file contains the code for loading hyphenation patterns into L<sup>A</sup>T<sub>E</sub>X. Most of this will end up in a file called `hyphen.ltx`. If you wish to customize your L<sup>A</sup>T<sub>E</sub>X system in respect of hyphenation patterns, write a file `hyphen.cfg`. If this file exists, it will be loaded instead of `hyphen.ltx`. See the comments below for additional information.

To produce the printed version of this file the following code is used. It can be extracted with the DOCSTRIP program, or one can run this file directly through L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

```
1 (*driver)
2 \documentclass{ltxdoc}
3 \begin{document}
4 \DocInput{lthyphen.dtx}
5 \end{document}
6 </driver>
```

The default file `hyphen.ltx` loads hyphenation patterns for US english. If you want to load additional or other hyphenation patterns, you should create a file `hyphen.cfg`. This is best done by starting from `hyphen.ltx`.

For backward compatibility, the default file, `hyphen.ltx`, first tries to load the file `hyphen.tex`. If this file exists, an information message is issued and the appropriate defaults for T<sub>E</sub>X's internal parameters are set: `\language` is initialized to 0, and `\lefthyphenmin` and `\righthyphenmin` to 2 and 3, respectively, to disallow x- or -xx breaks.

```
7 (*default)
8 \InputIfFileExists{hyphen.tex}%
9   {\message{Loading hyphenation patterns for US english.}}%
10   \language=0
11   \lefthyphenmin=2 \righthyphenmin=3 }%
```

Otherwise, since we cannot do anything without any hyphenation patterns, an error message is printed and the IniT<sub>E</sub>X run is terminated by invoking `\@@end` (which is the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> name for T<sub>E</sub>X's `\end` primitive).

```
12   {\errhelp{The configuration for hyphenation is incorrectly
13             installed.^^J%
14             If you don't understand this error message you need
15             to seek^^Jexpert advice.}%
16   \errmessage{OOPS! I can't find any hyphenation patterns for
17             US english.^^J \space Think of getting some or the
18             latex2e setup will never succeed}\@@end}
19 </default>
```

The following example describes the possible contents of a file `hyphen.cfg` that will load both US English and German hyphenation patterns, making the former the default. It sets `\language` to 0 for the US patterns and to 1 for the German patterns. Then `\language` is set to 0 to make this the default and the default values of `\lefthyphenmin` and `\righthyphenmin` are set.

```
\language=0
\input hyphen % (or \input ushyphen1 if the file has been renamed)
```

```
\language=1
\input ghyph31
\language=0
\lefthyphenmin=2
\righthyphenmin=3
\endinput
```

Another possibility is to use the package `babel`, by Johannes Braams. That package is distributed with a suitable `hyphen.cfg` file.

## File N

# lualatex.dtx

## 70 Overview

LuaTeX adds a number of engine-specific functions to TeX. Several of these require set up that is best done in the kernel or need related support functions. This file provides *basic* support for LuaTeX at the L<sup>A</sup>T<sub>Ε</sub>X 2<sub>ε</sub> kernel level plus as a loadable file which can be used with plain TeX and L<sup>A</sup>T<sub>Ε</sub>X.

This file contains code for both TeX (to be stored as part of the format) and Lua (to be loaded at the start of each job). In the Lua code, the kernel uses the namespace `luatexbase`.

The following `\count` registers are used here for register allocation:

```
\e@alloc@attribute@count Attributes (default 258)
\e@alloc@ccodetable@count Category code tables (default 259)
\e@alloc@luafunction@count Lua functions (default 260)
  \e@alloc@whatsit@count User whatsits (default 261)
  \e@alloc@bytecode@count Lua bytecodes (default 262)
  \e@alloc@luachunk@count Lua chunks (default 263)
```

(`\count 256` is used for `\newmarks` allocation and `\count 257` is used for `\newXeTeXintercharclass` with XeTeX, with code defined in `ltxfinal.dtx`). With any L<sup>A</sup>T<sub>Ε</sub>X 2<sub>ε</sub> kernel from 2015 onward these registers are part of the block in the extended area reserved by the kernel (prior to 2015 the L<sup>A</sup>T<sub>Ε</sub>X 2<sub>ε</sub> kernel did not provide any functionality for the extended allocation area).

## 71 Core TeX functionality

The commands defined here are defined for possible inclusion in a future L<sup>A</sup>T<sub>Ε</sub>X format, however also extracted to the file `lualatex.tex` which may be used with older L<sup>A</sup>T<sub>Ε</sub>X formats, and with plain TeX.

<code>\newattribute</code>	<code>\newattribute{&lt;attribute&gt;}</code> Defines a named <code>\attribute</code> , indexed from 1 ( <i>i.e.</i> <code>\attribute0</code> is never defined). Attributes initially have the marker value <code>-7FFFFFFF</code> ('unset') set by the engine.
<code>\newcatcodetable</code>	<code>\newcatcodetable{&lt;catcodetable&gt;}</code> Defines a named <code>\catcodetable</code> , indexed from 1 ( <code>\catcodetable0</code> is never assigned). A new catcode table will be populated with exactly those values assigned by IniTeX (as described in the LuaTeX manual).
<code>\newluafunction</code>	<code>\newluafunction{&lt;function&gt;}</code> Defines a named <code>\luafunction</code> , indexed from 1. (Lua indexes tables from 1 so <code>\luafunction0</code> is not available).
<code>\newwhatsit</code>	<code>\newwhatsit{&lt;whatsit&gt;}</code> Defines a custom <code>\whatsit</code> , indexed from 1.
<code>\newluabytecode</code>	<code>\newluabytecode{&lt;bytecode&gt;}</code>

	Allocates a number for lua bytecode register, indexed from 1.
<code>\newluachunkname</code>	<code>newluachunkname{⟨chunkname⟩}</code> Allocates a number for lua chunk register, indexed from 1. Also enters the name of the register (without backslash) into the <code>lua.name</code> table to be used in stack traces.
<code>\catcodetable@initex</code>	Predefined category code tables with the obvious assignments. Note that the
<code>\catcodetable@string</code>	<code>latex</code> and <code>atletter</code> tables set the full Unicode range to the codes predefined by
<code>\catcodetable@latex</code>	the kernel.
<code>\catcodetable@atletter</code>	<code>\setattribute{⟨attribute⟩}{⟨value⟩}</code>
<code>\setattribute</code>	<code>\unsetattribute{⟨attribute⟩}</code>
<code>\unsetattribute</code>	Set and unset attributes in a manner analogous to <code>\setlength</code> . Note that attributes take a marker value when unset so this operation is distinct from setting the value to zero.

## 72 Plain T<sub>E</sub>X interface

The `luatex` interface may be used with plain T<sub>E</sub>X using `\input{ltuatex}` this inputs `ltuatex.tex` which inputs `etex.src` (or `etex.sty` if used with L<sup>A</sup>T<sub>E</sub>X) if it is not already input, and then defines some internal commands to allow the `luatex` interface to be defined.

The `luatexbase` package interface may also be used in plain T<sub>E</sub>X, as before, by inputting the package `\input luatexbase.sty`. The new version of `luatexbase` is based on this `luatex` code but implements a compatibility layer providing the interface of the original package.

## 73 Lua functionality

### 73.1 Allocators in Lua

<code>new_attribute</code>	<code>luatexbase.new_attribute(⟨attribute⟩)</code> Returns an allocation number for the <code>⟨attribute⟩</code> , indexed from 1. The attribute will be initialised with the marker value <code>-0xFFFFFFFF</code> ('unset'). The attribute allocation sequence is shared with the T <sub>E</sub> X code but this function does <i>not</i> define a token using <code>\attributedef</code> . The attribute name is recorded in the <code>attributes</code> table. A metatable is provided so that the table syntax can be used consistently for attributes declared in T <sub>E</sub> X or lua.
<code>new_whatsit</code>	<code>luatexbase.new_whatsit(⟨whatsit⟩)</code> Returns an allocation number for the custom <code>⟨whatsit⟩</code> , indexed from 1.
<code>new_bytecode</code>	<code>luatexbase.new_bytecode(⟨bytecode⟩)</code> Returns an allocation number for a bytecode register, indexed from 1. The optional <code>⟨name⟩</code> argument is just used for logging.
<code>new_chunkname</code>	<code>luatexbase.new_chunkname(⟨chunkname⟩)</code> Returns an allocation number for a lua chunk name for use with <code>\directlua</code> and <code>\latelua</code> , indexed from 1. The number is returned and also <code>⟨name⟩</code> argument is added to the <code>lua.name</code> array at that index.

### 73.2 Lua access to T<sub>E</sub>X register numbers

<code>registernumber</code>	<code>luatexbase.registernumber(⟨name⟩)</code> Sometimes (notably in the case of Lua attributes) it is necessary to access
-----------------------------	---

a register *by number* that has been allocated by T<sub>E</sub>X. This package provides a function to look up the relevant number using LuaT<sub>E</sub>X's internal tables. After for example `\newattribute\myattrib`, `\myattrib` would be defined by (say) `\myattrib=\attribute15`. `luatexbase.registernumber("myattrib")` would then return the register number, 15 in this case. If the string passed as argument does not correspond to a token defined by `\attributedef`, `\countdef` or similar commands, the Lua value `false` is returned.

As an example, consider the input:

```
\newcommand\test[1]{%
\typeout{#1: \expandafter\meaning\csname#1\endcsname^^J
\space\space\space\space
\directlua{tex.write(luatexbase.registernumber("#1") or "bad input")}%
}}

\test{undefinedrubbish}

\test{space}

\test{hbox}

\test{@MM}

\test{@tempdima}
\test{@tempdimb}

\test{strutbox}

\test{sixt@@n}

\attributedef\myattr=12
\myattr=200
\test{myattr}
```

If the demonstration code is processed with LuaL<sup>A</sup>T<sub>E</sub>X then the following would be produced in the log and terminal output.

```
undefinedrubbish: \relax
      bad input
space: macro:->
      bad input
hbox: \hbox
      bad input
@MM: \mathchar"4E20
      20000
@tempdima: \dimen14
      14
@tempdimb: \dimen15
      15
strutbox: \char"B
      11
sixt@@n: \char"10
      16
```

```
myattr: \attribute12
12
```

Notice how undefined commands, or commands unrelated to registers do not produce an error, just return **false** and so print **bad input** here. Note also that commands defined by `\newbox` work and return the number of the box register even though the actual command holding this number is a `\chardef` defined token (there is no `\boxdef`).

### 73.3 Module utilities

`provides_module` `luatexbase.provides_module(<info>)`

This function is used by modules to identify themselves; the `info` should be a table containing information about the module. The required field `name` must contain the name of the module. It is recommended to provide a field `date` in the usual L<sup>A</sup>T<sub>E</sub>X format `yyyy/mm/dd`. Optional fields `version` (a string) and `description` may be used if present. This information will be recorded in the log. Other fields are ignored.

`module_info` `luatexbase.module_info(<module>, <text>)`

`module_warning` `luatexbase.module_warning(<module>, <text>)`

`module_error` `luatexbase.module_error(<module>, <text>)`

These functions are similar to L<sup>A</sup>T<sub>E</sub>X's `\PackageError`, `\PackageWarning` and `\PackageInfo` in the way they format the output. No automatic line breaking is done, you may still use `\n` as usual for that, and the name of the package will be prepended to each output line.

Note that `luatexbase.module_error` raises an actual Lua error with `error()`, which currently means a call stack will be dumped. While this may not look pretty, at least it provides useful information for tracking the error down.

### 73.4 Callback management

`add_to_callback` `luatexbase.add_to_callback(<callback>, <function>, <description>)` Registers the `<function>` into the `<callback>` with a textual `<description>` of the function. Functions are inserted into the callback in the order loaded.

`remove_from_callback` `luatexbase.remove_from_callback(<callback>, <description>)` Removes the callback function with `<description>` from the `<callback>`. The removed function and its description are returned as the results of this function.

`in_callback` `luatexbase.in_callback(<callback>, <description>)` Checks if the `<description>` matches one of the functions added to the list for the `<callback>`, returning a boolean value.

`disable_callback` `luatexbase.disable_callback(<callback>)` Sets the `<callback>` to **false** as described in the Lua<sub>T</sub><sub>E</sub>X manual for the underlying `callback.register` built-in. Callbacks will only be set to false (and thus be skipped entirely) if there are no functions registered using the callback.

`callback_descriptions` A list of the descriptions of functions registered to the specified callback is returned. `{}` is returned if there are no functions registered.

`create_callback` `luatexbase.create_callback(<name>, metatype, <default>)` Defines a user defined callback. The last argument is a default function of **false**.

`call_callback` `luatexbase.call_callback(<name>, ...)` Calls a user defined callback with the supplied arguments.

## 74 Implementation

```
1 <*2ekernel | tex | latexrelease>
2 <2ekernel | latexrelease>\ifx\directlua\@undefined\else
```

### 74.1 Minimum LuaTeX version

LuaTeX has changed a lot over time. In the kernel support for ancient versions is not provided: trying to build a format with a very old binary therefore gives some information in the log and loading stops. The cut-off selected here relates to the tree-searching behaviour of `require()`: from version 0.60, LuaTeX will correctly find Lua files in the `texmf` tree without ‘help’.

```
3 <latexrelease>\IncludeInRelease{2015/10/01}
4 <latexrelease>          {\newluafunction}{LuaTeX}%
5 \ifnum\luatexversion<60 %
6   \wlog{*****}
7   \wlog{* LuaTeX version too old for ltuatex support *}
8   \wlog{*****}
9   \expandafter\endinput
10 \fi
```

### 74.2 Older L<sup>A</sup>T<sub>E</sub>X/Plain T<sub>E</sub>X setup

```
11 <*tex>
```

Older L<sup>A</sup>T<sub>E</sub>X formats don’t have the primitives with ‘native’ names: sort that out. If they already exist this will still be safe.

```
12 \directlua{tex.enableprimitives("",tex.extraprimitives("luatex"))}
13 \ifx\e@alloc\@undefined
```

In pre-2014 L<sup>A</sup>T<sub>E</sub>X, or plain T<sub>E</sub>X, load `etex.{sty,src}`.

```
14 \ifx\documentclass\@undefined
15   \ifx\loccount\@undefined
16     \input{etex.src}%
17   \fi
18   \catcode'\@=11 %
19   \outer\expandafter\def\csname newfam\endcsname
20     {\alloc@8\fam\chardef\et@xmaxfam}
21 \else
22   \RequirePackage{etex}
23   \expandafter\def\csname newfam\endcsname
24     {\alloc@8\fam\chardef\et@xmaxfam}
25   \expandafter\let\expandafter\new@mathgroup\csname newfam\endcsname
26 \fi
```

#### 74.2.1 Fixes to etex.src/etex.sty

These could and probably should be made directly in an update to `etex.src` which already has some `luatex`-specific code, but does not define the correct range for `luatex`.

```
27 % 2015-07-13 higher range in luatex
28 \edef \et@xmaxregs {\ifx\directlua\@undefined 32768\else 65536\fi}
29 % luatex/xetex also allow more math fam
30 \edef \et@xmaxfam {\ifx\Umathchar\@undefined\sixt@@n\else\ccclvi\fi}
```

```

31 \count 270=\et@xmaxregs % locally allocates \count registers
32 \count 271=\et@xmaxregs % ditto for \dimen registers
33 \count 272=\et@xmaxregs % ditto for \skip registers
34 \count 273=\et@xmaxregs % ditto for \muskip registers
35 \count 274=\et@xmaxregs % ditto for \box registers
36 \count 275=\et@xmaxregs % ditto for \toks registers
37 \count 276=\et@xmaxregs % ditto for \marks classes

    and 256 or 16 fam. (Done above due to plain/LATEX differences in lAuatex.)
38 % \outer\def\newfam{\alloc@8\fam\chardef\et@xmaxfam}

    End of proposed changes to etex.src

```

### 74.2.2 luatex specific settings

Switch to global cf `luatex.sty` to leave room for inserts not really needed for luatex but possibly most compatible with existing use.

```

39 \expandafter\let\csname newcount\expandafter\expandafter\endcsname
40     \csname globcount\endcsname
41 \expandafter\let\csname newdimen\expandafter\expandafter\endcsname
42     \csname globdimen\endcsname
43 \expandafter\let\csname newskip\expandafter\expandafter\endcsname
44     \csname globskip\endcsname
45 \expandafter\let\csname newbox\expandafter\expandafter\endcsname
46     \csname globbox\endcsname

```

Define `\e@alloc` as in latex (the existing macros in `etex.src` hard to extend to further register types as they assume specific 26x and 27x count range. For compatibility the existing register allocation is not changed.

```

47 \chardef\e@alloc@top=65535
48 \let\e@alloc\chardef\chardef

49 \def\e@alloc#1#2#3#4#5#6{%
50   \global\advance#3\@ne
51   \e@ch@ck{#3}{#4}{#5}#1%
52   \allocationnumber#3\relax
53   \global#2#6\allocationnumber
54   \wlog{\string#6=\string#1\the\allocationnumber}}%

55 \gdef\e@ch@ck#1#2#3#4{%
56   \ifnum#1<#2\else
57     \ifnum#1=#2\relax
58       #1\@ccclvi
59       \ifx\count#4\advance#1 10 \fi
60     \fi
61     \ifnum#1<#3\relax
62     \else
63       \errmessage{No room for a new \string#4}%
64     \fi
65   \fi}%

```

Two simple L<sup>A</sup>T<sub>E</sub>X macros used in `lAtex.sty`.

```

66 \long\def\@gobble#1{}
67 \long\def\@firstofone#1{#1}

68 % Fix up allocations not to clash with |etex.src|.

```

```

69 \expandafter\csname newcount\endcsname\@alloc@attribute@count
70 \expandafter\csname newcount\endcsname\@alloc@ccodetable@count
71 \expandafter\csname newcount\endcsname\@alloc@luafunction@count
72 \expandafter\csname newcount\endcsname\@alloc@whatsit@count
73 \expandafter\csname newcount\endcsname\@alloc@bytecode@count
74 \expandafter\csname newcount\endcsname\@alloc@luachunk@count

```

End of conditional setup for plain T<sub>E</sub>X / old L<sup>A</sup>T<sub>E</sub>X.

```

75 \fi
76 \</tex>

```

### 74.3 Attributes

`\newattribute` As is generally the case for the LuaT<sub>E</sub>X registers we start here from 1. Notably, some code assumes that `\attribute0` is never used so this is important in this case.

```

77 \ifx\@alloc@attribute@count\@undefined
78   \countdef\@alloc@attribute@count=258
79 \fi
80 \def\newattribute#1{%
81   \@alloc@attribute\attributedef
82   \@alloc@attribute@count\m@ne\@alloc@top#1%
83 }
84 \@alloc@attribute@count=\z@

```

`\setattribute` Handy utilities.

```

\unsetattribute 85 \def\setattribute#1#2{#1=\numexpr#2\relax}
86 \def\unsetattribute#1{#1=-"7FFFFFFF\relax}

```

### 74.4 Category code tables

`\newcatcodetable` Category code tables are allocated with a limit half of that used by LuaT<sub>E</sub>X for everything else. At the end of allocation there needs to be an initialisation step. Table 0 is already taken (it's the global one for current use) so the allocation starts at 1.

```

87 \ifx\@alloc@ccodetable@count\@undefined
88   \countdef\@alloc@ccodetable@count=259
89 \fi
90 \def\newcatcodetable#1{%
91   \@alloc@catcodetable\chardef
92   \@alloc@ccodetable@count\m@ne{"8000}#1%
93   \initcatcodetable\allocationnumber
94 }
95 \@alloc@ccodetable@count=\z@

```

`\catcodetable@initex` Save a small set of standard tables. The Unicode data is read here in a group  
`\catcodetable@string` avoiding any global definitions: that needs a bit of effort so that in package/plain  
`\catcodetable@latex` mode there is no effect on any settings already in force.

```

\catcodetable@atletter 96 \newcatcodetable\catcodetable@initex
97 \newcatcodetable\catcodetable@string
98 \begingroup
99   \def\setrangeecatcode#1#2#3{%

```

```

100 \ifnum#1>#2 %
101 \expandafter\@gobble
102 \else
103 \expandafter\@firstofone
104 \fi
105 {%
106 \catcode#1=#3 %
107 \expandafter\setrangecatcode\expandafter
108 {\number\numexpr#1 + 1\relax}{#2}{#3}
109 }%
110 }
111 \@firstofone{%
112 \catcodetable\catcodetable@initex
113 \catcode0=12 %
114 \catcode13=12 %
115 \catcode37=12 %
116 \setrangecatcode{65}{90}{12}%
117 \setrangecatcode{97}{122}{12}%
118 \catcode92=12 %
119 \catcode127=12 %
120 \savecatcodetable\catcodetable@string
121 \endgroup
122 }%
123 \newcatcodetable\catcodetable@latex
124 \newcatcodetable\catcodetable@atletter
125 \begingroup
126 \let\ENDGROUP\endgroup
127 \let\begingroup\relax
128 \let\endgroup\relax
129 \let\global\relax
130 \let\gdef\def
131 \input{unicode-letters.def}%
132 \let\endgroup\ENDGROUP
133 \@firstofone{%
134 \catcode64=12 %
135 \savecatcodetable\catcodetable@latex
136 \catcode64=11 %
137 \savecatcodetable\catcodetable@atletter
138 }
139 \endgroup

```

## 74.5 Named Lua functions

`\newluafunction` Much the same story for allocating Lua<sub>TEX</sub> functions except here they are just numbers so are allocated in the same way as boxes. Lua index from 1 so once again slot 0 is skipped.

```

140 \ifx\e@alloc@luafunction@count\undefined
141 \countdef\e@alloc@luafunction@count=260
142 \fi
143 \def\newluafunction{%
144 \e@alloc@luafunction\e@alloc@chardef
145 \e@alloc@luafunction@count\m@ne\e@alloc@top
146 }

```

```
147 \e@alloc@luafunction@count=\z@
```

## 74.6 Custom whatsits

`\newwhatsit` These are only settable from Lua but for consistency are definable here.

```
148 \ifx\e@alloc@whatsit@count\@undefined
149 \countdef\e@alloc@whatsit@count=261
150 \fi
151 \def\newwhatsit#1{%
152   \e@alloc\whatsit\e@alloc@chardef
153   \e@alloc@whatsit@count\m@ne\e@alloc@top#1%
154 }
155 \e@alloc@whatsit@count=\z@
```

## 74.7 Lua bytecode registers

`\newluabytocode` These are only settable from Lua but for consistency are definable here.

```
156 \ifx\e@alloc@bytecode@count\@undefined
157 \countdef\e@alloc@bytecode@count=262
158 \fi
159 \def\newluabytocode#1{%
160   \e@alloc\luabytocode\e@alloc@chardef
161   \e@alloc@bytecode@count\m@ne\e@alloc@top#1%
162 }
163 \e@alloc@bytecode@count=\z@
```

## 74.8 Lua chunk registers

`\newluachunkname` As for bytecode registers, but in addition we need to add a string to the `lua.name` table to use in stack tracing. We use the name of the command passed to the allocator, with no backslash.

```
164 \ifx\e@alloc@luachunk@count\@undefined
165 \countdef\e@alloc@luachunk@count=263
166 \fi
167 \def\newluachunkname#1{%
168   \e@alloc\luachunk\e@alloc@chardef
169   \e@alloc@luachunk@count\m@ne\e@alloc@top#1%
170   {\escapechar\m@ne
171    \directlua{lua.name[\the\allocationnumber]="\string#1"}}%
172 }
173 \e@alloc@luachunk@count=\z@
```

## 74.9 Lua loader

Load the Lua code at the start of every job. For the conversion of T<sub>E</sub>X into numbers at the Lua side we need some known registers: for convenience we use a set of systematic names, which means using a group around the Lua loader.

```
174 (2ekernel)\everyjob\expandafter{%
175 (2ekernel) \the\everyjob
176 \begingroup
177 \attributedef\attributezero=0 %
178 \chardef \charzero =0 %
```

Note name change required on older luatex, for hash table access.

```

179 \countdef \CountZero =0 %
180 \dimendef \dimenzero =0 %
181 \mathchardef \mathcharzero =0 %
182 \muskipdef \muskipzero =0 %
183 \skipdef \skipzero =0 %
184 \toksdef \tokszero =0 %
185 \directlua{require("ltnuatex")}
186 \endgroup
187 <2ekernel>}
188 <latexrelease>\EndIncludeInRelease

189 % \changes{v1.0b}{2015/10/02}{Fix backing out of \TeX{ } code}
190 % \changes{v1.0c}{2015/10/02}{Allow backing out of Lua code}
191 <latexrelease>\IncludeInRelease{0000/00/00}
192 <latexrelease> \{\newluafunction\}{LuaTeX}%
193 <latexrelease>\let\@alloc@attribute@count\@undefined
194 <latexrelease>\let\newattribute\@undefined
195 <latexrelease>\let\setattribute\@undefined
196 <latexrelease>\let\unsetattribute\@undefined
197 <latexrelease>\let\@alloc@ccodetable@count\@undefined
198 <latexrelease>\let\newcatcodetable\@undefined
199 <latexrelease>\let\catcodetable@initex\@undefined
200 <latexrelease>\let\catcodetable@string\@undefined
201 <latexrelease>\let\catcodetable@latex\@undefined
202 <latexrelease>\let\catcodetable@atletter\@undefined
203 <latexrelease>\let\@alloc@luafunction@count\@undefined
204 <latexrelease>\let\newluafunction\@undefined
205 <latexrelease>\let\@alloc@luafunction@count\@undefined
206 <latexrelease>\let\newwhatsit\@undefined
207 <latexrelease>\let\@alloc@whatsit@count\@undefined
208 <latexrelease>\let\newluabytecode\@undefined
209 <latexrelease>\let\@alloc@bytecode@count\@undefined
210 <latexrelease>\let\newluachunkname\@undefined
211 <latexrelease>\let\@alloc@luachunk@count\@undefined
212 <latexrelease>\directlua{luatexbase.uninstall()}
213 <latexrelease>\EndIncludeInRelease

214 <2ekernel | latexrelease>\fi
215 </2ekernel | tex | latexrelease>

```

## 74.10 Lua module preliminaries

216 <\*lua>

Some set up for the Lua module which is needed for all of the Lua functionality added here.

**luatexbase** Set up the table for the returned functions. This is used to expose all of the public functions.

```

217 luatexbase = luatexbase or { }
218 local luatexbase = luatexbase

```

Some Lua best practice: use local versions of functions where possible.

```

219 local string_gsub = string.gsub
220 local tex_count = tex.count

```

```

221 local tex_setattribute = tex.setattribute
222 local tex_setcount     = tex.setcount
223 local texio_write_nl   = texio.write_nl

```

## 74.11 Lua module utilities

### 74.11.1 Module tracking

**modules** To allow tracking of module usage, a structure is provided to store information and to return it.

```

224 local modules = modules or { }

```

**provides\_module** Modelled on `\ProvidesPackage`, we store much the same information but with a little more structure.

```

225 local function provides_module(info)
226   if not (info and info.name) then
227     luatexbase_error("Missing module name for provides_modules")
228     return
229   end
230   local function spaced(text)
231     return text and (" " .. text) or ""
232   end
233   texio_write_nl(
234     "log",
235     "Lua module: " .. info.name
236       .. spaced(info.date)
237       .. spaced(info.version)
238       .. spaced(info.description)
239   )
240   modules[info.name] = info
241 end
242 luatexbase.provides_module = provides_module

```

### 74.11.2 Module messages

There are various warnings and errors that need to be given. For warnings we can get exactly the same formatting as from `TEX`. For errors we have to make some changes. Here we give the text of the error in the `LATEX` format then force an error from Lua to halt the run. Splitting the message text is done using `\n` which takes the place of `\MessageBreak`.

First an auxiliary for the formatting: this measures up the message leader so we always get the correct indent.

```

243 local function msg_format(mod, msg_type, text)
244   local leader = ""
245   local cont
246   if mod == "LaTeX" then
247     cont = string_gsub(leader, ".", " ")
248     leader = leader .. "LaTeX: "
249   else
250     first_head = leader .. "Module " .. msg_type
251     cont = "(" .. mod .. ")"
252     .. string_gsub(first_head, ".", " ")
253     first_head = leader .. "Module " .. mod .. " " .. msg_type .. ":"

```

```

254 end
255 if msg_type == "Error" then
256     first_head = "\n" .. first_head
257 end
258 if string.sub(text,-1) ~= "\n" then
259     text = text .. " "
260 end
261 return first_head .. " "
262     .. string_gsub(
263         text
264     .. "on input line "
265         .. tex.inputlineno, "\n", "\n" .. cont .. " "
266     )
267     .. "\n"
268 end

module_info Write messages.
module_warning 269 local function module_info(mod, text)
module_error 270     texio_write_nl("log", msg_format(mod, "Info", text))
271 end
272 luatexbase.module_info = module_info
273 local function module_warning(mod, text)
274     texio_write_nl("term and log", msg_format(mod, "Warning", text))
275 end
276 luatexbase.module_warning = module_warning
277 local function module_error(mod, text)
278     error(msg_format(mod, "Error", text))
279 end
280 luatexbase.module_error = module_error

Dedicated versions for the rest of the code here.
281 local function luatexbase_warning(text)
282     module_warning("luatexbase", text)
283 end
284 local function luatexbase_error(text)
285     module_error("luatexbase", text)
286 end

```

## 74.12 Accessing register numbers from Lua

Collect up the data from the T<sub>E</sub>X level into a Lua table: from version 0.80, LuaT<sub>E</sub>X makes that easy.

```

287 local luaregisterbasetable = { }
288 local registermap = {
289     attributezero = "assign_attr"    ,
290     charzero      = "char_given"    ,
291     CountZero     = "assign_int"    ,
292     dimenzero     = "assign_dimen"  ,
293     mathcharzero  = "math_given"    ,
294     muskipzero    = "assign_mu_skip",
295     skipzero      = "assign_skip"   ,
296     tokszero      = "assign_toks"   ,
297 }

```

```

298 local i, j
299 local createtoken
300 if tex.luatexversion > 79 then
301   createtoken = newtoken.create
302 end
303 local hashtokens = tex.hashtokens
304 local luatexversion = tex.luatexversion
305 for i,j in pairs (registermap) do
306   if luatexversion < 80 then
307     luaregisterbasetable[hashtokens()[i][1]] =
308       hashtokens()[i][2]
309   else
310     luaregisterbasetable[j] = createtoken(i).mode
311   end
312 end

```

**registernumber** Working out the correct return value can be done in two ways. For older LuaTeX releases it has to be extracted from the `hashtokens`. On the other hand, newer LuaTeX's have `newtoken`, and whilst `.mode` isn't currently documented, Hans Hagen pointed to this approach so we should be OK.

```

313 local registernumber
314 if luatexversion < 80 then
315   function registernumber(name)
316     local nt = hashtokens()[name]
317     if(nt and luaregisterbasetable[nt[1]]) then
318       return nt[2] - luaregisterbasetable[nt[1]]
319     else
320       return false
321     end
322   end
323 else
324   function registernumber(name)
325     local nt = createtoken(name)
326     if(luaregisterbasetable[nt.cmdname]) then
327       return nt.mode - luaregisterbasetable[nt.cmdname]
328     else
329       return false
330     end
331   end
332 end
333 luatexbase.registernumber = registernumber

```

## 74.13 Attribute allocation

**new\_attribute** As attributes are used for Lua manipulations its useful to be able to assign from this end.

```

334 local attributes=setmetatable(
335 {}),
336 {
337   __index = function(t,key)
338     return registernumber(key) or nil
339   end}
340 )

```

```

341 luatexbase.attributes=attributes
342 local function new_attribute(name)
343   tex_setcount("global", "e@alloc@attribute@count",
344               tex_count["e@alloc@attribute@count"] + 1)
345   if tex_count["e@alloc@attribute@count"] > 65534 then
346     luatexbase_error("No room for a new \\attribute")
347     return -1
348   end
349   attributes[name]= tex_count["e@alloc@attribute@count"]
350   texio_write_nl("Lua-only attribute " .. name .. " = " ..
351                 tex_count["e@alloc@attribute@count"])
352   return tex_count["e@alloc@attribute@count"]
353 end
354 luatexbase.new_attribute = new_attribute

```

## 74.14 Custom whatsit allocation

`new_whatsit` Much the same as for attribute allocation in Lua

```

355 local function new_whatsit(name)
356   tex_setcount("global", "e@alloc@whatsit@count",
357               tex_count["e@alloc@whatsit@count"] + 1)
358   if tex_count["e@alloc@whatsit@count"] > 65534 then
359     luatexbase_error("No room for a new custom whatsit")
360     return -1
361   end
362   texio_write_nl("Custom whatsit " .. (name or "") .. " = " ..
363                 tex_count["e@alloc@whatsit@count"])
364   return tex_count["e@alloc@whatsit@count"]
365 end
366 luatexbase.new_whatsit = new_whatsit

```

## 74.15 Bytecode register allocation

`new_bytecode` Much the same as for attribute allocation in Lua. The optional  $\langle name \rangle$  argument is used in the log if given.

```

367 local function new_bytecode(name)
368   tex_setcount("global", "e@alloc@bytecode@count",
369               tex_count["e@alloc@bytecode@count"] + 1)
370   if tex_count["e@alloc@bytecode@count"] > 65534 then
371     luatexbase_error("No room for a new bytecode register")
372     return -1
373   end
374   texio_write_nl("Lua bytecode " .. (name or "") .. " = " ..
375                 tex_count["e@alloc@bytecode@count"])
376   return tex_count["e@alloc@bytecode@count"]
377 end
378 luatexbase.new_bytecode = new_bytecode

```

## 74.16 Lua chunk name allocation

`new_chunkname` As for bytecode registers but also store the name in the `lua.name` table.

```

379 local function new_chunkname(name)

```

```

380 tex_setcount("global", "e@alloc@luachunk@count",
381               tex_count["e@alloc@luachunk@count"] + 1)
382 local chunkname_count = tex_count["e@alloc@luachunk@count"]
383 chunkname_count = chunkname_count + 1
384 if chunkname_count > 65534 then
385   luatexbase_error("No room for a new chunkname")
386   return -1
387 end
388 lua.name[chunkname_count]=name
389 texio_write_nl("Lua chunkname " .. (name or "") .. " = " ..
390               chunkname_count .. "\n")
391 return chunkname_count
392 end
393 luatexbase.new_chunkname = new_chunkname

```

## 74.17 Lua callback management

The native mechanism for callbacks in Lua allows only one per function. That is extremely restrictive and so a mechanism is needed to add and remove callbacks from the appropriate hooks.

### 74.17.1 Housekeeping

The main table: keys are callback names, and values are the associated lists of functions. More precisely, the entries in the list are tables holding the actual function as **func** and the identifying description as **description**. Only callbacks with a non-empty list of functions have an entry in this list.

```

394 local callbacklist = callbacklist or { }

```

Numerical codes for callback types, and name-to-value association (the table keys are strings, the values are numbers).

```

395 local list, data, exclusive, simple = 1, 2, 3, 4
396 local types = {
397   list      = list,
398   data      = data,
399   exclusive = exclusive,
400   simple    = simple,
401 }

```

Now, list all predefined callbacks with their current type, based on the LuaTeX manual version 0.80. A full list of the currently-available callbacks can be obtained using

```

\directlua{
  for i,_ in pairs(callback.list()) do
    texio.write_nl("- " .. i)
  end
}
\bye

```

in plain LuaTeX. (Some undocumented callbacks are omitted as they are to be removed.)

```

402 local callbacktypes = callbacktypes or {

```

Section 4.1.1: file discovery callbacks.

```
403 find_read_file      = exclusive,
404 find_write_file     = exclusive,
405 find_font_file      = data,
406 find_output_file    = data,
407 find_format_file    = data,
408 find_vf_file        = data,
409 find_map_file       = data,
410 find_enc_file       = data,
411 find_sfd_file       = data,
412 find_pk_file        = data,
413 find_data_file      = data,
414 find_opentype_file  = data,
415 find_truetype_file  = data,
416 find_type1_file     = data,
417 find_image_file     = data,
```

Section 4.1.2: file reading callbacks.

```
418 open_read_file      = exclusive,
419 read_font_file      = exclusive,
420 read_vf_file        = exclusive,
421 read_map_file       = exclusive,
422 read_enc_file       = exclusive,
423 read_sfd_file       = exclusive,
424 read_pk_file        = exclusive,
425 read_data_file      = exclusive,
426 read_truetype_file  = exclusive,
427 read_type1_file     = exclusive,
428 read_opentype_file  = exclusive,
```

Section 4.1.3: data processing callbacks.

```
429 process_input_buffer = data,
430 process_output_buffer = data,
431 process_jobname      = data,
432 token_filter         = exclusive,
```

Section 4.1.4: node list processing callbacks.

```
433 buildpage_filter    = simple,
434 pre_linebreak_filter = list,
435 linebreak_filter     = list,
436 post_linebreak_filter = list,
437 hpack_filter         = list,
438 vpack_filter         = list,
439 pre_output_filter    = list,
440 hyphenate            = simple,
441 ligaturing           = simple,
442 kerning              = simple,
443 mlist_to_hlist      = list,
```

Section 4.1.5: information reporting callbacks.

```
444 pre_dump            = simple,
445 start_run           = simple,
446 stop_run            = simple,
447 start_page_number   = simple,
448 stop_page_number    = simple,
```

```

449 show_error_hook      = simple,
450 show_error_message    = simple,
451 show_lua_error_hook    = simple,
452 start_file             = simple,
453 stop_file              = simple,

```

Section 4.1.6: PDF-related callbacks.

```

454 finish_pdffile = data,
455 finish_pdfpage = data,

```

Section 4.1.7: font-related callbacks.

```

456 define_font = exclusive,

```

Undocumented callbacks which are likely to get documented.

```

457 find_cidmap_file      = data,
458 pdf_stream_filter_callback = data,
459 }
460 luatexbase.callbacktypes=callbacktypes

```

**callback.register** Save the original function for registering callbacks and prevent the original being used. The original is saved in a place that remains available so other more sophisticated code can override the approach taken by the kernel if desired.

```

461 local callback_register = callback_register or callback.register
462 function callback.register()
463   luatexbase_error("Attempt to use callback.register() directly\n")
464 end

```

## 74.17.2 Handlers

The handler function is registered into the callback when the first function is added to this callback's list. Then, when the callback is called, then handler takes care of running all functions in the list. When the last function is removed from the callback's list, the handler is unregistered.

More precisely, the functions below are used to generate a specialized function (closure) for a given callback, which is the actual handler.

Handler for **data** callbacks.

```

465 local function data_handler(name)
466   return function(data, ...)
467     local i
468     for _,i in ipairs(callbacklist[name]) do
469       data = i.func(data,...)
470     end
471     return data
472   end
473 end

```

Handler for **exclusive** callbacks. We can assume `callbacklist[name]` is not empty: otherwise, the function wouldn't be registered in the callback any more.

```

474 local function exclusive_handler(name)
475   return function(...)
476     return callbacklist[name][1].func(...)
477   end
478 end

```

Handler for list callbacks.

```
479 local function list_handler(name)
480   return function(head, ...)
481     local ret
482     local alltrue = true
483     local i
484     for _,i in ipairs(callbacklist[name]) do
485       ret = i.func(head, ...)
486       if ret == false then
487         luatexbase_warning(
488           "Function 'i.description' returned false\n"
489           .. "in callback 'name'"
490         )
491         break
492       end
493       if ret ~= true then
494         alltrue = false
495         head = ret
496       end
497     end
498     return alltrue and true or head
499   end
500 end
```

Handler for simple callbacks.

```
501 local function simple_handler(name)
502   return function(...)
503     local i
504     for _,i in ipairs(callbacklist[name]) do
505       i.func(...)
506     end
507   end
508 end
```

Keep a handlers table for indexed access.

```
509 local handlers = {
510   [data]      = data_handler,
511   [exclusive] = exclusive_handler,
512   [list]      = list_handler,
513   [simple]     = simple_handler,
514 }
```

### 74.17.3 Public functions for callback management

Defining user callbacks perhaps should be in package code, but impacts on `add_to_callback`. If a default function is not required, may may be declared as `false`. First we need a list of user callbacks.

```
515 local user_callbacks_defaults = { }
```

`create_callback` The allocator itself.

```
516 local function create_callback(name, ctype, default)
517   if not name or
518     name == "" or
519     callbacktypes[name] or
```

```

520     not(default == false or type(default) == "function")
521     then
522         luatexbase_error("Unable to create callback " .. name)
523     end
524     user_callbacks_defaults[name] = default
525     callbacktypes[name] = types[ctype]
526 end
527 luatexbase.create_callback = create_callback

```

**call\_callback** Call a user defined callback. First check arguments.

```

528 local function call_callback(name,...)
529     if not name or
530         name == "" or
531         user_callbacks_defaults[name] == nil
532     then
533         luatexbase_error("Unable to call callback " .. name)
534     end
535     local l = callbacklist[name]
536     local f
537     if not l then
538         f = user_callbacks_defaults[name]
539         if l == false then
540             return nil
541         end
542     else
543         f = handlers[callbacktypes[name]](name)
544     end
545     return f(...)
546 end
547 luatexbase.call_callback=call_callback

```

**add\_to\_callback** Add a function to a callback. First check arguments.

```

548 local function add_to_callback(name, func, description)
549     if
550         not name or
551         name == "" or
552         not callbacktypes[name] or
553         type(func) ~= "function" or
554         not description or
555         description == "" then
556         luatexbase_error(
557             "Unable to register callback.\n\n"
558             .. "Correct usage:\n"
559             .. "add_to_callback(<callback>, <function>, <description>)"
560         )
561         return
562     end

```

Then test if this callback is already in use. If not, initialise its list and register the proper handler.

```

563     local l = callbacklist[name]
564     if l == nil then
565         l = { }
566         callbacklist[name] = l

```

If it is not a user defined callback use the primitive callback register.

```

567     if user_callbacks_defaults[name] == nil then
568         callback_register(name, handlers[callbacktypes[name]](name))
569     end
570 end

```

Actually register the function and give an error if more than one `exclusive` one is registered.

```

571 local f = {
572     func      = func,
573     description = description,
574 }
575 local priority = #l + 1
576 if callbacktypes[name] == exclusive then
577     if #l == 1 then
578         luatexbase_error(
579             "Cannot add second callback to exclusive function\n" ..
580             name .. "'")
581     end
582 end
583 table.insert(l, priority, f)

```

Keep user informed.

```

584 texio_write_nl(
585     "Inserting '" .. description .. "' at position "
586     .. priority .. " in '" .. name .. "'."
587 )
588 end
589 luatexbase.add_to_callback = add_to_callback

```

`remove_from_callback` Remove a function from a callback. First check arguments.

```

590 local function remove_from_callback(name, description)
591     if
592         not name or
593         name == "" or
594         not callbacktypes[name] or
595         not description or
596         description == "" then
597         luatexbase_error(
598             "Unable to remove function from callback.\n\n"
599             .. "Correct usage:\n"
600             .. "remove_from_callback(<callback>, <description>)"
601         )
602         return
603     end
604     local l = callbacklist[name]
605     if not l then
606         luatexbase_error(
607             "No callback list for '" .. name .. "'\n")
608     end

```

Loop over the callback's function list until we find a matching entry. Remove it and check if the list is empty: if so, unregister the callback handler.

```

609     local index = false
610     local i,j

```

```

611 local cb = {}
612 for i,j in ipairs(l) do
613   if j.description == description then
614     index = i
615     break
616   end
617 end
618 if not index then
619   luatexbase_error(
620     "No callback '" .. description .. "' registered for '" ..
621     name .. "'\n")
622   return
623 end
624 cb = l[index]
625 table.remove(l, index)
626 texio_write_nl(
627   "Removing '" .. description .. "' from '" .. name .. "'."
628 )
629 if #l == 0 then
630   callbacklist[name] = nil
631   callback_register(name, nil)
632 end
633 return cb.func,cb.description
634 end
635 luatexbase.remove_from_callback = remove_from_callback

```

`in_callback` Look for a function description in a callback.

```

636 local function in_callback(name, description)
637   if not name
638     or name == ""
639     or not callbacktypes[name]
640     or not description then
641     return false
642   end
643   local i
644   for _, i in pairs(callbacklist[name]) do
645     if i.description == description then
646       return true
647     end
648   end
649   return false
650 end
651 luatexbase.in_callback = in_callback

```

`disable_callback` As we subvert the engine interface we need to provide a way to access this functionality.

```

652 local function disable_callback(name)
653   if(callbacklist[name] == nil) then
654     callback_register(name, false)
655   else
656     luatexbase_error("Callback list for " .. name .. " not empty")
657   end
658 end
659 luatexbase.disable_callback = disable_callback

```

**callback\_descriptions** List the descriptions of functions registered for the given callback.

```
660 local function callback_descriptions (name)
661   local d = {}
662   if not name
663     or name == ""
664     or not callbacktypes[name]
665   then
666     return d
667   else
668     local i
669     for k, i in pairs(callbacklist[name] or {}) do
670       d[k] = i.description
671     end
672   end
673   return d
674 end
675 luatexbase.callback_descriptions = callback_descriptions
```

**uninstall** Unlike at the T<sub>E</sub>X level, we have to provide a back-out mechanism here at the same time as the rest of the code. This is not meant for use by anything other than `latexrelease`: as such this is *deliberately* not documented for users!

```
676 local function uninstall()
677   module_info(
678     "luatexbase",
679     "Uninstalling kernel luatexbase code"
680   )
681   callback.register = callback_register
682   luatexbase = nil
683 end
684 luatexbase.uninstall = uninstall
```

```
685  $\langle$ /lua $\rangle$ 
```

Reset the catcode of @.

```
686  $\langle$ tex $\rangle$ \catcode'\@=\etatcatcode\relax
```

# File O

## l<sup>t</sup>final.dtx

### 75 Final settings

This section contains the final settings for L<sup>A</sup>T<sub>E</sub>X. It initialises some debugging and typesetting parameters, sets the default `\catcodes` and `uc/lc` codes, and inputs the hyphenation file.

#### 75.1 Debugging

By default, L<sup>A</sup>T<sub>E</sub>X shows statistics:

```
1 <*2ekernel>
2 \tracingstats1
```

#### 75.2 Typesetting parameters

```
\@lowpenalty These are penalties used internally.
\@medpenalty 3 \newcount\@lowpenalty
\@highpenalty 4 \newcount\@medpenalty
5 \newcount\@highpenalty

\newmarks Allocate extended marks types if etex is active. Placed here at the end of the
format to increase compatibility with count allocations in earlier releases.
6 </2ekernel>
7 <*2ekernel | latexrelease>
8 <latexrelease>\IncludeInRelease{2015/01/01}%
9 <latexrelease> \newmarks}{Extended Allocation}%
10 \ifx\marks\@undefined\else
11 \def\newmarks{%
12 \e@alloc\marks \e@alloc@chardef{\count256}\m@ne\e@alloc@top}
13 \fi
14 </2ekernel | latexrelease>
15 <latexrelease>\EndIncludeInRelease
16 <latexrelease>\IncludeInRelease{0000/00/00}%
17 <latexrelease> \newmarks}{Extended Allocation}%
18 <latexrelease>\let\newmarks\@undefined
19 <latexrelease>\EndIncludeInRelease
20 <*2ekernel>

\newXeTeXintercharclass Allocate \XeTeXintercharclass types if xetex is active. previously defined in
xetex.ini.
21 </2ekernel>
22 <*2ekernel | latexrelease>
23 <latexrelease>\IncludeInRelease{2015/01/01}%
24 <latexrelease> \newXeTeXintercharclass}{Extended Allocation}%

Classes allocatedfrom 4 (1,2 and 3 are used by CJK), up to 254.
25 \ifx\XeTeXcharclass\@undefined
26 \else
```

```

27 \countdef\xe@alloc@intercharclass=257
28 \xe@alloc@intercharclass=\thr@@
29 \def\newXeTeXintercharclass{%
30 \e@alloc\XeTeXcharclass\chardef\xe@alloc@intercharclass\m@ne\@cclv}
31 \fi

32 /2kernel | latexrelease)
33 (latexrelease)\EndIncludeInRelease
34 (latexrelease)\IncludeInRelease{0000/00/00}%
35 (latexrelease) \newXeTeXintercharclass{Extended Allocation}%
36 (latexrelease) \ifx\XeTeXcharclass\@undefined
37 (latexrelease) \else
38 (latexrelease) \newcount\xe@alloc@intercharclass
39 (latexrelease) \xe@alloc@intercharclass=\thr@@
40 (latexrelease) \def\xe@alloc#1#2#3#4#5{\global\advance#1\@ne
41 (latexrelease) \xe@ch@ck#1#4#2%
42 (latexrelease) \allocationnumber#1%
43 (latexrelease) \global#3#5\allocationnumber
44 (latexrelease) \wlog{\string#5=\string#2\the\allocationnumber}}
45 (latexrelease) \def\xe@ch@ck#1#2#3{%
46 (latexrelease) \ifnum#1<#2\else
47 (latexrelease) \errmessage{No room for a new #3}%
48 (latexrelease) \fi}
49 (latexrelease) \def\newXeTeXintercharclass{%
50 (latexrelease) \xe@alloc@\xe@alloc@intercharclass
51 (latexrelease) \XeTeXcharclass\chardef\@cclv}
52 (latexrelease) \fi
53 (latexrelease)\EndIncludeInRelease
54 (*2kernel)

```

The default values of the picture and \fbox parameters:

```

55 \unitlength = 1pt
56 \fboxsep = 3pt
57 \fboxrule = .4pt

```

The saved value of TeX's \maxdepth:

```

58 \@maxdepth = \maxdepth

```

\vsize initialized because a \clearpage with \vsize < \topskip causes trouble.  
\@colroom and \@colht also initialized because \vsize may be set to them if a  
\clearpage is done before the \begin{document}

```

59 \vsize = 1000pt
60 \@colroom = \vsize
61 \@colht = \vsize

```

Initialise \textheight \textwidth and page style, to avoid internal errors if they  
are not set by the class.

```

62 \textheight=.5\maxdimen
63 \textwidth=\textheight
64 \ps@empty

```

### 75.3 Lccodes for hyphenation

For 7- and 8-bit engines the assumption of T1 encodings is the basis for the  
hyphenation patterns. That's not the case for the Unicode engines, where the

assumption is engine-native working. The file `unicode-letters.def` contains data extracted from the master Unicode Consortium information covering not only `\lccode` but also other related data. The `\lccode` part of that at least needs to be loaded before hyphenation is tackled: XeTeX follows the standard TeX route of building patterns into the format. LuaTeX doesn't require this data be loaded *here* but it does need to be loaded somewhere. Rather than test for the Unicode engines by name, the approach here is to look for the extended math mode handling both provide: any other engine developed in this area will presumably also provide `\Umathcode` (older XeTeX versions use `\XeTeXmathcode` so that is covered too).

```

65 \ifnum 0%
66   \ifx\Umathcode\@undefined\else 1\fi
67   \ifx\XeTeXmathcode\@undefined\else 1\fi
68   >\z@
69   \message{ Unicode character data,}
70   \input{unicode-letters.def}

```

There is one over-ride that makes sense here (see below for the same for 8-bit engines): setting the `lccode` for `-` to itself.

```

71 \lccode'\- ='\- % default hyphen char

```

The alternative is that a “traditional” engine is in use.

```

72 \else

```

We set things up so that hyphenation files can assume that the default (T1) `lccodes` are in use (at present this also sets up the `uccodes`). We temporarily define `\reserved@a` to apply `\reserved@c` to all the numbers in the range of its arguments.

```

73 \def\reserved@a#1#2{%
74   \@tempcnta#1\relax
75   \@tempcntb#2\relax
76   \reserved@b
77 }
78 \def\reserved@b{%
79   \ifnum\@tempcnta>\@tempcntb\else
80     \reserved@c\@tempcnta
81     \advance\@tempcnta\@ne
82     \expandafter\reserved@b
83   \fi
84 }

```

Depending on the TeX version, we might not be allowed to do this for non-ASCII characters.

```

85 \def\reserved@c#1{%
86   \count@=#1\advance\count@ by -"20
87   \uccode#1=\count@
88   \lccode#1=#1
89 }
90 \reserved@a{'\a}{'\z}
91 \ifnum\inputlineno=\m@ne\else
92   \reserved@a{"A0}{\BC}
93   \reserved@a{"E0}{\FF}
94 \fi

```

The upper case characters need their `\uccode` and `\lccode` values set, and their `\sfcode` set to 999.

```

95 \def\reserved@c#1{%
96   \count@=#1\advance\count@ by "20
97   \uccode#1=#1
98   \lccode#1=\count@
99   \sfcode#1=999
100 }
101 \reserved@a{'\A}{'\Z}
102 \ifnum\inputlineno=\m@ne\else
103   \reserved@a{"80}{"9C}
104   \reserved@a{"C0}{"DF}
105 \fi

```

Well, it would be nice if that were correct, but unfortunately, the Cork encoding contains some odd slots whose `uccode` or `lccode` isn't quite what you'd expect.

```

106 \uccode'\^^Y='I      % dotless i
107 \lccode'\^^Y='^^Y    % dotless i
108 \uccode'\^^Z='J      % dotless j, ae in OT1
109 \lccode'\^^Z='^^Z    % dotless j, ae in OT1
110 \ifnum\inputlineno=\m@ne\else
111   \lccode'\^^9d='i    % dotted I
112   \uccode'\^^9d='^^9d % dotted I
113   \lccode'\^^9e='^^9e % d-bar
114   \uccode'\^^9e='^^d0 % d-bar
115 \fi

```

Finally here is one that helps hyphenation in the OT1 encoding.

```

116 \lccode'\^^[='^^[    % oe in OT1

```

And we also set the `\lccode` of `\-` and `\textcompwordmark` so that they do not prevent hyphenation in the remainder of the word (as suggested by Lars Helström).

```

117 \lccode'\- ='\'      % default hyphen char
118 \lccode 127=127      % alternate hyphen char
119 \lccode 23 =23       % textcompwordmark in T1

```

End of the conditional to select either Unicode or T1 encoding defaults.

```

120 \fi

```

This is as good a place as any to active a few XeTeX-specific settings

```

121 \ifx\XeTeXuseglyphmetrics\undefined
122 \else
123   \XeTeXuseglyphmetrics=1 %
124   \XeTeXdashbreakstate=1 %
125 \fi

```

## 75.4 Hyphenation

The following code will be compiled into the format file. It checks for the existence of `hyphen.cfg` in inputs that file if found. Otherwise it inputs `hyphen.ltx`. Note that these are loaded in *before* the `\catcodes` are set, so local hyphenation files can use 8-bit input.

We try to load the customized hyphenation description file.

```

126 \InputIfFileExists{hyphen.cfg}

```

```

127      {\typeout{=====^^J%
128              Local configuration file hyphen.cfg used^^J%
129              =====}%
130      \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
131      }
132      {\input{hyphen.ltx}}
133 \let\@addtofilelist\@gobble

```

## 75.5 Font loading

Fonts loaded during the formatting process might already have changed the `\font@submax` from 0pt to something higher. If so, we put out a bold warning.

```

134 % \changes{v1.1c}{2000/08/23}{Fix typo in warning}
135 \ifdim \font@submax >\z@
136   \@font@warning{Size substitutions with differences\MessageBreak
137     up to \font@submax\space have occurred.\MessageBreak
138     \MessageBreak
139     Please check the transcript file
140     carefully\MessageBreak
141     and redo the format generation if necessary!
142     \@gobbletwo}%
143   \errhelp{Only stopped, to give you time to
144     read the above message.}
145   \errmessage{}

```

We reset the macro. Otherwise every user will get a warning on every job.

```

146 \def\font@submax{0pt}
147 \fi

```

## 75.6 Input encoding

We temporarily define `\reserved@a` to apply `\reserved@c` to all the numbers in the range of its arguments.

```

148 \def\reserved@a#1#2{%
149   \@tempcnta#1\relax
150   \@tempcntb#2\relax
151   \reserved@b
152 }
153 \def\reserved@b{%
154   \ifnum\@tempcnta>\@tempcntb\else
155     \reserved@c\@tempcnta
156     \advance\@tempcnta\@ne
157     \expandafter\reserved@b
158   \fi
159 }

```

Set the special catcodes (although some of these are useless, since an error will have occurred if the catcodes have changed). Note that `^^J` has catcode ‘other’ for use in warning messages.

```

160 \catcode'\ =10
161 \catcode'\# =6
162 \catcode'\$ =3
163 \catcode'\% =14
164 \catcode'\& =4

```

```

165 \catcode'\=0
166 \catcode'\^=7
167 \catcode'\_ =8
168 \catcode'\{=1
169 \catcode'\}=2
170 \catcode'\~=13
171 \catcode'\@=11
172 \catcode'\^^I=10
173 \catcode'\^^J=12
174 \catcode'\^^L=13
175 \catcode'\^^M=5

```

Set the ‘other’ catcodes.

```

176 \def\reserved@c#1{\catcode#1=12\relax}
177 \reserved@c{'\!}
178 \reserved@c{'\"}
179 \reserved@a{'\''}{'\?}
180 \reserved@c{'\[]}
181 \reserved@c{'\]}
182 \reserved@c{'\''}
183 \reserved@c{'\|}

```

Set the ‘letter’ catcodes.

```

184 \def\reserved@c#1{\catcode#1=11\relax}
185 \reserved@a{'\A'}{'\Z}
186 \reserved@a{'\a'}{'\z}

```

All the characters in the range 0–31 and 127–255 are illegal, *except* tab (`^^I`), nl (`^^J`), ff (`^^L`) and cr (`^^M`).

Now allow 8-bit characters, although their use in this way is strongly discouraged. See `inputenc.dtx` for a supported mechanism for 8-bit input.

```

187 \def\reserved@c#1{\catcode#1=15\relax}
188 \reserved@a{0}{'\^H}
189 \reserved@c{'\^K}
190 \reserved@a{'\^N'}{31}
191 %\ifnum\inputlineno=\m@ne
192   \catcode"7F=15
193 %\else
194 %   \reserved@a{"7F}{\^FF}
195 %\fi

```

## 75.7 Lccodes and uccodes

We now again set up the default (T1) uc/lccodes. The lower case characters need their `\uccode` and `\lccode` values set. Some of this is a repeat of the set-up before loading hyphenation files. Depending on the `TEX` version, we might not be allowed to do this for non-ASCII characters. For the Unicode engines (`XETEX` and `LuaTEX`) there is no need to do any of this: they use hyphenation data which does not alter any of the set up and so this entire block is skipped.

```

196 \ifnum 0%
197   \ifx\Umathcode\@undefined\else 1\fi
198   \ifx\XeTeXmathcode\@undefined\else 1\fi
199   >\z@
200 \else

```

```

201 \def\reserved@c#1{%
202   \count@=#1\advance\count@ by -"20
203   \uccode#1=\count@
204   \lccode#1=#1
205 }
206 \reserved@a{'\a}{'\z}
207 \ifnum\inputlineno=\m@ne\else
208   \reserved@a{"A0}{\BC}
209   \reserved@a{"E0}{\FF}
210 \fi

```

The upper case characters need their \uccode and \lccode values set, and their \sfcode set to 999.

```

211 \def\reserved@c#1{%
212   \count@=#1\advance\count@ by "20
213   \uccode#1=#1
214   \lccode#1=\count@
215   \sfcode#1=999
216 }
217 \reserved@a{'\A}{'\Z}
218 \ifnum\inputlineno=\m@ne\else
219   \reserved@a{"80}{\9C}
220   \reserved@a{"C0}{\DF}
221 \fi

```

Well, it would be nice if that were correct, but unfortunately, the Cork encoding contains some odd slots whose uccode or lccode isn't quite what you'd expect.

```

222 \uccode'\^^Y='^I      % dotless i
223 \lccode'\^^Y='^Y      % dotless i
224 \uccode'\^^Z='^J      % dotless j, ae in OT1
225 \lccode'\^^Z='^Z      % dotless j, ae in OT1
226 \ifnum\inputlineno=\m@ne\else
227   \lccode'\^^9d='^i    % dotted I
228   \uccode'\^^9d='^9d  % dotted I
229   \lccode'\^^9e='^9e  % d-bar
230   \uccode'\^^9e='^d0  % d-bar
231 \fi

```

Finally here is one that helps hyphenation in the OT1 encoding.

```

232 \lccode'\^^[='^^[    % oe in OT1
233 \fi % End of reset block for 8-bit engines

```

\MakeUppercase And whilst we're doing things with uc/lc tables, here are two commands to upper- and lower-case a string.

\@uclclist *Note* that this implementation is subject to change! At the moment we're not providing any way to extend the list of uc/lc commands, since finding a good interface is difficult. These commands have some nasty features, such as uppercasing mathematics, environment names, labels, etc. A much better long-term solution is to use all-caps fonts, but these aren't generally available.

```

234 \DeclareRobustCommand{\MakeUppercase}[1]{%
235   \def\i{I}\def\j{J}%
236   \def\reserved@a##1##2{\let##1##2\reserved@a}%
237   \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b\@gobble}%
238   \protected@edef\reserved@a{\uppercase{#1}}%

```

```

239     \reserved@a
240   }}
241 \DeclareRobustCommand{\MakeLowercase}[1]{%
242     \def\reserved@a##1##2{\let##2##1\reserved@a}%
243     \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b\@gobble}%
244     \protected@edef\reserved@a{\lowercase{#1}}%
245     \reserved@a
246   }}
247 \def\@uclclist{\oe\OE\o\O\ae\AE
248     \dh\DH\dj\DJ\l\L\ng\NG\ss\SS\th\TH}

```

The above code works, but has the nasty side-effect that if you say something like:

```

\markboth{\MakeUppercase\contentsname}
{\MakeUppercase\contentsname}

```

then the uppercasing is only done to the first letter of the contents name, since the mark expands out to:

```

\mark{\protect\MakeUppercase Table of Contents}
{\protect\MakeUppercase Table of Contents}

```

In order to get round this, we redefine `\MakeUppercase` and `\MakeLowercase` to grab their argument and brace it. This is a very low-level hack, and is *not* recommended practice! This is an instance of a general problem that makes it unsafe to grab arguments unbraced, and probably needs a more general solution. For the moment though, this hack will do:

```

249 \protected@edef\MakeUppercase#1{\MakeUppercase{#1}}
250 \protected@edef\MakeLowercase#1{\MakeLowercase{#1}}

```

## 75.8 Applying Patch files

Between major releases, small patches will be distributed in files `ltpatch.ltx` which must be added at this point.

Patch file code removed.

```

251 %\IfFileExists{ltpatch.ltx}
252 % {\typeout{=====^^J%
253 %     Applying patch file ltpatch.ltx^^J%
254 %     =====}}
255 % \def\fmtversion@topatch{unknown}
256 % \input{ltpatch.ltx}
257 % \ifx\fmtversion\fmtversion@topatch
258 % \ifx\patch@level\@undefined
259 % \typeout{^^J^^J^^J%
260 %     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
261 %     !! Patch file 'ltpatch.ltx' not suitable for this^^J%
262 %     !! version of LaTeX.^^J^^J%
263 %     !! Please check if initex found an old patch file:^^J%
264 %     !! --- if so, rename it or delete it, and redo the^^J%
265 %     !! initex run.^^J%
266 %     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J}%
267 % \batchmode \@@end
268 % \else

```

The code below adds the ‘patch level’ string to the first `\typeout` in the startup banner.

```

269 % \def\fmtversion@topatch{0}%
270 % \ifx\fmtversion@topatch\patch@level\else
271 % \def\reserved@a\typeout##1##2\reserved@a{%
272 % \typeout{##1 patch level \patch@level}##2}
273 % \everyjob\expandafter\expandafter\expandafter{%
274 % \expandafter\reserved@a\the\everyjob\reserved@a}
275 % \let\reserved@a\relax
276 % \the\everyjob
277 % \fi
278 % \fi
279 % \else
280 % \typeout{^^J^^J^^J%
281 % !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
282 % !! Patch file 'ltpatch.ltx' (for version <\fmtversion@topatch>)^^J%
283 % !! is not suitable for version <\fmtversion> of LaTeX.^^J^^J%
284 % !! Please check if initex found an old patch file:^^J%
285 % !! --- if so, rename it or delete it, and redo the^^J%
286 % !! initex run.^^J%
287 % !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J}%
288 % \batchmode \@@end
289 % \fi
290 % \let\fmtversion@topatch\relax
291 % }{}

```

## 75.9 Freeing Memory

`\reserved@a` And just to make sure nobody relies on those definitions of `\reserved@b` and `\reserved@b` friends. These macros are reserved for use in the kernel. *Do not use them as general scratch macros.*

```
292 \let\reserved@a\@filelist
293 \let\reserved@b=\@undefined
294 \let\reserved@c=\@undefined
295 \let\reserved@d=\@undefined
296 \let\reserved@e=\@undefined
297 \let\reserved@f=\@undefined
```

\toks

```
298 \toks0{}
299 \toks2{}
300 \toks4{}
301 \toks6{}
302 \toks8{}
```

`\errhelp` Empty the error help message, which may have some rubbish:  
303 `\errhelp{}`

### 75.10 Initialise file list

`\@providesfile` Initialise for use in the document. During initex a modified version has been used which leaves debugging information for `latexbug.tex`.

```

304 \def\@providesfile#1[#2]{%
305     \wlog{File: #1 #2}%
306     \expandafter\xdef\csname ver@#1\endcsname{#2}%
307 \endgroup}

```

`\@filelist` Reset `\@filelist` so files input while making the format are not listed. The list built up so far may take up a lot of memory and so it is moved to `\reserved@a` where it will be overwritten as soon as almost any L<sup>A</sup>T<sub>E</sub>X command is issued in a class file. However the `latexbug.tex` program will be able to access this information and insert it into a bug report.

```

308 \let\@filelist\@gobble
309 \def\@addtofilelist#1{\xdef\@filelist{\@filelist,#1}}%

```

## 75.11 Dumping the format

Finally we make `@` into a letter, ensure the format will be in the ‘normal’ error mode, and dump everything into the format file.

```

310 \makeatother
311 \errorstopmode
312 \dump
313 \</2ekernel>

```

# Change History

1985/11/04 ltmath.dtx LaTeX2.09	<code>\mathversion</code> : Test if version defined added. . . . . 147
General: produce warning message if line extends into margin. Doesn't warn about formula overprinting equation number. 264	
1989/04/10 ltffssbas.dtx v1.0a	1989/04/29 ltffssbas.dtx v1.0i
General: Starting with version numbers! <code>\ifmmode</code> added in <code>\math@group</code> . . . . . 139	General: Removed the <code>\halign</code> <code>\noalign</code> correction (wasn't bugfree) . . . . . 139
1989/04/10 ltffssbas.dtx v1.0b	1989/04/29 ltffssini.dtx v1.0f
General: <code>\preload@sizes</code> added. 139	General: Corrections to L <sup>A</sup> T <sub>E</sub> X tabular env. added. . . . . 208
<code>\wrong@fontshape</code> changed to define substitution font/shape macro. . . . . 139	1989/05/01 ltffssbas.dtx v1.0j
1989/04/10 ltffssini.dtx v1.0a	General: Default for <code>\base-linestretch</code> added. . . . . 139
General: Starting with version numbers <code>\newif</code> for <code>\@tempswa</code> added since this switch is unknown at the time when this file is read in. (latex.tex is loaded later.) <code>\math@famname</code> changed to <code>\math@version</code> . . . . . 208	1989/05/22 ltffssbas.dtx v1.0k
1989/04/14 ltffssbas.dtx v1.0c	General: Lines longer than 72 characters folded. . . . . 139
General: More documentation added. . . . . 139	1989/05/22 ltffssini.dtx v1.0g
1989/04/15 ltffssini.dtx v1.0b	General: Lines shortened to 72 characters . . . . . 208
General: <code>\mathfontset</code> renamed to <code>\mathversion</code> . . . . . 208	1989/09/14 ltffssbas.dtx v1.0m
1989/04/19 ltffssbas.dtx v1.0d	General: Global replacement: <code>\group</code> to <code>\mathgroup</code> . . . . 139
General: Even more doc. . . . . 139	<code>\mathversion</code> : Corrected typo: <code>\endscname</code> to <code>\endcsname</code> . . 147
1989/04/21 ltffssbas.dtx v1.0e	1989/11/07 ltffssini.dtx v1.0i
General: Documentation is fun! Parameters of <code>\define@mathalphabet</code> changed. 139	General: All family, series, and shape names abbreviated. . . 208
1989/04/21 ltffssini.dtx v1.0c	1989/11/08 ltffssbas.dtx v1.0o
General: Changed to conform to fam.tex. . . . . 208	General: First parameter of <code>\define@mathalphabet</code> and <code>\define@mathgroup</code> changed from string to control sequence. . . 139
1989/04/23 ltffssbas.dtx v1.0f	1989/11/14 ltffssbas.dtx v1.0p
General: % in <code>\getanddefinefonts</code> added. . . . . 139	<code>\math@version</code> : Math version prefix 'mv@' added. . . . . 147
1989/04/26 ltffssini.dtx v1.0d	1989/11/19 ltffssbas.dtx v1.0q
General: <code>\xpt</code> added. . . . . 208	<code>\define@newfont</code> : Group added. 149
1989/04/27 ltffssbas.dtx v1.0g	<code>\wrong@fontshape</code> : Instead of calling <code>\family\default@family</code> , etc. we directly set <code>\f@family</code> , etc. . . . . 152
General: Documentation revised. 139	1989/11/22 ltffssbas.dtx v1.0r
1989/04/27 ltffssini.dtx v1.0e	<code>\math@version</code> : <code>\def</code> → <code>\edef</code> for <code>\math@version</code> . . . . . 147
General: Definitions of L <sup>A</sup> T <sub>E</sub> X symbols corrected. . . . . 208	1989/11/25 ltffssbas.dtx v1.0s
1989/04/29 ltffssbas.dtx v1.0h	General: All <code>\edef\font@name</code> changed to <code>\xdef\font@name</code> . Necessary after introduction of <code>\begingroup/\endgroup</code> in v1.0q. . . . . 139
General: Documented problem with <code>\halign</code> , and <code>\noalign</code> . . . 139	extra// → + in <code>\extra@def</code> . . 139

1989/11/26 ltfssbas.dtx v1.0t	Macro <code>\no@alphabet@help</code> added . . . . .	139
<code>\select@group: \bgroup/\egroup</code> changed to <code>\begin-group/\endgroup</code> to avoid empty Ord atom on math list. 154	<code>\no@alphabet@error: Changed to error call</code> . . . . .	139
1989/12/02 ltfssini.dtx v1.1b	1990/01/25 ltfssini.dtx v1.1e	
General: <code>\rmmath</code> renamed to <code>\mathrm</code> . . . . .	<code>\nfss@text: Macro added.</code> . . . . .	211
1989/12/03 ltfssini.dtx v1.1c	1990/01/27 ltfssbas.dtx v1.2d	
General: Some internal macros renamed to make them inaccessible. . . . .	<code>\DeclarePreloadSizes: Font identifier set to <code>\relax</code>.</code> . . . . .	144
1989/12/05 ltfssbas.dtx v1.0u	1990/01/28 ltfssbas.dtx v1.2e	
<code>\addto@hook: \addto@hook</code> added. 157	<code>\mathgroup: \newfam</code> let to <code>\new@mathgroup.</code> . . . . .	139
1989/12/05 ltfssstrc.dtx v1.0u fam.dtx	1990/01/28 ltfssbas.dtx v1.2f	
<code>\every@math@size: Hook \every@size</code> added. . . . .	<code>\define@newfont: Added call to <code>\curr@fontshape</code> macro to allow substitution.</code> . . . . .	150
1989/12/13 ltfssstrc.dtx v1.0f	<code>\wrong@fontshape: Warning message slightly changed.</code> . . . . .	152
<code>\use@mathgroup: \expandafter</code> added before final <code>\fi.</code> . . . . .	1990/01/28 ltfssini.dtx v1.2b	
1989/12/16 ltfssbas.dtx v1.1a	<code>\em: Call to <code>\nomath</code> added.</code> . . . . .	209
<code>\select@group: \relax</code> in front added. . . . .	1990/02/08 ltfssini.dtx v1.1g	
Now four arguments. . . . .	General: Protected the commands <code>\family</code> , <code>\series</code> , <code>\shape</code> , <code>\size</code> , <code>\selectfont</code> , and <code>\mathversion.</code> . . . . .	208
Redefinition of alphabet now simpler. . . . .	1990/02/16 ltfssbas.dtx v1.2g	
Usage of ‘=’ macro added. . . . .	General: Support for changes of <code>\baselineskip</code> without changing the size. . . . .	139
1989/12/16 ltfssstrc.dtx v1.1a	<code>\math@version: \nomath</code> added. 147	
<code>\selectfont: Changed order of calls.</code> . . . . .	1990/02/16 ltfssstrc.dtx v1.0i	
<code>\use@mathgroup: Redefinition of alphabet now simpler.</code> . . . . .	<code>\selectfont: Changed <code>\f@size</code> to <code>\lcl@currsz</code> (see fam file).</code> 163	
Usage of ‘=’ macro added. . . . .	1990/02/18 ltfssstrc.dtx v1.0j	
1990/01/18 ltfssstrc.dtx v1.0h	General: Redefine unprotected version <code>\p@selectfont</code> instead of <code>\selectfont.</code> . . . . .	163
General: <code>\tracingfonts</code> meaning changed. . . . .	1990/03/14 ltfssstrc.dtx v1.0k	
1990/01/20 ltfssbas.dtx v1.2a	General: Added code for TeX3. . . . .	159
<code>\math@bgroup: Def. placed in this file.</code> . . . . .	<code>\extract@font: Added code for TeX3.</code> . . . . .	162
<code>\math@egroup: Def. placed in this file.</code> . . . . .	<code>\selectfont: Added code for TeX3.</code> . . . . .	163
<code>\select@group: Def for alph id changed.</code> . . . . .	1990/03/30 ltfssbas.dtx v1.2h	
1990/01/21 ltfssbas.dtx v1.2b	<code>\math@egroup: Changed to have one arg.</code> . . . . .	156
<code>\select@group: Code moved to <code>\use@mathgroup.</code></code> . . . . .	1990/03/30 ltfssstrc.dtx v1.2h	
1990/01/21 ltfssstrc.dtx v1.2b	<code>\use@mathgroup: Third argument removed (see <code>\math@egroup</code>).</code> 169	
<code>\use@mathgroup: Macro added to allow cleaner interface.</code> . . . . .	1990/04/01 ltfssbas.dtx v1.2i	
1990/01/23 ltfssbas.dtx v1.2c	General: Code added from <code>tracefnt.dtx.</code> . . . . .	139
General: <code>\no@version@warning</code> renamed to <code>\no@alphabet@error.</code> . . . . .	Support for TeX3. . . . .	139

- 1990/04/01 ltfsstrc.dtx v1.0l  
 General: Part of code moved to fam.dtx. . . . . 159  
`\tracingfonts`: Check if `\tracingfonts` already defined. . . 160
- 1990/04/01 ltfsstrc.dtx v1.0o  
`\tracingfonts`: Check if `\tracingfonts` defined removed again. . . . . 160
- 1990/04/02 ltfsini.dtx v1.1i  
 General: `\input` of files now handled by docstrip. . . . . 208
- 1990/04/05 ltfsstrc.dtx v1.0m  
`\selectfont`: Call `\tracingon` only if `\tracingfonts` greater than 3. . . . . 163
- 1990/05/05 ltfsstrc.dtx v1.0n  
`\selectfont`: `\tracingon` with new syntax. . . . . 163
- 1990/06/23 ltfsini.dtx v1.1k  
`\nfss@text`: Changed to `\mbox`. . 211
- 1990/06/24 ltfsbas.dtx v1.2j  
`\DeclarePreloadSizes`: Missing percent added. . . . . 143
- 1990/06/24 ltfsstrc.dtx v1.0o  
`\baselinestretch`: Moved to tracefnt.dtx. . . . . 166  
`\getanddefine@fonts`: `\Adding` tracing code. . . . . 170  
`\Macro` moved from fam.dtx. . 170  
 Adding debug code. . . . . 170  
`\use@mathgroup`: Tracing code added. . . . . 169
- 1990/06/30 ltfsbas.dtx v1.2l  
`\showhyphens`: Macro added. . . 157
- 1990/06/30 ltfsstrc.dtx v1.0p  
`\use@mathgroup`: Added `\relax` after math group number. . . 169
- 1990/07/07 ltfsstrc.dtx v1.0q  
`\getanddefine@fonts`: Group number added to tracing. . . 170  
`\math@egroup`: Tracing code added. . . . . 169  
`\use@mathgroup`: Group number added to tracing. . . . . 169
- 1990/08/27 ltfsstrc.dtx 1.0r  
`\type@restoreinfo`: Some extra tracing info. . . . . 165
- 1990/08/27 ltfsstrc.dtx v1.0r  
`\getanddefine@fonts`: Correcting missing name after `\tracingon`. . . . . 170
- 1991/03/28 ltfsini.dtx v1.1m  
`\copyright`: Extra braces added. 211
- 1991/03/30 ltfsini.dtx v1.2g  
`\newfont`: Definition added. . . . 210  
`\symbol`: Definition added. . . . 210
- 1991/07/24 ltmiscen.dtx LaTeX2.09  
`\@verbatim`: Added `\penalty\interlinepenalty` to definition of `\par` so that `\samepage` works. . . . . 255
- 1991/08/14 ltmath.dtx LaTeX2.09  
`\cases`: (RmS) inserted extra braces around entry for NFSS 261
- 1991/08/14 ltpictur.dtx LaTeX2.09  
 General: (RmS) inserted extra braces around entry for NFSS 321
- 1991/08/14 ltthm.dtx LaTeX2.09  
`\@endtheorem`: Moved `\itshape` after `\item` to make it work with NFSS . . . . . 342
- 1991/08/26 ltfsini.dtx v1.1n  
`\p@reset@font`: Macro introduced 211
- 1991/08/26 ltmiscen.dtx LaTeX2.09  
`\@verbatim`: `\@@par` added . . . . 255
- 1991/08/26 ltpictur.dtx LaTeX2.09  
`\endpicture`: (RmS & FmI) extra boxing level around `\@picbox` to guard against unboxing in math mode (proposed by John Hobby) . . . . . 320
- 1991/08/26 ltplain.dtx LaTeX2.09  
`\tracingall`: Added `\errorcontextlines=\maxdimen`, suggested by J. Schrod . . . . . 29
- 1991/09/29 ltboxes.dtx LaTeX2.09  
`\@mpfootnotetext`: (RmS) added `\reset@font` . . . . . 293
- 1991/09/29 ltfloat.dtx LaTeX2.09  
`\@footnotetext`: (RmS) added `\reset@font` . . . . . 371
- 1991/09/29 ltmath.dtx LaTeX2.09  
`\@eqnnum`: RmS: `\reset@font` added. . . . . 264
- 1991/09/29 ltsect.dtx LaTeX2.09  
`\@dottedtocline`: (RmS) added `\reset@font` for page number 352
- 1991/10/17 ltcntrl.dtx LaTeX2.09  
`\@tfor`: (RmS) `\xdef` replaced by `\def` (See FmI's array.doc) . . 54
- 1991/10/25 ltbibl.dtx LaTeX2.09  
`\@citex`: added `\reset@font`, suggested by Bernd Raichle. . . 376
- 1991/11/01 ltfloat.dtx LaTeX2.09  
`\footnote`: (RmS) Added `\let\protect\noexpand` in `\footnote`, `\footnotemark`,

- and `\footnotetext`, since  
`\xdef` is used . . . . . 370
- 1991/11/04 `ltlists.dtx` LaTeX2.09  
`\makelabel`: (RmS) added default  
definition for `\makelabel`, to  
produce an error message. . . 281
- 1991/11/04 `ltplain.dtx` RmS  
General: Removed `\itemitem`  
since never needed/useful with  
 $\text{\LaTeX}$ . . . . . 28
- 1991/11/06 `ltbibl.dtx` LaTeX2.09  
`\@citex`: added code to remove a  
leading blank . . . . . 376
- 1991/11/13 `ltbibl.dtx` LaTeX2.09  
`\@bibitem`: Changed counter `enumi`  
to `enumiv`, as it says in the com-  
ment above . . . . . 376
- 1991/11/21 `ltfssini.dtx` v1.1o  
`\p@reset@font`: Added extra  
braces for robustness. . . . . 211  
Changed to protected version of  
macro. . . . . 211
- 1991/11/22 `ltfloat.dtx` LaTeX2.09  
`\footnote`: (RmS) Added  
`\let\protect\noexpand` in  
`\@xfootnote`, `\@xfoot-`  
`notemark`, and `\@xfootnote-`  
`text` . . . . . 370
- 1991/11/22 `ltlists.dtx` LaTeX2.09  
`\@item`: (RmS) Changed second  
call to `\makelabel` to `\un-`  
`hbox\@tempboxa`. Avoids prob-  
lems with side effects in `\make-`  
`label` and is more efficient. . 281
- 1991/11/27 `ltfssbas.dtx` v1.3a  
General: All `\family`, `\shape` etc.  
renamed to `\fontfamily` etc. 139
- 1991/11/27 `ltfssini.dtx` v1.2a  
General: All `\family`, `\shape` etc.  
renamed to `\fontfamily` etc. 208
- 1992/01/06 `ltfssini.dtx` v1.2c  
General: added `slitex` code . . . . 208
- 1992/01/10 `ltbibl.dtx` LaTeX2.09  
`\@bibitem`: Changed `\c@enumiv` to  
`\value of \@listctr` . . . . . 376
- 1992/01/10 `ltmath.dtx` LaTeX2.09  
`equation`: RmS: put `\hbox` around  
`\@eqnnum` to typeset the equa-  
tion number in text mode (as in  
the `eqnarray` env.) . . . . . 264
- 1992/01/10 `ltthm.dtx` LaTeX2.09  
`\@othm`: (RmS) Check for existence  
of theorem environment . . . . 341
- 1992/01/14 `ltbibl.dtx` LaTeX2.09  
`\@biblabel`: removed `\hfill` . . . 378
- 1992/01/14 `ltsect.dtx` 0.0  
`\@starttoc`: (RmS) added `\imme-`  
`diat` to `\openout` as all `\write`  
commands are also executed  
`\immediate` . . . . . 351
- 1992/02/26 `ltbibl.dtx` LaTeX2.09  
`\@lbibitem`: Added `\hfill` to re-  
store left-alignment of bibliog-  
raphy labels in alpha style . . 376
- 1992/03/18 `ltdefns.dtx` LaTeX2.09  
General: (RmS) changed input  
channel from 0 to `\@inputcheck`  
to avoid conflicts with other  
channels allocated by `\newread` 36
- 1992/03/18 `ltfloat.dtx` LaTeX2.09  
`\@xympar`: (RmS) added  
`\global\@ignorefalse` . . . . 366  
`\end@float`: (RmS) changed `\@es-`  
`phack` to `\@Esphack` . . . . . 360
- 1992/03/18 `ltlists.dtx` 0.0  
General: RmS: added `\@nmbrlist-`  
`false` . . . . . 278
- 1992/03/18 `ltmiscen.dtx` LaTeX2.09  
`\begin`: Changed `\@ignoretrue`  
to `\@ignorefalse` (as docu-  
mented) . . . . . 253
- 1992/03/21 `ltfssini.dtx` v1.2d  
General: Renamed `\text` to  
`\nfss@text` to make it inter-  
nal. . . . . 208
- 1992/05/12 `ltfssbas.dtx` v1.3c  
`\extract@alph@from@version`:  
Macro added. . . . . 155  
`\select@group`: Added call to `\ex-`  
`tract@alph@from@version`. . 155
- 1992/07/26 `ltfssbas.dtx` v1.9a  
`\curr@fontshape`: . . . . . 149  
`\DeclareFontShape`: Introduced  
`\DeclareFontShape` . . . . . 140  
`\define@newfont`: . . . . . 149  
`\math@fonts`: . . . . . 154  
`\select@group`: . . . . . 154, 155  
`\split@name`: Added splitting into  
`\f@encoding`. . . . . 149  
`\wrong@fontshape`: . . . . . 152
- 1992/07/26 `ltfsssrc.dtx` v2.0b  
`\s@fct@`: . . . . . 178  
`\s@fct@sub`: . . . . . 179  
`\selectfont`: . . . . . 163  
`\try@simple@size`: . . . . . 172, 173  
`\try@size@range`: . . . . . 176  
`\use@mathgroup`: . . . . . 169

- 1992/08/14 ltbibl.dtx LaTeX2.09  
`\@citex`: added missing argument  
braces around `\hbox`, found by  
Ed Sznyter . . . . . 376
- 1992/08/14 ltboxes.dtx LaTeX2.09  
`\endminipage`: (RmS) replaced  
`\vskip`-`\lastskip` by `\unskip`  
(proposed by FMi) . . . . . 293
- 1992/08/17 ltbibl.dtx LaTeX2.09  
`\@citex`: simplified code for remov-  
ing leading blanks in citation  
key (proposed by Frank Jensen  
and Kresten Krab Thorup) . 376
- 1992/08/19 ltsect.dtx 0.0  
`\@xsect`: (RmS) corrected bug:  
stretch and shrink in argu-  
ment to `\hskip` previously not  
negated . . . . . 347
- 1992/08/19 ltthm.dtx LaTeX2.09  
`\@othm`: (RmS) Changed error mes-  
sage to complain about unde-  
fined counter . . . . . 341
- 1992/08/20 ltssini.dtx v1.4b  
`\@setsize`: Added `\@currsize`. . 210
- 1992/08/24 ltdefs.dtx LaTeX2.09  
`\@ifnextchar`: (Rms) `\@ifnextchar`  
didn't work if its first argument  
was an equal sign. . . . . 46
- 1992/08/24 ltmiscen.dtx LaTeX2.09  
`\begin`: Added code to `\begin` to  
remember line number. Used by  
`\@badend` to display position of  
non-matching `\begin`. . . . . 253
- `\verb`: Changed `\verb` and `\sverb`  
to work correctly in math mode 256
- 1992/08/25 ltsect.dtx LaTeX2.09  
`\@sect`: (FMi) replaced explicit set-  
ting of `\@svsec` by call to  
`\@seccntformat` . . . . . 346
- 1992/09/18 ltlists.dtx LaTeX2.09  
General: (RmS) Added warning if  
`\item` is used in math mode 279
- 1992/09/18 lttab.dtx LaTeX2.09  
`\@array`: Changed `\par` to `\@empty`  
to avoid starting new row e.g.  
after `\hline` . . . . . 307
- 1992/09/19 ltfsstrc.dtx v2.0c  
`\try@simple@size`: . . . . . 172
- 1992/09/21 ltssini.dtx v1.4d  
`\not@math@alphabet`: Macro de-  
fined. . . . . 209
- 1992/09/22 ltssbas.dtx v1.91a  
General: Introduced `\tf@size` for  
math size. . . . . 139
- 1992/09/22 ltfsstrc.dtx v2.1a  
`\getanddefine@fonts`: Introduced  
`\tf@size` for math size. . . . . 170
- 1992/11/13 ltssini.dtx v?  
`\hexnumber@`: Made expandable. . 210
- 1992/11/23 ltcounts.dtx LaTeX2.09  
`\stepcounter`: Replaced `{}` in  
`\stepcounter` by `\begingroup`  
`\endgroup` to avoid adding an  
empty ord in math mode . . . 133
- 1992/11/26 ltboxes.dtx LaTeX2.09  
`\@mpfootnotetext`: (RmS) added  
protection for `\edef` . . . . . 293
- 1992/11/26 ltfloat.dtx LaTeX2.09  
`\@footnotetext`: (RmS) added pro-  
tection for `\edef` . . . . . 371
- `\footnote`: (RmS) Changed all to  
`'def'protect'noexpand'protect'noexpand`  
. . . . . 370
- 1992/12/03 ltssini.dtx v?  
`\hexnumber@`: Make it accept coun-  
ters. . . . . 210
- 1993/03/08 preload.dtx v2.0b  
General: Added 12pt preloads . . 232
- 1993/03/18 ltssbas.dtx v2.0c  
General: Changed all `\@tempdima`  
in `\@tempdimb` to avoid killing  
`\numberline` . . . . . 139
- 1993/03/18 ltfsstrc.dtx v2.1b  
General: Changed all `\@tempdima`  
in `\@tempdimb` to avoid killing  
`\numberline` . . . . . 159
- Changed all `\@tempdimb` in  
`\@tempdimx` to avoid killing  
`\numberline` . . . . . 159
- 1993/03/18 ltfsstrc.dtx v2.1c  
`\DeclareSizeFunction`: Added all  
args to avoid blanks problems 175
- 1993/04/09 lterror.dtx v1.0e  
`\@latexerr`: Mention The Compan-  
ion . . . . . 60
- 1993/04/11 lterror.dtx v1.0f  
`\@latexerr`: Remove setting of er-  
rorcontextlines . . . . . 60
- 1993/05/05 ltfnctcmd.dtx v2.0b  
General: Removed all LaTeX re-  
lated cmds . . . . . 236
- 1993/05/16 ltssbas.dtx v2.0e  
`\showhyphens`: Use `\reset@font` 157
- 1993/07/16 ltfsstrc.dtx v2.1h  
General: Changed layout of info  
messages . . . . . 159
- 1993/07/17 ltoutenc.dtx 1.0d  
General: changed `\catcoding @` . 94

- 1993/08/03 ltmiscen.dtx LaTeX2.09  
`\enddocument`: Changed redefinition of `\global` to redefinition of `\setckpt`. . . . . 250
- 1993/08/05 ltpictur.dtx LaTeX2.09  
`\circle`: (RmS) Added error message if `\circle` is used in math mode. . . . . 335
- 1993/08/05 ltsect.dtx LaTeX2.09  
`\sect`: (RmS) Made sure that `\protect` works correctly in expansion of `\the` counter . . . 346
- 1993/08/05 ltspace.dtx LaTeX2e  
`\hspace`: (RmS) Removed superfluous `\leavevmode` in `\hspace` and `\hspace`, as suggested by CAR. . . . . 78
- 1993/08/05 lttab.dtx latex2e  
`\tabular*`: Replaced `\expandafter\def` by `\namedef`. 307
- 1993/08/06 ltbibl.dtx LaTeX2.09  
`\citex`: Moved writing to .aux file in loop over citation keys so that leading blanks are removed there as well. . . . . 376
- 1993/08/13 ltoutenc.dtx 1.0f  
General: Protected against active @ sign. . . . . 94
- 1993/08/13 preload.dtx v2.0c  
General: Added `\relax` at end of font names. . . . . 233
- 1993/08/16 ltoutenc.dtx 1.0g  
General: Needs space after `\string` 94
- 1993/08/18 ltfssdcl.dtx v2.0e  
`\new@mathversion`: Exchanged names of encodings in warning message of `\SetSymbolFont`. 193
- 1993/09/02 ltfssstrc.dtx v2.1i  
General: Corrected name of sgen size function. . . . . 159
- 1993/09/03 ltmiscen.dtx LaTeX2.09  
`\verbatim@nolig@list`: Replaced `\noligs` by extensible list . 256
- 1993/09/07 ltmiscen.dtx LaTeX2.09  
`\verb@balance@group`: (RmS) Changed definition of `\verb` so that it detects a missing second delimiter. . . . . 256
- 1993/09/08 ltmiscen.dtx LaTeX2.09  
`\enddocument`: Added warning in case of undefined references. 250
- 1993/09/15 ltfssbas.dtx v2.0g  
`\DeclareFontEncoding`: Corrected: `\default@T` to `\default@M`. . 142
- 1993/09/15 ltfssstrc.dtx v2.1j  
General: Corrected spelling of `\noexpand`. . . . . 159
- 1993/09/19 lterror.dtx LaTeX2.09  
`\@invalidchar`: (RmS) Error message for invalid input characters. . . . . 62
- 1993/11/02 ltmath.dtx LaTeX2.09  
General: RmS: Corrected description of `\@eqnset`, moved `\@eqnset` accordingly and removed extra `\tabskip` assignment. . . . . 264
- 1993/11/03 ltmath.dtx LaTeX2e  
General: RmS: Initialized `\everycr` to empty . . . . . 264
- 1993/11/03 ltpictur.dtx LaTeX2.09  
General: (RmS) changed `\halign` to `\ialign` to initialize `\tabskip` and `\everycr` . . . . 321
- 1993/11/11 ltfssini.dtx v2.1a  
`\normalfont`: Macro added . . . . 211
- 1993/11/11 ltfssstrc.dtx v2.2a  
General: Option concept added for LaTeX2e . . . . . 159
- 1993/11/14 ltclass.dtx v0.2a  
`\@current`: Name changed from `\@currentextension` . . . . . 455
- `\@fileswithoptions`: Moved resetting of `\default@ds`, `\ds@` and `\@declaredoptions` here, from the end of `\ProcessOptions`. 463
- `\@reset@options`: macro added . . 465
- `\AtEndDocument`: Included extension in the generated macro name for package and class hooks. . . . . 465
- `\documentstyle`: Added `\RequirePackage` `\@unusedoptionlist` stuff. . . . . 461
- `\g@addto@macro`: Made global . . 465
- `\NeedsTeXFormat`: made more robust for alternative syntax for other formats. . . . . 462
- `\ProcessOptions*`: Optimise ‘empty option’ code. . . . . 459
- Stop adding the global option list inside class files. . . . . 459
- 1993/11/15 ltclass.dtx v0.2b  
`\documentstyle`: Modified to match `\ProcessOption*` . . . 461
- `\ProcessOptions*`: Star form added. . . . . 459

1993/11/17 ltclass.dtx v0.2c	1993/11/22 ltlength.dtx LaTeX2e
\@fileswith@pti@ns: Macro	\@settodim: Macro added . . . . . 138
added . . . . . 464	\@settopoint: Macro added . . . . . 138
\@badrequireerror: Macro added 466	\settodepth: Macro added . . . . . 138
\@fileswithoptions: Added trap	\settoheight: Macro added . . . . . 138
for two \LoadClass commands. 464	1993/11/22 ltlogos.dtx LaTeX2e
\@twoloadclasserror: Macro	\LaTeXe: Macro added . . . . . 80
added . . . . . 466	1993/11/23 ltclass.dtx v0.2g
\CurrentOption: Name changed	\@use@ption: Name changed from
from \@curroption . . . . . 455	\@executeoption . . . . . 460
\DeclareOption*: Error checking	General: Various macros now
added . . . . . 458	moved to latex.tex. . . . . 455
\NeedsTeXFormat: Name changed	Warnings and errors now directly
from \NeedsFormat . . . . . 462	coded. . . . . 455
\ProcessOptions*: restoring	1993/11/23 ltdefs.dtx LaTeX2e
\@fileswith@pti@ns added. 459	\@argdef: Macro added . . . . . 37
1993/11/18 ltclass.dtx v0.2d	\ifundefined: Redefined to re-
\documentstyle: Modified \Re-	move a trailing \fi . . . . . 46
quirePackage stuff. . . . . 461	\@newcommand: Macro added . . . . . 37
\ExecuteOptions: Use \Curren-	\@newenv: Macro interface changed 40
tOption not \reserved@a .. 460	\@xargdef: Macro interface
\NeedsTeXFormat: \fmtname	changed . . . . . 37
\fmtversion not \@... . . . . 462	\@yargd@f: Avoid \@?@? token .. 38
1993/11/21 ltfiles.dtx LaTeX2e	Macro interface changed . . . . . 38
\@missingfileerror: Stop infinite	\newcommand: Macro reimple-
looping on \@er@ext . . . . . 88	mented and extended . . . . . 37
1993/11/21 ltmiscen.dtx v0.9a	\renewcommand: Macro reimple-
\@verbatim: use \verbatim@font	mented and extended . . . . . 39
instead of \tt . . . . . 255	\renewenvironment: Macro reim-
\verb: Use \verbatim@font instead	plemented and extended . . . . . 40
of \tt. . . . . 256	\two@digits: Macro added . . . . . 34
\verbatim@font: Macro added .. 256	1993/11/23 ltoutput.dtx v0.1a
1993/11/22 ltclass.dtx v0.2f	\paperheight: Register added .. 395
\@fileswithoptions: Made the de-	\paperwidth: Register added ... 395
fault [] not [\@unknownversion] 463	1993/11/23 ltoutput.dtx v0.1c
Made the initial version [] not	\@enlargepage: Command added 435
[\@unknownversion] . . . . . 463	\@kludgeins: Insert added . . . . . 435
\@ifclasslater: Added //00 so	\@makecol: Command changed .. 405
parsing never produces a run-	\@specialoutput: Command
away argument. . . . . 457	changed . . . . . 398
General: \@unknownversion re-	\enlargethispage*: Commands
moved . . . . . 468	added . . . . . 435
1993/11/22 ltdefs.dtx LaTeX2e	1993/11/24 ltfntcmd.dtx v2.1a
\@minus: Macro added . . . . . 35	\maybe@ic@: Use \t@st@ic . . . . . 241
\@plus: Macro added . . . . . 35	\t@st@ic: Macro added . . . . . 241
\CheckCommand: Macro added ... 41	1993/11/24 ltfssini.dtx v2.1a
\providecommand: Macro added . 41	General: Removed \xpt stuff ... 211
1993/11/22 lterror.dtx LaTeX2e	1993/11/24 ltlogos.dtx LaTeX2e
\@errorcontextlines: Macro	\LaTeX: Macro changed . . . . . 80
added . . . . . 59	1993/11/28 ltclass.dtx v0.2h
1993/11/22 ltfiles.dtx LaTeX2e	\@twoclasseserror: Macro added 466
\listfiles: Removed checking for	General: Assorted commands now
\@unknownversion . . . . . 89	in the kernel removed. . . . . 455

Directory syntax checing moved to dircheck.dtx . . . . .	455	<code>\raisebox</code> : redefined to support <code>\height</code> . . . . .	294
Primitive filenames now terminated by space not <code>\relax</code> . . . . .	455	<code>\sbox</code> : color support . . . . .	288
<code>\endfilecontents</code> : Don't globally allocate a write stream (always use 15) . . . . .	466	extra group . . . . .	288
1993/11/28 ltfiles.dtx LaTeX2e		<code>\setcolor</code> : color support . . . . .	287
<code>\@missingfileerror</code> : Use filename parser from dircheck . . . . .	88	macro added . . . . .	287
1993/11/29 ltoutput.dtx v1.0b		1993/12/03 ltclass.dtx v0.2i	
<code>\@makecol</code> : <code>\@makespecialcolbox</code> added . . . . .	405	<code>\@cls@pkg</code> : Name changed to avoid clash with output routine. . . . .	465
<code>\@makespecialcolbox</code> : Command added . . . . .	406	General: <code>\@onlypreamble</code> : Many commands declared. . . . .	455
1993/11/29 ltplain.dtx LaTeX2e		Removed obsolete <code>\@documentclass</code> . . . . .	455
General: All accents in decimals; suggested by Paul Taylor . . . . .	28	1993/12/03 lterror.dtx v1.0b	
1993/11/30 ltoutput.dtx v1.0c		<code>\@latexerr</code> : Set <code>\c@errorcontextlines</code> to -1 . . . . .	60
<code>\fl@tracemessage</code> : Commands added . . . . .	437	1993/12/03 ltfssini.dtx v2.1a	
1993/12/01 fontdef.dtx v2.1a		General: update for LaTeX2e . . . . .	208
General: Update for LaTeX2e . . . . .	214	1993/12/04 ltfiles.dtx v0.9b	
1993/12/01 ltoutput.dtx v1.0e		<code>\@iinput</code> : Macro reimplemented . . . . .	88
<code>\@reinserts</code> : Command added . . . . .	406	<code>\@input</code> : Macro reimplemented . . . . .	88
1993/12/03 ltboxes.dtx v0.1a		<code>\IfFileExists</code> : Macro added . . . . .	87
<code>\@argsbox</code> : macro removed . . . . .	294	<code>\input</code> : Macro reimplemented . . . . .	88
<code>\@begin@tempboxa</code> : macro added . . . . .	286	<code>\InputIfFileExists</code> : Macro added . . . . .	88
<code>\@end@tempboxa</code> : macro added . . . . .	286	1993/12/05 ltfloat.dtx LaTeX2e	
<code>\@iirsbox</code> : redefined to support <code>\height</code> . . . . .	295	<code>\@dblfloatplacement</code> : Command changed . . . . .	363
<code>\@imakebox</code> : macro modified . . . . .	286	<code>\@xfloat</code> : Command changed . . . . .	357
<code>\@irsbox</code> : redefined to support <code>\height</code> . . . . .	294	1993/12/05 ltoutput.dtx v1.0f	
<code>\@isavebox</code> : color support . . . . .	288	<code>\@addtobot</code> : Command changed . . . . .	417
extra group . . . . .	288	<code>\@addtocurcol</code> : Command changed . . . . .	418
<code>\@isavepicbox</code> : extra group . . . . .	288	<code>\@addtodblcol</code> : Command changed . . . . .	429
<code>\@makebox</code> : default changed from x to c . . . . .	286	<code>\@addtonextcol</code> : Command changed . . . . .	425
<code>\@makepicbox</code> : macro modified . . . . .	287	<code>\@addtotoporbot</code> : Command changed . . . . .	417
<code>\@savebox</code> : default c not x . . . . .	288	<code>\@boxfpsbit</code> : Command added . . . . .	440
<code>\bm@b</code> : macros added . . . . .	286	<code>\@flcheckspace</code> : Command added . . . . .	442
<code>\endlrbox</code> : macro added . . . . .	288	<code>\@flsetnum</code> : Command added . . . . .	441
<code>\fbox</code> : extra group . . . . .	289	<code>\@flsettextmin</code> : Command added . . . . .	442
<code>\lrbox</code> : color support . . . . .	288	<code>\@flstop</code> : Commands added . . . . .	438
macro added . . . . .	288	<code>\@flupdates</code> : Command added . . . . .	443
<code>\makebox</code> : modified . . . . .	285	<code>\@fpsadddefault</code> : Command added . . . . .	439
<code>\mbox</code> : extra group . . . . .	286	<code>\@getfpsbit</code> : Command added . . . . .	440
<code>\minipage</code> : Redefined to support extra optional arguments . . . . .	292	<code>\@opcol</code> : Command changed . . . . .	404
<code>\newsavebox</code> : Pass the whole of arg 1 to <code>\@ifdefinable</code> . . . . .	287	Hook added . . . . .	404
<code>\parbox</code> : Redefined to support extra optional arguments . . . . .	290	<code>\@outputpage</code> : Command changed . . . . .	408
		<code>\@resetfps</code> : Command added . . . . .	441
		<code>\@setfloattypecounts</code> : Command added . . . . .	439

- `\@setfpsbit`: Command added . 440
- `\@shipoutsetup`: Command added 408
- `\@startcolumn`: Command changed ..... 412
- `\@startdblcolumn`: Command changed ..... 412
- `\@testfp`: Command added .... 440
- `\@textfloatsheight`: Commands added ..... 439
- `\@topnewpage`: Commands changed ..... 397
- `\@tryfcolumn`: Command changed 413
- `\@writesetup`: `\@startpagehook` added ..... 408
- `\output`: Command changed ... 398
- 1993/12/06 ltclass.dtx v0.2k
  - `\ExecuteOptions`: Preserve `\CurrentOption`. .... 460
- 1993/12/06 ltoutput.dtx v1.0f
  - `\@specialoutput`: Unboxing of 255 added to rescue writes ..... 398
- 1993/12/06 ltoutput.dtx v1.0g
  - `\@topnewpage`: `\@floatplacement` placement bug fixed ..... 397
- 1993/12/07 ltclass.dtx v0.2l
  - `\ProvidesFile`: Macro added ... 458
- 1993/12/07 ltclass.dtx v0.2m
  - `\@fileswithoptions`: Reset `\CurrentOption` ..... 463
- 1993/12/07 ltoutenc.dtx 1.1
  - General: Protected all special characters with `\string`. .... 94
- 1993/12/07 ltoutenc.dtx v1.1
  - General: Made all character numbers decimal. .... 91
  - Removed a lot of equal signs and the like. .... 91
- 1993/12/08 ltboxes.dtx v0.1b
  - `\@begin@tempboxa`: Extra braces for color support (braces removed from other macros) .. 286
  - `\@irsbox`: fix typo ..... 294
  - `\@parboxto`: `\endgraf` added due to extra group in `\@begin@tempboxa` ..... 291
  - `\lrbox`: move `\endpefalse` out of the inner group ..... 288
- 1993/12/08 ltfnctcmd.dtx v2.1b
  - General: Macros `\rm`, `\bf` and `\sf` moved to classes.dtx ..... 243
- 1993/12/08 ltlists.dtx LaTeX2e
  - `\@item`: use `\sbox` to support colour ..... 281
- 1993/12/08 ltspc.dtx LaTeX2e
  - `\@bsphack`: Command reimplemented ..... 71
  - Command reimplemented; late birthday present for Chris ... 71
  - `\@vbsphack`: Command added .. 73
- 1993/12/09 ltboxes.dtx v0.1c
  - `\@irsbox`: fix another typo ..... 294
- 1993/12/09 ltclass.dtx v0.2n
  - `\documentstyle`: input 209 compatibility file. .... 461
- 1993/12/09 ltfiles.dtx v0.9e
  - `\document`: Hook added ..... 83
- 1993/12/09 ltmiscen.dtx v0.9e
  - `\enddocument`: Hook added .... 250
- 1993/12/10 ltoutenc.dtx v1.2
  - General: Added source code for `tlenc.sty`. .... 91
- 1993/12/11 ltfnctcmd.dtx v3.0a
  - General: Complete reworking of all text commands, using just one creator function ..... 236
  - italic correction now put in front of penalty before glue ..... 236
  - newcommands replaced by defs 236
  - newfontswitch command corrected and changed ..... 236
- `\DeclareTextFontCommand`: Macro changed ..... 238
- `\emph`: Macro changed ..... 239
- `\fix@penalty`: Macro added .... 241
- `\maybe@ic`: Macro name changed 240
- `\maybe@ic@`: Macro and name changed ..... 240
- `\sw@slant`: Macro changed ..... 241
- `\textup`: Macros changed ..... 239
- 1993/12/11 ltmath.dtx v0.9g
  - General: Added a group around the first argument of `\frac` to prevent changes (for example font changes) from modifying the contents of the second argument. .... 264
- 1993/12/11 ltoutenc.dtx v1.2a
  - General: Corrected for `tlenc`, `math`. .... 91
- 1993/12/11 ltsect.dtx LaTeX2e
  - `\@author`: Added default ..... 343
  - `\@title`: Added default ..... 343
- 1993/12/11 ltxref.dtx LaTeX2e
  - `\@setref`: Macro added ..... 246
  - `\pageref`: Macro reimplemented . 246
  - `\ref`: Macro reimplemented .... 246

- 1993/12/12 ltoutput.dtx v1.0h
- `\@cflb`: `boxmaxdepth` setting moved ..... 411
  - defs changed to `lets` ..... 411
  - `\@cflt`: name changed ..... 411
  - `\@docclearpage`: defs changed to `lets` ..... 403, 404
  - `\@makecol`: defs changed to `lets` . 405
  - `\@resetfyps`: Warnings added: minimal ..... 441
  - `\@startdblcolumn`: defs changed to `lets` ..... 413
  - `\@topnewpage`: braces removed .. 397
  - `\@tryfcolumn`: defs changed to `lets` 414
  - `\fl@tracemessage`: Commands changed ..... 437
- 1993/12/13 ltclass.dtx v0.2o
- General: Removed setting `\errorcontextlines` (now in `latex.tex`) ..... 455
  - `\documentstyle`: compatibility file now `latex209.sty`. .... 461
  - `\usepackage`: Fixed error handling 462
- 1993/12/13 ltdirchk.dtx v0.2a
- General: on the ‘docstrip’ pass, do not check `openin` path ..... 10
  - `\IfFileExists`: Removed interactive prompting for current directory syntax ..... 10
  - `\strip@prefix`: modified, name changed from `\stripmeaning`. . 5
- 1993/12/13 ltlists.dtx latex2e
- `\trivlist`: Initialised `\@itemlabel` ..... 278
- 1993/12/13 ltmiscen.dtx v0.9h
- `\@noligs`: Readdded `\@noligs` ... 257
  - `\@verbatim`: Readdded `\@noligs` . 255
  - Removed optional argument of `\item` ..... 255
  - `center`: Removed optional argument of `\item` ..... 254
  - `flushleft`: Removed optional argument of `\item` ..... 254
  - `flushright`: Removed optional argument of `\item` ..... 254
- 1993/12/13 ltoutenc.dtx v1.2b
- General: Corrected file name in driver code. .... 91
- 1993/12/13 lttab.dtx latex2e
- `\tabbing`: Removed optional argument of `\item` ..... 302
- 1993/12/14 ltoutput.dtx v1.0i
- General: Section added to declare all parameters ..... 448
- 1993/12/15 ltboxes.dtx v0.1d
- `\@iminipage`: Changed default from ‘c’ to ‘s’ ..... 292
  - `\@iparbox`: Changed default from ‘c’ to ‘s’ ..... 291
  - `\minipage`: Changed default from ‘c’ to ‘s’ ..... 292
  - extra space removed. .... 292
  - `\parbox`: Changed default from ‘c’ to ‘s’ ..... 290
- 1993/12/15 ltclass.dtx v0.2p
- General: Removed extra ‘.’s from `\@@warnings` ..... 455
- 1993/12/16 ltlogos.dtx LaTeX2e
- `\LaTeXe`: Extended logo by DPC 80
- 1993/12/16 ltmath.dtx v0.9i
- `\@eqnocr`: use `\refstepcounter` instead of shortcut ..... 265
  - General: use `\refstepcounter` instead of shortcut ..... 264
- 1993/12/16 ltmiscen.dtx v0.9i
- General: `\literal` added ..... 257
- 1993/12/16 ltpage.dtx LaTeX2e
- `\mark`: Init `\mark` at begin document ..... 381
- 1993/12/16 ltspac.dtx LaTeX2e
- `\@bsphack`: Corrected optimisation :- ) ..... 71
- 1993/12/16 lttab.dtx latex2e
- `\@xhline`: Measure from middle of vertical rules ..... 316
- 1993/12/17 ltclass.dtx v0.2q
- `\@documentclasshook`: Macro added ..... 455
  - `\@fileswithoptions`: Add `\@compatibility` hook ..... 463
  - `\documentstyle`: Match Alan’s new code. .... 461
- 1993/12/17 ltoutenc.dtx 1.3
- General: Added this section .... 94
  - Removed all the hackery for use in `\DeclareFontEncoding`, and redid everything using `\DeclareTextFoo`. .... 105, 107
  - Removed the catcode hackery, since the file is only read as a package in the preamble, and removed all the messages on the screen, which just confuse users. Replaced them by the appropriate `\ProvidesPackage` commands. Added `XXXenc`. .... 94

- 1993/12/17 ltoutenc.dtx v1.3  
 General: Added `\EncodingSpecificAccent`, `\EncodingSpecificAccentedLetter` and `\EncodingSpecificCommand`. . . . . 91  
 Made Rokicki's encoding a proper encoding scheme rather than a variant of OT1. . . . . 91
- 1993/12/17 ltoutput.dtx v1.0j  
`\@opcol`: Hook removed . . . . . 404  
`\@specialoutput`: Page room test added . . . . . 399  
`\@topnewpage`: check for vsize too small added . . . . . 397  
 Page room test added . . . . . 398  
`\@writesetup`: —and then removed . . . . . 408  
`\fl@tracemessage`: tracefloatvals made a document command 437
- 1993/12/17 ltpage.dtx LaTeX2e  
`\mark`: Removed init `\mark` at begin document, since it doesn't work. . . . . 381  
`\rightmark`: Stopgap solution to mark `\leftmark` and `\rightmark` work without initializing mark until the problem is solved. . . . . 380
- 1993/12/18 ltoutenc.dtx 1.3b  
 General: Fixed typos with `\ProvidesPackage` lines. Added the `\NeedsTeXFormat` line. Added the last argument to `\DeclareEncoding`. Moved the use of the encodings to after their declaration. . . . . 94  
 Replaced the missing last argument to `\DeclareFontEncoding`. . . . . 105, 107
- 1993/12/18 ltoutenc.dtx 1.3c  
 General: Rewrote for the new syntax of `\EncodingSpecific`. . . . . 105, 107  
 Split `\EncodingSpecificAccent` up into `\EncodingSpecific` and `\DeclareAccent`. . . . . 94
- 1993/12/18 ltoutenc.dtx v1.3a  
 General: Replaced OT3 by XXX 91
- 1993/12/18 ltoutenc.dtx v1.3b  
 General: Corrected typos. . . . . 91  
 Replaced the missing last argument to `\DeclareFontEncoding`. . . . . 91
- 1993/12/18 ltoutenc.dtx v1.3c  
 General: A new syntax, separating accent-definitions from encoding-specific definitions, and allowing encoding-specific `\chardef`, `\let`, etc. . . . . 91  
 Rewrote for the new syntax of `\EncodingSpecific`. . . . . 91
- 1993/12/18 ltoutenc.dtx v1.3d  
 General: Some T1 stuff had drifted into the OT1 file. . . . . 91
- 1993/12/18 ltpage.dtx LaTeX2e  
`\sloppy`: Added `\emergencysretch` . . . . . 381
- 1993/12/19 ltclass.dtx v0.2r  
`\endfilecontents`: Different message when ignoring a file . . . 466
- 1993/12/19 ltfntcmd.dtx v3.0b  
 General: `\@pdef` command added 236  
 Added by ASAJ. . . . . 243  
 Made `\@newfontswitch` produce an error if command already exists, and added `\@renewfontswitch`, ASAJ . . . . . 236  
 Other tidying . . . . . 236  
 Some more tidying done . . . . . 236  
 Untidying added, so this is now a TEMPORARY version. . . 236  
 Wording changes by CAR. . . 243  
`\DeclareOldFontCommand`: Corrected and tidied . . . . . 243  
`\DeclareTextFontCommand`: Corrected and tidied . . . . . 238
- 1993/12/19 ltspace.dtx LaTeX2e  
`\@bsphack`: There seem to be problems with selfmade birthday presents . . . . . 72
- 1993/12/20 ltdefs.dtx LaTeX2e  
`\@reargdef`: Kept old version of `\@reargdef`, for array.sty . . . 39
- 1993/12/20 ltfiles.dtx v0.9m  
`\@obsoletefile`: Added this command, removed `@oldfilewarning` . . . . . 89
- 1994/01/05 fontdef.dtx v2.1d  
 General: Removed nf prefix from file names. . . . . 216
- 1994/01/13 ltmath.dtx v0.9o  
`\@eqnocr`: correcting 0.9i . . . . . 265  
 General: correcting 0.9i . . . . . 264
- 1994/01/14 ltdirchk.dtx v0.2d  
`\IfFileExists`: Close the texsys.aux output stream . . . 10

- 1994/01/15 ltfiles.dtx v0.9o  
`\document`: move `\@preamblecmds`  
 after document hook ..... 84
- 1994/01/17 ltclass.dtx v0.2s  
`\@fileswithoptions`: Modify to re-  
 duce parameter stack usage .  
 ..... 463, 464  
 General: Added many more `\on-`  
`lypreamble` commands .... 455  
 Wrapped long lines to column 72 455
- 1994/01/17 ltfiles.dtx LaTeX2e  
`\listfiles`: New Version, adds  
 ‘.tex’ if needed, and lines up  
 columns ..... 89
- 1994/01/17 ltffsbas.dtx v2.1a  
 General: New math font setup .. 139  
`\curr@math@size`: New math font  
 setup ..... 148  
`\everydisplay`: New math font  
 setup ..... 148  
`\everymath`: New math font setup 148  
`\frozen@everydisplay`: New math  
 font setup ..... 148  
`\frozen@everymath`: New math  
 font setup ..... 148  
`\math@version`: New math font  
 setup ..... 147
- 1994/01/17 ltffsini.dtx v2.1e  
`\not@math@alphabet`: Message  
 changed ..... 209
- 1994/01/17 ltffstrc.dtx v2.3a  
 General: New math font setup .. 159  
`\check@mathfonts`: New math font  
 setup ..... 168  
`\glb@currsize`: New math font  
 setup ..... 165  
`\restglb@settings`: New math  
 font setup ..... 168
- 1994/01/18 ltbibl.dtx LaTeX2e  
`\bibliography`: Use `\@input@` so  
 include files are listed. .... 377
- 1994/01/18 ltclass.dtx v0.2t  
`\ifclassloaded`: Fix typo  
`\@pkgetension` ..... 456
- 1994/01/18 ltfiles.dtx v0.9p  
`\@iffilenameonpath`: Macro added .. 87  
`\@input`: do not use a different def-  
 inition for `\input@path` .... 88  
`\@input@`: Macro added ..... 88  
`\IfFileExists`: New Definition .. 87  
`\include`: Use `\@input@` so include  
 files are listed. .... 86  
`\InputIfFileExists`: New Defini-  
 tion ..... 88
- 1994/01/18 ltffsini.dtx v2.1f  
`\not@math@alphabet`: Message cor-  
 rected ..... 209
- 1994/01/18 ltmiscen.dtx v0.9p  
`\@verbatim`: Add `\global\@inlabelfalse`  
 ..... 255  
 Only add `\penalty` if in hmode 255
- 1994/01/19 fontdef.dtx v2.1e  
 General: Added missing setting for  
 symbols in bold version. ... 219
- 1994/01/19 ltdirchk.dtx v0.2e  
`\IfFileExists`: name changed  
 from `\test` ..... 9  
`\input@path`: No longer check that  
 an empty group is in the path 11  
`\strip@prefix`: name changed  
 from `\strip@meaning`, to match  
 NFSS. .... 5
- 1994/01/19 ltmath.dtx v1.0n classes  
`\mathindent`: Deferred setting of  
`\mathindent` ..... 267
- 1994/01/20 ltdirchk.dtx v0.2f  
 General: `\@copytexsys` and the  
 texsys.new file removed ..... 9  
 Modify all of ltxcheck ..... 13  
`\IfFileExists`: `\@copytexsys` re-  
 moved ..... 10
- 1994/01/21 ltclass.dtx v0.2u  
`\documentstyle`: compatibility file  
 now latex209.def. .... 461
- 1994/01/21 ltdirchk.dtx v0.2g  
 General: Improve documentation,  
 reorganise docstrip module ... 1  
`\filename@parse`: Minor changes,  
 and add Mac version (:) .... 11  
`\today`: Name changed from  
`\stamp`, to save memory ..... 9
- 1994/01/21 ltfloat.dtx LaTeX2e  
`\@xfloat`: Added missing percent  
 characters. .... 357
- 1994/01/21 ltmiscen.dtx v0.9s  
`\verbatim@font`: Removed unnec-  
 essary category code hackery. 256
- 1994/01/24 ltdirchk.dtx v0.2h  
`\IfFileExists`: Stop testing once  
 texsys.aux has been found .... 9
- 1994/01/24 ltpage.dtx LaTeX2e  
`\pagestyle`: (DPC) Complain if  
 pagestyle is undefined. .... 379
- 1994/01/25 ltdirchk.dtx v0.2i  
 General: Protect against looping on  
`\@@input` and `\@@end`. .... 2

- 1994/01/25 ltfsbas.dtx v2.1b  
 $\mathsf{math@version}$ : Corrections for  
math setup ..... 148
- 1994/01/25 ltmath.dtx LaTeX2e  
 $\mathsf{bordermatrix}$ : Removed  
 $\mathsf{p@renwd}$ : ..... 261
- 1994/01/26 ltfsstrc.dtx v2.3c  
 $\mathsf{check@mathfonts}$ : Correct trace  
info placement ..... 168  
 $\mathsf{restglb@settings}$ : Correct trace  
info placement ..... 168
- 1994/01/27 ltfntcmd.dtx v3.1a  
 $\mathsf{nocorrlst}$ : Only ., used as de-  
fault for cm fonts ..... 242
- 1994/01/29 ltclass.dtx v0.2v  
 $\mathsf{unprocessedoptions}$ : Macro  
added. .... 466  
 $\mathsf{fileswithoptions}$ : All options  
raise error if no  $\mathsf{ProcessOp-}$   
 $\mathsf{tions}$  appears ..... 464
- 1994/01/31 ltclass.dtx v0.2w  
 $\mathsf{g@addto@macro}$ : Use toks register  
to avoid ‘hash’ problems ... 465
- 1994/01/31 ltfiles.dtx v0.9t  
 $\mathsf{document}$ : set  $\mathsf{normalsize}$  or  
 $\mathsf{normalsize}$  if necessary .... 84
- 1994/01/31 ltfntcmd.dtx v3.1b  
General:  $\mathsf{normalsize}$  no longer  
defined ..... 236
- 1994/02/01 ltpage.dtx LaTeX2e  
 $\mathsf{pagestyle}$ : (DPC) Modify to get  
nicer error message ..... 379  
 $\mathsf{thispagestyle}$ : (DPC) Modify to  
get nicer error message .... 380
- 1994/02/02 ltclass.dtx v0.2x  
 $\mathsf{fileswithoptions}$ : Only run the  
hook and options check if the  
file was loaded. .... 464
- 1994/02/03 ltoutput.dtx v1.0k  
 $\mathsf{makespecialcolbox}$ : correct mis-  
takes in the documentation . 407
- 1994/02/07 ltclass.dtx v0.2y  
 $\mathsf{fileswithoptions}$ : Run  $\mathsf{com-}$   
 $\mathsf{patibility}$  on the first class to  
start (not the first to finish) 463  
 $\mathsf{ifclasswith}$ : Add extra ,s so  
‘two’ is not matched with  
‘twocolumn’ ..... 457  
 $\mathsf{ProcessOptions*}$ : Add extra ,s  
so ‘two’ is not matched with  
‘twocolumn’ ..... 459, 460
- 1994/02/07 ltfsbas.dtx v2.1c  
 $\mathsf{DeclareFontEncoding}$ : revert cat-  
code settings earlier ..... 141
- $\mathsf{DeclareFontShape}$ : revert catcode  
settings earlier ..... 140
- 1994/02/08 ltoutput.dtx v1.0k  
 $\mathsf{makespecialcolbox}$ : box-  
maxdepth setting added ... 407  
boxmaxdepth setting removed 406  
General: Documentation and tasks  
tidied. .... 382
- 1994/02/10 ltclass.dtx v0.2z  
 $\mathsf{documentclasshook}$ : Changed  
the name from  $\mathsf{compatibil-}$   
 $\mathsf{ity}$  to  $\mathsf{documentclasshook}$ ,  
and added the check for  
whether  $\mathsf{normalsize}$  has been  
defined. ASAJ. .... 455
- $\mathsf{fileswithoptions}$ : Renamed  
 $\mathsf{compatibility}$  to  $\mathsf{docu-}$   
 $\mathsf{mentclasshook}$ . ASAJ. .... 463
- 1994/02/10 ltfsbas.dtx v2.1d  
 $\mathsf{addto@hook}$ : Made  $\mathsf{addto@hook}$   
long. .... 157
- 1994/02/10 ltfsncmp.dtx v2.1d  
 $\mathsf{scan@fontshape}$ : scan away stuff  
after pt ..... 182
- 1994/02/22 ltssini.dtx v2.1g  
General: Correct error message .. 211
- 1994/02/24 ltfsbas.dtx v2.1e  
 $\mathsf{DeclareFontShape}$ : Separate  
restoration of catcodes for fd  
cmds ..... 140
- $\mathsf{define@newfont}$ : Separate  
restoration of catcodes for fd  
cmds ..... 150
- $\mathsf{nfss@catcodes}$ : Separate restora-  
tion of catcodes for fd cmds . 150
- 1994/02/25 ltdirchk.dtx v0.2j  
General: Remove need for drv file . 1
- 1994/03/01 ltdirchk.dtx v0.2k  
General: Add unstripped module,  
so that dircheck.dtx may be  
used with initex ..... 1
- 1994/03/02 ltboxes.dtx v0.1e  
General: Add 2kernel module .. 285  
Remove need for drv file .... 285
- 1994/03/02 ltclass.dtx v0.3a  
General: Remove need for driver  
file ..... 455
- 1994/03/03 ltboxes.dtx v0.1f  
 $\mathsf{irsbox}$ : Replaced a missing  
 $\mathsf{else}$  ..... 294
- 1994/03/04 ltfloat.dtx v1.0a  
General: Initial version, split from  
latex.dtx ..... 353

1994/03/04 ltsect.dtx v1.0a	1994/03/07 ltpictur.dtx v0.1a
General: Initial version, split from latex.dtx . . . . . 343	General: Initial version, split from latex.dtx . . . . . 318
1994/03/04 lttab.dtx v1.0a	Long lines wrapped to 72 columns . . . . . 318
General: Initial version, split from latex.dtx . . . . . 296	1994/03/07 ltsect.dtx v1.0a
1994/03/04 ltvers.dtx v1.0a	$\backslash$ @hangfrom: (DPC)Extra groups for colour . . . . . 349
General: Initial version, split from latex.dtx . . . . . 32	1994/03/07 lttab.dtx v1.0a
1994/03/07 ltboxes.dtx v0.1a	General: Long lines wrapped to 72 columns . . . . . 296
$\backslash$ @mpfootnotetext: Extra group for colour . . . . . 293	1994/03/08 ltclass.dtx v0.3b
1994/03/07 ltboxes.dtx v1.0a	General: Modify driver code into ‘new style’ . . . . . 455
General: Unify format with other Kernel files . . . . . 285	1994/03/08 ltdirchk.dtx v1.0a
1994/03/07 ltdefns.dtx v1.0a	General: Reorganise driver module into ‘new style’ . . . . . 1
$\backslash$ @@italiccorr: Macro added . . . 35	1994/03/08 ltplain.dtx v1.0a
1994/03/07 ltfiles.dtx v1.0a	General: Remove need for a driver file. . . . . 14
General: Initial version, split from latex.dtx . . . . . 81	1994/03/10 ltfssbas.dtx v2.2f
Long lines wrapped to 72 columns . . . . . 81	$\backslash$ math@egroup: Changed $\backslash$ beginningroup/ $\backslash$ endgroup to $\backslash$ bgroup/ $\backslash$ egroup. . . . . 156
1994/03/07 ltfinal.dtx v0.1a	1994/03/11 ltfssdcl.dtx v2.1b
General: Add code from the old dump.dtx . . . . . 501	$\backslash$ DeclareSymbolFontAlphabet@: Added check against use of alphabet switch outside of math mode. . . . . 207
Initial version, split from latex.dtx . . . . . 494	$\backslash$ SetMathAlphabet@: Changed parameter template in temporary macro to catch check add below. . . . . 198
move code here from lhyphen.dtx . . . . . 497	1994/03/12 ltclass.dtx v0.3c
Remove oldcomments environment . . . . . 494	$\backslash$ @fileswithoptions: Do not use $\backslash$ @pr@videpackage to avoid typeout . . . . . 464
use $\backslash$ InputIfFileExists not $\backslash$ IfFileExists . . . . . 497	General: Change name from docclass to ltclass . . . . . 455
1994/03/07 ltfloat.dtx v1.0a	$\backslash$ ProvidesFile: Add $\backslash$ wlog . . . 458
$\backslash$ @endfloatbox: (DPC) Extra group for colour . . . . . 362	$\backslash$ ProvidesPackage: Add $\backslash$ wlog .. 457
$\backslash$ @footnotetext: (DPC) Extra group for colour . . . . . 371	use $\backslash$ @gtempa . . . . . 457
$\backslash$ @xfloat: (DPC) Extra group for colour . . . . . 358	1994/03/12 ltdefns.dtx v1.0b
1994/03/07 lthyphen.dtx v0.1c	$\backslash$ @reargdef: New defn, in terms of $\backslash$ @yargdef . . . . . 39
General: move the 2kernel code to ltfinal.dtx . . . . . 470	$\backslash$ @yargd@f: Name changed from $\backslash$ XXX@argdef . . . . . 38
1994/03/07 ltlength.dtx v1.0a	1994/03/12 ltdirchk.dtx v1.0b
$\backslash$ @settodim: (DPC) Extra group for colour . . . . . 138	General: Change name from dircheck.dtx . . . . . 1
1994/03/07 ltlists.dtx v1.0a	Minor edits to the typeouts in ltcheck . . . . . 1
General: Initial version, split from latex.dtx . . . . . 270	
Long lines wrapped to 72 columns . . . . . 270	
1994/03/07 ltpage.dtx v1.0a	
General: Initial version, split from ltherest.dtx . . . . . 379	

- 1994/03/12 ltfloat.dtx v1.0b  
   \@savemarbox: (DPC) Extra group for colour . . . . . 365  
   \@xympar: (DPC) Extra bgroup for colour . . . . . 366
- 1994/03/12 ltplain.dtx v1.0b  
   General: Name changed from lplain.  
   The end of an era . . . . . 14
- 1994/03/12 ltplain.dtx v1.0e  
   General: Replaced remaining width, height, depth by L<sup>A</sup>T<sub>E</sub>X macro names to save tokens. . 14
- 1994/03/13 ltcntrl.dtx v1.0c  
   \@tfor: (DPC) Add \@tfor so a single group is correctly treated. . . . . 54
- 1994/03/13 ltfiles.dtx LaTeX2e  
   \@addtofilelist: Macro added . 89  
   \listfiles: Reset \@addtofilelist at begin document . . . . . 89
- 1994/03/13 ltfiles.dtx v0.3b  
   \InputIfFileExists: Use new cmd  
   \@addtofilelist . . . . . 88
- 1994/03/13 ltssbas.dtx v2.1g  
   General: add 2kernel module to omit repeated code . . . . . 139
- 1994/03/13 ltssdcl.dtx v2.1c  
   General: add 2kernel module to omit repeated code . . . . . 185
- 1994/03/14 ltboxes.dtx v1.0b  
   \@isavebox: Use \@color@setgroup 288  
   \@isavepicbox: Use \@color@setgroup . . . . . 288  
   \color@begingroup: macro added for colour support . . . . . 287  
   \color@endgroup: macro added for colour support . . . . . 287  
   \lrbox: Use \@color@setgroup . . 288  
   \sbox: Use \@color@setgroup . . . 288
- 1994/03/14 ltfloat.dtx 1.0c  
   \@xympar: (DPC) Use \@color@begingroup . . . . . 366
- 1994/03/14 ltfloat.dtx v1.0c  
   \@endfloatbox: (DPC) Use \@color@endgroup . . . . . 362  
   \@footnotetext: (DPC) Use \@color@begingroup, add \endgraf . . . . . 371  
   \@savemarbox: (DPC) Use \@color@begingroup . . . . . 365  
   \@xfloat: (DPC) Use \@color@begingroup . . . . . 358
- 1994/03/15 ltfiles.dtx LaTeX2e  
   \@missingfileerror: Quit on x or X just like a real error . . . . . 88
- 1994/03/15 ltfmtcmd.dtx v3.2a  
   General: Adapted to mass formatting . . . . . 236  
   Changed \ / to \@italiccorr 236  
   Removed \@renewfontswitch . 236  
   Removed defs of short-forms and all sizes except \normalize . 236
- 1994/03/15 ltoutput.dtx v1.0l  
   \@addtocurcol: Changed \addvspace to \vskip . . . . . 420, 424  
   \@combinedblfloats: Removed boxmaxdepth setting. . . . . 412  
   \@makecol: \maxdepth changed to \@maxdepth . . . . . 405  
   Removed boxmaxdepth setting. 406  
   \@makespecialcolbox: Removed boxmaxdepth setting. . . . . 407  
   \@topnewpage: Corrected and amended warning message . . 397  
   Warning added: it should be improved . . . . . 398  
   General: Added some warnings when page gets full of top floats. . . . . 382  
   Driver added and further tidying. . . . . 382  
   Removed duplicated code and corrected docstrip options. . . 382  
   Some boxmaxdepth settings removed. . . . . 382
- 1994/03/16 ltclass.dtx v0.3f  
   General: Add pkgindoc package . 468
- 1994/03/16 ltfiles.dtx LaTeX2e  
   \listfiles: Move this code directly into \document . . . . . 89
- 1994/03/16 ltfiles.dtx v1.0c  
   \document: (DPC) directly add file list settings . . . . . 84
- 1994/03/16 ltmiscen.dtx v1.0b  
   \@verbatim: Remove \global\@inlabelfalse again. . . . . 255
- 1994/03/28 ltalloc.dtx v1.0d  
   General: Redefinition of ‘new’ allocations removed. . . . . 49
- 1994/03/28 ltdirchk.dtx v1.0d  
   General: Improve documentation . 1
- 1994/03/28 lterror.dtx v1.0d  
   \@invalidchar: (DPC) Comment out (use catcode15 instead) . . 62  
   General: Remove test for \inputlineno undefined. . . . . 59

- 1994/03/28 ltfiles.dtx v1.0d  
   \document: (DPC) Use \normal-size not \@normalsize ..... 84  
   (DPC) remove \@normalsize check ..... 84
- 1994/03/28 ltfloat.dtx v1.0b  
   \@caption: Use \normalsize not \@normalsize ..... 356  
   General: Split further from ltherest.dtx ..... 353
- 1994/03/28 ltlists.dtx v1.0b  
   General: Improve documentation 269
- 1994/03/28 ltmiscen.dtx v1.0c  
   General: Improve Documentation 249
- 1994/03/28 ltplain.dtx v1.0c  
   \newlanguage: Remove some \outer declarations. .... 16
- 1994/03/28 ltsect.dtx v1.0b  
   General: Split further from ltherest.dtx ..... 343
- 1994/03/28 lttab.dtx v1.0b  
   General: Improve documentation 296
- 1994/03/28 ltthm.dtx v1.0a  
   General: Initial version, split from latex.dtx ..... 339
- 1994/03/29 ltcounts.dtx v1.0c  
   General: Create file from parts of ltmiscen and ltherest. .... 132
- 1994/03/29 ltlength.dtx v1.0c  
   General: Create file ltcntlen from parts of ltmiscen and ltherest. 138
- 1994/03/29 ltmiscen.dtx v1.0d  
   General: Remove counter macros to ltcntlen ..... 249
- 1994/03/29 ltpageno.dtx v1.0c  
   General: Create file ltcntlen from parts of ltmiscen and ltherest. 244
- 1994/03/29 ltxref.dtx v1.0c  
   General: Create file ltcntlen from parts of ltmiscen and ltherest. 245
- 1994/03/31 ltbibl.dtx v1.0a  
   General: Initial version of ltidxbib.dtx, split from ltherest.dtx ..... 375
- 1994/03/31 ltidxglo.dtx v1.0a  
   General: Initial version of ltidxbib.dtx, split from ltherest.dtx ..... 373
- 1994/04/09 ltcounts.dtx v1.0d  
   \@newctr: \@nocnterr now has counter name argument .... 133  
   \addtocounter: \@nocnterr now has counter name argument . 133
- \setcounter: \@nocnterr now has counter name argument .... 133
- \stepcounter: Use \addtocounter to have name checked ..... 133
- 1994/04/09 ltthm.dtx v1.0b  
   \@othm: Use standard counter error message (FMi) ..... 341
- 1994/04/11 ltclass.dtx v0.3g  
   \endfilecontents: Add star form, dont write \endinput at the end of the file. .... 466
- \ProvidesFile: Protect against weird catcodes. .... 458
- 1994/04/11 ltfssbas.dtx v2.1h  
   General: Added \defaultscriptratio and \defaultscriptscriptratio. ASAJ. . 139
- \defaultscriptratio: Macro added ..... 157
- \defaultscriptscriptratio: Macro added ..... 157
- 1994/04/12 ltboxes.dtx v1.0c  
   General: Remove \@acci, now defined in ltplain.dtx ..... 291
- Remove \@dischyph, now defined in ltinit.dtx ..... 291
- 1994/04/12 ltdefs.dtx v1.0g  
   \@dischyph: Define \@dischyph, was previously in ltboxes.dtx . 35
- 1994/04/12 ltplain.dtx v1.0d  
   General: Define \@acci ..... 28
- 1994/04/12 ltvers.dtx v1.0b  
   General: Have version info generated automatically. .... 32
- 1994/04/14 ltfntcmd.dtx v3.2b  
   General: Macros renamed to non-private forms, JB ..... 236
- \DeclareOldFontCommand: Renamed from \@newfontswitch 242
- 1994/04/15 ltboxes.dtx v1.0d  
   \@isavebox: Added missing percent character. .... 288
- 1994/04/17 ltcounts.dtx v1.0e  
   \@newctr: Use \@nocounterr instead of \@nocnterr ..... 133
- \addtocounter: Use \@nocounterr instead of \@nocnterr ..... 133
- \setcounter: Use \@nocounterr instead of \@nocnterr ..... 133
- 1994/04/17 lterror.dtx v1.0h  
   \@nocounterr: New name for error message, old error message (without arg) kept ..... 60

- 1994/04/17 ltthm.dtx v1.0c  
`\@othm`: Use new std counter error message (FMi) ..... 341
- 1994/04/18 ltfinal.dtx v0.1b  
 General: Initialise `\textheight`, `\textwidth` and page style . 495
- 1994/04/18 ltfloat.dtx v1.0d  
`\@footnotetext`: (DPC) Remove Colour support ..... 371  
`\@savemarbox`: (DPC) Remove Colour support ..... 365
- 1994/04/18 ltssbas.dtx v2.1i  
 General: Macro `\no@alphabet@help` removed again ..... 139  
`\calculate@math@sizes`: Changed message to log only ..... 157  
`\no@alphabet@error`: Use std LaTeX error macro ..... 139
- 1994/04/18 ltssdcl.dtx ???  
`\DeclareMathAlphabet`: Pass correct arg (2 not 3) ..... 196
- 1994/04/18 ltssdcl.dtx v2.1d  
 General: Removed surplus `\no@alphabet@error` (see fam.dtx) ..... 185
- 1994/04/18 ltssstrc.dtx v2.3d  
 General: Changed to new error/warning scheme ..... 159  
`\font@submax`: Changed dimen to macro ..... 176  
`\fontsubfuzz`: Changed dimen to macro ..... 176  
`\subst@size`: `\font@submax` and `\fontsubfuzz` now macros .. 177
- 1994/04/19 ltpage.dtx v1.0b  
 General: Improve documentation 379
- 1994/04/20 ltfntcmd.dtx v3.3a  
 General: Documentation up-dated 236  
 New implementation of `\nocorr` 236  
`\check@nocorr@`: Macros added . 239  
`\maybe@ic@`: `\nocorr` etc removed from list of tokens to check, leaving only punctuation characters ..... 241
- 1994/04/20 ltmiscen.dtx v1.0e  
`\enddocument`: Changed logic for producing warning messages 251
- 1994/04/21 ltboxes.dtx v1.0e  
`\@iiiminipage`: Extra `\bgroup` for colour ..... 292  
`\@mpfootnotetext`: Extra `\endgraf` for colour ..... 293  
`\endminipage`: Extra `\egroup` for colour ..... 293
- 1994/04/21 ltfinal.dtx v0.1c  
 General: Added comments, set the catcodes of 128–255. .... 494
- 1994/04/22 ltssini.dtx v2.1g  
`\not@math@alphabet`: Message changed again ..... 209
- 1994/04/23 ltfinal.dtx v0.1d  
 General: Check that `\font@submax` is still zero ..... 494
- 1994/04/24 ltoutput.dtx v1.0m  
`\@resetfyps`: Number 2 changed to `\tw@` ..... 441  
 Warning changed ..... 441  
`\@specialoutput`: Message changed to give more info and ‘top’ removed ..... 399  
`\@topnewpage`: Message changed to give more info ..... 398  
 Warning message removed as it will be generated later ..... 397  
 General: Changed `\@normalsize` to `\normalsize`. .... 382  
 Corrected unverbbed commands in documentation. .... 382  
 Removed some long lines and other aesthetic changes. .... 382  
 Warning messages changed/corrected. .... 382
- 1994/04/24 ltpictur.dtx v0.1b  
 General: Removed surplus spaces after `\hbox to` in several cases ..... 318
- 1994/04/25 ltclass.dtx v0.3h  
 General: Removed spurious extra ‘s’ at the end of error messages 455
- 1994/04/25 ltfloat.dtx v1.0e  
`\@largefloatcheck`: Changed warning message to give more info ..... 362  
 Command added ..... 362  
 General: Changed warning messages ..... 353  
 Removed obsolete tracing code 353
- 1994/04/27 ltssstrc.dtx v2.3e  
 General: Corrected item that was forgotten in last change. ... 159
- 1994/04/28 lterror.dtx v1.0j  
`\@inmatherr`: Macro added .... 62
- 1994/04/28 lterror.dtx v1.1c  
`\@inmatherr`: Replaced `\noexpand` with `\protect`. .... 62
- 1994/04/28 ltssdcl.dtx v2.1e  
 General: Removed all `\uppercase` in hex num parsing macros . 185

- 1994/04/28 ltlists.dtx v1.0c  
 General: Replaced `\@ltxnomath` by  
`\@inmatherr` ..... 279
- 1994/04/28 ltpictur.dtx v0.1c  
 General: bezier curves added ... 336  
`\multiput`: (DPC) Ignore spaces  
 between `)`( ..... 320  
 (DPC) Macro added ..... 320  
`\picture`: (DPC) Ignore spaces be-  
 fore `(` ..... 319
- 1994/04/28 ltplain.dtx v1.0g  
 General: Turn off overfull box trac-  
 ing in log ..... 24
- 1994/04/29 ltclass.dtx v1.0a  
 General: Change version number to  
 1 (no other change) ..... 455
- 1994/04/29 ltmiscen.dtx v1.0f  
`\@verbatim`: `\leavevmode` added 255  
 Change to `\everypar` added .. 255
- 1994/04/29 ltoutenc.dtx 1.4a  
 General: Removed `\EncodingSpe-`  
`cific`. Renamed all the com-  
 mands. Added `\DeclareTextG-`  
`lyph` and `\UndeclareTextCom-`  
`mand`. ..... 94  
 Removed Rokicki's OT1 variant  
 encoding. Moved the driver to  
 the top. .... 94
- 1994/04/30 ltfntcmd.dtx v3.3b  
 General: Documentation up-dated  
 and tidied ..... 236  
 Prefix `frag@` changed to `frag` in  
`\@protecteddef` ..... 236  
 Title changed ..... 236  
 Warning changed to info message  
 in `\@protecteddef` ..... 236
- 1994/04/30 ltoutput.dtx v1.0n  
`\@activechar@info`: `\@ac-`  
`tivechar@warning` changed to  
`\@activechar@info` ..... 408  
`\@combinedblfloats`: Removed  
 rule in `topnewpage` case .... 412  
`\@emptycol`: Empty column action  
 added: `\@emptycol` ..... 397  
`\@flsetnum`: Rogue space removed 441  
`\@specialoutput`: Cut-off point  
 changed to `2\baselineskip` . 399  
 Empty column action added:  
`\@emptycol` ..... 399  
 Extra empty column added for  
 twocolumn case ..... 399  
 Extra empty column added for  
 twocolumn case (wrong, see be-  
 low) ..... 399
- `\@topnewpage`: Added setting of  
`\col@number` ..... 397  
 Cut-off point changed to  
`3\baselineskip` ..... 398  
 Empty column action added:  
`\@emptycol` ..... 398  
 Message changed for Frank ... 398  
 General: `\@activechar@warning`  
 changed to an info message. . 382  
 Added `\col@number`. .... 382  
 Documentation tidied. .... 382  
 Empty column action added. . 382  
 Fixed bug from `\dblfigrule`  
 with `\@topnewpage`. .... 382  
 Full of floats action improved. . 382  
`\col@number`: Added `\col@number` 395  
`\onecolumn`: Added setting of  
`\col@number` ..... 396
- 1994/05/01 lterror.dtx v1.0k  
`\@latexerr`: (CAR) Added draft  
`\@latexinfo`. .... 60
- 1994/05/01 ltoutenc.dtx 1.4a  
 General: Added the `\a` command. 102  
 Added the `\SaveAtCatcode`  
 and `\RestoreAtCatcode` com-  
 mands. .... 105  
 Removed the `uc/lc` table settings,  
 since the T1 `uc/lc` table is now  
 the default. .... 111  
 Rewrote for the new syntax. 105, 107
- 1994/05/01 ltoutenc.dtx v1.4a  
 General: Removed Rokicki's encod-  
 ing. .... 91  
 Renamed the commands, re-  
 moved the `\EncodingSpecific`  
 command. Turned all slots into  
 decimal. Added `\a`. .... 91
- 1994/05/02 ltcntrl.dtx v1.0l  
`\@break@tfor`: Macro added (from  
`ltfiles.dtx`) ..... 54
- 1994/05/02 ltfiles.dtx v1.0f  
`\@iffileonpath`: `\@break@loop` re-  
 named to `\@break@tfor` .... 87  
`\@obsoletefile`: Make `\@on-`  
`lypreamble` ..... 89
- 1994/05/02 ltfinal.dtx v0.1e  
 General: Added setting the 'letter'  
 catcodes. .... 499  
 Added setting the 'other' cat-  
 codes. .... 499  
 Added setting the special cat-  
 codes. .... 498  
 Made slot 127 illegal ..... 499  
 Set all the catcodes ..... 494

- 1994/05/02 ltfinal.dtx v0.1f  
 General: Set the catcode of control-J. . . . . 498
- 1994/05/02 ltmiscen.dtx v1.0g  
 General: Changed 91 to 1991 and moved some bits . . . . . 249
- 1994/05/02 ltoutput.dtx v1.0o  
 $\backslash\text{resetthfps}$ : Code shortened . . . 441  
 General: Code of  $\backslash\text{resetthfps}$  shortened. . . . . 382
- 1994/05/03 ltbibl.dtx v1.0b  
 $\backslash\text{nocite}$ : Make  $\backslash\text{nocite}$  issue a warning for an undefined citation key. . . . . 377
- 1994/05/03 ltfinal.dtx v0.1f  
 General: Set the catcode of control-J to be ‘other’, for use in messages. . . . . 494
- 1994/05/03 ltfloat.dtx v1.0f  
 General: (CAR) Added  $\backslash\text{largefloatcheck}$  . . . . . 353  
 Removed unnecessary braces from arguments of  $\backslash\text{ifnextchar}$  . . . . . 353  
 $\backslash\text{enddblfloat}$ :  $\backslash\text{largefloatcheck}$  added . . . . . 361  
 $\backslash\text{endfloat}$ : (CAR) Added  $\backslash\text{largefloatcheck}$  . . . . . 360
- 1994/05/03 ltfsdcl.dtx v2.1f  
 General: Renamed  $\backslash\text{DeclareMathDelimiter}$  to  $\backslash\text{DeclareMathDelimiter}$  . . . . . 185
- 1994/05/03 ltlists.dtx v1.0d  
 $\backslash\text{item}$ :  $\backslash\text{hskip}$  changed to  $\backslash\text{kern}$  280  
 General: Removed superfluous braces . . . . . 279
- 1994/05/03 ltmiscen.dtx v1.0h  
 $\backslash\text{centercr}$ :  $\backslash\text{badcrerr}$  replaced by  $\backslash\text{noInerr}$  . . . . . 254
- 1994/05/03 lttab.dtx v1.0d  
 $\backslash\text{endpbox}$ : Use  $\backslash\text{finalstrut}$  based on depth of  $\backslash\text{arstrutbox}$  . . . . . 317
- 1994/05/04 ltclass.dtx v1.0b  
 $\backslash\text{NeedsTeXFormat}$ : Changed wording of the warning . . . . . 462
- 1994/05/04 lterror.dtx v1.0m  
 $\backslash\text{badcrerr}$ : Error message removed . . . . . 62
- 1994/05/05 ltbibl.dtx v1.0c  
 $\backslash\text{citex}$ : Set switch for warning and end of run. . . . . 376  
 $\backslash\text{nocite}$ : Do not write page number in  $\backslash\text{nocite}$  warning message. 377
- Set switch for warning and end of run. . . . . 377
- 1994/05/05 ltfinal.dtx v0.1g  
 General: Added empty errhelp. . . 494  
 $\backslash\text{errhelp}$ : Set error help empty. . 502
- 1994/05/05 ltftcmd.dtx v3.3c  
 $\backslash\text{@@math@egroup}$ : Corrected  $\backslash\text{@fontswitch}$  and added saved versions . . . . . 243  
 General: Corrected  $\backslash\text{@fontswitch}$  236
- 1994/05/05 ltmiscen.dtx v1.0i  
 General: Removed braces from  $\text{ifnextchar}$  and  $\text{ifstar}$  arguments 249
- 1994/05/07 lttab.dtx v1.0c  
 $\backslash\text{@maxtab}$ : Changed  $\backslash\text{@firsttab}$  to  $\backslash\text{chardef}$  . . . . . 300  
 Changed  $\backslash\text{@maxtab}$  to  $\backslash\text{chardef}$  300  
 General: Removed definition of  $\backslash+$  296  
 Removed surplus braces from  $\backslash\text{ifnextchar}$  constructs . . . 296
- 1994/05/08 ltftcmd.dtx v3.3d  
 General: Removed  $\backslash\text{undefinedfonterror}$  . . . . . 236  
 $\backslash\text{normalsize}$ : Removed  $\backslash\text{undefinedfonterror}$  . . . . . 243
- 1994/05/09 ltftcmd.dtx v3.3f  
 General: Replaced all  $\backslash\text{next}$  by  $\backslash\text{let@token}$  and undo change 3.3e, whatever that was. . . . 236
- 1994/05/10 ltdefs.dtx v1.0n  
 General: (ASAJ) Added  $\backslash\text{DeclareProtectedCommand}$ . . . . . 34  
 Added  $\backslash\text{DeclareProtectedCommand}$  . . . . . 42  
 Added  $\backslash\text{makeatletter}$  and  $\backslash\text{makeatother}$  ASAJ. . . . . 47  
 Removed braces around  $\backslash\text{ifundefined}$  argument. ASAJ. . . . 39
- 1994/05/10 lterror.dtx v1.0n  
 $\backslash\text{@latexerr}$ : (ASAJ) Added extra blank lines to  $\backslash\text{@latexerr}$ . . . . 60
- 1994/05/10 ltmiscen.dtx v1.0j  
 $\backslash\text{@sverb}$ : Slight change in error message text. . . . . 256
- 1994/05/11 ltboxes.dtx v1.0f  
 $\backslash\text{@begin@tempboxa}$ : Use new  $\backslash\text{color@setgroup}$  concept. . . 286  
 $\backslash\text{@iiiminipage}$ : Use new  $\backslash\text{color@setgroup}$  concept. . . 292  
 $\backslash\text{@mpfootnotetext}$ : Use new  $\backslash\text{color@setgroup}$  concept. . . 293  
 Use new  $\backslash\text{normalcolor}$  and  $\backslash\text{@finalstrut}$ . . . . . 293

- General: Superfluous braces removed from several commands 285
- `\color@setgroup`: macro added for colour support ..... 287
- `\endminipage`: Use new `\color@setgroup` concept. . . 293
- 1994/05/11 `ltclass.dtx` v1.0c
- `\endfilecontents`: Add checks for form feed and tab ..... 466
- 1994/05/11 `ltdirchk.dtx` v1.0e
- General: Add `\ProvidesFile` as used in fd files. .... 4
- 1994/05/11 `lterror.dtx` v1.0o
- `\@latexerr`: (ASAJ) Removed one of the extra blank lines to `\@latexerr`. .... 60
- 1994/05/11 `ltlogos.dtx` v1.0o
- `\LaTeX`: Use `\DeclareProtectedCommand`. ASAJ. .... 80
- `\LaTeXe`: Use `\DeclareProtectedCommand`. ASAJ. .... 80
- 1994/05/11 `ltoutenc.dtx` 1.5a
- General: Made T1 and OT1 generate packages rather than def files. Renamed the ‘package’ module to ‘teststy’. .... 94
- 1994/05/11 `ltoutenc.dtx` v1.5a
- General: Reimplemented `\DeclareTextCommand` using `\@changed@cmd` and `\DeclareProtectedCommand`. .... 94
- Renamed the commands again. Made the encoding part of the command syntax. Added the `\DeclareTextCommand` interface. Used `\DeclareProtectedCommand`. .... 91
- `\DeclareTextAccent`: Reimplemented using `\DeclareTextCommand`. .... 97
- 1994/05/11 `ltspace.dtx` v1.0o
- `\,`: Use `\DeclareRobustCommand`. ASAJ. .... 78
- `\hspace`: Use `\DeclareRobustCommand`. ASAJ. .... 78
- 1994/05/12 `ltboxes.dtx` v1.0g
- `\@finalstrut`: macro added .... 295
- `\fbox`: New definition, merged with `\framebox` ..... 289
- `\framebox`: Merged `\fbox` and `\framebox` ..... 289
- `\normalcolor`: macro added for colour support ..... 287
- 1994/05/12 `ltdefns.dtx` v1.0p
- General: (ASAJ) Fixed a bug with `\relax` which was using `\@gobble` before defining it. .... 34
- Fixed a bug with `\relax` which was using `\@gobble` before defining it. .... 42
- 1994/05/12 `ltfssbas.dtx` v2.1j
- General: New `baselinestretch` concept ..... 139
- Replaced hand-protected commands by `\DeclareRobustCommand` defs ..... 139
- `\f@linespread`: New macro .... 147
- `\fontencoding`: Use `\DeclareRobustCommand`. .... 145
- `\fontfamily`: Use `\DeclareRobustCommand`. .... 146
- `\fontseries`: Use `\DeclareRobustCommand`. .... 146
- `\fontshape`: Use `\DeclareRobustCommand`. .... 146
- `\fontsize`: Redefined to use `\set@fontsize` ..... 147
- `\linespread`: New macro ..... 147
- `\mathversion`: Use `\DeclareRobustCommand`. .... 147
- 1994/05/12 `ltfssdcl.dtx` v2.1g
- General: Allow `\relax` as undefined command ..... 185
- Allow `\relax`’ed cmds to be declared ..... 185
- 1994/05/12 `ltfssini.dtx` v2.1i
- General: Moved `\fontencoding` to `fam.dtx` ..... 208
- Moved `\fontfamily` to `fam.dtx` 208
- Moved `\fontseries` to `fam.dtx` 208
- Moved `\fontshape` to `fam.dtx` 208
- Moved `\fontsize` to `fam.dtx` . 208
- Moved `\mathversion` to `fam.dtx` 208
- Moved `\selectfont` to `tracefmt.dtx` ..... 208
- 1994/05/12 `ltfssstrc.dtx` v2.3f
- `\selectfont`: Use `\DeclareRobustCommand` ..... 163
- 1994/05/12 `ltoutenc.dtx` 1.5a
- General: Removed the `\SaveAtCatcode` and `\RestoreAtCatcode` commands. .... 105
- Rewrote for the new syntax. 105, 107
- 1994/05/12 `ltoutput.dtx` v1.0p
- `\@writsetup`: `\normalcoloradded` ..... 408

- General: `\normalcolor` added in various places (DPC). . . . . 382
- 1994/05/13 ltboxes.dtx v1.0h  
`\arrayparboxrestore`: New accent system, use `\let` not `\def` 292
- 1994/05/13 ltcounts.dtx v1.0f  
 General: Removed `\@Ialph` . . . . 134  
 Removed `\@ialph` . . . . . 134
- 1994/05/13 ltdefns.dtx v1.0q  
 General: (ASAJ) Renamed `\DeclareProtectedCommand` to `\DeclareRobustCommand`. Removed `\@if@short@command`. . 34  
 (ASAJ) Replaces `\space` by ‘ ’ in `\csname`. . . . . 34  
 Renamed `\DeclareProtectedCommand` to `\DeclareRobustCommand`. Removed `\@if@short@command`. Moved to after the definition of `\@gobble`. . . . . 42
- 1994/05/13 ltdefns.dtx v1.0r  
 General: (ASAJ) Added logging message to `\DeclareProtectedCommand`. . . . . 34  
 Added logging message to `\DeclareProtectedCommand`. . . . 42
- 1994/05/13 ltdefns.dtx v1.0s  
 General: (ASAJ) Added `\@backslashchar`. . . . . 34  
 (ASAJ) Coded `\@ifdefinable` more efficiently. . . . . 34  
 Coded more efficiently, thanks to FMi. . . . . 39
- 1994/05/13 ltfiles.dtx LaTeX2e  
`\listfiles`: Stop `\listfiles` being run twice . . . . . 89
- 1994/05/13 ltfiles.dtx v1.0g  
`\document`: Added execution of `\every@size` . . . . . 84
- 1994/05/13 ltfinal.dtx v0.1h  
 General: Added package `otlenc`, and defined `\@acci`, `\@accii` and `\@acciii`. . . . . 494
- 1994/05/13 ltfinal.dtx v1.0h  
 General: Added output enc stuff . 501
- 1994/05/13 ltfloat.dtx v1.0g  
`\@footnotetext`: (DPC) Add new style colour support: `\normalcolor` . . . . . 371  
 (DPC) Use `\@finalstrut` . . . . 371  
`\@xfloat`: (DPC) Use `\normalcolor` . . . . . 358
- 1994/05/13 ltfontcmd.dtx v3.3g  
 General: Replaced `\@protecteddef` by `\DeclareRobustCommand` . 236
- 1994/05/13 ltfsbas.dtx v2.1k  
 General: Remove File identification ‘typeout’ . . . . . 139
- 1994/05/13 ltfsbas.dtx v2.1l  
`\DeclareFontEncoding`: Init encoding change command . . . . 142  
`\define@newfont`: Use `\@input@` for fd files . . . . . 150
- 1994/05/13 ltfsdcl.dtx v2.1h  
 General: Removed file identification typeout . . . . . 185
- 1994/05/13 ltfsini.dtx v2.1j  
 General: Removed file identification typeout . . . . . 208
- 1994/05/13 ltfsstrc.dtx v2.3g  
 General: Removed typeouts as `\ProvidesPackage` writes to log. . . . . 159
- 1994/05/13 ltoutenc.dtx v1.5b  
 General: Added `\{`, `\}` and `\$`. . . 91  
 Renamed `\DeclareProtectedCommand` to `\DeclareRobustCommand`. . . . . 91  
 Replaces `\space` by ‘ ’ in `\csname`. . . . . 91
- 1994/05/13 ltpictur.dtx v0.1d  
 General: Removed surplus braces from `\@if..` constructions . . 318
- 1994/05/13 lttab.dtx v1.0d  
`\@contfield`: Colour support . . . 302  
`\@startfield`: Colour support . . 301  
`\@stopfield`: Colour support . . . 301  
`\a`: moved to ltoutenc . . . . . 300
- 1994/05/14 fontdef.dtx v2.1f  
 General: Removed .def files. . . . 216
- 1994/05/14 ltfsbas.dtx v2.1m  
`\enc@update`: Macro added . . . . 146
- 1994/05/14 ltfsbas.dtx v2.1n  
 General: Set defaults for all `\f@...` 147  
`\DeclareErrorFont`: Don’t set `\f@encoding` . . . . . 151  
`\DeclareFontEncoding`: Log if encoding is redeclared . . . . 142  
 Only init enc change cmd when new encoding . . . . . 142
- 1994/05/14 ltfsini.dtx v2.1k  
 General: Init error font just before checking for fontdef.cfg . . . . 212  
`\p@reset@font`: Remove surplus braces . . . . . 211

- 1994/05/14 ltfsstrc.dtx v2.3h  
     `\selectfont`: Added `\enc@update` 164
- 1994/05/14 ltoutenc.dtx 1.5d  
     General: Moved the driver to the top. . . . . 94
- 1994/05/14 ltoutenc.dtx v1.5c  
     General: Added the fontenc package . . . . . 118  
     Added the fontenc package. . . 91  
     Fixed a bug which caused an infinite loop if `\f@encoding` was incorrectly set. . . . . 91, 94  
     Moved fontsmpl to its own dtx file. . . . . 91
- 1994/05/14 ltoutenc.dtx v1.5d  
     General: Rewrote `\DeclareTextCommand` to define its argument to use the current encoding by default, rather than the encoding provided to `\DeclareTextCommand`. . . . . 91, 94  
     Tidied up the documentation. . 91
- 1994/05/14 ltoutenc.dtx v1.5e  
     General: Replaced `\ENC@cmd` by `\ENC-cmd`. . . . . 91
- 1994/05/15 ltfsbas.dtx v2.1o  
     General: encoding cmds changed to enc-cmd . . . . . 139
- 1994/05/16 ltalloc.dtx v1.1a  
     General: (ASAJ) Split from ltinit.dtx. . . . . 49
- 1994/05/16 ltctrl.dtx v1.0a  
     General: (ASAJ) Split from ltinit.dtx. . . . . 51
- 1994/05/16 ltdefns.dtx v1.1a  
     General: (ASAJ) Split from ltinit.dtx. . . . . 34
- 1994/05/16 lterror.dtx v1.1a  
     General: (ASAJ) Completely new error interface. . . . . 55  
     (ASAJ) Split from ltinit.dtx. . 55
- 1994/05/16 ltfinal.dtx v1.0i  
     General: moved output enc stuff to lfonts . . . . . 501
- 1994/05/16 ltfsbas.dtx v2.1p  
     `\fontsize`: Pass `\baselinestretch` not `\f@linespread` . . . . . 147  
     `\linespread`: Remove surplus braces . . . . . 147
- 1994/05/16 ltfsini.dtx v2.1m  
     `\@acciii`: Define saved versions of accents . . . . . 213
- 1994/05/16 ltlogos.dtx v1.1a  
     General: (ASAJ) Split from ltinit.dtx. . . . . 80
- 1994/05/16 ltmath.dtx v1.0k  
     `\ensuremath`: Use `\DeclareRobustCommand` and add extra braces in math mode . . . . . 266
- 1994/05/16 ltoutenc.dtx 1.5h  
     General: `\pounds` was still using u rather than ui shape. . . . . 105
- 1994/05/16 ltoutenc.dtx v1.5f  
     General: enc files now have uc encoding name parts (FMi) . . . . 91  
     Revert code so that the encoding given is used in `\DeclareTextCommand` (FMi) . . . . . 91
- 1994/05/16 ltoutenc.dtx v1.5g  
     General: Made fontenc.sty use the new mixed-case encoding files. 91  
     Removed the lowercasing of the filename. . . . . 118
- 1994/05/16 ltoutenc.dtx v1.5h  
     General: Added `\NG`, `\ng`, `\TH`, `\th`, `\DH`, `\dh`, `\DJ` and `\dj`. . . . . 91  
     Added `\r` (ring accent) and `\k` (ogonek) accents. . . . . 91  
     Fixed a bug with `\pounds`. . . 91  
     Removed `\P` from the OT1 definitions file. . . . . 91
- 1994/05/16 ltoutenc.dtx v1.5i  
     General: Fixed a bug with `\d`. . . 91
- 1994/05/16 ltoutput.dtx v1.0q  
     `\@writsetup`: Changed setting of accents (FMi): with the new encoding setup they can use `\let`. It could also use the new internal commands? . . . . . 409  
     General: Changed setting of accents (FMi). . . . . 382
- 1994/05/16 ltpar.dtx v1.1a  
     General: (ASAJ) Split from ltinit.dtx. . . . . 64
- 1994/05/16 ltplain.dtx v1.0h  
     General: Comment out encoding specific commands . . . . . 28  
     Remove `\@acci` and friends again . . . . . 28  
     Remove unnecessary def for `\item` . . . . . 28  
     `\loop`: Use Kabelschacht method 26  
     `\m@th`: Remove unnecessary space 27
- 1994/05/16 ltspc.dtx v1.1a  
     General: (ASAJ) Split from ltinit.dtx. . . . . 66

1994/05/17 ltclass.dtx v1.0e	Replaced <code>\defaultencoding</code> with <code>\encodingdefault</code> . . . . .	91
<code>\@use@option</code> : Execute option after removing from list, not before		460
1994/05/17 ltdefns.dtx 1.1b	General: (ASAJ) Added the <code>\@protect@...</code> commands. . . . .	43
1994/05/17 ltdefns.dtx v1.1b	General: (ASAJ) Added definitions for <code>protect</code> . . . . .	34
	(ASAJ) Removed warnings and logging to <code>lterror.dtx</code> . . . . .	34
	Added the discussion of protected commands, defined the values that <code>\protect</code> should have. . . . .	43
1994/05/17 ltdefns.dtx v1.1c	General: (ASAJ) Redid definitions for <code>protect</code> . . . . .	34
1994/05/17 lterror.dtx v1.1b	General: (ASAJ) Moved error stuff from <code>ltdefns.dtx</code> . . . . .	55
1994/05/17 ltssini.dtx v2.1n	<code>\copyright</code> : Really add extra braces . . . . .	211
	<code>\nfss@text</code> : Added braces to allow use in subscripts . . . . .	211
1994/05/17 ltmath.dtx v1.0i	General: Replaced <code>\let</code> by <code>\gdef</code> , for indirect definition. . . . .	262
1994/05/17 ltoutenc.dtx v1.5j	General: Added braces to <code>\pounds</code> so it works as a subscript. . . .	91
1994/05/18 ltdefns.dtx 1.1c	General: (ASAJ) Renamed the commands, and removed one which is no longer needed. . . .	43
1994/05/18 ltdefns.dtx v1.1c	General: Redid the discussion and definitions, in line with the proposed new setting of <code>\protect</code> in the output routine. . . . .	43
1994/05/18 ltfinal.dtx v0.1j	General: Corrected the <code>lccode</code> for <code>d-bar</code> . . . . .	494
1994/05/18 ltlogos.dtx v1.1b	General: (ASAJ) Added the $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ logo. . . . .	80
	(ASAJ) Made the $\mathrm{I}\mathrm{T}_{\mathrm{E}}\mathrm{X}2_{\epsilon}$ logo use the text font ‘2’ rather than the math font ‘2’. . . . .	80
1994/05/18 ltoutenc.dtx v1.5k	General: Made dotted-i produce ‘i’. . . . .	91
	Removed braces from <code>\pounds</code> and <code>\dollar</code> . . . . .	91
	General: Initial version of <code>ltbibl.dtx</code> , split from <code>ltidxbib.dtx</code> . . . . .	375
1994/05/19 ltcounts.dtx v1.1a	General: Extracted file from <code>lcntlen</code> . . . . .	132
1994/05/19 ltdefns.dtx v1.1d	General: (RmS) Added definitions for <code>\@namedef</code> and <code>\@nameuse</code> again. . . . .	34
1994/05/19 ltfinal.dtx v0.1k	General: Removed <code>\makeat...</code> . . . .	494
1994/05/19 ltidxglo.dtx v1.1a	General: Initial version of <code>ltidxglo.dtx</code> , split from <code>ltidxbib.dtx</code> . . . .	373
1994/05/19 ltlength.dtx v1.1a	General: Extract file <code>ltlength</code> from <code>lcntlen</code> . . . . .	138
1994/05/19 ltpageno.dtx v1.1a	General: Extract file <code>ltpageno</code> from <code>lcntlen</code> . . . . .	244
1994/05/19 ltplain.dtx v0.1k ltfinal	<code>\showoutput</code> : used <code>\maxdimen</code> not 99999 . . . . .	29
	<code>\showoverfull</code> : used <code>\@one</code> not 1 . . . .	29
1994/05/19 ltxref.dtx v1.1a	General: Extract file <code>ltxref</code> from <code>lcntlen</code> . . . . .	245
1994/05/19 fontdef.dtx v2.1g	General: Removed <code>\DeclareFontEncoding</code> for <code>otl</code> and <code>tl</code> and input <code>.def</code> files instead . .	216
1994/05/2 ltdefns.dtx v1.1f	<code>\renewcommand</code> : Removed surplus <code>\space</code> in error . . . . .	39
	<code>\renewenvironment</code> : Removed surplus <code>\space</code> in error . . . . .	40
1994/05/20 ltdefns.dtx v1.1e	General: Changed command name from <code>\@checkcommand</code> to <code>\CheckCommand</code> . . . . .	34
	<code>\CheckCommand</code> : Changed name from <code>\@checkcommand</code> to <code>\CheckCommand</code> . . . . .	41
1994/05/20 lterror.dtx v1.1c	General: (ASAJ) Added <code>\@latex@info@no@line</code> . . . . .	55
	(ASAJ) Added missing full stops. . . . .	55
	(ASAJ) Fixed a bug with <code>\@in-matherr</code> . . . . .	55

- 1994/05/20 ltfinal.dtx v0.1l  
General: Use new font warning commands ..... 498
- 1994/05/20 ltfloat.dtx v1.0h  
\@endfloatbox: Restore outer value of @nobreak switch. . . 362
- 1994/05/20 ltfntcmd.dtx v3.3h  
General: Use new error commands 236
- 1994/05/20 ltfsbas.dtx v2.1q  
General: Use new error commands 139
- 1994/05/20 ltfsstrc.dtx v2.3i  
General: Use new error command names ..... 159
- 1994/05/20 ltmiscen.dtx v1.0l  
\@writefile: Added correct setting of \protect. .... 251
- 1994/05/20 ltmiscen.dtx v1.0m  
General: Use new warning commands ..... 249
- 1994/05/20 ltoutput.dtx v1.0s  
\@writsetup: Added setting of \protect during \shipout. . 408  
General: Added setting of \protect during \shipout. .... 382
- 1994/05/20 ltpage.dtx v1.0d  
\markright: Changed setting for \protect. .... 380
- 1994/05/20 ltsect.dtx v1.0c  
General: Correct setting of \protect. .... 351  
\addcontentsline: Correct setting of \protect. .... 351
- 1994/05/21 ltbibl.dtx v1.1b  
General: Use new warning commands ..... 375
- 1994/05/21 lterror.dtx v1.1d  
General: (ASAJ) Made the error commands robust. .... 55
- 1994/05/21 ltfiles.dtx v1.0h  
General: Use new error commands 81
- 1994/05/21 ltlists.dtx v1.0f  
General: Use new error commands 269
- 1994/05/21 ltmiscen.dtx v1.0n  
General: Use new error commands 249
- 1994/05/21 ltsect.dtx v1.0d  
General: Use new error commands 343
- 1994/05/21 lttab.dtx v1.0f  
General: Use new error commands 296
- 1994/05/21 ltxref.dtx v1.1b  
General: Use new warning commands ..... 245  
\newlabel: Use new warning commands ..... 246
- 1994/05/22 ltclass.dtx v1.0f  
General: Use new warning and error commands ..... 451
- 1994/05/22 ltdefs.dtx v1.1f  
General: Use new warning and error cmds ..... 34
- 1994/05/22 lterror.dtx v1.1e  
General: (ASAJ) Replaced bgroup by begingroup in error messages, to stop extra mathords creeping into math mode. . . . 55
- 1994/05/22 lterror.dtx v1.2a  
General: (ASAJ) Made \GenericError, \GenericWarning and \GenericInfo robust. .... 55  
(ASAJ) Replaced \@generic@message and \@generic@error by \GenericError, \GenericWarning and \GenericInfo. 55  
(ASAJ) Replaced \ and tilde by \MessageBreak and \space. . . 55  
(ASAJ) Replaces \string by \protect in some messages. . 55
- 1994/05/22 lterror.dtx v1.2d  
\GenericError: (DPC) Alternative version added for old TeXs . . 55  
(DPC) New version using long command name. .... 55
- 1994/05/22 ltfloat.dtx v1.0i  
General: Use new warning commands ..... 353
- 1994/05/22 ltoutput.dtx v1.0t  
General: Changed warnings and infos to new commands. .... 382
- 1994/05/22 ltpictur.dtx v0.1e  
General: Use new warning cmds . 318
- 1994/05/23 ltclass.dtx v1.0h  
\NeedsTeXFormat: Don't stop completely when format is wrong 462  
\usepackage: Remove argument if possible ..... 462
- 1994/05/23 ltdirchk.dtx v1.0f  
General: Document \TeXversion 1
- 1994/05/23 ltfsstrc.dtx v2.3j  
General: Removed def of \f@warn@break ..... 176
- 1994/05/23 ltoutput.dtx v1.0u  
\@activechar@info: Added \MessageBreak ..... 408  
\@writsetup: Changed resetting of \protect after shipout to use \aftergroup ..... 408  
General: Added \MessageBreak. . 382

Changed resetting of <code>\protect</code> after shipout. . . . .	382	1994/05/26 ltplain.dtx v1.1p	
1994/05/24 ltterror.dtx v1.2e		<code>\underbar</code> : (DPC) changed to use <code>\sbox</code> . . . . .	27
<code>\@latex@info@no@line</code> : Macro added . . . . .	58	1994/05/26/16 ltmiscen.dtx v1.0r	
1994/05/24 ltterror.dtx v1.2f		General: <code>\literal</code> removed . . . .	257
General: (DPC) wrap long lines .	55	1994/05/29 ltfsdcl.dtx v2.1j	
1994/05/24 ltfontcmd.dtx v3.3i		General: Use new error commands	185
General: Tidying and typos fixed	236	1994/05/31 ltfinal.dtx v1.0n	
1994/05/24 ltmiscen.dtx v1.0q		General: Renamed <code>lthyphen.*</code> to <code>lthyphen.*</code> . . . . .	494
<code>\@currentvline</code> : Use <code>\@empty</code> as outer default . . . . .	253	1994/06/01 ltboxes.dtx v1.0i	
1994/05/25 ltdirchk.dtx v1.0g		<code>\@framebox</code> : Macro added. . . . .	290
<code>\filename@parse</code> : Mac parser had " typo for : . . . . .	12	<code>\@ifframebox</code> : New version, so <code>\width</code> is correct in <code>\framebox</code>	289
1994/05/25 ltfontcmd.dtx v3.3j		<code>\fbox</code> : New version, using <code>\@framebox</code> . . . . .	289
General: Insertion of <code>\aftergroups</code> to implement <code>\nocorr</code> moved to the end of the group . . . . .	236	<code>\framebox</code> : New version, so <code>\width</code> is correct in <code>\framebox</code> . . . .	289
<code>\check@icr</code> : Macros added . . . .	239	1994/06/01 ltlogos.dtx v1.1d	
<code>\check@nocorr@</code> : Insertion of <code>\aftergroups</code> moved and defaults set up for efficiency . . . . .	239	<code>\LaTeX</code> : Add <code>\m@th</code> to force math size calculations . . . . .	80
<code>\DeclareTextFontCommand</code> : <code>\expandafter</code> inserted . . . . .	238	1994/06/01 ltoutput.dtx v1.0w	
Insertion of <code>\aftergroups</code> moved . . . . .	238	General: Tidied up typesetting. .	382
1994/05/25 ltoutput.dtx v1.0v		1994/06/08 ltfinal.dtx v1.0m	
General: Extra documentation. .	382	General: Add patch file system . .	501
1994/05/25 ltsect.dtx v1.0e		1994/06/09 ltfinal.dtx v1.0n	
<code>\@dottedtocline</code> : Put braces around argument 4 (the actual toc entry) to avoid font (and possibly other) changes leaking out to the leaders. . . . .	352	General: For $\TeX$ 2, do not set codes for higher half of character table. . . . .	496, 499
1994/05/25 ltthm.dtx v1.0c		1994/06/09 ltfontcmd.dtx v3.3k	
General: Modify documentation .	339	General: Tidying and typos fixed in documentation . . . . .	236
1994/05/25 ltvers.dtx v1.0d		1994/06/18 ltfontcmd.dtx v3.3l	
General: Remove PRELIMINARY TEST RELEASE from startup banner (spring is here) . . . .	32	General: Added check for empty text . . . . .	236
1994/05/25 ltxref.dtx v1.1c		<code>\check@nocorr@</code> : Added check for empty text . . . . .	239
General: Modify documentation .	245	1994/06/22 ltfontcmd.dtx v3.3m	
1994/05/26 ltfiles.dtx LaTeX2e		General: Removed space from <code>\nfss@text</code> . . . . .	236
<code>\@missingfileerror</code> : Modify message format . . . . .	88	Renamed <code>\check@nocorr</code> . . . .	236
1994/05/26 ltlogos.dtx v1.1c		<code>\check@nocorr@</code> : Renamed <code>\check@nocorr</code> to <code>\text@command</code> to improve <code>\long</code> error message	239
General: Remove <code>\SLiTeX</code> logo . .	80	<code>\DeclareTextFontCommand</code> : Removed space from <code>\nfss@text</code>	238
1994/05/26 ltplain.dtx v1.1m		1994/06/22 ltmath.dtx v1.2t classes	
<code>\iterate</code> : (CAR) added <code>\long</code> . .	26	<code>\mathindent</code> : Set <code>\mathindent</code> at the end of the class instead of at begin document . . . . .	267
<code>\underbar</code> : (CAR/FMi) changed to use box <code>\tw@</code> . . . . .	27	1994/07/20 ltlogos.dtx v1.1e	
		<code>\LaTeX</code> : Save a few tokens . . . .	80
		<code>\LaTeXe</code> : Save a few tokens . . . .	80

- 1994/07/20 ltpage.dtx v1.0h  
`\sloppy`: Save a few tokens . . . . 381
- 1994/09/16 ltffsbas.dtx v2.1s  
`\nfss@catcodes`: Reset [ and ] as  
well, just in case . . . . . 151
- 1994/10/07 ltoutenc.dtx v1.5l  
General: Moved the ogonek accent. 91
- 1994/10/11 ltdirchk.dtx v1.0h  
`\TeXversion`: Check for TeX3.14 13  
General: Modify all of ltxcheck  
again . . . . . 13
- 1994/10/12 ltsect.dtx v1.0f  
General: Doc. typos . . . . . 343
- 1994/10/14 fontdef.dtx v2.2a  
General: New coding . . . . . 214
- 1994/10/14 ltffsini.dtx v2.2a  
General: New coding for cfg files . 208
- 1994/10/14 ltmiscen.dtx v1.0s  
General: Move math to other file 249
- 1994/10/14 ltplain.dtx v1.1a  
General: Moved code to other files. 14
- 1994/10/15 ltffsbas.dtx v2.1t  
`\extract@alph@from@version`:  
Warn if math alpha is used out-  
side math . . . . . 156
- 1994/10/18 ltboxes.dtx v1.0j  
`\@frameb@x`: `\leavevmode` added 290  
`\@ifframebox`: `\leavevmode` moved  
to `\@frameb@x` . . . . . 289  
`\@parboxto`: Macro added to re-  
move misuse of `\@empty` . . . 291  
General: stuff from ltpatch done . 285  
`\fbox`: `\long` added . . . . . 289  
`\mbox`: `\long` added . . . . . 286  
`\sbox`: `\long` added . . . . . 288
- 1994/10/18 ltclass.dtx v1.0j  
General: Move `\listfiles` to lt-  
files.dtx . . . . . 468
- 1994/10/18 ltdefs.dtx v1.2a  
`\@star@or@long`: macro added . . 37  
General: Add extra test for `\end-`  
`graf` . . . . . 34  
Add star-forms for all commands 34  
`\renew@environment`: reset end  
command . . . . . 40
- 1994/10/18 ltfiles.dtx v1.0i  
`\listfiles`: code moved here from  
ltclass . . . . . 89
- 1994/10/18 ltoutenc.dtx v1.5l  
General: Added new definitions of  
`\patterns` and `\hyphenation`. 101
- 1994/10/18 ltoutenc.dtx v1.5m  
General: Added new definitions of  
`\patterns` and `\hyphenation`. 91
- 1994/10/18 ltsect.dtx v1.0g  
`\@dottedtocline`: Added `\normal-`  
`color` for page number . . . . 352  
General: Added `\normalcolor` . . 343
- 1994/10/19 ltffsbas.dtx v2.1t  
`\DeclareFontEncoding`: Add miss-  
ing `\relax`. . . . . 141
- 1994/10/23 ltfsstrc.dtx v23.k  
`\every@math@size`: Renamed to  
`\every@math@size` . . . . . 166
- 1994/10/23 ltmath.dtx v1.0l  
`\@eqnnum`: Added `\normalcolor`  
since `\eqno` introduces a sub-  
group of the displayed math  
group . . . . . 264  
`\ensuremath`: Remove extra braces:  
but see p 168 of Leslie's book 266
- 1994/10/24 ltboxes.dtx v1.0k  
`\fbox`: Inner braces added (to fix la-  
tex/1061) . . . . . 289
- 1994/10/25 fontdef.dtx v2.2c  
General: Added OMSenc.def . . . 216
- 1994/10/25 ltboxes.dtx v1.0l  
`\@isavepicbox`: missing percent  
(moved from ltpatch) . . . . 288
- 1994/10/25 ltdefs.dtx v1.2b  
General: Documentation improve-  
ments . . . . . 34
- 1994/10/25 ltoutenc.dtx 1.6a  
General: Added `\textdollar`,  
`\textlbrace`, `\textrbrace`,  
`\textsterling`, `\textunder-`  
`line`. . . . . 107  
Removed `\textlbrace`, `\textr-`  
`brace`, `\textunderline` to give  
them their proper names. . . 107
- 1994/10/25 ltoutenc.dtx v1.6a  
General: Added `\Provide-`  
`TextCommand`, `\UseTextSym-`  
`bol`, `\UseTextAccent`, `\De-`  
`clareTextSymbolDefault`,  
`\DeclareTextAccentDefault`,  
`\DeclareTextCommandDefault`,  
and `\ProvideTextCommandDe-`  
`fault`. . . . . 91  
Added the `\Provide` commands,  
and the default definitions. . . 94  
Added the defaults. . . . . 102  
Added the files OTlenc.def,  
Tlenc.def and OMSenc.def. . 102  
Added the OMS encoding. . . . 111
- 1994/10/27 ltoutenc.dtx 1.6b  
General: Added `\textasci-`  
`icircum` `\textasciitilde`

- `\textbackslash`    `\textbar`  
`\textbraceleft`   `\textbraceright`    `\textcompwordmark`   `\textemdash`   `\textendash`   `\textexclamdown`  
`\textgreater`   `\texthyphenchar`   `\texthyphen`   `\textless`   `\textquestiondown`   `\textquotedblleft`   `\textquotedblright`   `\textquotedbl`   `\textquoteleft`   `\textquoteright`  
`\textunderscore`   `\textvisiblespace` ..... 107  
 Added:   `\textemdash`   `\textendash`   `\textexclamdown`   `\texthyphenchar`   `\texthyphen`   `\textquestiondown`   `\textquotedblleft`   `\textquotedblright`   `\textquoteleft`   `\textquoteright` ..... 105  
 1994/10/27 ltoutenc.dtx v1.5d  
   General:    Rewrote `\DeclareTextSymbol` to define its argument to use the current encoding by default, to fit with `\DeclareTextCommand`. .... 95  
 1994/10/27 ltoutenc.dtx v1.6b  
   General: Added `\textbackslash`. 111  
   Added more defaults for OT1. 102  
   Removed the enc.def files .... 91  
   Removed the files OT1enc.def, T1enc.def and OMSenc.def. . 102  
   Renamed `\textlbrace` to `\textbraceleft` and `\textrbrace` to `\textbraceright`. . 111  
 1994/10/29 ltmath.dtx v1.0m  
   General: ASAJ: Added `\DeclareMathOperator`. .... 258  
   ASAJ: Tidied up documentation. .... 262  
 1994/10/29 ltmath.dtx v1.0m  
   General: ASAJ: Added `\mathellipsis`, `\mathdollar` and `\mathsterling`. .... 262  
   ASAJ: Removed `\dag`, `\ddag`. . 262  
   ASAJ: Renamed `\S` and `\P` to `\mathsection` and `\mathparagraph` and made them `\mathchardefs`. .... 262  
 1994/10/29 ltoutenc.dtx v1.6c  
   General: Added commands like `\dots` for use in text and math. 102  
   Renamed `\P`, `\S`, `\dag` and `\ddag` to `\textparagraph`, `\textsection`, `\textdagger` and `\textdaggerdbl`. .... 91  
 1994/10/30 ltdefs.dtx v1.2c  
   `\@onelevel@sanitize`: Macro added ..... 47  
   General: (CAR)\@onelevel@sanitize added ..... 34  
 1994/10/30 ltdefs.dtx v1.2f  
   General: (DPC)\newwrite's moved to ltfiles ..... 34  
 1994/10/30 ltmath.dtx v1.0n  
   General: ASAJ: Moved the new commands to ltoutenc. .... 262  
 1994/10/30 ltoutenc.dtx v1.6d  
   General: Added `\DeclareTextCompositeCommand`. .... 91  
   Added `\textcircled`. 91, 103, 111  
   Added `\t`. .... 103  
   Added math commands. .... 91  
   Added OML encoding. ... 91, 103  
   Added the OML encoding. ... 112  
   Made `\textless` and `\textgreater` come from OML. .... 103  
   Moved math commands here from ltmath. .... 105  
   Removed `\textregistered`. .. 103  
   Rewrote `\copyright` to use `\textcircled`. .... 103  
 1994/10/31 fontdef.dtx v2.2d  
   General: Added OMLenc.def ... . 216  
 1994/10/31 fontdef.dtx v2.2e  
   General: ... and moved further down ..... 216  
 1994/10/31 ltfloat.dtx v1.1a  
   `\@dblfloat`: Major changes since two-column and one-column cases merged ..... 357  
   `\@dblflset`: Macro added ..... 356  
   Major changes to parameter parsing, setting of local variables, etc; two-column and one-column cases merged; space hacks moved ..... 356  
   `\@endfloatbox`: (DPC/CAR) Extra box added to remove colour resetting from vmode ..... 362  
   `\@floatboxreset`: Macro added . 360  
   `\@footnotetext`: (DPC/CAR) Move colour setting to output routine ..... 371  
   `\@savemarbox`: (DPC/CAR) Extra box added for colour ..... 365  
   `\@setfps`: Macro added ..... 357

\@xdblfloat: Macros removed: \@dbflt, \@xdblfloat . . . . .	362	\makeglossary to \nofiles. ASAJ. . . . .	85
\@xfloat: (DPC/CAR) Extra box added to remove colour reset- ting from vmode . . . . .	358	\protected@write: Macro added ASAJ. . . . .	85
Major changes, removing set- ting of local variables, space hacks etc; two-column and one- column cases merged . . . . .	357	1994/11/04 ltfloat.dtx v1.1b \@footnotetext: (ASAJ) Added \protected@edef. . . . .	371
Reset hook added . . . . .	358	\footnotemark: Added \pro- tected@xdef to \foot- notemark. . . . .	371
\@xympar: (DPC/CAR) Extra box added since needed for floats	366	1994/11/04 ltidxglo.dtx v1.1b \@wrglossary: Added \pro- tected@write to \@wrglos- sary. . . . .	374
\fps@dbl: Macro added . . . . .	357	\@wrindex: Added \pro- tected@write to \@wrindex.	374
1994/10/31 ltoutput.dtx v1.1a \@makecol: (DPC/CAR) Colour re- setting moved to here . . . . .	405	General: Removed \if@files from \makeindex. . . . .	373
\@topnewpage: (DPC/CAR) Extra box added to remove colour re- setting from vmode . . . . .	397	\makeglossary: Removed \if@files from \makeglos- sary. . . . .	374
(DPC/CAR) Use \color@begingroup for colour . . . . .	397	1994/11/04 ltmiscen.dtx v1.0t \@writefile: Removed setting of \protect. ASAJ. . . . .	251
(DPC/CAR) Use \normalcolor	397	1994/11/04 ltoutenc.dtx v1.6f General: Added \_. . . . .	104
1994/11/02 ltoutenc.dtx v1.6d General: Wrapped lines longer than 70 characters. . . . .	91	Added \mathunderscore. . . . .	105
1994/11/03 ltclass.dtx v1.0k General: Move \@missingfileer- ror to ltfiles . . . . .	455	1994/11/04 ltpage.dtx v1.0e \markright: Added \@unexpand- able@protect. ASAJ. . . . .	380
1994/11/03 ltidrchk.dtx v1.0i General: Generate an error if la- tex.ltx not used with clean ini- tex . . . . .	1	1994/11/04 ltsect.dtx 1.0h \@sect: (ASAJ) Added \pro- tected@edef. . . . .	346
1994/11/03 ltfiles.dtx v1.0j \@missingfileerror: Move here from ltclass . . . . .	88	General: (ASAJ) Added \pro- tected@xdef to \thanks. . . . .	343
1994/11/04 ltboxes.dtx v1.0m \@mpfootnotetext: Added \pro- tected@edef. ASAJ. . . . .	293	1994/11/04 ltsect.dtx v1.0h General: Added \protected@write to \addtocontents. ASAJ. . . . .	351
1994/11/04 ltdefns.dtx v1.2e General: Added \set@display@protect to \typeout. ASAJ. . . . .	34	\addcontentsline: Added \pro- tected@write to \addcon- tentsline. ASAJ. . . . .	351
Added commands for setting and restoring \protect. ASAJ. . . . .	44	1994/11/04 lttab.dtx v1.0h \@mkpream: (ASAJ) Added \@unexpandable@protect to \@mkpream. . . . .	313
Rewrote protected short com- mands using \x@protect. ASAJ. . . . .	43	\multicolumn: (ASAJ) added \set@typeset@protect. . . . .	309
1994/11/04 lterror.dtx v1.2g General: Added \set@display@protect to \Generic* commands. ASAJ. . . . .	55	1994/11/04 ltxref.dtx v1.1d \label: (ASAJ) Added \pro- tected@write . . . . .	247
1994/11/04 ltfiles.dtx v1.0k \nofiles: Added setting of \pro- tected@write, \makeindex and		\refstepcounter: (ASAJ) Added \protected@edef . . . . .	247

1994/11/05 ltboxes.dtx v1.0n	
\@mpfootnotetext: Colour resetting for footnotes moved to endminipage: as for main page. .	293
\color@endbox: macro added for colour support	287
\color@hbox: macro added for colour support	287
\endminipage: Colour resetting for footnotes moved to here: as for main page.	293
1994/11/05 ltboxes.dtx v1.0o	
\@mpfootnotetext: Colour groups restored here.	293
1994/11/05 ltfloat.dtx v1.1c	
\@dblflset: Add compatibility with old version of \xfloat.	356
\@endfloatbox: Use new \color@hbox concept.	362
\@footnotetext: Removed \normalcolor (again)	371
\@savemarbox: Use new \color@hbox concept.	365
\@setfps: Add compatibility with old version of \xfloat.	357
\xfloat: Add compatibility with old version of \xfloat: but the arguments, provided at exorbitant cost, are now completely ignored	357
Use new \color@hbox concept.	358
\@xympar: Use new \color@hbox concept.	366
1994/11/05 ltoutenc.dtx v1.6g	
General: Added setting of \@typeset@protect to \patterns and \hyphenation.	101
1994/11/05 ltoutput.dtx v1.1b	
\@topnewpage: Use new \color@hbox concept.	397
\@writsetup: Change protect settings for new-style, protect-free aux-files.	408
Use new \color@hbox concept.	408
1994/11/05 ltoutput.dtx v1.1c	
\@beginDvi: Added macro	411
\@beginDvibox: Added macro	394
\@writsetup: Add new \AtBeginDvi concept	408
\AtBeginDvi: Added macro	394
1994/11/06 ltssbas.dtx v2.1u	
\cf@encoding: New macro	147
\DeclareFixedFont: Renamed \every@size to \every@math@size.	140
1994/11/06 ltssini.dtx v2.2b	
\@setsize: Use \@typeset@protect	210
1994/11/06 ltssstrc.dtx v2.3k	
\glb@currsiz: New implementation	165
\try@simples: New implementation	176
\try@size@substitution: New implementation	176
\tryis@simple: New implementation	176
1994/11/07 fontdef.dtx v2.2f	
General: (DPC) Add \DeclareMathSizes declarations	219
(DPC) Updated to use \ProvidesFile	216
1994/11/07 ltfiles.dtx v1.0l	
\@unused: move here from ltdefs, remove duplicate \mainaux	83
1994/11/07 ltfiles.dtx v1.0m	
\document: Renamed \every@size to \every@math@size.	84
1994/11/07 preload.dtx v2.1e	
General: (DPC) Updated to use \ProvidesFile	232
1994/11/09 ltboxes.dtx v1.0p	
\@finalstrut: Revert \finalstrut to 2.09 equivalent (from lt-patch)	295
General: more colour changes...	285
1994/11/09 ltssbas.dtx v2.1v	
\@vpt: (DPC) macros added, from setsize.dtx	157
(DPC) reduce save stack usage latex/1742	157
1994/11/10 ltbibl.dtx v1.1c	
General: Fix \nocite{*}	375
\nocite: Fix \nocite{*}	377
1994/11/10 ltmath.dtx v1.2v classes	
eqnarray: Added value of \parskip to \abovedisplayskip to compensate for negative \topsep	268
1994/11/10 ltoutput.dtx v1.1e	
\@writsetup: Modify \protect setting	408
1994/11/10 ltplain.dtx v1.1b	
General: (CAR) added patch to \loop.	14
\iterate: (CAR) added extra \relax	26

1994/11/11 ltspac.dtx v1.2a	1994/11/17 ltdirchk.dtx v1.0j
<code>\:</code> (DPC) Make robust . . . . . 70	General: <code>\@tempa</code> to <code>\reserved@a</code> . 1
1994/11/12 ltfntcmd.dtx v3.3o	1994/11/17 lterror.dtx v1.2h
<code>\normalsize</code> : Added <code>\Message-</code>	General: <code>\@tempa</code> to <code>\reserved@a</code> 55
<code>Break</code> . . . . . 243	1994/11/17 ltfiles.dtx v1.0n
1994/11/12 ltlists.dtx v1.2b ltspac	General: <code>\@tempa</code> to <code>\reserved@a</code> 81
<code>\endtrivlist</code> : Changed order of	1994/11/17 ltfinal.dtx v1.0o
tests to make <code>\@noitemerror</code>	General: <code>\@tempa</code> to <code>\reserved@a</code> 494
correct: end of an era. . . . . 278	1994/11/17 ltfloat.dtx v1.1e
1994/11/12 ltmiscen.dtx v1.0u	General: <code>\@tempa</code> to <code>\reserved@a</code> 353
<code>center</code> : Changed end macro to	1994/11/17 ltfntcmd.dtx v3.3p
<code>\def</code> : safer and consistent . . 254	General: <code>\@tempa</code> to <code>\reserved@a</code> 236
<code>flushleft</code> : Changed end macro to	1994/11/17 ltfssbas.dtx v2.1w
<code>\def</code> : safer and consistent . . 254	General: <code>\@tempa</code> to <code>\reserved@a</code> 139
<code>flushright</code> : Changed end macro to	1994/11/17 ltfssdcl.dtx v2.1m
<code>\def</code> : safer and consistent . . 254	General: <code>\@tempa</code> to <code>\reserved@a</code> 185
1994/11/12 ltplain.dtx v1.1c	1994/11/17 ltfsstrc.dtx v2.3l
General: Comment out more encod-	General: <code>\@tempa</code> to <code>\reserved@a</code> 159
ing specific commands . . . . . 28	1994/11/17 ltmath.dtx v1.0o
1994/11/12 ltspac.dtx v1.2b	General: <code>\@tempa</code> to <code>\reserved@a</code> 258
<code>\addpenalty</code> : Corrected error mes-	1994/11/17 ltmiscen.dtx v1.0v
sage . . . . . 75	General: <code>\@tempa</code> to <code>\reserved@a</code> 249
<code>\addvspace</code> : Corrected error mes-	1994/11/17 ltoutenc.dtx v1.6h
sage . . . . . 75	General: (DPC) <code>\@tempa</code> to <code>\re-</code>
1994/11/13 ltspac.dtx v1.2c	<code>served@a</code> . . . . . 91
<code>\addpenalty</code> : Recorrected error	1994/11/17 ltoutput.dtx v1.1h
message . . . . . 75	General: <code>\@tempa</code> to <code>\reserved@a</code> 382
<code>\addvspace</code> : Recorrected error mes-	1994/11/17 ltpictur.dtx v1.0f
sage . . . . . 75	General: <code>\@tempa</code> to <code>\reserved@a</code> 318
1994/11/14 ltoutput.dtx v1.1f	1994/11/17 ltsect.dtx v1.0i
<code>\@beginDvi</code> : Use normal box regis-	General: <code>\@tempa</code> to <code>\reserved@a</code> 343
ter: why a box? . . . . . 411	1994/11/17 lttab.dtx v1.0j
<code>\@beginDviBox</code> : Use normal box	General: <code>\@tempa</code> to <code>\reserved@a</code> 296
register: why a box? . . . . . 394	1994/11/18 ltboxes.dtx v1.0r
<code>\@writsetup</code> : Modify new <code>\AtBe-</code>	<code>\color@vbox</code> : macro added for
<code>ginDvi</code> concept . . . . . 408	colour support . . . . . 287
General: Removed old definition of	1994/11/18 ltfinal.dtx v1.0n
<code>\@testfp</code> . . . . . 382	General: re-allow slots 127–255 . . 499
1994/11/14 ltspac.dtx v1.2d	1994/11/18 ltfssbas.dtx v2.1x
<code>\:</code> (DPC) Macro modified . . . . . 70	General: (DPC) use <code>\reserved@f</code>
1994/11/14 lttab.dtx v1.0i	not <code>\next</code> . . . . . 139
<code>\tabularnewline</code> : (DPC) Macro	1994/11/18 ltfssdcl.dtx v2.1m
added . . . . . 308	<code>\DeclareMathDelimiter</code> : (DPC)
1994/11/16 fontdef.dtx v2.2h	<code>\expandafter</code> instead of <code>\next</code> 201
General: (DPC) Removed <code>\{</code> and	1994/11/18 ltfsstrc.dtx v2.3m
<code>\}</code> . . . . . 216	General: <code>\next</code> to <code>\reserved@f</code> . 159
1994/11/17 ltboxes.dtx v1.0q	1994/11/18 ltmath.dtx v1.0p
General: <code>\@tempa</code> to <code>\reserved@a</code> 285	<code>\phantom</code> : (DPC) colour support 260
1994/11/17 ltclass.dtx v1.0l	(DPC) use <code>\expandafter</code> instead
General: <code>\@tempa</code> to <code>\reserved@a</code> 451	of <code>\next</code> . . . . . 260
1994/11/17 ltcntrl.dtx v1.0b	<code>\prime@s</code> : (DPC) use <code>\@let@token</code>
General: <code>\@tempa</code> to <code>\reserved@a</code> 51	instead of <code>\next</code> and <code>\ex-</code>
1994/11/17 ltdefs.dtx v1.0g	<code>pandafter</code> instead of <code>\nxt</code> . . 262
General: <code>\@tempa</code> to <code>\reserved@a</code> 34	<code>\smash</code> : (DPC) colour support . . 260

(DPC) use <code>\expandafter</code> instead of <code>\next</code> . . . . .	260	<code>\listfiles</code> : Use <code>\@dofilelist</code> .	89
1994/11/21 <code>ltfloat.dtx</code> v1.1f		<code>\nofiles</code> : There is no <code>\@gobblethree</code> . . . . .	85
<code>\@endfloatbox</code> : Added reset of minipage flag . . . . .	362	1994/11/30 <code>ltfssbas.dtx</code> v2.1y	
Corrected position of <code>\outer@nobreak</code> . . . . .	362	<code>\fontshape</code> : Use <code>\@currentcmd</code> in <code>\@enc@update</code> . ASAJ. . . . .	146
<code>\@marginparreset</code> : Macro added	365	1994/11/30 <code>ltmath.dtx</code> 1.0q	
<code>\@savemarbox</code> : Added <code>\@setminipage</code> etc . . . . .	365	General: ASAJ: <code>\DeclareMathOperator</code> moved to <code>AMSTeX</code> . . . . .	258
Added resetting of size and font	365	1994/11/30 <code>ltmiscen.dtx</code> v1.0w	
Changed to <code>\color@vbox</code> . . . . .	365	<code>\enddocument</code> : (DPC) Do warnings even for <code>\nofiles</code> . . . . .	251
Use <code>\@setnobreak</code> etc . . . . .	365	(DPC) Use <code>\@dofilelist</code> . . . . .	251
<code>\@setminipage</code> : Macro added . . . . .	360	1994/11/30 <code>ltoutenc.dtx</code> 1.7a	
<code>\@setnobreak</code> : Macro added . . . . .	360	General: Redefined <code>\a</code> for the new scheme. . . . .	102
<code>\@xfloat</code> : Added <code>\@setminipage</code>	358	1994/11/30 <code>ltoutenc.dtx</code> v1.6g	
Added resetting of size and font	358	General: Removed new definitions of <code>\patterns</code> and <code>\hyphenation</code> , since encoding-specific commands now expand in the mouth. . . . .	101
Changed to <code>\color@vbox</code> so that large floats overflow at the bottom . . . . .	358	1994/11/30 <code>ltoutenc.dtx</code> v1.7a	
Missing percents reinserted after 4, 8: these are not numbers. . . . .	357	General: Added new code for encoding-specific commands. These now expand in the mouth, which means that ligaturing and kerning can happen. . . . .	91
Use <code>\@setnobreak</code> . . . . .	358	Always load the <code>enc.def</code> file, so that the default encoding for the commands will change. . . . .	118
<code>\@xympar</code> : Changed to <code>\color@vbox</code> . . . . .	366	Redefined <code>\@changed@cmd</code> to expand in the mouth. . . . .	95
1994/11/21 <code>ltoutput.dtx</code> v1.1i		Removed <code>\@changed@x@mouth</code> since <code>\@changed@x</code> now expands in the mouth. . . . .	95
<code>\@addtocurcol</code> : Added <code>\if@nobreak</code> test before float box . . . . .	420, 424	Rewrote <code>\@text@composite</code> so it allows an empty argument, or an argument containing lots of commands. . . . .	97
<code>\@specialoutput</code> : Added <code>\if@nobreak</code> test . . . . .	401	1994/12/01 <code>ltfinal.dtx</code> v1.0p	
<code>\@topnewpage</code> : Changed to <code>\color@vbox</code> . . . . .	397	General: Renamed <code>lthyphen.*</code> to <code>hyphen.*</code> . . . . .	494
1994/11/22 <code>ltfssdcl.dtx</code> v2.1o		1994/12/01 <code>lthyphen.dtx</code> v1.0g	
General: wrap long lines . . . . .	185	General: Rename <code>lthyphen.ltx/cfg</code> to <code>hyphen.ltx/cfg</code> . . . . .	470
1994/11/22 <code>ltoutenc.dtx</code> v1.6i		1994/12/01 <code>ltplain.dtx</code> v1.1g	
General: Corrected <code>\dots</code> so that there's no kerning in monowidth fonts. . . . .	91	General: (DPC) More doc changes	14
Corrected typo with <code>\mathunderscore</code> . . . . .	91	1994/12/02 <code>fontdef.dtx</code> v2.2i	
Fixed empty accents. Again. . . . .	91	General: Commented out <code>\ldots</code> . ASAJ. . . . .	214
1994/11/24 <code>ltdefns.dtx</code> v1.2h		1994/12/02 <code>ltfssini.dtx</code> v2.2c	
<code>\@newenv</code> : Added test for <code>\endgraf</code>	40	<code>\copyright</code> : <code>\copyright</code> is now in <code>ltoutenc</code> . ASAJ . . . . .	211
1994/11/25 <code>ltplain.dtx</code> v1.1f			
General: (DPC) Comment out lots of obsolete code . . . . .	14		
1994/11/26 <code>ltfloat.dtx</code> v1.1b			
<code>\footnote</code> : (ASAJ) Added <code>\protected@xdef</code> . . . . .	370		
1994/11/28 <code>ltcntrl.dtx</code> v1.0c			
General: Documentation improvements . . . . .	51		
1994/11/30 <code>ltfiles.dtx</code> v1.0o			
<code>\@dofilelist</code> : Macro added . . . . .	90		

1994/12/02 ltlists.dtx v1.0e	
<code>\@trivlist</code> : RmS: Added check for looping . . . . .	277
1994/12/02 ltoutenc.dtx 1.7b	
General: Redefined <code>\a</code> properly. . .	102
1994/12/02 ltoutenc.dtx v1.7b	
General: Fixed a bug with <code>\a</code> . . .	91
1994/12/04 lthyphen.dtx v1.0h	
General: Documentation edits for /1989 . . . . .	470
1994/12/05 ltoutenc.dtx v1.7c	
General: Added braces to <code>\textcircled</code> . . . . .	91
1994/12/06 ltssbas.dtx v2.1z	
<code>\DeclareFontEncoding</code> : use <code>\nfss@catcodes</code> . . . . .	141
<code>\nfss@catcodes</code> : Added tab char as well . . . . .	151
1994/12/08 ltoutenc.dtx v1.7d	
General: Added <code>\null</code> and <code>\sh@ft</code> to <code>\b</code> and <code>\d</code> . . . . .	91
1994/12/08 ltab.dtx v1.0k	
<code>\@array</code> : Add <code>\tabularnewline</code> . .	308
<code>\tabularnewline</code> : (DPC) Made it <code>\relax</code> . . . . .	308
1994/12/09 ltbibl.dtx v1.1d	
<code>\bibliographystyle</code> : (DPC) Allow use in preamble. . . . .	377
1994/12/10 ltfloat.dtx v1.1g	
<code>\@dblfloat</code> : Old version reinstated temporarily . . . . .	357
<code>\@dblflset</code> : Macro removed temporarily . . . . .	356
Old version reinstated temporarily . . . . .	356
<code>\@setfps</code> : Macro removed temporarily . . . . .	357
<code>\@xdblfloat</code> : Macros reinserted temporarily . . . . .	362
<code>\@xfloat</code> : Old version reinstated temporarily . . . . .	357
Sanitisation added temporarily . . . . .	357
General: Some temps reinserted temporarily . . . . .	353
<code>\fps@dbl</code> : Macro removed temporarily . . . . .	357
1994/12/10 ltfntcmd.dtx v3.3q	
<code>\@@math@egroup</code> : Don't read arguments . . . . .	243
<code>\check@nocorr@</code> : Use <code>\space</code> command for comparison . . . . .	239
1994/12/10 ltssdcl.dtx v2.1p	
<code>\document@select@group</code> : Surround with braces (add fourth	
arg) . . . . .	189
<code>\select@group</code> : Surround with braces (add fourth arg) . . . .	187
1994/12/10 ltoutenc.dtx v1.7e	
General: Added documentation for the OML encoding. . . . .	91
Replaced width with <code>\@width</code> and ditto height in vrules. . . .	91
1994/12/14 ltoutenc.dtx v1.7f	
General: Added braces to <code>\copyright</code> so it works unbraced in subscripts. . . . .	91
Added check for math mode in <code>\@changed@cmd</code> . . . . .	91
Commented out <code>\textasciicircum</code> , <code>\textasciitilde</code> , <code>\textbackslash</code> , <code>\textbar</code> , <code>\textgreater</code> , <code>\textthyphenchar</code> , <code>\texthyphen</code> and <code>\textless</code> to save memory. . . . .	91
1995/01/12 ltmath.dtx v1.2y classes	
<code>\@eqnnum</code> : Added <code>\normalcolor</code> . .	266
1995/03/03 ltoutenc.dtx 1.7g	
General: Corrected an error in documentation referring to the tabular rather than the tabbing environment. . . . .	102
1995/04/02 ltfntcmd.dtx v3.3r	
<code>\@@math@egroup</code> : Read them again to be able to add <code>\relax</code> . . .	243
1995/04/02 ltssdcl.dtx v2.1q	
<code>\document@select@group</code> : fix problem for pr/1275 . . . . .	189
<code>\select@group</code> : fix problem for pr/1275 . . . . .	187
<code>\set@mathdelimiter</code> : fix pr/1329 .	203
1995/04/02 ltssini.dtx v2.2d	
<code>\not@math@alphabet</code> : add <code>\noexpand</code> to second part of message .	209
1995/04/21 ltclass.dtx v1.0m	
<code>\DeclareOption*</code> : Made long /1498 . . . . .	458
<code>\endfilecontents</code> : Close input check stream: latex/1487 . . .	466
1995/04/21 ltfinal.dtx v1.0q	
General: Allow initial patch level 0 .	502
1995/04/21 ltoutenc.dtx v1.7h	
General: Added <code>\null</code> <code>\k</code> latex/1274 . . . . .	91
1995/04/22 ltfiles.dtx v1.0p	
<code>\includeonly</code> : Allow blanks in argument . . . . .	85

1995/04/22 ltmiscen.dtx v1.0x	
General: Removed extra def of	
\@gobble . . . . .	249
1995/04/23 ltsect.dtx v1.0j	
\addcontentsline: Use \con-	
tentsline internally. . . . .	351
1995/04/24 ltbibl.dtx v1.1e	
\@citex: Add \mbox to undefined	
case: latex/1239. . . . .	376
1995/04/24 ltbibl.dtx v1.1f	
\bibcite: Make \@onlypreamble	
/1388. . . . .	376
1995/04/24 ltcntrl.dtx v1.0d	
\@for: Dont expand second argu-	
ment with \edef: /1317 (DPC)	53
1995/04/24 ltoutput.dtx v1.1j	
\fl@tracemessage: Do not add to	
kernel unless ‘trace’ specified	437
1995/04/24 ltoutput.dtx v1.1l	
\@begindvibox: Add \vbox la-	
tex/1392 . . . . .	394
\@writeseup: Reset \ latex/1451	
(DPC) . . . . .	409
1995/04/24 ltpage.dtx v1.0f	
\fussy: reset \emergencystretch	
latex/1344 . . . . .	381
1995/04/24 ltplain.dtx v1.1h	
\newlanguage: Remove remaining	
\outer declarations. . . . .	16
1995/04/24 ltxref.dtx v1.1e	
\newlabel: Make \@onlypreamble	
for /1388. . . . .	246
1995/04/25 ltdefns.dtx v1.2i	
\@check@c: Make \long for la-	
tex/1346 . . . . .	42
\new@environment: Parse argu-	
ments slowly but safely /1507	40
1995/04/25 ltfiles.dtx v1.0q	
\document: Removed execution of	
\every@size latex/1407 . . . .	84
1995/04/25 ltsect.dtx v1.0k	
\@dottedtocline: Added \hbox	
around dots. . . . .	352
1995/04/27 ltboxes.dtx v1.0s	
\@frameb@x: Move \leavevmode for	
graphics/1512 . . . . .	290
\@iframebox: Move \leavevmode	
for graphics/1512 . . . . .	289
\@iirsbox: Move \leavevmode for	
graphics/1512 . . . . .	295
\@irsbox: Move \leavevmode for	
graphics/1512 . . . . .	294
\fbbox: Move \leavevmode for	
graphics/1512 . . . . .	289
\raisebox: Move \leavevmode for	
graphics/1512 . . . . .	294
1995/04/27 ltfiles.dtx v1.0r	
\document: Added \global to sup-	
port groups in hook . . . . .	84
1995/04/27 ltmiscen.dtx v1.0y	
\enddocument: \@checkend moved	
after hook . . . . .	250
1995/04/27 ltplain.dtx v1.1i	
General: Move \hang and \textin-	
dent to latex209.def . . . . .	28
1995/04/29 ltcntrl.dtx v1.0e	
General: Moved init of \protect	
to ltdefns.dtx . . . . .	54
Removed unused defs for \@set-	
protect and \@resetprotect	54
1995/04/29 ltdefns.dtx v1.2j	
\protect: Init \protect here . . .	45
1995/04/29 ltpar.dtx v1.1b	
General: (TO) Comments clean-	
up. . . . .	64
1995/05/02 ltsect.dtx v1.0l	
\@dottedtocline: Don’t reset to	
\rmfamily . . . . .	352
1995/05/03 ltsect.dtx v1.0m	
General: TO: Promoted documen-	
tation to doc.sty standard . .	343
1995/05/06 ltsect.dtx 1.0n	
\@seccntformat: Use \quad instead	
of \hskip . . . . .	348
\@sect: Added \relax after \@sec-	
cntformat just in case . . . .	346
1995/05/07 ltboxes.dtx v1.0t	
General: Use \hb@xt@ . . . . .	285
1995/05/07 ltdefns.dtx v1.2k	
\hb@xt@: Macro added . . . . .	35
1995/05/07 ltmath.dtx v1.0r	
General: Use \hb@xt@ . . . . .	258
1995/05/07 ltoutput.dtx v1.1m	
General: Use \hb@xt@. . . . .	382
1995/05/07 ltpictur.dtx v1.0g	
General: Use \hb@xt@ . . . . .	318
1995/05/07 ltplain.dtx v1.1j	
General: Use \hb@xt@ . . . . .	14
1995/05/07 ltsect.dtx v1.0o	
General: Use \hb@xt@ . . . . .	343
1995/05/07 lttab.dtx v1.0l	
General: Use \hb@xt@ . . . . .	296
1995/05/08 ltbibl.dtx v1.1g	
\@citex: Use \@firstofone . . . .	376
\bibitem: Removed unnecessary	
braces . . . . .	376
\nocite: Use \@firstofone . . . .	377

- 1995/05/08 ltdefs.dtx v1.2k  
`\typein`: Use `\@firstofone` . . . . . 36
- 1995/05/08 ltdefs.dtx v1.2l  
`\typein`: Remove unnecessary braces . . . . . 36  
Replace `\def` by `\let` . . . . . 36
- 1995/05/08 ltfssstrc.dtx v2.3n  
`\ifnot@nil`: Use `\@firstofone` . 171
- 1995/05/11 fontdef.dtx v2.2j  
General: Updates to some plain macros . . . . . 214
- 1995/05/12 ltclass.dtx v1.0n  
`\DeclareOption*`: Use `\toks@` to remove need to double hash /1557 . . . . . 458
- 1995/05/12 ltfloat.dtx v1.1h  
`\@footnotemark`: Add `\nobreak` to allow hyphenation. latex/1605 371
- 1995/05/12 ltpictur.dtx v1.0h  
`\pictur@`: Macro added for latex/1355 . . . . . 319
- 1995/05/12 ltvers.dtx v1.0e  
General: Add autoload docstrip guards . . . . . 32  
Check for format older than 1 year . . . . . 32
- 1995/05/13 ltfssstrc.dtx v2.3o  
General: Use single hash mark in `\DeclareOption` . . . . . 160
- 1995/05/16 ltfloat.dtx v1.1i  
`\@makefnmark`: Now use `\textsuperscript`. . . . . 369  
`\textsuperscript`: Command added./pr1503 . . . . . 370  
`\thefootnote`: Streamlined parts of code. . . . . 369
- 1995/05/17 ltboxes.dtx v1.0u  
`\@irsbox`: Removed surplus braces 294
- 1995/05/17 ltclass.dtx v1.0o  
`\g@addto@macro`: Make long for latex/1522 . . . . . 465
- 1995/05/17 ltlists.dtx v1.0g  
`\@item`: Removed surplus braces . 281  
`\@nbitem`: Removed surplus braces 281  
`enumerate`: Use `\thr@@` and remove surplus braces . . . . . 282  
`itemize`: Use `\thr@@` . . . . . 283
- 1995/05/18 ltfloat.dtx v1.1j  
`\@makefnmark`: Added `\normalfont`. . . . . 369  
`\thempfootnote`: Added `\itshape`. . . . . 369
- 1995/05/19 ltpictur.dtx v1.1a  
General: Support autoloading feature . . . . . 318
- 1995/05/20 ltcounts.dtx v1.1b  
`\@definecounter`: Streamlined code . . . . . 134  
`\@fnsymbol`: Allowing both text and math . . . . . 135  
`\fnsymbol`: Streamlined code . . . 134
- 1995/05/20 ltcounts.dtx v1.1c  
`\@definecounter`: And do it right 134
- 1995/05/20 ltfloat.dtx v1.1k  
`\@makefnmark`: Moved `\normalfont` back and use `\@textsuperscript` . . . . . 369  
Moved `\normalfont` to `\textsuperscript` . . . . . 369  
`\textsuperscript`: Use `\normalfont`. . . . . 370
- 1995/05/21 ltfssdcl.dtx v2.1t  
`\DeclareMathRadical`: Allow for undefined cs names . . . . . 204
- 1995/05/21 ltlists.dtx v1.0f  
General: Moved to doc.sty standard . . . . . 269
- 1995/05/21 ltmath.dtx v1.0r  
`\@sqrt`: Use `\sqrtsign` . . . . . 264  
General: Remove `\mathhexbox` from this file . . . . . 261  
Update some plain macros . . . 258  
`\lefteqn`: Use `\rlap` . . . . . 266  
`\r@t`: Use `\sqrtsign` instead of `\sqrt` . . . . . 260
- 1995/05/21 ltoutenc.dtx v1.7h  
`\@inmathwarn`: Added several `\@onlypreamble` . . . . . 95
- 1995/05/21 ltoutenc.dtx v1.7j  
General: Updated some plain macros . . . . . 105
- 1995/05/21 ltplain.dtx v1.1j  
General: Moved some code to other files . . . . . 14
- 1995/05/22 ltplain.dtx v1.1k  
General: Definitions of `\footins` and `\footnoterule` moved to ltfloat. . . . . 29
- 1995/05/22 lttab.dtx v1.1a  
General: Support autoloading feature . . . . . 296
- 1995/05/23 ltfssini.dtx v2.2e  
`\newfont`: Font assignment made local again. . . . . 210

1995/05/24 ltdefs.dtx v1.1l		<code>\InputIfFileExists:</code> (CAR)	
<code>\newif:</code> (DPC) New implementation	41	added <code>\long</code>	88
1995/05/24 ltdefs.dtx v1.2m		<code>\nofiles:</code> (CAR) added <code>\long</code>	85
<code>\typein:</code> (DPC) New implementation	36	<code>\protected@write:</code> (CAR) added <code>\long</code>	85
1995/05/24 ltfloat.dtx v1.1l		1995/05/25 ltfloat.dtx v1.1m	
<code>\@textsuperscript:</code> Command added.	370	<code>\@savemarbox:</code> (CAR) Resettings moved to hook	365
General: Moved definition of <code>\footins</code> and <code>\footnoterule</code> from <code>ltplain</code> .	369	<code>\@xfloat:</code> (CAR) Resettings moved to hook	358
<code>\textsuperscript:</code> Use <code>\@textsuperscript</code>	370	1995/05/25 ltlists.dtx v1.0i	
1995/05/24 ltfsbas.dtx v3.0a		<code>\endtrivlist:</code> Macros moved from <code>ltspace.dtx</code>	278
General: (DPC) Make file from previous file, <code>fam.dtx</code> 1995/05/20 v2.2d	139	1995/05/25 ltmath.dtx v1.3c classes	
<code>\mathgroup:</code> (DPC) No need to redefine <code>\newfam</code> as not outer	139	<code>\@eqnnum:</code> replace <code>\reset@font\rmfamily</code> with <code>\normalfont</code> (PR 1578)	266
1995/05/24 ltfscomp.dtx v3.0a		1995/05/25 ltspc.dtx v1.2f	
General: (DPC) Make file from previous file, <code>fam.dtx</code> 1995/05/20 v2.2d	181	<code>\@vbsphack:</code> (CAR) not used so 'removed'.	73
1995/05/24 ltfsdcl.dtx v3.0a		<code>\@vspacer:</code> (CAR) <code>\@restorepar</code> added to avoid possible infinite tail recursion caused by a typo in the argument.	76
General: (DPC) Make file from previous file, <code>latint.dtx</code> 1995/05/21 v2.1t	185	(CAR) macros modified to be more efficient	76
1995/05/24 ltfsini.dtx v3.0a		General: Macros moved to <code>ltlists.dtx</code>	66
General: (DPC) Make file from previous file, <code>lfont.dtx</code> 1995/05/23 v2.2e	208	1995/05/26 ltdefs.dtx v1.2n	
<code>\cal:</code> (DPC) Remove definition	213	<code>\@gobblefour:</code> (CAR) Added <code>\longs</code>	42
<code>\mit:</code> (DPC) Remove definition	213	1995/05/26 ltmath.dtx v1.0s	
1995/05/24 ltfsstrc.dtx v3.0a		<code>\@eqnnum:</code> Removed <code>\rmfamily</code> (PR 1578), replaced <code>\reset@font</code> with <code>\normalfont</code>	264
General: (DPC) Make file from previous file, <code>tracefnt</code> 1995/05/16 v2.3o	159	1995/05/26 ltpage.dtx v1.0g	
1995/05/24 ltfsstrc.dtx v3.0b		<code>\ps@plain:</code> removed <code>\rmfamily</code> (PR 1578)	380
General: (DPC) Fix <code>\ProvidesFile</code> usage	159	1995/05/27 ltfsbas.dtx v3.0b	
1995/05/25 ltclass.dtx v1.0p		<code>\mathgroup:</code> (FMI) But a need to define <code>\new@mathgroup</code>	139
<code>\endfilecontents:</code> Delete <code>\filecontents</code> after preamble	466	1995/06/05 fontdef.dtx v2.2k	
1995/05/25 ltfiles.dtx v1.0s		General: Moved math commands from <code>ltoutenc.dtx</code> .	229
<code>\document:</code> Added check for <code>\topskip</code> zero	84	1995/06/05 ltfinal.dtx v1.0r	
1995/05/25 ltfiles.dtx v1.0t		General: Added <code>\MakeUppercase</code> and <code>\MakeLowercase</code> .	494
<code>\iffilenameonpath:</code> (CAR) added <code>\long</code>	87	1995/06/05 ltoutenc.dtx v1.7k	
<code>\document:</code> Corrected typo	84	<code>\@inmathwarn:</code> Removed <code>\protected@cmd</code> and replaced with explicit <code>\noexpand</code> .	95
<code>\IfFileExists:</code> (CAR) added <code>\long</code>	87	General: Allowed <code>\ProvideTextCommandDefault</code> after the preamble.	97

Commented out <code>\textless</code> and <code>\textgreater</code> . . . . .	103	1995/06/28 ltmath.dtx v1.0t	
Moved math commands to fontdef.dtx. . . . .	105	General: minor doc edits . . . . .	258
Save some tokens in <code>\textvisible</code> and <code>\textunderscore</code> . . . . .	103	1995/07/02 ltplain.dtx v1.1n	
1995/06/06 ltfinal.dtx v1.0s		General: Removed surplus ‘by’ and ‘=’ in various places . . . . .	14
General: Made <code>\MakeUppercase</code> and <code>\MakeLowercase</code> brace their argument. . . . .	494	<code>\offinterlineskip</code> : Replaced 1000 by <code>\@m</code> . . . . .	27
1995/06/09 ltoutenc.dtx v1.7l		<code>\showoutput</code> : Use <code>\showoverfull</code> to save space . . . . .	29
<code>\DeclareTextComposite</code> : Rewrote <code>\DeclareTextComposite</code> to define the composite as a no-argument command rather than a two-argument command. . . . .	98	<code>\tracingall</code> : Use <code>\showoutput</code> to save space . . . . .	29
1995/06/11 ltspc.dtx v1.2g		1995/07/03 ltdefns.dtx v1.2o	
<code>\restorecr</code> : (CAR) <code>\relax</code> added to stop silent eating of *. . . . .	79	<code>\set@typeset@protect</code> : Use <code>\@typeset@protect</code> for init . . . . .	45
1995/06/13 ltfinal.dtx v1.0t		1995/07/03 ltfntcmd.dtx v3.3s	
General: Add patch level string more carefully . . . . .	502	<code>\t@st@ic</code> : Use clean interface for jump . . . . .	241
Call <code>\errorstopmode</code> . . . . .	503	1995/07/05 ltfntcmd.dtx v3.3s	
1995/06/13 ltpictur.dtx v1.1b		<code>\t@st@ic</code> : Renamed from <code>\test@next</code> . . . . .	241
General: Use <code>\ProvidesFile</code> in autoload . . . . .	318	1995/07/05 ltspc.dtx v1.2h	
1995/06/14 lttab.dtx v1.1b		<code>\@gnewline</code> : Use <code>\break</code> . . . . .	70
General: Use <code>\ProvidesFile</code> in autoload . . . . .	296	<code>\@no@pgbk</code> : Macro replaces <code>\@pgbk</code> and <code>\@nopgbk</code> . . . . .	69
1995/06/15 ltfsbas.dtx v3.0c		<code>\nopagebreak</code> : Reimplemented both using <code>\@no@pgbk</code> . . . . .	69
General: (DPC) minor documentation changes . . . . .	139	1995/07/09 ltctrl.dtx v1.0f	
1995/06/15 ltfscomp.dtx v3.0b		<code>\@iforloop</code> : Reimplemented using Kabelschacht method . . . . .	53
General: (DPC) minor documentation edits . . . . .	181	<code>\@iwhiledim</code> : Reimplemented using Kabelschacht method . . . . .	52
1995/06/15 ltssdcl.dtx v3.0b		<code>\@iwhilenum</code> : Reimplemented using Kabelschacht method . . . . .	52
General: (DPC) minor documentation changes . . . . .	185	<code>\@iwhiles</code> : Reimplemented using Kabelschacht method . . . . .	52
1995/06/19 ltbibl.dtx v1.1h		<code>\@tfor</code> : Reimplemented using Kabelschacht method . . . . .	54
<code>\biblecite</code> : Call <code>\@newl@bel</code> so repeated keys produce better warning. . . . .	376	1995/07/09 ltlists.dtx v1.0j	
1995/06/19 ltclass.dtx v1.0q		<code>enumerate</code> : Use <code>\expandafter</code> . . . . .	282
<code>\documentclass</code> : Dont redefine <code>\usepackage</code> in compat mode for /1634 . . . . .	461	<code>itemize</code> : Use <code>\expandafter</code> . . . . .	283
1995/06/19 ltxref.dtx v1.1e		1995/07/12 ltpictur.dtx v1.1d	
<code>\newlabel</code> : Use <code>\@newl@bel</code> to share code with <code>\biblecite</code> . . . . .	246	General: allow 2e commands in 209 mode. latex/1737 . . . . .	318
1995/06/28 ltfsini.dtx v3.0b		1995/07/13 ltdefns.dtx v1.0p	
General: (DPC) Fix documentation typos . . . . .	208	General: Updates to documentation . . . . .	34
		1995/07/13 ltfiles.dtx v1.0u	
		General: Updates to docu . . . . .	81
		1995/07/13 ltfsbas.dtx v3.0d	
		<code>\@defaults</code> : macro added . . . . .	154
		<code>\@defaults</code> : macro added . . . . .	154
		General: minor documentation changes . . . . .	139

<code>\wrong@fontshape:</code> Change a macro not a switch to flag default font substitutions . . . .	153	<code>set \do globally</code> . . . . .	84
1995/07/13 ltmiscen.dtx v1.0z		<code>set \topskip globally</code> . . . . .	84
<code>\@centercr:</code> Use <code>\nobreak</code> . . . .	254	1995/08/24 ltfssbas.dtx v3.0f	
<code>\@writefile:</code> Added missing percent and use <code>\relax</code> in the THEN case . . . . .	251	General: Added autoload code . .	139
<code>\@xobeysp:</code> Use <code>\nobreak</code> . . . .	255	1995/08/24 ltfssstrc.dtx v3.0c	
General: Improve Documentation	249	General: Macro <code>\gobble@font@spec</code> removed . . . . .	171
<code>\enddocument:</code> Set <code>\@setckpt</code> to <code>\@gobbletwo</code> instead of defining it by hand . . . . .	250	<code>\tryis@simple:</code> . . . . .	178
Shorten redefinition of <code>\biblecite</code> and <code>\newlabel</code> . . . . .	250	1995/08/25 ltoutput.dtx v1.1p	
Use <code>\@defaultsubs</code> instead of switch . . . . .	251	General: Support autoloading feature (FMi). . . . .	382
1995/07/14 ltbibl.dtx v1.1i		1995/09/01 lterror.dtx v1.2i	
<code>\biblecite:</code> Remove <code>\@onlypreamble</code> so still defined in new <code>\enddocument</code> . . . . .	376	General: Add autoload support . .	55
1995/07/14 ltxref.dtx v1.1g		1995/09/01 ltplain.dtx v1.1m	
<code>\newlabel:</code> Remove <code>\@onlypreamble</code> so still defined in new <code>\enddocument</code> . . . . .	246	<code>\empty:</code> Use <code>\let</code> to save space . .	26
1995/07/19 ltfssini.dtx v3.0d		<code>\I:</code> Use <code>\let</code> to save space . . . .	25
General: (DPC) TeX2 support . .	212	1995/09/14 ltplain.dtx v1.1o	
1995/07/20 ltboxes.dtx v1.0v		General: Moved <code>\multispan</code> to <code>lttab.dtx</code> . . . . .	14
<code>\@isavebox:</code> Use <code>\sbox</code> . . . . .	288	1995/09/14 lttab.dtx v1.1c	
<code>\@isavepicbox:</code> Use <code>\sbox</code> . . . .	288	<code>\cline:</code> (DPC) New implementation . . . . .	316
1995/07/21 ltoutput.dtx v1.1o		1995/09/15 ltfssini.dtx v3.0e	
<code>\@writesetup:</code> Command added .	408	General: (DPC) Modify TeX2 message . . . . .	212
New, experimental, versions: need in-lining . . . . .	408	1995/09/19 ltmiscen.dtx v1.1a	
1995/08/09 ltmath.dtx v1.0u		<code>\verb:</code> Put <code>\@noligs</code> after <code>\verbatim@font</code> where it belongs. . .	256
General: Added code for class options <code>leqno</code> and <code>fleqn</code> . . . . .	266	1995/10/01 ltfiles.dtx LaTeX2e	
1995/08/11 ltlength.dtx v1.1b		<code>\@addtofilelist:</code> Macro added . .	89
General: Doc typos fixed for latex/753 . . . . .	138	1995/10/02 ltdefs.dtx v1.2q	
1995/08/16 ltcntrl.dtx v1.0g		<code>\@ifnch:</code> Use <code>\@let@token</code> for internal/924, save <code>\reserved@e</code> .	47
<code>\@break@tfor:</code> Made long . . . . .	54	<code>\@ifnextchar:</code> Use <code>\@let@token</code> .	46
<code>\@forloop:</code> Made defs long . . . . .	53	<code>\@protected@testopt:</code> Macro added . . . . .	38
<code>\@fornoop:</code> Made defs long . . . . .	53	<code>\@testopt:</code> Macro added . . . . .	38
<code>\@iforloop:</code> Made defs long . . . .	53	<code>\@xargdef:</code> New implementation, using <code>\@test@opt</code> . . . . .	37
<code>\@iwhiledim:</code> Made defs long . . . .	52	1995/10/02 ltplain.dtx v1.1p	
Removed <code>\@whilenoop</code> . . . . .	52	General: Move <code>\newif</code> to <code>ltdefs</code> . .	22
<code>\@iwhilenum:</code> Made defs long . . . .	52	1995/10/03 fontdef.dtx v2.2l	
Removed <code>\@whilenoop</code> . . . . .	52	General: <code>\@@sqrt</code> from patch file for /1701 . . . . .	214
<code>\@iwhilesw:</code> Removed <code>\@whileswnoop</code> . . . . .	52	1995/10/03 ltdefs.dtx v1.2r	
<code>\@tfor:</code> Made defs long . . . . .	54	<code>\typein:</code> Add missing <code>\@typein</code> for /1710 (from patch file) . . . . .	36
1995/08/16 ltfiles.dtx v1.0v		1995/10/03 ltpictur.dtx v1.1e	
<code>\document:</code> set <code>\@maxdepth</code> . . . .	84	General: New autoload code . . . .	318
		1995/10/04 ltfssbas.dtx v3.0g	
		General: Modify autoload code . .	139
		1995/10/04 ltfssstrc.dtx v3.0d	
		General: (DPC) Modify autoload code . . . . .	159

1995/10/04 lttab.dtx v1.1d	1995/10/16 ltthm.dtx v1.0g
General: Modify autoload support 296	General: Revert to previous
1995/10/06 ltfiles.dtx v1.0w	\newtheorem behaviour . . . . 339
\@missingfileerror: Autoload error . . . . . 88	1995/10/17 ltclass.dtx v1.0r
1995/10/09 lterror.dtx v1.2j	\@providesfile: Delay definition of \ProvidesFile till ltfinal 458
General: Modify autoload support 55	\ProcessOptions*: Reset \CurrentOption for graphics/1873 . . 460
1995/10/09 ltoutenc.dtx v1.7m	1995/10/17 ltdirchk.dtx v1.0l
\@inmathwarn: Autoload error . . 96	General: Modify initex version of \ProvidesFile . . . . . 4
1995/10/10 ltssbas.dtx v3.0h	1995/10/17 ltfinal.dtx v1.0v
\showhyphens: Use \normalfont and make colour safe, and autoloadable . . . . . 157	\@providesfile: reset macro . . . 502
1995/10/10 ltssdcl.dtx v3.0c	\reserved@b: reset here after the \input above . . . . . 502
\non@alpherr: (DPC) autoload error message . . . . . 189	1995/10/17 ltplain.dtx v1.1s
1995/10/10 ltplain.dtx v1.1r	\eject: Move \supereject to compat file . . . . . 27
General: Autoload tracing code . 14	1995/10/17 lttab.dtx v1.1e
1995/10/10 ltthm.dtx v1.0f	\@cline: (DPC) Use \@multicnt 316
General: Make \newtheorem ‘only preamble’ . . . . . 339	\@multispan: (DPC) Macro added. . . . . 316
1995/10/11 ltoutput.dtx v1.1r	1995/10/19 ltfinal.dtx v1.0w
\clearpage: Added a check so that it does not lose the argument of \twocolumn[...] . . . . . 396	\@filelist: Move after \reserved@a setting:-) . . . . . 503
1995/10/16 ltbibl.dtx v1.1j	1995/10/20 ltbibl.dtx v1.1k
\cite: (DPC) Make robust . . . . 376	\@citex: Removed refundefined flag . . . . . 376
1995/10/16 ltboxes.dtx v1.0w	\nocite: Removed refundefined flag . . . . . 377
General: Clarify makebox description . . . . . 285	1995/10/20 ltclass.dtx v1.0s
1995/10/16 ltdefs.dtx v1.2u	\@begindocumenthook: Make setting conditional, for autoload version . . . . . 465
\@ifstar: (DPC) New implementation, for /1910 . . . . . 47	1995/10/20 ltssbas.dtx v3.0i
\new@command: (DPC) Use \@testopt /1911 . . . . . 37	General: (DPC) Modify autoload code, change \undefined . . . 139
\new@environment: (DPC) Use \@testopt /1911 . . . . . 40	1995/10/20 ltfsstrc.dtx v3.0e
\typein: (DPC) Use \@testopt /1911 . . . . . 36	General: (DPC) Modify autoload code . . . . . 159
1995/10/16 ltssini.dtx v3.0f	1995/10/22 ltssbas.dtx v3.0j
\p@reset@font: Added \relax after \usefont, as the latter eats up spaces. . . . . 211	General: (RmS) New size function macro \genb@sfcnt needs to be disabled at \document. . . . 139
1995/10/16 ltmath.dtx v1.0y	1995/10/22 ltfsstrc.dtx v3.0f
\@yeqncr: (DPC) Use \@testopt /1911 . . . . . 265	General: Added ‘genb’ and ‘sgenb’ size functions to support new DC font naming scheme. . . . 159
\sqrt: (DPC) Make robust /1808 264	1995/10/23 lttab.dtx v1.1f
1995/10/16 ltspc.dtx v1.2j	\@settab: (CAR)Ensure that \@hightab increases by at most one . . . . . 303
\nolinebreak: (DPC) Use \@testopt /1911 . . . . . 69	
\nopagebreak: (DPC) Use \@testopt /1911 . . . . . 69	

- `\@startline`: (CAR)Ensure that `\@nxttabmar` is never larger than `\@hightab` ..... 301
- `\poptabs`: (CAR)Ensure that `\@curtab` is never larger than `\@hightab` ..... 304
- `\tabbing`: (CAR)Make `\@hightab` consistently a local variable . 302
- 1995/10/24 `ltfiles.dtx` v1.1a
  - `\document`: Removed multiplelabels switch ..... 83
  - Removed refundefined switch . 84
- 1995/10/24 `ltfssbas.dtx` v3.0k
  - `\@@defaultsubs`: macro removed 154
  - `\wrong@fontshape`: Make this code inline since it happens only here ..... 153
- 1995/10/24 `ltmiscen.dtx` v1.1b
  - `\enddocument`: Changed logic for producing warning messages and removed switch ..... 251
  - Use `\@refundundefined` instead of switch ..... 251
- 1995/10/24 `ltxref.dtx` v1.1h
  - `\@multiplelabels`: Switch for multiplelabels removed ..... 247
  - `\@newl@bel`: Switch for multiplelabels replaced by inline code . 246
  - `\@refundundefined`: Switch for refundundefined replaced ..... 246
  - `\@setref`: Switch for refundundefined renamed ..... 246
  - `\if@multiplelabels`: Macro removed ..... 247
- 1995/10/25 `ltalloc.dtx` v1.1b
  - General: General doc improvements ..... 49
- 1995/10/25 `ltfloat.dtx` v1.1n
  - `\@endfloatbox`: (CAR) macro added: to unify code for double and single versions ..... 362
  - `\end@dblfloat`: (CAR) unify code for double and single versions 361
  - `\end@float`: (CAR) unify code for double and single versions .. 360
- 1995/10/25 `ltidxglo.dtx` v1.1d
  - General: Doc cleanup ..... 373
- 1995/10/25 `ltsect.dtx` v1.0q
  - `\subparagraphmark`: Use `\let` not `\def` to save space. .... 350
- 1995/10/27 `ltpictur.dtx` v1.1f
  - General: Move initialisation to kernel from autoload file ..... 336
- 1995/10/31 `ltboxes.dtx` v1.0x
  - `\@finalstrut`: Add `\nobreak` in horiz mode to allow hyphenation. internal/1931 ..... 295
- 1995/11/01 `fontdef.dtx` v2.2m
  - General: add `\nfss@catcodes` for internal/1932 ..... 217
- 1995/11/01 `ltdirchk.dtx` v1.0n
  - General: Initialise `\@ad-dtofilelist` to `\@gobble` .... 4
- 1995/11/01 `ltfinal.dtx` v1.0x
  - General: (DPC) Switch meaning of `\@addtofilelist` for cfg files 497
- 1995/11/01 `ltfssbas.dtx` v3.0m
  - `\DeclareFontShape`: (DPC) Test for `\relax` not `\undefined`, internal/1933 ..... 140
- 1995/11/01 `ltfssini.dtx` v3.0g
  - General: (DPC) Switch meaning of `\@addtofilelist` for cfg files 212
- 1995/11/02 `ltfssbas.dtx` v3.0n
  - `\wrong@fontshape`: (DPC) Remove extra space with `\string` for latex/1676 ..... 152
- 1995/11/02 `ltoutenc.dtx` v1.7n
  - General: Changed internal name `\a` to `\@tabacckludge` to protect against redefinition by malicious users. .... 102
- 1995/11/07 `ltlists.dtx` v1.0k
  - `\@doendpe`: Enclosed `\setbox0` assignment by a group so that it leaves the contents of box 0 intact. .... 279
- 1995/11/07 `ltoutenc.dtx` v1.7o
  - General: Added `\leavevmode` at start of `\c`, otherwise the output routine might be invoked within the macro. .... 105
  - Changed `\char32` to `\@xxxii` (two tokens less). .... 106
  - Replaced octal number 27 by decimal number 23 to protect against the quote character being active. .... 106
  - Replaced some 0's by `\z@` (faster). .... 106
- 1995/11/10 `ltoutput.dtx` v1.1s
  - `\@shipoutsetup`: Command removed ..... 408
  - `\@writesetup`: Command removed 408
  - In-lined ..... 408

1995/11/14 ltclass.dtx v1.0t	1995/12/04 ltspac.dtx v1.2k
\@unprocessedoptions: Allow	\nobreakspace: (Macro added .. 78
empty option ..... 466	1995/12/04 ltspac.dtx v1.2l
\@loadwithoptions: macro added 461	\@xobeysp: (braces added to defini-
\LoadClassWithOptions: macro	tion of tilde ..... 78
added ..... 461	1995/12/04 preload.dtx v2.4e
\RequirePackageWithOptions:	General: Ulrik Vieth. added 12pt
macro added ..... 461	OMS and OML preloads /1989 234
1995/11/17 ltfsbas.dtx v3.0m	1995/12/05 ltdefns.dtx 1.2w
\@wrong@font@char: (DPC) Macro	\@unexpandable@noexpand: Re-
added. latex/1676 ..... 154	moved as never used. inter-
\define@newfont: Redefine \type-	nal/1733 ..... 43
out latex/1676 ..... 149	1995/12/05 ltfiles.dtx v1.1c
\wrong@fontshape: Support	\document: \ignorespaces added
\@wrong@font@char latex/1676	for latex/1933 ..... 85
..... 152	1995/12/05 ltfloat.dtx v1.1n
1995/11/17 ltoutenc.dtx v1.7p	\@textsuperscript: Use \ensure-
\UseTextSymbol: Support	math for latex/1984. .... 370
\@wrong@font@char latex/1676 99	1995/12/05 ltoutenc.dtx v1.7v
1995/11/18 ltoutenc.dtx v1.7q	\@inmathwarn: Changed \TextSym-
\UseTextSymbol: Modify message	bolUnavailable text ..... 96
slightly ..... 99	1995/12/06 ltfsbas.dtx v3.00
1995/11/21 fontdef.dtx v2.2n	\nfss@catcodes: Reset hat, for
General: Incorporate changed fig-	typeouts etc in fd files .... 151
ures, as in plain.tex ..... 228	1995/12/07 ltbibl.dtx v1.1l
1995/11/27 ltfsbas.dtx v3.0n	\@citex: Restored name of
\nfss@catcodes: Reset hash, for	\G@refundefinedtrue ..... 376
definitions in fd files ..... 151	1995/12/07 ltfloat.dtx v1.1m
1995/11/28 ltfloat.dtx v1.1n	\@textsuperscript: Move \m@th
General: documentation fixes ... 353	out of the \ensuremath for la-
1995/11/28 ltfstrc.dtx v3.0g	tex/1984. .... 370
General: documentation fixes ... 159	1995/12/07 ltxref.dtx v1.1i
1995/11/28 ltoutenc.dtx v1.7r	\@setref: Switch for refundefined
General: Added math mode checks	restored ..... 246
to text commands. .... 95	\G@refundefinedtrue: Renamed
doc fixes ..... 91	(back) from \G@refundefined 246
Renamed \@changed@x@err to	1995/12/11 ltoutenc.dtx v1.7w
\TextSymbolUnavailable. ... 95	General: Modified \copyright .. 103
1995/11/29 ltoutenc.dtx v1.7t	1995/12/13 ltdefns.dtx 1.2x
General: Added \textasci-	\:- Documentation changed. ... 34
icircum, \textasciitilde,	1996/01/10 ltfiles.dtx v1.1d
\textbackslash, \textbar,	\@iffilenamepath: Change argument
\textgreater and \textless. 107	handling to not require doubled
Added \textasciicircum,	hash. latex/2024 ..... 87
\textasciitilde, \textreg-	1996/01/20 ltidxglo.dtx v1.1e
istered and \texttrademark. 103	\makeglossary: Make no-op after
Added \textbackslash and	use pr/2048 ..... 374
\textbar. .... 103, 111	\makeindex: Make no-op after use
Added \textless and	pr/2048 ..... 374
\textgreater. .... 103, 112	1996/01/20 ltspac.dtx v1.2m
1995/12/01 ltoutenc.dtx v1.7u	\vspace: Made robust ..... 76
General: Made \SS a Default,	1996/03/25 ltmath.dtx v1.1a
rather than having the default	\@ensuredmath: Macro added for
point to the OT1 definition. 103	amslatex/2104 ..... 266

<code>\ensuremath</code> : Reimplement for amslatex/2104 . . . . .	266	1996/05/23 ltoutenc.dtx v1.7z	
1996/04/18 ltpage.dtx v1.0i		<code>\@strip@args</code> : <code>\expandafter</code>	
General: Improve documentation	379	added to match other changes	
1996/04/22 ltmiscen.dtx v1.1c		for latex/2133 . . . . .	99
General: Improve Documentation	249	<code>\add@accent</code> : macro added. la-	
1996/04/22 ltspace.dtx v1.2n		tex/2133 . . . . .	97
General: Documentation Improve-		<code>\DeclareTextAccent</code> : Reimple-	
ments . . . . .	66	mented using <code>\add@accent</code> to	
1996/04/22 lttab.dtx v1.1g		save space latex/2133 . . . . .	97
<code>\@tabclassz</code> : (DPC) Extra <code>\hskip</code>		<code>\DeclareTextCompositeCommand</code> :	
keeps <code>tabcolsep</code> in empty		Modified to cope with new	
columns internal/2122 . . . . .	314	<code>\add@accent</code> command: re-	
1996/04/23 ltcounts.dtx v1.1d		quired removal of check for one	
General: Documentation improve-		argument-command . . . . .	98
ments . . . . .	132	1996/05/24 ltoutput.dtx v1.1t	
1996/04/24 ltfiles.dtx v1.1e		<code>\@specialoutput</code> : Check that	
<code>\document</code> : (DPC) Reset <code>\AtBe-</code>		<code>\@colroom</code> is less than <code>\vsize</code> ,	
<code>ginDocument</code> eg for latex/1297	84	indicating that a float has been	
1996/05/08 ltfstrc.dtx v3.0h		added . . . . .	399
<code>\math@egroup</code> : Use <code>\bgroup</code> instead		Cut-off point changed to	
of <code>\begingroup</code> to match a ker-		<code>1.5\baselineskip</code> . . . . .	399
nel change made in 1994!! . .	169	<code>\@topnewpage</code> : Cut-off point	
1996/05/09 ltfncmd.dtx v3.3t		changed to <code>2.5\baselineskip</code>	398
<code>\check@icr</code> : Default definitions		1996/05/25 ltoutput.dtx v1.1u	
added . . . . .	239	<code>\@specialoutput</code> : Correct the	
1996/05/17 fontdef.dtx v2.2o		above check . . . . .	399
General: <code>\@@sqrt</code> removed, at last		1996/06/03 ltmiscen.dtx v1.1d	
. . . . .	214, 227	<code>\@verbatim</code> : Exchanged the fol-	
1996/05/17 ltfiles.dtx v1.1f		lowing two code lines so that	
<code>\nofiles</code> : added <code>\write</code> to <code>\pro-</code>		<code>\dospecials</code> cannot reset the	
<code>tected@write</code> for latex/2146 .	85	category code of characters han-	
1996/05/18 ltoutenc.dtx v1.7x		dled by <code>\@noligs</code> . . . . .	255
General: Produce error if encoding		General: Move setting of verbatim	
not found. pr/2054 . . . . .	118	font and <code>\@noligs</code> . . . . .	249
1996/05/21 ltoutenc.dtx v1.7y		<code>\verb</code> : Put setting of verbatim	
General: Corrected error message		font after <code>\dospecials</code> so that	
(CAR) . . . . .	118	<code>\dospecials</code> cannot reset the	
1996/05/21 ltsect.dtx v1.0s		category code of characters han-	
<code>\@sect</code> : (DPC) Added extra braces		dled by <code>\@noligs</code> . . . . .	256
for internal/2148 . . . . .	347	1996/06/10 ltboxes.dtx v1.0y	
(DPC) Moved brace to allow		<code>\@parboxto</code> : (DPC) Changed <code>\end-</code>	
commands like <code>\MakeUppercase</code>		<code>graf</code> to <code>\@@par</code> . . . . .	291
in 6th argument. Changed <code>\par</code>		1996/06/10 ltsect.dtx v1.0t	
to <code>\endgraf</code> to allow non-long		<code>\@sect</code> : (DPC) Changed <code>\endgraf</code>	
commands. internal/2148 . .	346	to <code>\@@par</code> . . . . .	346
<code>\@ssect</code> : (DPC) Added extra		<code>\@ssect</code> : (DPC) Changed <code>\endgraf</code>	
braces for internal/2148 . . .	349	to <code>\@@par</code> . . . . .	349
(DPC) Moved brace to allow		1996/06/13 ltdirchk.dtx v1.0r	
commands like <code>\MakeUppercase</code>		General: documentation improve-	
in 4th argument. Changed <code>\par</code>		ments mainly from inter-	
to <code>\endgraf</code> to allow non-long		nal/2174 . . . . .	1
commands. internal/2148 . .	349	1996/06/14 lttab.dtx v1.1h	
		<code>\@tabclassz</code> : (DPC) Change	

both\z@skip to lsp for latex/2160 . . . . .	314	Remove unnecessary \global before \@nobreak... . . . .	281
1996/06/22 ltspac.dtx v1.2o		1996/07/26 ltmath.dtx v1.1b	
General: Documentation of problems added . . . . .	66	General: Removed \global before \@ignoretrue in various places. . . . .	258
1996/07/10 ltfinal.dtx v1.0y		1996/07/26 ltmiscen.dtx v1.1e	
\toks: Free up memory from scratch registers /2213 . . . .	502	\@ignorefalse: put \global into definition . . . . .	250
1996/07/19 ltoutenc.dtx v1.8a		\begin: remove \global before \@ignore... . . . .	253
\@strip@args: Use char 0 not @ as carrier for \lowercase /2197 .	99	\end: remove \global before \@ignore... . . . .	253
1996/07/26 ltboxes.dtx v1.0z		\ignorespacesafterend: user level macro added . . . . .	250
\if@minipage: put \global into definition . . . . .	292	1996/07/26 ltoutput.dtx v1.1v	
1996/07/26 ltclass.dtx v1.0u		\@testfp: remove \global before \@test... . . . .	440
\@classoptionslist: made only preamble . . . . .	455	\@xtryfc: remove \global before \@test... . . . .	414
\@unusedoptionlist: made only preamble . . . . .	455	\@ztryfc: remove \global before \@test... . . . .	416
1996/07/26 ltdefs.dtx v1.2y		\clearpage: add number of missing percents . . . . .	396
\@reargdef: third arg picked up by \@yargdef . . . . .	39	1996/07/26 ltplain.dtx v1.1t	
\renew@command: use \noexpand instead of \string . . . . .	39	\sh@ft: replace \dimen\z@ by \dimen@ . . . . .	28
use \relax in place of empty arg	39	1996/07/26 ltsect.dtx v1.0u	
\renew@environment: use \relax in place of empty arg . . . . .	40	\@starttoc: removed \global before \@nobreak... . . . .	351
1996/07/26 ltfloat.dtx v1.1n		\@xsect: Removed \global before \@nobreak... . . . .	348
\@endfloatbox: remove unnecessary \global before \@minipage... .	362	1996/07/26 ltspac.dtx v1.2p	
\@savemarbox: remove unnecessary \global before \@minipage... .	365	\if@nobreak: put \global inside definition . . . . .	71
\@setminipage: remove unnecessary \global before \@minipage... .	360	1996/07/27 ltssbas.dtx v3.0q	
\@setnobreak: remove unnecessary \global before \@nobreak... .	360	General: \if@inmath switch removed . . . . .	148
1996/07/26 ltssbas.dtx v3.0p		1996/07/27 ltspac.dtx v1.2q	
\@DeclareMathSizes: use faster \if test . . . . .	144	General: Further documentation of problems . . . . .	66
\nfss@catcodes: omit \relax as not needed . . . . .	151	1996/07/27 ltspac.dtx v1.2r	
1996/07/26 ltssdcl.dtx v3.0e		General: Correct documentation of problems . . . . .	66
\init@restore@version: Removed \ifrestore@version switch and replaced by \init@restore@version . . .	189	1996/08/02 ltfloat.dtx v1.1o	
1996/07/26 ltssstrc.dtx v3.0i		\@xympar: Remove \global before \@ignore... . . . .	366
\init@restore@glb@settings: macro added replacing \if@inmath switch . . . . .	168	1996/08/02 ltsect.dtx v1.0v	
1996/07/26 ltlists.dtx v1.0l		\@afterheading: Removed \global before \@nobreak... .	349
\@item: Remove unnecessary \global before \@minipage... .	280	1996/08/02 ltspac.dtx v1.2s	
		\@Esphack: Remove \global before \@ignore... . . . .	73

- 1996/08/25 ltssbas.dtx v3.0r  
`\nfss@catcodes`: Reset the acute, grave and double quote chars as well ..... 151
- 1996/09/21 ltoutput.dtx v1.1w  
`\@writeseup`: Added `\@parboxrestore` and made consequent deletions: wait for the howls of protest ..... 408
- 1996/09/25 ltdirchk.dtx v1.0t  
 General: Move ltxcheck to separate file ..... 13
- 1996/09/28 ltmiscen.dtx v1.1f  
`\@xobeysp`: Moved to ltspc.dtx ..... 255
- 1996/09/28 ltspc.dtx v1.2t  
`\@xobeysp`: Moved from ltmiscen.dtx and redefined to use `\nobreakspace` ..... 78
- 1996/09/29 ltfiles.dtx v1.1g  
`\document`: Added disabling of `\@nodocument` ..... 85
- 1996/09/29 ltoutput.dtx v1.1x  
`\newpage`: Checks for noskipsec and inlabel added ..... 396
- 1996/09/29 ltsect.dtx 1.0w  
`\@noskipsectrue`: Added documentation ..... 344
- 1996/09/30 ltoutput.dtx v1.1y  
`\newpage`: Checks for noskipsec and inlabel removed pending further tests ..... 396
- 1996/10/04 ltclass.dtx v1.0v  
`\RequirePackageWithOptions`: Reset `\@unprocessedoptions` for /2269 ..... 461
- 1996/10/05 ltfiles.dtx v1.1h  
`\@clubpenalty`: Added setting its value ..... 83
- 1996/10/08 ltfntcmd.dtx v3.3u  
`\DeclareTextFontCommand`: Removed `\check@icr` when in vmode since it causes various errors (see pr/2157) ..... 238
- 1996/10/21 lttab.dtx v1.1i  
`\@array`: Use `\set@typeset@protect` ..... 308  
 General: Moved the code associated with `\@mkpream` into the group provided by the box, for robustness (latex/2183) ..... 307  
`\multicolumn`: Make `\multicolumn` long (latex/2180) ..... 309
- `\tabbing`: Moved the `\indent` so that the `\everypar` can remove it when necessary; this is needed because the code for items in lists has changed (see pr/22111) ..... 302
- 1996/10/23 ltlists.dtx v1.0m  
`\@item`: `\@nobreak...` moved into the `\everypar` and not executed unconditionally, see above .. 281  
`\kern...` changed to `\setbox...` ..... 280  
 Added setting of `\clubpenalty` and set `\@nobreakfalse` only when necessary ..... 281
- 1996/10/23 ltsect.dtx v1.0x  
`\@xsect`: Replaced `\hskip...` with `\setbox...` as used in `\afterheading` ..... 348
- 1996/10/24 ltboxes.dtx v1.1a  
`\@arrayparboxrestore`: Added local settings of flags: dangerous! 291  
`\@iiiminipage`: Use it or lose it (`\@setminpage`): Frank will want to lose it ..... 292
- 1996/10/24 ltfloat.dtx v1.1p  
`\@floatboxreset`: Added local settings of flags: dangerous! ... 360  
`\@marginparreset`: Added local settings of flags: dangerous! . 366  
`\@xfloat`: Added `\@nodocument` to trap floats in the preamble . 357
- 1996/10/24 ltoutput.dtx v1.1z  
`\@addtocurcol`: Added `\nobreak`, etc as appropriate ..... 420, 424  
`\@specialoutput`: Added `\nobreak` as appropriate ..... 401  
`\@topnewpage`: Added `\@nodocument` to trap `\twocolumn` in the preamble ..... 397  
`\newpage`: Better checks for noskipsec and inlabel added, plus nobreak ..... 396
- 1996/10/25 ltlists.dtx v1.0n  
`\endtrivlist`: Change `\indent` to `\leavevmode` ..... 278  
 Reset flags explicitly ..... 278
- 1996/10/25 ltoutput.dtx v1.2a  
`\newpage`: Reset all flags explicitly 396
- 1996/10/26 ltlists.dtx v1.0o  
`\endtrivlist`: Correct typo .... 278
- 1996/10/27 ltoutenc.dtx v1.8c  
`\@strip@args`: Removed macro . 98  
 General: Added `\r A` ..... 106

Added <code>\textasteriskcentered</code> ..... 103, 111	1996/11/20 <code>ltvers.dtx</code> v1.0f General: Check for old format modified /2319 ..... 32
Corrected syntax descriptions . 92	
Removed <code>\aa</code> and <code>\AA</code> 102, 106, 107	1996/11/23 <code>ltoutenc.dtx</code> v1.8e General: Corrected description .. 92 Extended description ..... 93
1996/10/28 <code>ltplain.dtx</code> v1.1u General: (CAR) More doc changes 14	1996/11/28 <code>ltvers.dtx</code> v1.0g General: Check for old format modified /2319 ..... 32
<code>\dotfill</code> : Removed math mode . 29	1996/12/06 <code>ltdirchk.dtx</code> v1.0u <code>\IfFileExists</code> : *** removed from various messages for GNU Make. internal/2338 ..... 10
1996/10/29 <code>ltplain.dtx</code> v1.1v <code>\dotfill</code> : Got arithmetic correct (CAR) ..... 29	1996/12/06 <code>ltfloat.dtx</code> v1.1r <code>\caption</code> : Call <code>\setminpage</code> if needed. latex/2318 ..... 356
1996/10/29 <code>ltspace.dtx</code> v1.2u <code>\gnewline</code> : Added macro ..... 70	1996/12/06 <code>ltfssini.dtx</code> v3.0h General: (DPC) Remove *** from messages internal/2338 .... 212
<code>\no@lnbk</code> : Macro replaces <code>\lnbk</code> and <code>\no!lnbk</code> ..... 69	1996/12/17 <code>ltclass.dtx</code> v1.0w <code>\g@addto@macro</code> : Use <code>\begingroup</code> to save making a <code>mathord</code> .. 465
<code>\:</code> : Corrected and rationalised code 70	1996/12/20 <code>ltsect.dtx</code> v1.0z <code>\dottedtocline</code> : Added <code>\nobreak</code> for latex/2343 ..... 352
<code>\nolinebreak</code> : Reimplemented both using <code>\no@lnbk</code> ..... 69	1997/01/08 <code>fontdef.dtx</code> v2.2q General: Use <code>\DeclareMathDelimiter</code> to set delimiter codes . 222 <code>\mathparagraph</code> : Define using <code>\DeclareMathSymbol</code> ..... 229
1996/10/31 <code>ltfinal.dtx</code> v1.0z General: Added extra <code>\lcode</code> , hoping it does no harm in T1 (pr/1969) ..... 497, 500	1997/01/08 <code>ltfiles.dtx</code> v1.1j <code>\@include</code> : reset <code>\deadcycles</code> latex/2365 ..... 86
1996/10/31 <code>ltlists.dtx</code> v1.0p <code>\@trivlist</code> : Added check for missing item in outer list ..... 277	1997/01/08 <code>ltmath.dtx</code> v1.1d <code>\root</code> : (DPC) Remove spurious space tokens from plain $\TeX$ definition /2359 ..... 260
1996/10/31 <code>ltsect.dtx</code> v1.0y General: Corrected and tidied documentation; removed long lines 343	1997/02/05 <code>ltclass.dtx</code> v1.0x <code>\g@addto@macro</code> : missing percent /2402 ..... 465
1996/11/03 <code>ltplain.dtx</code> v1.1w <code>\dotfill</code> : Saved tokens by using <code>\hb@xt@</code> ..... 29	1997/02/21 <code>ltlists.dtx</code> v1.0r <code>\@item</code> : <code>\ifvoid</code> check added for <code>\noindent</code> . latex/2414 .... 280
1996/11/04 <code>lterror.dtx</code> v1.2m <code>\@nodocument</code> : Always define <code>\@nodocument</code> in kernel, so that it can be cleared by <code>\document</code> . 61	1997/03/21 <code>ltcounts.dtx</code> v1.1e <code>\fnsymbol</code> : Use <code>\mathsection</code> and <code>\mathparagraph</code> . latex/2445 134
1996/11/04 <code>ltlists.dtx</code> v1.0q <code>\@trivlist</code> : Moved check for missing item: only checked when not inlabel flag is false ..... 277	1997/04/14 <code>ltfiles.dtx</code> v1.1k <code>\document</code> : Set the document space factor defaults. latex/2404 ... 84 <code>\normalsfcodes</code> : Macro added (from patch file) latex/2404 .. 85
1996/11/05 <code>ltfiles.dtx</code> v1.1i <code>\nofiles</code> : Standard <code>\if@nobreak</code> test added ..... 85	
1996/11/09 <code>ltmath.dtx</code> v1.1c <code>\@ensuredmath</code> : Made long, as it was before. /2104 ..... 266	
1996/11/18 <code>ltfssbas.dtx</code> v3.0s <code>\define@newfont</code> : (DPC) lowercase fd file names. internal/1044 . 150	
1996/11/18 <code>ltoutenc.dtx</code> v1.8d General: (DPC) lowercase external file names. internal/1044 ... 118	
1996/11/20 <code>fontdef.dtx</code> v2.2p General: lowercase fd and enc.def file names /1044 ..... 214	

- 1997/04/14 ltoutput.dtx v1.2b  
`\@writsetup`: Call `\normalsf-codes` (from patch file) latex/2404 ..... 410  
 Move `\label` and `\index` (from patch file) ..... 410
- 1997/04/24 ltbibl.dtx v1.1m  
`\@citex`: `\@empty` to avoid primitive error on empty cite keys. latex/2432 ..... 376
- 1997/04/30 ltoutenc.dtx v1.9a  
 General: Changed `\textsc` to `\scshape` ..... 104  
 Introduced `\textcopyright` and modified `\copyright` ..... 103  
 Introduced `\textcopyright` and modify `\copyright` ..... 104  
 Modified `\textunderscore`, removing `\mathunderscore` .. 103  
 Modified `\underscore`, removing `\mathunderscore` ..... 104
- 1997/04/30 ltoutenc.dtx v1.9b  
 General: Added `\leavevmode` to `\textunderscore` ..... 103
- 1997/05/04 ltoutenc.dtx v1.9c  
 General: Added ‘hex index tabs’ . 109  
 Added TS1 encoding v2.2.beta 114
- 1997/05/07 ltoutenc.dtx v1.9d  
 General: Added `\leavevmode` to `\textcompwordmark` ..... 103
- 1997/05/07 ltspc.dtx v1.2v  
`\newline`: Made completely robust. .... 70
- 1997/05/29 ltfstrc.dtx v3.0j  
 General: Replaced `\` by `\MessageBreak`, as suggested by Donald Arseneau. .... 161
- 1997/05/29 ltlogos.dtx v1.1f  
`\LaTeXe`: Added `\m@th` so that the  $\text{\LaTeX}_\varepsilon$  logo works with non-zero values of `\mathsurround`. 80
- 1997/06/16 ltdirchk.dtx v1.0v  
 General: documentation improvements mainly from internal/2520 ..... 1
- 1997/06/16 ltfloat.dtx v1.1s  
 General: documentation fixes ... 353
- 1997/06/16 ltfntcmd.dtx v3.3v  
 General: Fix typo in documentation. .... 236
- 1997/08/05 ltoutenc.dtx v1.9e  
 General: Corrected order of arguments in `\UseTextSymbol` example. .... 92
- 1997/08/29 ltoutenc.dtx v1.9f  
 General: Added OT4 encoding, provided by Marcin Woliński. ... 91
- 1997/09/09 ltdefns.dtx v1.2z  
`\providecommand`: Use `\begin-group` to avoid generating math ords if used in math mode. pr/2573 ..... 41
- 1997/09/15 ltpictur.dtx v1.1g  
`\@getcirc`: Warn if lines become invisible pr/2524 ..... 334  
`\@picturewarn`: Macro added pr/2524 ..... 334  
`\@sline`: Warn if lines become invisible pr/2524 ..... 325
- 1997/10/06 ltcounts.dtx v1.1f  
`\@Roman`: Change `\@Roman` to be fully expandable, so that the result is written properly to files. 135  
`\@slowromancap`: Macro added. . 135
- 1997/10/08 ltlogos.dtx v1.1h  
`\LaTeX`: Simplify macro (force loading of suitable math fonts once). .... 80
- 1997/10/10 ltclass.dtx v1.0y  
`\endfilecontents`: `\@currentenv` in banner ..... 467  
`\reserved@c` not `\verbatim@out` to save a csname ..... 467  
 Check for text before or after `\end` environment. latex/2636 467  
 Use `\@gobbletwo` ..... 467
- 1997/10/17 ltfntcmd.dtx v3.3w  
`\check@nocorr@`: Check for vertical mode moved here, from `\DeclareTextFontCommand` (see PR/2646). .... 240  
`\DeclareTextFontCommand`: Reinstalled `\check@icr` as check is now done in `\check@nocorr@` (see PR/2646). .... 238
- 1997/10/20 ltfinal.dtx v1.1a  
`\@uclclist`: Removed `\aa` and `\AA` from `\@uclclist` as these are macros. .... 500
- 1997/10/21 ltdefns.dtx v1.2z1  
`\renewcommand`: Use `\begin-group/\endgroup` rather than braces for grouping, to avoid generating empty math atom. 39
- 1997/10/21 ltfssbas.dtx v3.0t  
`\define@newfont`: Move `\makeatletter` to `\nfss@catcodes`. ... 150

- `\nfss@catcodes`: Moved `\makeatletter` from `\try@load@font@shape`. . . . . 151
- 1997/11/09 `ltoutput.dtx` v1.2c
  - `\@specialoutput`: Remove incorrect code: only one `\@emptycol` is needed here . . . . . 399
  - `\@topnewpage`: Documentation of `vsize` check enhanced . . . . . 397
- 1997/11/13 `ltfssdcl.dtx` v3.0f
  - `\DeclareSymbolFont`: (DPC) Really update `\group@list` dont leave new version in `\toks@. latex/2661` . . . . . 193
  - `\stepcounter`: (DPC) Remove as never used. (Re)defined in `ltcounts` . . . . . 187
- 1997/11/19 `ltfloat.dtx` v1.1t
  - `\@footnotetext`: Missing percent, again . . . . . 371
- 1997/11/19 `ltoutput.dtx` v1.2d
  - `\@vtryfc`: Reindent code, to be understandable(DPC). . . . . 414
- 1997/11/20 `ltfssdcl.dtx` v3.0g
  - `\document@select@group`: (DPC) inline use of `\stepcounter` (faster, and saves a `csname` per math version as no reset list) . . . . . 189
  - `\select@group`: (DPC) inline use of `\stepcounter` (faster, and saves a `csname` per math version as no reset list) . . . . . 187
- 1997/11/23 `ltoutenc.dtx` v1.9g
  - General: Use `\textperthousand`, `\textpertenthousand` and `\textfractionsolidus` not `\textpermill`, `\textpertenmill` and `\textfraction`. /2673 . . . . . 114
- 1997/12/17 `ltoutenc.dtx` v1.9h
  - General: Added `\textperthousand` and `\textpertenthousand` . . . . . 107
  - Added code for `textcomp.sty`. . . . . 118
  - Added section. . . . . 118
  - Added `textcomp.sty`. . . . . 91
  - As in OT1, Added `\leavevmode` at start of `\c`, otherwise the output routine might be invoked within the macro. . . . . 107
  - Changed to decimal codes in `\oalign`. . . . . 116
  - Changed to decimal codes. . . . . 112
  - Documentation changes and additions. . . . . 91
  - Example corrected, braces removed. . . . . 91
  - Removed default settings, see next section. . . . . 114
- 1997/12/19 `ltoutenc.dtx` v1.9i
  - General: Documentation corrections. . . . . 91
- 1997/12/20 `fontdef.dtx` v2.2s
  - General: Added documentation . . . . . 216
- 1997/12/31 `ltoutenc.dtx` v1.9k
  - General: Further correction . . . . . 92
- 1998/01/12 `ltoutenc.dtx` v1.9k
  - General: Added `\ProvidesPackage` for `textcomp.sty` . . . . . 91
  - Adding missing braces and `\ushape`. . . . . 116
- 1998/01/16 `ltoutenc.dtx` v1.9m
  - General: fixed decimal codes. `latex/2734` . . . . . 112
- 1998/03/04 `ltdefns.dtx` v1.2z2
  - `\@xargdef`: Unnecessary `\expandafter` removed: `pr/2758` . . . . . 38
- 1998/03/05 `ltoutenc.dtx` v1.9n
  - General: Added `masc/fem` ords as in `pr/2579` . . . . . 104
- 1998/03/20 `ltdefns.dtx` v1.2z3
  - `\@thirdofthree`: Macro added . . . . . 42
- 1998/03/20 `ltoutenc.dtx` v1.9o
  - General: Added various `\UndeclareTextCommand` declarations for `pr/2783` . . . . . 128
  - Documentation added about order of decls . . . . . 94
  - Documentation added for `pr/2783` . . . . . 93
  - Load decls after defaults for speed. . . . . 128
- `\UndeclareTextCommand`: Macro added for `pr/2783` . . . . . 100
- 1998/03/21 `ltclass.dtx` v1.0z
  - General: Added to documentation of `filecontents` . . . . . 451
- 1998/03/21 `ltclass.dtx` v1.1a
  - `\@providesfile`: Allow `&`. `Internal/2702` . . . . . 458
  - General: Correct to new onlypreamble command list . . . . . 468
- 1998/03/25 `ltfssbas.dtx` v3.0u
  - `\showhyphens`: Suppress unnecessary error when used in preamble . . . . . 157
- 1998/04/11 `fontdef.dtx` v2.2t
  - General: Added `\mathring` accent (`pr2785`) . . . . . 227

- 1998/04/15 fontdef.dtx v2.2u  
General: Use new syntax for `\DeclareMathDelimiter` . . . . . 222
- 1998/04/15 ltssdcl.dtx v3.0h  
`\@xxDeclareMathDelimiter:`  
Macro added (pr/2662) . . . . . 201
- 1998/04/17 fontdef.dtx v2.2v  
General: Reinsert symbol defs for < and > chars. . . . . 222
- 1998/04/18 fontdef.dtx v2.2w  
General: Reinsert symbol def for / char. . . . . 222
- 1998/05/07 ltclass.dtx v1.1b  
`\@fileswithoptions:` Modify help message for latex/2805 . . . . . 464
- 1998/05/18 lttab.dtx v1.1j  
`\@endpbox:` Use `\setlength` to set `\hsize`, so that the changes in the calc package apply here. . . . . 317  
`\tabular*`: Use `\setlength`, so that calc extensions apply. . . . . 307
- 1998/05/20 ltfinal.dtx v1.1b  
General: Set up lccodes before loading hyphenation files: pr/2639 . . . . . 496  
Set up uc/lccodes after loading hyphenation files: pr/2639 . . . . . 499
- 1998/05/28 ltterror.dtx v1.2n  
`\@notdefinable:` Added message re 'end...' pr/1555 . . . . . 60
- 1998/06/04 ltboxes.dtx v1.1c  
`\@rule:` Support calc-expressions . . . . . 294
- 1998/06/12 ltoutenc.dtx v1.9p  
General: Corrected 130 and 131, see pr/2834 . . . . . 117  
Renamed `\textmacron` pr/2840 . . . . . 118, 124
- 1998/06/12 ltoutenc.dtx v1.9q  
`\add@accent:` Explicitly set `\spacefactor` after `\accent` (pr/2877) . . . . . 98
- 1998/06/18 lttab.dtx v1.1k  
General: Small addition to documentation . . . . . 296
- 1998/07/06 lttab.dtx v1.1l  
General: Small correction to documentation . . . . . 296
- 1998/08/17 ltboxes.dtx v1.1e  
General: (RmS) Minor Documentation fixes. . . . . 284
- 1998/08/17 ltclass.dtx v1.1c  
General: (RmS) Minor documentation fixes. . . . . 451
- 1998/08/17 ltdirchk.dtx v1.0w  
General: (RmS) Documentation improvements. . . . . 1
- 1998/08/17 ltfontcmd.dtx v3.3x  
General: (RmS) Minor documentation fixes. . . . . 236
- 1998/08/17 ltssbas.dtx v3.0v  
General: (RmS) Documentation fixes. . . . . 139
- 1998/08/17 ltssdcl.dtx v3.0i  
General: (RmS) Corrected minor glitches in changes entries. . . . . 185
- 1998/08/17 ltssini.dtx v3.0i  
General: (RmS) Minor documentation fixes. . . . . 208
- 1998/08/17 ltlogos.dtx v1.1i  
General: (RmS) Minor documentation fixes. . . . . 80
- 1998/08/17 ltmath.dtx v1.1c  
General: (RmS) Minor documentation fixes. . . . . 258
- 1998/08/17 ltmiscen.dtx v1.1g  
General: (RmS) Minor documentation fixes. . . . . 249
- 1998/08/17 ltspc.dtx v1.2w  
General: Documentation fixes. . . . . 66
- 1998/08/17 preload.dtx v2.1g  
General: (RmS) Minor documentation fixes. . . . . 232
- 1998/09/19 ltoutenc.dtx v1.9r  
`\a:` Added `\string` (pr/2878) . . . . . 102
- 1998/11/13 lttab.dtx v1.1m  
`\@array:` Check for hmode to see if something went wrong during parsing (pr/2884) . . . . . 308
- 1999/01/05 fontdef.dtx v2.2x  
General: Need special protection for character > in `\changes` entry. . . . . 214
- 1999/01/06 ltssbas.dtx v3.0w  
`\DeclareFontEncoding:` Added `\LastDeclaredEncoding` to support cyrillic integration (pr/2988) . . . . . 142  
`\LastDeclaredEncoding:` Added `\LastDeclaredEncoding` to support cyrillic integration (pr/2988) . . . . . 142
- 1999/01/06 ltoutenc.dtx v1.9r  
`\@strip@args:` New impl for latex/2930 . . . . . 99  
General: Minor documentation fix. . . . . 116
- 1999/01/06 ltoutput.dtx v1.2e  
`\@makecol:` Added negative vskip, as when processing output-

- box below: suggested by Fred Bartlett pr/2892 . . . . . 405
- 1999/01/07 ltdefns.dtx v1.3a
- `\@ifnextchar`: made long . . . . . 46
- `\@newenvb`: made long and brace optional arg. latex/2896 . . . . 40
- `\@testopt`: made long and brace optional arg. latex/2896 . . . . 38
- 1999/01/07 ltdefns.dtx v1.3b
- `\@ifnextchar`: extra `\long`. latex/2902 . . . . . 46
- 1999/01/07 ltoutenc.dtx v1.9r
- General: Hackery to allow using fontenc several times . . . . . 120
- Hackery to temp support cyrillic uc/lc . . . . . 119
- 1999/01/13 ltoutenc.dtx v1.9s
- `\@strip@args`: Simplified solution for latex/2930 . . . . . 99
- 1999/01/18 ltdefns.dtx v1.3c
- `\@yargd@f`: New implementation DPC /2942 . . . . . 38
- 1999/02/09 ltdefns.dtx v1.3d
- `\@yargd@f`: catch bad argument forms by re-inserting #3 . . . . 38
- 1999/02/12 ltffsini.dtx v3.0j
- `\oldstylenums`: Use `\rmdefault` instead of `cmm` (pr/2954) . . . 210
- 1999/02/24 ltoutenc.dtx v1.9t
- General: Corrected hackery cyrillic uc/lc list . . . . . 119
- 1999/03/01 ltdefns.dtx v1.3e
- `\@ifnextchar`: remove extra `\long`. internal/2967 . . . . . 46
- 1999/04/15 ltpictur.dtx v1.1h
- `\@getlarrow`: Replaced octal number, CAR . . . . . 326
- `\@upvector`: Replaced octal number, CAR . . . . . 327
- General: Replaced octal number, CAR . . . . . 326, 327
- Replaced octal numbers, CAR . . 318
- 1999/04/19 ltfloat.dtx v1.1u
- `\caption`: Made caption an error outside a float: latex/2815 . . 356
- 1999/04/27 ltboxes.dtx v1.1f
- `\@parboxto`: (CAR) Changed `\@empty` to `\relax` as flag for natural width: pr/2975 . . . . 291
- 1999/04/29 ltdefns.dtx v1.3f
- `\@yargd@f`: Full expansion and conversion needed for digit in new version, see pr/3013 . . . . . 38
- New macro added . . . . . 38
- 1999/06/10 ltoutenc.dtx v1.9u
- General: Ensure that we also forget old options (pr/2888) . . . . . 120
- 1999/06/12 ltoutenc.dtx v1.9v
- General: Extend `\@uclclist` only once . . . . . 120
- 1999/10/09 ltmath.dtx v1.1e
- `\active@math@prime`: Macro added, see PR 3104. . . . . 262
- `\prime@s`: Introduce `\active@math@prime`. . . . . 262
- 1999/10/09 ltoutput.dtx 1.2f
- `\@activechar@info`: Reset definition of active prime character (used in math mode) . . . . . 408
- 1999/10/28 ltoutenc.dtx v1.9w
- `\add@accent`: Give `\accent@spacefactor` a default definition (pr/3084) . . . . . 98
- 1999/12/08 ltoutenc.dtx v1.9x
- General: Changed `\CYRRHOOK` and `\cyrrhook` to `\CYRRHK` and `\cyrrhk` as name changed in the cyrillic bundle for naming consistency with other “hook” glyphs. . . . . 119
- 2000/01/07 ltmiscen.dtx v1.1h
- `\@verbatim`: Disable hyphenation even if the font allows it. . . . 255
- 2000/01/15 ltpictur.dtx v1.1i
- `\@upvector`: Removed space at end-of-line, CAR . . . . . 327
- 2000/01/30 ltfntcmd.dtx v3.3y
- `\DeclareTextFontCommand`: Use `\hmode@bgroup` now (pr/3160) . 238
- 2000/01/30 ltoutenc.dtx v1.9y
- General: Use `\hmode@bgroup` where applicable (pr/3160) . . . . . 105–107, 111, 113, 114, 116
- `\add@accent`: Use `\hmode@bgroup` where applicable (pr/3160) . . 97
- `\hmode@bgroup`: Macro added . . . 98
- 2000/01/30 ltoutenc.dtx v1.9z
- `\@use@text@encoding`: Macro reimplemented (pr/3160) . 99, 100
- `\add@accent`: Macro reimplemented (pr/3160) . . . . . 97
- `\hmode@start@before@group`: Macro added (pr/3160) . . . . 100
- 2000/05/19 ltmiscen.dtx v1.1i
- `\enddocument`: Reset `\AtEndDocument` for latex/3060 . . . . . 250

2000/05/26 ltpage.dtx v1.0j	2001/01/07 ltoutput.dtx v1.2j
\@markright: Reimplementation to fix expansion error (pr/3203). . . . . 380	\@writeseup: And do it in the right macro (pr/3286) . . . . . 409
\leftmark: Use \@empty instead of brace group (pr/3203). . . . . 380	2001/02/16 ltxref.dtx v1.1k
\markright: Reimplementation to fix expansion error (pr/3203). 380	\@newl@bel: Added an extra grou- plevel (PR3250), jlb . . . . . 246
\rightmark: Use \@empty instead of brace group (pr/3203). . . . . 380	2001/05/25 ltclass.dtx v1.1d
2000/06/02 ltpage.dtx v1.0k	\@providesfile: Explicitly set cat- code of \endlinechar to 10 (pr/3334) . . . . . 458
\@markright: Small adjustment to give slightly less expansion, CAR . . . . . 380	2001/05/25 ltdirchk.dtx v1.0x
\markright: Small adjustment to give slightly less expansion, CAR . . . . . 380	General: Explicitly set catcode of \endlinechar to 10 (pr/3334) . 4
Tidied 1.0j reimplementation, CAR . . . . . 380	2001/05/28 ltoutenc.dtx v1.93
2000/07/11 ltmiscen.dtx v1.1j	General: Added composites for compatibility with T1, pr/3295 106
\enddocument: Fix typo in warning 251	Changed the effect of \.\i, pr/3295 . . . . . 109
2000/07/12 ltoutput.dtx 1.2g	2001/06/02 fontdef.dtx v2.2y
General: Ensure that rule is in \normalcolor . . . . . 446	General: Provide default cfg files (pr/3264) . . . . . 230
2000/07/12 ltoutput.dtx 1.2i	2001/06/04 fontdef.dtx v2.2z
\@makecol: Removed negative vskip, as it gives unacceptable results when the depth is large: pr/3189 . . . . . 405	General: Guard against math active equal and pipe sign in \models (pr/3333) . . . . . 226
2000/07/19 ltoutput.dtx v1.2h	Guard against math active equal sign in \Relbar (pr/3333) . . 226
\@writeseup: Reset and re- store \@ifnewlist for inter- nal/3231 . . . . . 409	2001/06/04 ltclass.dtx v1.1e
2000/08/30 ltoutenc.dtx v1.91	\@providesfile: But only if it is a char (pr/3334) . . . . . 458
\@use@text@encoding: Rearranged but no change to final code, CAR (pr/3160) . . . . . 99	2001/06/04 ltdirchk.dtx v1.0y
\add@accent: Rearranged but no change to final code, CAR (pr/3160) . . . . . 97	General: But only if it is a char (pr/3334) . . . . . 4
2000/09/01 ltfinal.dtx v1.1d	2001/06/04 ltpictur.dtx v1.1j
\errhelp: Set error help empty at very end (pr/449 done cor- rectly). . . . . 502	\@sline: Don't warn for exactly zero pr/3318 . . . . . 325
2000/09/24 ltfloat.dtx v1.2b	2001/06/04 ltvers.dtx v1.0i
\end@dblfloat: FMI: use output routine to defer float . . . . . 361	General: Check for old format dis- abled . . . . . 32
2000/09/24 ltoutput.dtx v1.2b	2001/06/05 ltoutenc.dtx v1.94
\@doclearpage: FMI: ensure \do- clearpage is called again until all floats are output. . . . . 403	General: Text composite Com- mands need kludges for ‘,’ – see tlb1903.lvt . . . . . 106
2000/09/24 ltoutput.dtx v1.2n	2001/08/26 ltclass.dtx v1.1f
\@addtocurcol: FMI: test for wide float was in wrong place . . . 419	\@providesfile: Readdded setting of space char (pr/3353) . . . . 458
	2002/02/24 ltplain.dtx v1.1x
	\loggingall: Macro added . . . . 29
	\loggingoutput: Macro added . . 29
	\showoutput: Use newly added \loggingoutput . . . . . 29
	\tracingall: Use newly added \loggingoutput . . . . . 29

- 2002/06/16 ltoutenc.dtx v1.95  
 General: Added `\textbardbl` (pr/3400) ..... 111  
 Added default for `\textbardbl` (pr/3400) ..... 103
- 2002/06/17 ltoutenc.dtx v1.95  
 General: Corrected `\c` for T1 (pr/3442) ..... 107  
 Definition of `\textexclamdown` changed (pr/3368) ..... 105  
 Definition of `\textquestiondown` changed (pr/3368) .... 105
- 2002/06/18 ltoutenc.dtx v1.95  
 General: Changed def for `\textregistered` to avoid small caps (pr/3420) ..... 104
- 2002/10/01 ltfloat.dtx v1.1v  
`\thempfootnote`: Use braces around `\itshape` to keep font change local (pr/3460). .... 369
- 2002/10/02 ltssbas.dtx v3.0x  
`\DeclareFontSubstitution`:  
 Adding `\LastDeclaredEncoding` introduced a bug as on some occasions that macro name was stored in the internal lists instead of the actual encoding. (pr/3459) ..... 142
- 2002/10/28 ltlists.dtx v1.0s  
`\endtrivlist`: Check for math mode (pr/3437) ..... 278
- 2002/10/28 ltoutenc.dtx v1.96  
 General: coding change, to follow bug fix by DEK in plain.tex (pr/3469) ..... 106, 113
- 2002/12/13 ltbibl.dtx v1.1n  
`\citex`: Added `\leavevmode` in case citation is at start of paragraph (pr/3486) ..... 376
- 2003/01/01 ltfntcmd.dtx v3.3z  
 General: Code checked and documentation extended by Chris 238
- 2003/05/18 ltbibl.dtx v1.1o  
`\nocite`: Check if we are after `\document` ..... 377
- 2003/08/27 ltpictur.dtx v1.1k  
`\@bezier`: added missing displacement pr/3566 ..... 338  
`\@sline`: check for `\@linechar` being empty pr/3570 ..... 325
- 2003/10/13 ltfinal.dtx v1.1e  
 General: Added extra `\lccode` for `\-` and `\textcompwordmark` . 497
- 2003/12/16 ltoutput.dtx v1.2k  
`\@makecol`: Ensure that `\@elt` has a defined state (pr/3586) ... 405
- 2003/12/30 ltpictur.dtx v1.1j  
`\@getcirc`: issue warning if circle size can't be met pr/3473 .. 334
- 2004/01/03 ltoutenc.dtx v1.99b  
 General: Added `\textogonekcentered` (pr/3532) ..... 107  
 Added composites for `\k` (pr/3532) ..... 111  
 Use `\ooalign` for `\k` (pr/3532) 107
- 2004/01/04 ltbibl.dtx v1.1p  
`\nocite`: Changed error message 377
- 2004/01/04 ltoutenc.dtx v1.99c  
 General: More adjustments for `ogonek` (pr/3532) ..... 107
- 2004/01/23 ltdefs.dtx v1.1g  
`\@newenva`: Use kernel version of `\@ifnextchar` (pr/3501) .... 40  
`\@testopt`: Use kernel version of `\@ifnextchar` (pr/3501) .... 38  
`\@xargdef`: Use kernel version of `\@ifnextchar` (pr/3501) .... 37  
`\@xdblarg`: Use kernel version of `\@ifnextchar` (pr/3501) .... 47
- 2004/01/23 ltdefs.dtx v1.3g  
`\kernel@ifnextchar`: Added macro (pr/3501) ..... 46
- 2004/01/28 ltclass.dtx v1.1g  
`\@providesfile`: Use kernel version of `\@ifnextchar` (pr/3501) . 458
- 2004/01/28 ltvers.dtx v1.0k  
 General: Check for old format made 5 years (pr/3601) ..... 32
- 2004/02/02 fontdef.dtx v2.3  
 General: Many things from here on made robust ..... 226
- 2004/02/04 fontdef.dtx v2.3a  
 General: Added `bigtriangle` synonyms for `stmaryrd` ..... 224
- 2004/02/04 ltspc.dtx v1.3  
`\nobreakdashes`: (Macro added . 77
- 2004/02/06 ltoutenc.dtx v1.99d  
`\@inmathwarn`: New command added to fix severe bug: pr/3563 ..... 95
- 2004/02/07 ltoutput.dtx v1.2l  
`\@doclearpage`: Empty kludgeins box if necessary, pr/3528 ... 403
- 2004/02/13 ltoutenc.dtx v1.99e  
 General: Documentation fixes: typos ..... 91

2004/02/15 ltbibl.dtx v1.1q	change (pr/3975) . . . . .	194
\@cite@ofmt: Added hook with default value \hbox . . . . .	2009/09/24 ltvers.dtx v1.0l	
\@citex: Changed to use a hook with default value \hbox . . .	General: Stop checking for old format . . . . .	32
2004/02/15 ltspaced.dtx v1.3a	2009/10/20 ltfssdcl.dtx v3.0m	
\nobreakdashes: (Added spacefactor setting . . . . .	\in@: More robust thanks to Heiko. . . . .	185
2004/10/20 ltoutput.dtx v1.2m	2009/10/28 ltoutenc.dtx v1.99k	
\@makecol: Removed dead code . . . . .	General: Added Latin Modern and TeX Gyre subsets . . . . .	130
2005/07/27 ltfssdcl.dtx v3.0j	2009/11/04 ltoutenc.dtx v1.99l	
\DeclareMathAlphabet: (MH) Make document commands robust . . . . .	General: Added more Latin Modern and TeX Gyre subsets . . . . .	130
\DeclareSymbolFontAlphabet: (MH) Make document commands robust . . . . .	2009/12/14 ltfntcmd.dtx v3.4a	
\new@mathalphabet: (MH) Make document commands robust . . . . .	\ifmaybe@ic: Macro added . . . . .	240
\non@alpherr: (MH) Change because command is now properly robust . . . . .	\maybe@ic@: Use switch \ifmaybe@ic instead of \if@tempswa . . . . .	240
\SetMathAlphabet: (MH) Make document commands robust . . . . .	\t@st@ic: Use switch \ifmaybe@ic instead of \if@tempswa . . . . .	241
2005/09/27 ltoutenc.dtx v1.99g	2010/08/17 ltmiscen.dtx v1.1k	
General: Replace \sh@ft by \ltx@sh@ft . . . . .	\enddocument: Use braces around \input arg (pr/4124) . . . . .	251
2005/09/27 ltplain.dtx v1.1y	2010/08/17 ltmiscen.dtx v1.1l	
\ltx@sh@ft: New macro . . . . .	\enddocument: Change of plan: use @@input instead (pr/4124) . . . . .	251
\sh@ft: Macro no longer used but left for compatibility . . . . .	2011/05/08 ltfssdcl.dtx v3.0n	
2005/11/08 ltoutenc.dtx v1.99h	\in@: Simplified thanks to Bruno. . . . .	185
General: Added \ij and \IJ from babel. (pr/3771) . . . . .	2011/08/19 ltclass.dtx v1.1i	
2005/11/10 ltmath.dtx v1.1g	\@ifclasswith: Re-jig definition after more stringent \in@ test. . . . .	457
\[: (MH) Fixed potential problem in \[ (pr/3399). . . . .	2011/09/03 ltfssdcl.dtx v3.0o	
General: (MH) Minor documentation fixes. . . . .	\new@mathversion: (Will) Remove \global before \newcount (unnecessary and caused etex bug). . . . .	192
2006/05/18 ltboxes.dtx v1.1g	2012/01/20 ltplain.dtx v2.0b	
\@parboxto: Ensure \@parboxto holds the value of \@tempdimb not the register itself (pr/3867) . . . . .	\loggingall: etex tracing if available . . . . .	29
2006/09/13 ltoutput.dtx v1.1m	2013/07/07 ltclass.dtx v1.1i	
General: Ensure that rule is in \normalcolor . . . . .	General: Correctly describe how the date in \@ifpackagelater is used . . . . .	454
2007/08/05 ltclass.dtx v1.1h	2014/04/18 ltoutput.dtx v1.1o	
\@fileswithoptions: Prevent loss of brackets PR/3965 . . . . .	General: Handle infinite glue from \enlargethispage (pr/4023) . . . . .	446
2007/08/06 ltentrl.dtx v1.0h	2014/04/24 ltoutput.dtx v1.2n	
\@fornoop: Really make defs long . . . . .	\fl@tracemessage: Renamed internal trace commands; provide as package . . . . .	437
2007/08/31 ltfssdcl.dtx v3.0l	2014/04/27 ltfloating.dtx v1.2b	
\SetSymbolFont@: Font warning changed to info for encoding	\end@dblfloat: Inline the code to allow some coexistence with packages that hook into	

<code>\end@float</code> and do not know about the algorithm change .	361	<code>\savebox</code> : Make Robust (latexrelease) . . . . .	287
2014/06/10 <code>lfloat.dtx</code> v1.2b		2015/01/08 <code>ltdfn.dtx</code> v1.4a	
<code>\end@dblfloat</code> : missing <code>\fi</code> added	361	<code>\MakeRobust</code> : Added macro . . . .	45
2014/12/30 <code>ltfinal.dtx</code> v2.0a		2015/01/08 <code>ltnlength.dtx</code> v1.1c	
<code>\newmarks</code> : macro added . . . . .	494	<code>\setlength</code> : to ensure first length argument is terminated. (latexrelease) . . . . .	138
<code>\newXeTeXintercharclass</code> : macro added . . . . .	494	2015/01/08 <code>ltmath.dtx</code> v1.1h	
2014/12/30 <code>lfloat.dtx</code> v1.2a		<code>\]</code> : Make Robust (latexrelease) . .	262
<code>\@textsubscript</code> : Command added (latexrelease) . . . . .	370	<code>\]</code> : Make Robust (latexrelease) . .	263
<code>\textsubscript</code> : Command added (latexrelease) . . . . .	370	2015/01/09 <code>ltfssini.dtx</code> v3.1a	
2014/12/30 <code>ltfssbas.dtx</code> v3.0y		<code>\em</code> : Allow <code>\emph</code> to produce small caps (latexrelease) . . . . .	209
<code>\mathgroup</code> : move allocation to <code>ltplain</code> . . . . .	139	<code>\eminnershape</code> : macro added (latexrelease) . . . . .	209
2014/12/30 <code>ltoutput.dtx</code> v1.2m		2015/01/09 <code>ltspace.dtx</code> v1.1h	
General: Command updated (latexrelease) . . . . .	446	<code>\addpenalty</code> : Donald Arseneau's fix from PR/377703 (latexrelease) . . . . .	75
2014/12/30 <code>ltplain.dtx</code> v2.0a		2015/01/10 <code>ltcounts.dtx</code> v1.1h	
<code>\e@alloc</code> : macro added . . . . .	18	<code>\@fnsymbol</code> : Unse <code>\TextOrMath</code> (latexrelease) . . . . .	135
<code>\e@alloc@chardef</code> : macro added . . . .	17	<code>\@stpelt</code> : Reset all within counters in one go (latexrelease) . . . .	133
<code>\e@alloc@top</code> : macro added . . . .	17	2015/01/11 <code>ltcounts.dtx</code> v1.1h	
<code>\e@ch@ck</code> : macro added . . . . .	19	<code>\TextOrMath</code> : Add command to solve robustness issues (pr/3752) (latexrelease) . . . .	136
<code>\extrafloats</code> : macro added . . . .	19	2015/01/11 <code>lfloat.dtx</code> v1.2b	
<code>\newlanguage</code> : New engine-specific allocation scheme (latexrelease) . . . . .	17	<code>\dblfloatplacement</code> : float order in 2-column (latexrelease) . .	363
2014/12/30 <code>ltspace.dtx</code> v1.3b		<code>\@xfloat</code> : Check for valid option (latexrelease) . . . . .	357
<code>\@</code> : <code>\@</code> discards spaces when moving (pr3039)(latexrelease) . . . . .	78	<code>\end@dblfloat</code> : float order in 2-column (latexrelease) . . . .	361
2015/01/03 <code>ltdfn.dtx</code> v1.4a		2015/01/11 <code>ltfssbas.dtx</code> v3.0y	
<code>\typein</code> : use modified definition in <code>luatex</code> . . . . .	36	<code>\@DeclareMathSizes</code> : Allow arbitrary units (latexrelease) . . .	144
2015/01/03 <code>ltdirchk.dtx</code> v1.1		2015/01/11 <code>ltspace.dtx</code> v1.3d	
General: Enable extra primitives when <code>LuaTeX</code> is used . . . . .	3	<code>\@Esphack</code> : Allow hyphenation (Donald Arseneau pr/3498) (latexrelease) . . . . .	73
2015/01/03 <code>ltfinal.dtx</code> v2.0a		<code>\@esphack</code> : Allow hyphenation (Donald Arseneau pr/3498) (latexrelease) . . . . .	72
General: Skip resetting codes with Unicode engines . . . . .	499	2015/01/14 <code>ltoutput.dtx</code> v1.2n	
Unicode data loading added . .	495	<code>\@addtocurcol</code> : float order in 2-column (latexrelease) . . . .	418
2015/01/07 <code>ltvers.dtx</code> v1.0n		<code>\@addtodblcol</code> : float order in 2-column (latexrelease) . . . .	429
<code>\IncludeInRelease</code> : macro added . .	33	<code>\@addtonextcol</code> : float order in 2-column (latexrelease) . . . .	425
2015/01/08 <code>ltboxes.dtx</code> v1.1h			
<code>\framebox</code> : Make Robust (latexrelease) . . . . .	289		
<code>\makebox</code> : Make Robust (latexrelease) . . . . .	285		
<code>\parbox</code> : Make Robust (latexrelease) . . . . .	290		
<code>\raisebox</code> : Make Robust (latexrelease) . . . . .	294		
<code>\rule</code> : Make Robust (latexrelease) . .	293		

<code>\@docclearpage</code> : Empty kludgeins box if necessary, pr/3528 ...	402	2015/02/21 lterror.dtx v1.2o	
float order in 2-column (latexrelease) .....	402	General: Removed autoload support .....	55
<code>\@startdblcolumn</code> : float order in 2-column (latexrelease) .....	413	2015/02/21 ltfiles.dtx v1.1m	
<code>\@xtryfc</code> : float order in 2-column (latexrelease) .....	414	General: Removed autoload support .....	81
<code>\@ztryfc</code> : float order in 2-column (latexrelease) .....	416	2015/02/21 ltfssbas.dtx v3.0z	
2015/01/14 ltspaced.dtx v1.3e		General: Removed autoload code	139
<code>\addpenalty</code> : Avoid adding redundant skips (DPC) .....	75	2015/02/21 ltfsscnp.dtx v3.0d	
2015/01/17 ltvers.dtx v1.0m		General: Removed autoload code	181
<code>\IncludeInRelease</code> : modified with <code>\@currname</code> .....	33	2015/02/21 ltfssdcl.dtx v3.0p	
2015/01/19 ltvers.dtx v1.0o		General: Removed autoload code	185
<code>\IncludeInRelease</code> : Optional argument .....	33	2015/02/21 ltfssstrc.dtx v3.0k	
2015/01/20 ltoutput.dtx v1.2m		General: Removed autoload code	159
<code>\fl@tracemessage</code> : Reset <code>\IncludeInRelease</code> flags .....	438	2015/02/21 ltoutenc.dtx v1.99m	
2015/01/22 ltvers.dtx v1.0p		General: Removed autoload code	91
General: Preserve any <code>\everyjob</code> material inserted by a loader ( <code>.ini</code> file) .....	33	2015/02/21 ltoutput.dtx v1.2n	
2015/01/23 ltfinal.dtx v2.0b		General: Removed autoload code	382
<code>\newmarks</code> : use reserved count 256	494	<code>\f@depth</code> : macro added(latexrelease) .....	401
<code>\newXeTeXintercharclass</code> : use reserved count 257 .....	494	2015/02/21 ltpictur.dtx v1.1k	
2015/01/23 ltplain.dtx v2.0c		General: Removed autoload code	318
<code>\extrafloats</code> : reserve counts 256–265 .....	19	2015/02/21 ltplain.dtx v2.0e	
2015/01/24 ltfinal.dtx v2.0c		General: Removed autoload code	14
General: Skip T1-code entirely with Unicode engines .....	495	2015/02/21 lttab.dtx v1.1n	
2015/02/03 ltfinal.dtx v2.0d		General: Removed autoload code	296
General: Set <code>\lccode</code> for - with Unicode engines .....	496	2015/02/21 ltvers.dtx v1.0r	
2015/02/16 ltoutenc.dtx v1.99m		General: Removed autoload code	32
General: Added <code>\textcommabelow</code> latex/4414 .....	104	2015/02/22 ltfsscnp.dtx v3.0e	
Added lmtt (Heiko Oberdiek) latex/4415 .....	130	General: Moved all code into latexrelease - obsolete commands are no longer automatically part of the kernel .....	181
2015/02/19 ltvers.dtx v1.0q		2015/03/02 ltplain.dtx v2.0f	
<code>\IncludeInRelease</code> : Swap argument order .....	33	<code>\e@mathgroup@top</code> : macro added	18
2015/02/20 ltplain.dtx v2.0d		<code>\newlanguage</code> : allow 255 math groups in Unicode engines ...	17
<code>\loggingall</code> : Spell commands correctly :-)	29	2015/03/10 ltplain.dtx v2.0g	
2015/02/21 ltdefs.dtx v1.4b		<code>\hideoutput</code> : macro added .....	30
General: Removed autoload support .....	34	<code>\loggingall</code> : Reorganise to be less noisy .....	29
		<code>\tracingnone</code> : macro added .....	30
		2015/03/18 ltfssdcl.dtx v3.0q	
		<code>\DeclareSymbolFont</code> : Restrict Symbol fonts to 0-15 .....	193
		<code>\document@select@group</code> : Introduce <code>\e@mathgroup@top</code> ...	189
		<code>\select@group</code> : Introduce <code>\e@mathgroup@top</code> .....	187
		2015/03/26 ltfinal.dtx v2.0d	
		General: Use renamed <code>unicode-letters.def</code> .....	495

2015/04/07 ltffsbas.dtx v3.1a	2015/08/30 ltplain.dtx v2.1a
\wrong@fontshape: Try loading fd	\newinsert: new \newinsert im-
file if family has changed . . . 152	plementation . . . . . 21
2015/04/28 ltfinal.dtx v2.0f	2015/09/24 ltluatex.dtx v1.0a
\newXeTeXintercharclass: define	call_callback: Function added . 490
\xe@alloc@intercharclass for	callback.register: Function
compatibility with older xelatex	modified . . . . . 488
initialisation . . . . . 494	callback_descriptions: Function
2015/05/10 ltlists.dtx v1.0t	added . . . . . 493
\@doendpe: Explicitly reset	\catcodetable@atletter: Macro
\clubpenalty before clearing	added . . . . . 478
\everypar; see also pr/0462	\catcodetable@initex: Macro
and pr/4065 . . . . . 279	added . . . . . 478
2015/06/19 ltfinal.dtx v2.0g	\catcodetable@latex: Macro
\newmarks: Use -1 for first range	added . . . . . 478
to get contiguous allocation . 494	\catcodetable@string: Macro
\newXeTeXintercharclass: Use -1	added . . . . . 478
for first range to get contiguous	add_to_callback: Function added 490
allocation . . . . . 494	remove_from_callback: Function
2015/06/19 ltplain.dtx v2.0h	added . . . . . 491
General: delete spurious old defini-	new_attribute: Function added . 484
tion of \newtoks . . . . . 22	disable_callback: Function
\@e@alloc: extra braces in case ar-	added . . . . . 492
guments not single token . . . 19	in_callback: Function added . . 492
\newlanguage: Use -1 for first	\newattribute: Macro added . . 478
range to get contiguous alloca-	\newcatcodetable: Macro added 478
tion . . . . . 17	\newluabytecode: Macro added . 480
2015/06/23 ltfinal.dtx v2.0h	\newluachunkname: Macro added 480
General: set \patch@level in ltvers	\newluafunction: Macro added . 479
rather than in ltfinal/ltpatch 501	\newwhatsit: Macro added . . . 480
2015/06/23 ltvers.dtx v1.0t	module_error: Function added . . 483
General: set \patch@level in ltvers	module_info: Function added . . 483
rather than in ltfinal/ltpatch . 32	module_warning: Function added 483
2015/08/06 ltplain.dtx v2.0i	modules: Function modified . . . 482
\extrafloats: Add \string in case	create_callback: Function added 489
argument is not an unexpand-	provides_module: Function added 482
able primitive . . . . . 19	luatexbase: Table added . . . . 481
2015/08/23 ltdirchk.dtx v1.2	2015/10/02 ltdirchk.dtx v1.2a
General: Do not use luatex prefix . 3	General: Allow backing out of un-
2015/08/23 ltvers.dtx v1.0v	prefixed names . . . . . 3
General: Allow negative patchlevel	2015/10/02 ltluatex.dtx v1.0e
for pre-release . . . . . 33	uninstall: Function added . . . 493

# Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

## Symbols

$\backslash!$ . . . . .	b352, b354, <u>z144</u> , O177	k203, k219, p464, p466, p467, C199, C200, C201, C211, K10, K11
$\backslash"$ . . . . .	l176, l286, l324, l362, l370, l443, l475, l502, l510, l516, l520, l526, l530, l536, l542, l549, l550, l556, l560, l600, l643, o350, O178	$\backslash@@defaultsubs$ . . . . . <u>o440</u>
$\backslash\#$ . . . . .	a57, a70, b6, b14, b419, d314, o337, O161	$\backslash@@enc@update$ . . . . . l133, o224, <u>o228</u>
$\backslash\$$ . . . . .	a69, b4, b13, d313, l256, l349, l356, l432, l655, l662, O162	$\backslash@@end$ . . . . . a64, a217, d8, k183, k184, s135, y39, y49, M18, O267, O288
$\backslash\%$ . . . . .	a70, a100, a102, a122, b14, b417, d314, l393, l395, o339, L499, L500, O163	$\backslash@@endpbox$ . . . . . C166, C197, <u>C345</u>
$\backslash\&$ a69, b5, b13, b418, d313, L118, O164		$\backslash@@eqnocr$ . . . . . z273, z291, <u>z294</u> , <u>z399</u>
$\backslash'$ . b439, l177, l287, l326, l360, l367, l445, l455, l461, l463, l466, l468, l476, l482, l488, l490, l493, l495, l503, l507, l514, l518, l523, l528, l531, l533, l540, l545, l546, l553, l558, l561, l601, l645, l664, l666, l667, l668, l671, l673, l674, l675, l677, l678, o349, s168, t172, y145, z151, B236, C61, K558, O179		$\backslash@@fileswith@ptions$ . . . . . L191, <u>L361</u>
$\backslash($ . . . . .	<u>z168</u> , z238	$\backslash@@hyph$ . . . . . d9
$\backslash)$ . . . . .	b439, <u>z168</u> , z239	$\backslash@@hyphenation$ . . . . . l155
$\backslash*$ . . . . .	o342, <u>z148</u> , L433, L501	$\backslash@@if@newlist$ . . . . . K568, K615
$\backslash+$ . . . . .	C61	$\backslash@@if@definable$ . . . . . <u>d109</u> , l17
$\backslash,$ . b353, b355, <u>i267</u> , t414, y145, z7, z8, z40, z108, z110, z113, z127, z144		$\backslash@@input$ a63, d7, k162, k163, k172, y19
$\backslash-$ . . . . .	b321, <u>d9</u> , d11, i258, l321, l322, l438, l639, l640, o344, y145, B235, C61, O71, O117	$\backslash@@italiccorr$ . . . . . <u>d12</u> , v96, v100
$\backslash.$ . . . . .	b352, b354, k39, l178, l288, l357, l358, l376, l451, l452, l478, l479, l505, l602, l669, l676, o343	$\backslash@@line$ . . . . . <u>B368</u>
$\backslash/$ . . . . .	a92, d12, o291, o345, L117	$\backslash@@math@bgroup$ . . . . . v114, v121
$\backslash:$ . . . . .	b353, b355, d306, d307, <u>z149</u>	$\backslash@@math@egroup$ . . . . . <u>v111</u> , <u>v111</u>
$\backslash;$ . . . . .	b353, b355, t408, z128, <u>z144</u>	$\backslash@@par$ . . . . . 64, <u>d6</u> , h4, y49, y104, y108, y111, A82, A85, B217, B234, C172, F50, F101, K231
$\backslash<$ . . . . .	l439, l593, o340, y145, C60, C98	$\backslash@@patterns$ . . . . . l155
$\backslash=$ l179, l289, l375, l603, s168, B236, C60		$\backslash@@protect$ . . . . . d231, d237, d246
$\backslash>$ l436, l594, o341, y145, <u>z144</u> , z149, C60		$\backslash@@startpbox$ . . . . . C166, C197, <u>C345</u>
$\backslash?$ . . . . .	b352, b354, O179	$\backslash@@underline$ . . . . . B325, B328, B329
$\backslash@$ . . . . .	a60, d308, d309, g19, <u>i270</u> , j2, L24, L32, N18, N686, O171	$\backslash@@unprocessedoptions$ . . . . . L343, <u>L405</u>
$\backslash@@$ . . . . .	a302, a303, f15, f19, f20, f21, f22, f24, f27, f28, f30, f31, File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx, t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx, y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx, D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx, I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx, N=ltluatex.dtx, O=ltfinal.dtx	$\backslash@@warning$ . . . . . g170
		$\backslash@Alph$ . . . . . m49, <u>m65</u>
		$\backslash@DeclareMathDelimiter$ . . . . . r676, <u>r695</u>
		$\backslash@DeclareMathSizes$ . . . . . o171, o172, o174
		$\backslash@Esphack$ . . . . . i96, G201, G223, G241
		$\backslash@IncludeInRelease$ . . . . . c57, c58
		$\backslash@IncludeInRelease$ . . . . . c55, c56, c57
		$\backslash@M$ . . . . . <u>b21</u> , b396, b397, d24, d26, i27, i28, i29, i30, i31, i32, i33, i34, i57, p393, p406, z283, A194, C56, F50, F83, F101, F113, F154, K171, K172, K232
		$\backslash@MM$ . . . . . <u>b21</u> , G418, K273
		$\backslash@Mi$ . . . . . <u>e3</u> , K143
		$\backslash@Mii$ . . . . . <u>e3</u> , G53, G122, G194, G216, G241, G315, K269, K1069, K1236
		$\backslash@Miii$ . . . . . <u>e3</u> , G55, G124, G317, K272
		$\backslash@Miv$ e6, G195, G201, G217, G223, K246
		$\backslash@Roman$ . . . . . m47, <u>m53</u>
		$\backslash@TeXversion$ . . . . . 1, 6, <u>a297</u> , g28
		$\backslash@acci$ . . . . . <u>s168</u> , B236

File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx, t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx, y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx, D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx, I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx, N=ltluatex.dtx, O=ltfinal.dtx

- \@accii ..... [s168](#), [B236](#)
- \@acciii ..... [s168](#), [B236](#)
- \@acol ..... [C141](#),  
[C151](#), [C221](#), [C222](#), [C234](#), [C235](#),  
[C238](#), [C255](#), [C268](#), [C276](#), [C286](#)
- \@acolampacol ..... [C219](#), [C236](#),  
[C238](#), [C245](#), [C253](#), [C285](#), [C288](#)
- \@activechar@info ..... [K549](#)
- \@addamp ..... [C212](#), [C221](#),  
[C222](#), [C237](#), [C251](#), [C286](#), [C287](#)
- \@addfield ..... [C43](#),  
[C53](#), [C75](#), [C82](#), [C114](#), [C125](#), [C127](#)
- \@addmarginpar ..... [K305](#), [K1721](#)
- \@addtobot ..... [K885](#), [K972](#),  
[K1039](#), [K1091](#), [K1200](#), [K1259](#)
- \@addtocurcol .... [K302](#), [K976](#), [K1875](#)
- \@addtodblcol ..... [K764](#), [K1472](#)
- \@addtofilelist [a96](#), [a98](#), [k54](#), [k162](#),  
[k200](#), [s125](#), [s143](#), [s147](#), [s154](#),  
[s157](#), [s164](#), [s167](#), [O130](#), [O133](#), [O308](#)
- \@addtonextcol .. [K763](#), [K1296](#), [K1876](#)
- \@addtopreamble .... [C270](#), [C283](#),  
[C289](#), [C290](#), [C291](#), [C293](#), [C305](#)
- \@addtoreset ..... [m16](#), [m39](#), [m44](#)
- \@addtotoporbot ..... [K922](#),  
[K1085](#), [K1253](#), [K1345](#), [K1434](#)
- \@afterheading ..... [F75](#), [F108](#)
- \@afterindentfalse ..... [F28](#)
- \@afterindenttrue .. [F26](#), [F107](#), [F153](#)
- \@alph ..... [m48](#), [m61](#), [G379](#)
- \@ampacol .... [C219](#), [C236](#), [C247](#), [C288](#)
- \@arabic ..... [m43](#), [m45](#), [m51](#), [G377](#)
- \@argarraycr ..... [C176](#), [C177](#)
- \@argdef ..... [d57](#)
- \@argsbox ..... [B348](#)
- \@argtabularcr ..... [C183](#), [C184](#)
- \@array ..... [C154](#), [C155](#)
- \@arrayacol ..... [C141](#), [C219](#)
- \@arrayclassiv ..... [C142](#), [C290](#)
- \@arrayclassz ..... [C141](#), [C236](#)
- \@arraycr ..... [C143](#), [C174](#), [C176](#)
- \@arrayparboxrestore [B231](#), [B245](#), [C343](#)
- \@arrayrule ..... [C268](#),  
[C270](#), [C274](#), [C276](#), [C278](#), [C305](#)
- \@arstrut ..... [C165](#), [C198](#), [C302](#)
- \@arstrutbox . [C158](#), [C191](#), [C302](#), [C344](#)
- \@author ..... [F5](#)
- \@auxout .... [k81](#), [k87](#), [k103](#), [k118](#),  
[x33](#), [F145](#), [I7](#), [I8](#), [I19](#), [I29](#), [I37](#), [I43](#)
- \@backslashchar .....  
.... [d195](#), [g189](#), [g191](#), [t185](#), [L466](#)
- \@badcrrr ..... [g231](#)
- \@badend ..... [g202](#), [y65](#)
- File Key: [a=ltdirchk.dtx](#), [b=ltplain.dtx](#), [c=ltvers.dtx](#), [d=ltdefns.dtx](#),  
[e=ltalloc.dtx](#), [f=ltcntrl.dtx](#), [g=lterror.dtx](#), [h=ltpar.dtx](#), [i=ltspace.dtx](#),  
[j=ltlogos.dtx](#), [k=ltfiles.dtx](#), [l=ltoutenc.dtx](#), [m=ltcounts.dtx](#), [n=ltlength.dtx](#),  
[o=ltfssbas.dtx](#), [p=ltfssstrc.dtx](#), [q=ltfsscmp.dtx](#), [r=ltfssdcl.dtx](#), [s=ltfssini.dtx](#),  
[t=fontdef.dtx](#), [u=preload.dtx](#), [v=ltfntcmd.dtx](#), [w=ltpageno.dtx](#), [x=ltxref.dtx](#),  
[y=ltmiscen.dtx](#), [z=ltmath.dtx](#), [A=ltlists.dtx](#), [B=ltboxes.dtx](#), [C=lttab.dtx](#),  
[D=ltpictur.dtx](#), [E=ltthm.dtx](#), [F=ltsect.dtx](#), [G=ltfloat.dtx](#), [H=ltidxglo.dtx](#),  
[I=ltbibl.dtx](#), [J=ltpage.dtx](#), [K=ltoutput.dtx](#), [L=ltclass.dtx](#), [M=lthyphen.dtx](#),  
[N=ltluatex.dtx](#), [O=ltfinal.dtx](#)
- \@badlinearg ..... [g221](#), [D58](#),  
[D67](#), [D68](#), [D72](#), [D116](#), [D121](#), [D132](#)
- \@badmath .. [g205](#), [z172](#), [z174](#), [z179](#),  
[z181](#), [z189](#), [z201](#), [z206](#), [z215](#),  
[z227](#), [z232](#), [z323](#), [z335](#), [z351](#), [z360](#)
- \@badpoptabs ..... [g209](#), [C74](#), [C136](#)
- \@badrequireerror ..... [L131](#), [L413](#)
- \@badtab ..... [g212](#),  
[C22](#), [C76](#), [C97](#), [C103](#), [C110](#), [C133](#)
- \@begin@tempboxa .....  
[B26](#), [B41](#), [B155](#), [B217](#), [B349](#), [B357](#)
- \@begindocumenthook .....  
..... [k48](#), [k51](#), [L372](#), [L386](#), [I33](#)
- \@begindvi ..... [K594](#), [K620](#)
- \@begindvibox ..... [K93](#), [K621](#)
- \@beginparpenalty .....  
.. [i30](#), [z326](#), [z338](#), [z364](#), [A23](#), [A170](#)
- \@begintheorem ..... [E30](#), [E35](#)
- \@bezier ..... [D311](#), [D312](#)
- \@bibitem ..... [I3](#), [I8](#)
- \@biblabel ..... [I4](#), [I54](#)
- \@bitor ..... [K15](#),  
[K791](#), [K811](#), [K847](#), [K870](#), [K937](#),  
[K1021](#), [K1031](#), [K1179](#), [K1190](#),  
[K1332](#), [K1419](#), [K1537](#), [K1662](#)
- \@botlist ..... [K65](#),  
[K358](#), [K360](#), [K405](#), [K407](#), [K627](#),  
[K648](#), [K657](#), [K658](#), [K899](#), [K902](#),  
[K937](#), [K939](#), [K1031](#), [K1033](#),  
[K1190](#), [K1192](#), [K1832](#), [K1859](#)
- \@botnum ..... [G278](#),  
[K116](#), [K896](#), [K897](#), [K902](#), [K906](#),  
[K913](#), [K1368](#), [K1373](#), [K1461](#),  
[K1468](#), [K1824](#), [K1851](#), [K1893](#)
- \@botroom ..... [G279](#),  
[K117](#), [K899](#), [K902](#), [K1825](#), [K1852](#)
- \@boxfpsbit .... [K1941](#), [K1943](#), [K1948](#)
- \@break@tfor ..... [f31](#), [k157](#), [v81](#)
- \@bsphack ..... [i9](#), [i63](#), [i218](#),  
[i234](#), [x32](#), [G52](#), [G121](#), [G314](#),  
[H6](#), [H18](#), [H23](#), [H35](#), [K1792](#), [I39](#)
- \@caption ..... [G12](#), [G14](#)
- \@capttype ..... [G5](#), [G9](#),  
[G12](#), [G40](#), [G88](#), [G109](#), [G157](#), [K1905](#)
- \@car ..... [36](#), [d40](#), [j14](#), [I77](#)
- \@carcube ..... [d42](#), [d112](#)
- \@cclv ..... [b16](#), [K274](#), [K278](#),  
[K356](#), [K357](#), [K386](#), [K403](#), [K404](#),  
[K433](#), [K457](#), [K461](#), [K462](#), [O30](#), [O51](#)
- \@ccarvi ..... [b21](#), [b57](#), [b68](#), [b77](#),  
[b79](#), [b83](#), [b143](#), [b157](#), [N30](#), [N58](#)
- \@cdr ..... [36](#), [d40](#), [d287](#), [d288](#)
- \@centercr ..... [y68](#), [y76](#), [y83](#), [y89](#)

- \@centering ..... [z253](#),  
                   [z254](#), [z261](#), [z264](#), [z267](#), [z392](#), [z396](#)  
 \@cflb ..... [K624](#)  
 \@cflt ..... [K624](#)  
 \@changed@cmd .. [l3](#), [l63](#), [l173](#), [o96](#), [o232](#)  
 \@changed@x ..... [l3](#), [l161](#), [l169](#)  
 \@changed@x@mouth ..... [l161](#), [l169](#)  
 \@charlb ..... [k121](#), [k129](#)  
 \@charrb ..... [k123](#), [k129](#)  
 \@chclass [C232](#), [C233](#), [C294](#), [C307](#), [C312](#)  
 \@check@c ..... [d166](#), [d168](#)  
 \@check@eq ..... [d172](#), [d173](#), [d177](#)  
 \@checkend ..... [y11](#), [y61](#), [y64](#)  
 \@chnum ..... [C240](#),  
                   [C259](#), [C294](#), [C309](#), [C310](#), [C311](#)  
 \@circ ..... [D283](#), [D284](#), [D285](#)  
 \@circle ..... [D271](#), [D272](#)  
 \@circlefnt ..... [D37](#), [D39](#),  
                   [D232](#), [D250](#), [D276](#), [D291](#), [D306](#)  
 \@cite ..... [l16](#), [l52](#)  
 \@cite@ofmt ..... [l24](#), [l53](#)  
 \@citea ..... [l15](#), [l17](#)  
 \@citeb ..... [l16](#), [l18](#), [l19](#),  
                   [l20](#), [l23](#), [l24](#), [l41](#), [l42](#), [l43](#), [l44](#), [l45](#)  
 \@citex ..... [l13](#), [l14](#)  
 \@classi ..... [C232](#), [C266](#)  
 \@classii ..... [C232](#), [C280](#)  
 \@classiii ..... [C232](#), [C285](#)  
 \@classiv ..... [C142](#), [C153](#), [C233](#)  
 \@classoptionslist .....  
                   [l9](#), [l160](#), [l171](#), [l288](#), [l289](#), [l524](#)  
 \@classv ..... [C233](#), [C291](#)  
 \@classz ..... [C141](#), [C152](#), [C232](#)  
 \@cline ..... [C326](#)  
 \@clnht ..... [D74](#), [D75](#), [D83](#),  
                   [D85](#), [D87](#), [D97](#), [D104](#), [D130](#), [D300](#)  
 \@clnwd [D76](#), [D82](#), [D86](#), [D88](#), [D89](#), [D300](#)  
 \@cls@pkg ..... [l94](#), [l95](#),  
                   [l323](#), [l352](#), [l389](#), [l398](#), [l400](#), [l417](#)  
 \@clsextension .....  
                   ... [l16](#), [l41](#), [l52](#), [l70](#), [l101](#),  
                   [l127](#), [l144](#), [l160](#), [l170](#), [l210](#),  
                   [l225](#), [l233](#), [l287](#), [l356](#), [l364](#), [l390](#)  
 \@clubpenalty .....  
                   . [k9](#), [k19](#), [A128](#), [A196](#), [F89](#), [F118](#)  
 \@colht ..... [k16](#), [G277](#),  
                   [G279](#), [G282](#), [G288](#), [G289](#),  
                   [G302](#), [G303](#), [K121](#), [K205](#), [K216](#),  
                   [K225](#), [K226](#), [K361](#), [K373](#), [K408](#),  
                   [K421](#), [K448](#), [K479](#), [K509](#), [K515](#),  
                   [K519](#), [K529](#), [K534](#), [K616](#), [K687](#),  
                   [K725](#), [K769](#), [K794](#), [K813](#), [K853](#),  
                   [K875](#), [K1552](#), [K1678](#), [K2006](#), [O61](#)  
 \@colnum ..... [G280](#), [K118](#),  
                   [K905](#), [K950](#), [K1019](#), [K1020](#),  
                   [K1048](#), [K1056](#), [K1098](#), [K1177](#),  
                   [K1178](#), [K1210](#), [K1222](#), [K1266](#),  
                   [K1330](#), [K1331](#), [K1368](#), [K1373](#),  
                   [K1417](#), [K1418](#), [K1460](#), [K1467](#),  
                   [K1820](#), [K1847](#), [K1886](#), [K2061](#)  
 \@colroom ..... [k17](#),  
                   [K122](#), [K226](#), [K247](#), [K248](#), [K259](#),  
                   [K262](#), [K361](#), [K408](#), [K687](#), [K904](#),  
                   [K949](#), [K1015](#), [K1018](#), [K1047](#),  
                   [K1172](#), [K1176](#), [K1209](#), [K1326](#),  
                   [K1329](#), [K1412](#), [K1416](#), [K1821](#),  
                   [K1848](#), [K2016](#), [K2021](#), [K2066](#), [O60](#)  
 \@combinedblfloats [K660](#), [K2119](#), [K2158](#)  
 \@combinefloats ..... [K475](#), [K624](#)  
 \@comdblfleft ..... [K660](#)  
 \@comflelt ..... [K630](#), [K646](#), [K660](#)  
 \@cons ..... [36](#), [b180](#), [b197](#), [d39](#),  
                   [m44](#), [G193](#), [G215](#), [G239](#), [G359](#),  
                   [K211](#), [K798](#), [K817](#), [K833](#), [K857](#),  
                   [K859](#), [K879](#), [K881](#), [K1051](#),  
                   [K1119](#), [K1215](#), [K1288](#), [K1361](#),  
                   [K1451](#), [K1554](#), [K1577](#), [K1680](#),  
                   [K1705](#), [K1722](#), [K1723](#), [K2067](#)  
 \@contfield ..... [C50](#), [C126](#), [C138](#)  
 \@ctrerr .. [g198](#), [m64](#), [m68](#), [m82](#), [m90](#)  
 \@curfield ..... [C16](#), [C41](#),  
                   [C47](#), [C51](#), [C52](#), [C54](#), [C119](#), [C120](#)  
 \@curline .....  
                   [C16](#), [C27](#), [C39](#), [C44](#), [C53](#), [C54](#),  
                   [C55](#), [C79](#), [C80](#), [C92](#), [C117](#), [C118](#)  
 \@curr@enc ..... [l114](#), [l116](#)  
 \@currbox ..... [b247](#), [b248](#), [b249](#),  
                   [G60](#), [G91](#), [G95](#), [G129](#), [G160](#),  
                   [G164](#), [G193](#), [G214](#), [G215](#),  
                   [G239](#), [G261](#), [G263](#), [G265](#),  
                   [G323](#), [G326](#), [G331](#), [G335](#), [K187](#),  
                   [K188](#), [K199](#), [K200](#), [K202](#), [K203](#),  
                   [K211](#), [K285](#), [K286](#), [K763](#), [K764](#),  
                   [K1012](#), [K1014](#), [K1022](#), [K1045](#),  
                   [K1049](#), [K1051](#), [K1066](#), [K1107](#),  
                   [K1119](#), [K1167](#), [K1170](#), [K1207](#),  
                   [K1212](#), [K1215](#), [K1232](#), [K1277](#),  
                   [K1288](#), [K1320](#), [K1336](#), [K1350](#),  
                   [K1361](#), [K1403](#), [K1440](#), [K1451](#),  
                   [K1491](#), [K1495](#), [K1506](#), [K1512](#),  
                   [K1514](#), [K1518](#), [K1523](#), [K1532](#),  
                   [K1541](#), [K1547](#), [K1554](#), [K1577](#),  
                   [K1612](#), [K1616](#), [K1628](#), [K1635](#),  
                   [K1637](#), [K1641](#), [K1647](#), [K1657](#),  
                   [K1672](#), [K1680](#), [K1705](#), [K1723](#),  
                   [K1732](#), [K1911](#), [K1912](#), [K1941](#),  
                   [K1971](#), [K1976](#), [K2022](#), [K2025](#),  
 File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=lt pictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx

K2037, K2045, K2062, K2067  
 \@currdir 1, 6, a103, a125, a127, a133,  
     a135, a141, a143, a148, a150,  
     a160, a173, a238, a251, a264, L443  
 \@currentcmd ..... l25, o236  
 \@currentlabel ..... x34,  
     x37, x40, z257, z377, B298, G420  
 \@currenvir .....  
     g203, y3, y55, y65, A112, B102,  
     L460, L466, L474, L478, L484  
 \@currenvline ... g203, y56, y66, B103  
 \@current L15, L23, L31, L100, L101,  
     L144, L153, L160, L170, L220,  
     L229, L314, L315, L320, L321,  
     L326, L332, L336, L338, L340,  
     L342, L344, L345, L348, L354,  
     L356, L364, L382, L390, L406, L407  
 \@currlist .....  
     G193, G215, G359, K67, K285,  
     K362, K365, K409, K412, K1722  
 \@currname ..... c60, c67,  
     k211, k212, L14, L22, L30, L92,  
     L94, L100, L153, L229, L313,  
     L315, L338, L340, L342, L344,  
     L345, L382, L398, L400, L407, L417  
 \@currnamestack ..... L20  
 \@curroptions .....  
     ... L153, L161, L183, L407, L408  
 \@currsiz e ..... s72  
 \@currtype ..... K126,  
     K788, K789, K790, K791, K808,  
     K809, K810, K811, K937,  
     K1021, K1031, K1179, K1190,  
     K1332, K1419, K1537, K1662,  
     K1911, K1913, K1914, K1917  
 \@curtab ..... C11,  
     C26, C75, C76, C77, C83, C84,  
     C87, C91, C92, C96, C131, C132  
 \@curtabmar ..... C11, C25,  
     C26, C38, C44, C78, C91, C95, C96  
 \@d@r ..... a156, a157  
 \@dashbox ..... D175, D176,  
     D177, D178, D179, D182, D185,  
     D187, D196, D198, D199, D200,  
     D201, D204, D207, D210, D302  
 \@dashcnt ..... D169, D170,  
     D171, D172, D173, D174, D184,  
     D186, D189, D190, D191, D192,  
     D194, D195, D206, D209, D302  
 \@dashdim ..... D168, D169, D170,  
     D171, D173, D176, D178, D179,  
     D180, D184, D186, D188, D189,  
     D190, D191, D194, D198, D200,  
     D201, D202, D208, D211, D302  
 File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=ltterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx  
 \@date ..... F7  
 \@dbflt ..... G32, G268  
 \@dblarg .... 35, d311, F37, F125, G12  
 \@dbldeferlist ..... G239,  
     K74, K419, K424, K426, K726,  
     K733, K734, K1662, K1665,  
     K1705, K1707, K1836, K1864  
 \@dblfloat ..... G31  
 \@dblfloatplacement .....  
     k25, G275, G284, K375, K423,  
     K1817, K1844, K2124, K2164  
 \@dblflset ..... G26  
 \@dblpbot ..... G294, G308, K2208  
 \@dblpsep ..... G293, G307, K2208  
 \@dblpftop ..... G292, G306, K2208  
 \@dbltoplist .....  
     K69, K206, K209, K211, K371,  
     K372, K419, K420, K665, K669,  
     K671, K672, K1549, K1554,  
     K1674, K1680, K1835, K1862  
 \@dbltopnum ..... G287, G301,  
     K114, K134, K212, K214, K676,  
     K1488, K1489, K1553, K1556,  
     K1564, K1584, K1589, K1609,  
     K1610, K1679, K1683, K1691,  
     K1712, K1717, K1828, K1855  
 \@dbltoproom .. G288, G290, G302,  
     G304, K115, K1491, K1494,  
     K1495, K1504, K1505, K1508,  
     K1511, K1514, K1518, K1522,  
     K1526, K1531, K1551, K1612,  
     K1615, K1616, K1625, K1626,  
     K1627, K1630, K1634, K1637,  
     K1641, K1646, K1650, K1655,  
     K1656, K1677, K1829, K1856  
 \@dec@text@cmd ..... l3  
 \@declaredoptions .....  
     L8, L134, L157, L173, L188, L370  
 \@declareoption ... L132, L133, L141  
 \@defaultsubs .. o394, o428, o440, y26  
 \@defaultunits ..... o179, o183,  
     o184, o185, o200, o262, p133, p135  
 \@defdefault@ds ... L132, L137, L142  
 \@deferlist ..... K68,  
     K74, K358, K367, K368, K371,  
     K376, K378, K384, K405, K414,  
     K416, K688, K696, K697, K708,  
     K713, K714, K1021, K1024,  
     K1119, K1121, K1179, K1182,  
     K1288, K1290, K1332, K1334,  
     K1361, K1363, K1419, K1421,  
     K1451, K1453, K1537, K1539,  
     K1577, K1579, K1834, K1861

`\@definecounter` ..... [m12](#), [m36](#), [z242](#), [A227](#), [A228](#),  
[A229](#), [A230](#), [E8](#), [E16](#), [G376](#), [G378](#)  
`\@depth` ..... [d13](#), [p145](#),  
[t464](#), [t465](#), [t467](#), [t468](#), [B324](#),  
[B367](#), [C160](#), [C192](#), [D106](#), [D157](#),  
[D160](#), [D175](#), [D182](#), [D345](#), [K1761](#)  
`\@dir` .... [a155](#), [a158](#), [a160](#), [a162](#), [a163](#)  
`\@dischyph` ..... [d11](#), [B235](#)  
`\@docclearpage` ..... [K270](#), [K345](#)  
`\@documentclasshook` ..... [L3](#), [L292](#)  
`\@doendpe` ..... [y62](#), [A123](#)  
`\@dofilelist` ..... [k209](#), [k225](#), [y21](#)  
`\@donoparitem` ..... [A144](#), [A158](#)  
`\@dot` ..... [D271](#), [D284](#)  
`\@dotsep` ..... [F160](#)  
`\@dottedtocline` ..... [F149](#)  
`\@downline` ..... [D154](#), [D158](#), [D163](#)  
`\@downvector` ..... [D125](#), [D163](#)  
`\@eha` ..... [d255](#), [g174](#), [g192](#),  
[g194](#), [g196](#), [g204](#), [g206](#), [g236](#),  
[k88](#), [l52](#), [l949](#), [l959](#), [o25](#), [o67](#),  
[o109](#), [o152](#), [o218](#), [o273](#), [p106](#),  
[r25](#), [r70](#), [r99](#), [r161](#), [r192](#), [r293](#),  
[r314](#), [r346](#), [r387](#), [r432](#), [r437](#),  
[r492](#), [r600](#), [r604](#), [r608](#), [r642](#),  
[r652](#), [r736](#), [r741](#), [r744](#), [r776](#),  
[r779](#), [r833](#), [r836](#), [r839](#), [r906](#),  
[r912](#), [v129](#), [y54](#), [K1786](#), [K1802](#), [I47](#)  
`\@ehb` ..... [g174](#), [g199](#), [g224](#),  
[g226](#), [g228](#), [K208](#), [K364](#), [K411](#)  
`\@ehc` ..... [d105](#),  
[d132](#), [g174](#), [g231](#), [g234](#), [g240](#),  
[g242](#), [y130](#), [y141](#), [z298](#), [A220](#), [F4](#)  
`\@ehd` . [g174](#), [g201](#), [g208](#), [g211](#), [g213](#),  
[g219](#), [r118](#), [C89](#), [C98](#), [G6](#), [L258](#)  
`\@elt` .... [d39](#), [k122](#), [m20](#), [m35](#), [K8](#),  
[K11](#), [K15](#), [K27](#), [K30](#), [K31](#), [K32](#),  
[K33](#), [K38](#), [K39](#), [K40](#), [K41](#), [K42](#),  
[K43](#), [K44](#), [K45](#), [K47](#), [K51](#), [K57](#),  
[K58](#), [K59](#), [K60](#), [K472](#), [K630](#),  
[K641](#), [K646](#), [K656](#), [K668](#), [K670](#),  
[K698](#), [K715](#), [K735](#), [K754](#), [K767](#),  
[K774](#), [K825](#), [K828](#), [K837](#), [K1808](#)  
`\@empty` ..... [f14](#)  
`\@emptycol` .....  
... [K172](#), [K219](#), [K222](#), [K251](#), [K255](#)  
`\@end@tempboxa` .....  
[B35](#), [B44](#), [B160](#), [B230](#), [B355](#), [B365](#)  
`\@enddocumenthook` .. [y10](#), [L372](#), [L387](#)  
`\@endfloatbox` [G190](#), [G211](#), [G236](#), [G248](#)  
`\@endparenv` ..... [A120](#), [A123](#)  
`\@endparpenalty` .....  
... [i31](#), [z327](#), [z339](#), [z365](#), [A23](#), [A124](#)  
**File Key:** a=lt`dirchk`.dtx, b=lt`plain`.dtx, c=lt`vers`.dtx, d=lt`defs`.dtx,  
e=lt`alloc`.dtx, f=lt`cntrl`.dtx, g=lt`error`.dtx, h=lt`par`.dtx, i=lt`space`.dtx,  
j=lt`logos`.dtx, k=lt`files`.dtx, l=lt`outenc`.dtx, m=lt`counts`.dtx, n=lt`length`.dtx,  
o=lt`ssbas`.dtx, p=lt`ssstrc`.dtx, q=lt`sscmp`.dtx, r=lt`ssdcl`.dtx, s=lt`ssini`.dtx,  
t=lt`fontdef`.dtx, u=lt`preload`.dtx, v=lt`fontcmd`.dtx, w=lt`pageno`.dtx, x=lt`xref`.dtx,  
y=lt`miscen`.dtx, z=lt`math`.dtx, A=lt`lists`.dtx, B=lt`boxes`.dtx, C=lt`ttab`.dtx,  
D=lt`pictur`.dtx, E=lt`thm`.dtx, F=lt`tsect`.dtx, G=lt`float`.dtx, H=lt`idxglo`.dtx,  
I=lt`bibl`.dtx, J=lt`page`.dtx, K=lt`output`.dtx, L=lt`class`.dtx, M=lt`hyphen`.dtx,  
N=lt`luatex`.dtx, O=lt`final`.dtx

`\@endpbox` ..... [C166](#),  
[C197](#), [C227](#), [C292](#), [C343](#), [C346](#)  
`\@endpfalse` ..... [y59](#), [A129](#),  
[A131](#), [A135](#), [A136](#), [A138](#), [B105](#)  
`\@endpeltrue` ..... [A138](#)  
`\@endpetrue` ..... [A124](#), [A126](#), [A134](#)  
`\@endtheorem` ..... [E13](#), [E19](#), [E25](#), [E35](#)  
`\@enlargepage` .. [K1771](#), [K1776](#), [K1778](#)  
`\@ensuredmath` ..... [z309](#), [z311](#)  
`\@enumctr` ..... [A234](#), [A237](#), [A238](#)  
`\@enumdepth` .. [A226](#), [A232](#), [A233](#), [A234](#)  
`\@eqcnt` ..... [z250](#),  
[z295](#), [z300](#), [z379](#), [z394](#), [z395](#), [z397](#)  
`\@eqncr` ... [z262](#), [z280](#), [z301](#), [z302](#), [z381](#)  
`\@eqnnum` .. [z244](#), [z245](#), [z299](#), [z313](#), [z372](#)  
`\@eqnset` ..... [z250](#), [z393](#)  
`\@eqnswfalse` ..... [z279](#)  
`\@eqnswtrue` .... [z252](#), [z258](#), [z300](#), [z378](#)  
`\@eqpen` ..... [z250](#), [z283](#), [z285](#), [z292](#)  
`\@err@` ..... [g37](#),  
[g41](#), [g44](#), [g52](#), [g64](#), [g68](#), [g71](#), [g79](#)  
`\@esphack` . [i11](#), [i69](#), [i223](#), [i240](#), [x35](#),  
[G365](#), [H17](#), [H19](#), [H34](#), [K1794](#), [I50](#)  
`\@evenfoot` ..... [J12](#), [J15](#), [K584](#)  
`\@evenhead` ..... [J12](#), [J15](#), [K583](#)  
`\@expandtwoargs` .....  
... [d193](#), [L74](#), [L159](#), [L173](#), [L197](#)  
`\@expast` ..... [C200](#), [C228](#)  
`\@failedlist` .....  
... [K752](#), [K775](#), [K791](#), [K798](#),  
[K811](#), [K817](#), [K833](#), [K847](#), [K870](#)  
`\@fcolmadefalse` ..... [K743](#)  
`\@fcolmadetrue` ..... [K831](#)  
`\@filef@und` ... [k144](#), [k154](#), [k162](#), [k172](#)  
`\@filelist` .....  
... [k53](#), [k199](#), [k200](#), [k211](#), [s125](#),  
[s143](#), [s154](#), [s164](#), [O130](#), [O292](#), [O308](#)  
`\@filesfalse` ..... [k64](#)  
`\@fileswith@ptions` ..... [L131](#),  
[L191](#), [L283](#), [L284](#), [L286](#), [L310](#), [L361](#)  
`\@fileswith@ptions` .....  
... [L278](#), [L279](#), [L281](#), [L285](#)  
`\@fileswithoptions` .....  
... [L210](#), [L217](#), [L225](#), [L276](#)  
`\@filestrue` ..... [k7](#)  
`\@finalstrut` . [B302](#), [B366](#), [C344](#), [G425](#)  
`\@firstampfalse` ... [C215](#), [C238](#), [C255](#)  
`\@firstamptrue` ..... [C223](#)  
`\@firstcolfirstmark` .....  
... [K2101](#), [K2102](#), [K2106](#)  
`\@firstcoltopmark` .... [K2099](#), [K2107](#)  
`\@firstcolumnfalse` .... [K2091](#), [K2136](#)  
`\@firstcolumntrue` .....  
... [k22](#), [K105](#), [K181](#), [K2110](#), [K2142](#)

- \@firstofone ..... [d188](#), [k47](#),  
[l68](#), [l113](#), [p300](#), [r53](#), [r81](#), [r142](#),  
[r172](#), [r687](#), [y9](#), [z307](#), [C331](#), [G10](#),  
[N67](#), [N103](#), [N111](#), [N133](#), [I18](#), [I42](#)
  - \@firstoftwo [a82](#), [d188](#), [d283](#), [d310](#),  
[k155](#), [l97](#), [l921](#), [l937](#), [m100](#),  
[m105](#), [r691](#), [x19](#), [J16](#), [L48](#), [L64](#), [L77](#)
  - \@firstttab [C2](#), [C63](#), [C64](#), [C65](#), [C95](#), [C107](#)
  - \@flcheckspace ... [K899](#), [K935](#), [K2012](#)
  - \@flfail ..... [K775](#),  
[K826](#), [K847](#), [K857](#), [K870](#), [K879](#)
  - \@float ..... [G26](#), [G32](#)
  - \@floatboxreset ... [G101](#), [G170](#), [G174](#)
  - \@floatpenalty .....  
... [G3](#), [G53](#), [G55](#), [G58](#), [G122](#),  
[G124](#), [G127](#), [G191](#), [G194](#),  
[G199](#), [G201](#), [G212](#), [G216](#),  
[G221](#), [G223](#), [G237](#), [G241](#),  
[G315](#), [G317](#), [G321](#), [G325](#), [G359](#)
  - \@floatplacement [k25](#), [G275](#), [K156](#),  
[K183](#), [K227](#), [K451](#), [K1818](#), [K1845](#)
  - \@flsetnum ..... [K896](#),  
[K932](#), [K1019](#), [K1177](#), [K1330](#),  
[K1417](#), [K1488](#), [K1609](#), [K1980](#)
  - \@flsettextmin ..... [K995](#),  
[K1147](#), [K1316](#), [K1399](#), [K1996](#)
  - \@flstop ..... [K1882](#)
  - \@flsucceed .....  
... [K768](#), [K776](#), [K825](#), [K859](#), [K881](#)
  - \@fltovf ..... [g227](#), [G93](#), [G162](#), [G326](#)
  - \@flupdates ..... [K902](#), [K947](#), [K2058](#)
  - \@flushglue .....  
[e17](#), [y77](#), [y83](#), [y90](#), [y103](#), [A76](#), [B242](#)
  - \@fnsymbol ..... [m50](#), [m69](#)
  - \@font@info ..... [o98](#), [o136](#), [o142](#),  
[o300](#), [o317](#), [o476](#), [p30](#), [p38](#), [p46](#),  
[p74](#), [p87](#), [p126](#), [p154](#), [p168](#),  
[p179](#), [p193](#), [p209](#), [p215](#), [p228](#),  
[p235](#), [p242](#), [p247](#), [p257](#), [p269](#),  
[p281](#), [p445](#), [p457](#), [p462](#), [p469](#),  
[p494](#), [p502](#), [r202](#), [r217](#), [r251](#),  
[r297](#), [r366](#), [r372](#), [r416](#), [r429](#),  
[r512](#), [r591](#), [r633](#), [r726](#), [r874](#), [r903](#)
  - \@font@warning [o4](#), [o390](#), [o395](#), [o422](#),  
[o429](#), [p19](#), [p33](#), [p41](#), [p49](#), [p61](#),  
[p77](#), [p430](#), [p444](#), [p456](#), [p461](#),  
[p468](#), [p493](#), [p501](#), [q30](#), [y23](#), [O136](#)
  - \@fontswitch ..... [v109](#), [v111](#)
  - \@footnotemark .....  
... [G407](#), [G413](#), [G431](#), [G437](#), [G438](#)
  - \@footnotetext ..... [B272](#),  
[G407](#), [G413](#), [G414](#), [G447](#), [G453](#)
  - \@for [f16](#), [k99](#), [k211](#), [L78](#), [L157](#), [L171](#),  
[L183](#), [L188](#), [L203](#), [L408](#), [I16](#), [I41](#)
  - \@forloop ..... [f19](#), [f20](#)
  - \@fornoop ..... [f15](#), [f23](#), [f29](#)
  - \@fortmp ..... [f17](#), [f18](#), [f26](#)
  - \@fpbot .... [G294](#), [G308](#), [K773](#), [K2202](#)
  - \@fpmin ..... [G282](#), [G291](#), [G305](#),  
[K120](#), [K830](#), [K1826](#), [K1853](#), [K2073](#)
  - \@fps ..... [G41](#), [G42](#), [G44](#),  
[G47](#), [G64](#), [G110](#), [G111](#), [G113](#),  
[G116](#), [G133](#), [K1903](#), [K1905](#), [K1908](#)
  - \@fpsadddefault .....  
... [G45](#), [G48](#), [G114](#), [G117](#), [K1900](#)
  - \@fpsep ..... [G293](#), [G307](#),  
[K771](#), [K780](#), [K852](#), [K874](#), [K2202](#)
  - \@fpstype [K893](#), [K914](#), [K915](#), [K929](#),  
[K960](#), [K961](#), [K985](#), [K987](#), [K990](#),  
[K992](#), [K1043](#), [K1099](#), [K1100](#),  
[K1135](#), [K1138](#), [K1141](#), [K1144](#),  
[K1205](#), [K1267](#), [K1268](#), [K1306](#),  
[K1308](#), [K1311](#), [K1313](#), [K1387](#),  
[K1390](#), [K1393](#), [K1396](#), [K1485](#),  
[K1500](#), [K1502](#), [K1520](#), [K1529](#),  
[K1565](#), [K1566](#), [K1606](#), [K1621](#),  
[K1623](#), [K1643](#), [K1653](#), [K1692](#),  
[K1693](#), [K1896](#), [K1912](#), [K1914](#),  
[K1916](#), [K1919](#), [K1920](#), [K1921](#),  
[K1923](#), [K1924](#), [K1928](#), [K1929](#),  
[K1931](#), [K1932](#), [K1966](#), [K1968](#),  
[K1970](#), [K1982](#), [K1984](#), [K1998](#),  
[K2000](#), [K2030](#), [K2033](#), [K2044](#)
  - \@fptop .... [G292](#), [G306](#), [K770](#), [K2202](#)
  - \@frameb@x ..... [B132](#), [B159](#), [B161](#)
  - \@framebox ..... [B139](#), [B146](#), [B149](#)
  - \@framepicbox ..... [B139](#), [B146](#), [B182](#)
  - \@freelist . [b180](#), [b197](#), [b247](#), [G60](#),  
[G129](#), [G323](#), [G324](#), [K29](#), [K34](#),  
[K48](#), [K56](#), [K187](#), [K473](#), [K642](#),  
[K657](#), [K671](#), [K776](#), [K1722](#), [K1723](#)
  - \@getcirc ..... [D222](#), [D246](#), [D274](#)
  - \@getfpsbit .....  
[K890](#), [K926](#), [K1482](#), [K1603](#), [K1939](#)
  - \@getllarrow ..... [D123](#), [D131](#), [D133](#)
  - \@getlinechar ..... [D69](#), [D108](#)
  - \@getpen ..... [i7](#), [i10](#), [i21](#), [i55](#)
  - \@getrrarrow ..... [D124](#), [D131](#), [D140](#)
  - \@glossaryfile ..... [H21](#), [H22](#), [H31](#)
  - \@gnewline ..... [i46](#), [i48](#), [i49](#)
  - \@gobble ..... [d88](#),  
[d110](#), [d185](#), [d195](#), [d213](#), [d217](#),  
[d252](#), [d258](#), [d261](#), [d270](#), [f6](#), [f9](#),  
[g101](#), [g127](#), [g153](#), [g162](#), [i42](#),  
[i298](#), [k54](#), [k199](#), [l29](#), [l899](#), [o391](#),  
[o424](#), [p299](#), [q26](#), [r28](#), [r30](#), [r255](#),  
[r266](#), [r330](#), [r377](#), [r378](#), [r407](#),  
[r413](#), [r421](#), [r426](#), [r444](#), [r458](#),
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=ltterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx

- r468, r477, r490, r507, r516,  
 r594, r636, r729, r792, r865,  
 r896, s147, s157, s167, F126,  
 F127, F128, F129, F130, F146,  
 G7, K590, K591, K592, K837,  
 K1810, K2074, L246, L429,  
 L453, L458, N66, N101, O133,  
 O237, O243, O308, I11, I25, I26  
 \@gobble@IncludeInRelease c64, c71, c74  
 \@gobblecr ..... i296, i297  
 \@gobblefour ..... d185,  
 r24, r252, r368, r370, r374,  
 r376, r386, r390, r514, r566, L460  
 \@gobbletwo ..... d152, d153,  
 d185, f12, k26, o396, o430, r132,  
 y16, y24, J11, J13, L452, O142  
 \@gtempa .....  
 d103, d104, d158, d160, k180,  
 k181, k183, k184, k185, C3, C5,  
 C6, C7, C8, L91, L92, L102, L104  
 \@halfwidth ..... D2, D38,  
 D40, D41, D106, D156, D159,  
 D175, D182, D196, D206, D209,  
 D308, D330, D343, D344, D345  
 \@halignto .. C143, C147, C150, C164  
 \@hangfrom ..... F49, F100, F121  
 \@height ..... b390,  
 d13, i228, i236, l242, l244, p144,  
 t246, t464, t465, t467, t468,  
 B116, B121, B168, B178, B324,  
 B367, C159, C192, C318, C335,  
 D106, D157, D160, D175, D182,  
 D198, D205, D268, D344, K1761  
 \@highpenalty ..... i56, O3  
 \@hightab ... C11, C21, C23, C63,  
 C75, C84, C85, C100, C131, C132  
 \@hline ..... D60, D105, D122  
 \@holdpg ..... K129, K274,  
 K276, K277, K282, K283, K284  
 \@hspace ..... i282, i283  
 \@hspacecr ..... i282, i284  
 \@hvector ..... D118, D122  
 \@icentercr ..... y71, y72  
 \@iden ..... d191  
 \@if ..... d148, d149, d151  
 \@if@ptions ..... L74, L76, L89  
 \@if@ptions .. L69, L70, L73, L75, L321  
 \@ifatmargin ..... C55, C95  
 \@ifclasslater ..... 454, L51  
 \@ifclassloaded ..... 454, L40  
 \@ifclasswith ..... 454, L69  
 \@ifdefinable ..... 35, d61,  
 d63, d107, d109, d215, l14, l17,  
 m11, n3, s68, B69, E7, E15, E22  
 File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx  
 \@iffileonpath ..... k140, k148  
 \@ifl@aded .. L40, L41, L44, L50, L320  
 \@ifl@t@r ..... L56, L59, L66, L267  
 \@ifl@ter ..... l882,  
 l883, L51, L52, L55, L58, L348  
 \@ifl@ter@@ ..... l882, l883  
 \@ifnch ..... d293, d295, d307  
 \@ifnextchar .....  
 ... 35, a93, d289, d294, d310,  
 i44, i297, k163, m13, p365,  
 y70, z248, A143, B9, B11, B18,  
 B20, B25, B46, B75, B76, B82,  
 B83, B89, B93, B138, B139,  
 B145, B146, B150, B183, B191,  
 B199, B205, B209, B250, B254,  
 B258, B309, B314, B336, B343,  
 B347, C57, C154, C176, C183,  
 D10, D42, D53, D238, E3, E5,  
 E28, G27, G268, G328, G405,  
 G428, G445, K183, K1884,  
 L97, L262, L277, L282, I3, I13  
 \@iforloop ..... f21, f22  
 \@ifpackagelater ..... 454, L51  
 \@ifpackageloaded .. 454, K1868, L40  
 \@ifpackagewith ..... 454, L69  
 \@ifframebox ..... B151, B152, B153  
 \@ifframepicbox ..... B183, B184  
 \@ifstar 35, d50, d310, i38, i212, i282,  
 o171, q121, y69, y136, z282,  
 C56, C175, C182, D52, D271,  
 F35, F125, K1766, L132, L154  
 \@ifundefined ..... 35, d104,  
 d111, d131, d138, d160, d171,  
 d252, d258, d281, l901, m3, m7,  
 m16, o65, o151, p378, r287, x23,  
 y44, y53, E21, J3, J7, L38,  
 L122, L184, L490, L493, I20, I44  
 \@ignorefalse .... y4, y58, y63, G364  
 \@ignoretrue ..... i106,  
 i119, y4, y7, z241, z244, z276, z402  
 \@iiiminipage .....  
 .. B252, B256, B259, B260, B261  
 \@iiiparbox ..... B193, B201,  
 B207, B210, B211, B212, B289  
 \@iiiminipage ..... B255, B257  
 \@iinput ..... k163, k164  
 \@iiparbox ..... B206, B208  
 \@iirsbox ..... B347, B356  
 \@imakebox ..... B25, B40, B91  
 \@imakepicbox ... B46, B47, B96, B185  
 \@iminipage ..... B251, B253  
 \@include ..... k89, k90  
 \@index ..... H18, H19, H35  
 \@indexfile ..... H4, H5, H14

- `\inlabelfalse` [A28](#), [A104](#), [A184](#), [K166](#)
  - `\inlabeltrue` [A28](#), [A178](#)
  - `\inmatherr` [g237](#), [A112](#), [A142](#), [D271](#)
  - `\inmathwarn` [l3](#)
  - `\input` [k28](#), [k93](#), [k171](#), [F135](#)
  - `\input@` [k108](#), [k173](#), [o327](#), [I31](#)
  - `\inputcheck` [a65](#), [a186](#), [a187](#), [a190](#), [a198](#), [d25](#), [d32](#), [k3](#), [k135](#), [k136](#), [k143](#), [k152](#), [k153](#), [k156](#), [L440](#), [L441](#), [L448](#)
  - `\insertfalse` [K983](#), [K1133](#), [K1304](#), [K1385](#), [K1480](#), [K1601](#)
  - `\inserttrue` [K909](#), [K954](#), [K1071](#), [K1239](#), [K1559](#), [K1686](#)
  - `\invalidchar` [g242](#)
  - `\iparbox` [B192](#), [B200](#), [B204](#)
  - `\irsbox` [B336](#), [B343](#), [B347](#), [B348](#)
  - `\isavebox` [B89](#), [B90](#)
  - `\isavepicbox` [B94](#), [B95](#)
  - `\ishortstack` [D43](#), [D51](#)
  - `\istackcr` [D53](#), [D54](#)
  - `\itabcr` [C57](#), [C58](#)
  - `\item` [A143](#), [A156](#)
  - `\itemdepth` [A241](#), [A243](#), [A244](#), [A245](#)
  - `\itemfudge` [C38](#), [C44](#), [C71](#)
  - `\itemitem` [A245](#), [A248](#)
  - `\itemlabel` [A44](#), [A96](#), [A143](#)
  - `\itempenalty` [i32](#), [A23](#), [A175](#)
  - `\iwhiledim` [f7](#)
  - `\iwhilenum` [f3](#)
  - `\iwhilesw` [f10](#)
  - `\ixpt` [o504](#)
  - `\ixstackcr` [D52](#)
  - `\killglue` [D22](#), [D30](#), [D36](#)
  - `\kludgeins` [K293](#), [K294](#), [K295](#), [K297](#), [K350](#), [K351](#), [K397](#), [K398](#), [K476](#), [K492](#), [K493](#), [K499](#), [K500](#), [K501](#), [K510](#), [K526](#), [K530](#), [K540](#), [K1762](#), [K1793](#)
  - `\labels` [A27](#), [A146](#), [A147](#), [A189](#), [A206](#), [A207](#)
  - `\largefloatcheck` [G192](#), [G213](#), [G238](#), [G260](#)
  - `\lastchclass` [C223](#), [C233](#), [C234](#), [C236](#), [C244](#), [C267](#), [C281](#), [C285](#), [C294](#), [C307](#), [C308](#)
  - `\latex@error` [d105](#), [d132](#), [d253](#), [g136](#), [g172](#), [g188](#), [g194](#), [g196](#), [g199](#), [g201](#), [g203](#), [g206](#), [g208](#), [g210](#), [g213](#), [g217](#), [g222](#), [g226](#), [g228](#), [g230](#), [g231](#), [g233](#), [g236](#), [g240](#), [g242](#), [k88](#), [l50](#), [o6](#), [o25](#), [o67](#), [o109](#), [o152](#), [o218](#), [o273](#), [p105](#), [q100](#), [q111](#), [r23](#), [r68](#), [r97](#), [r117](#), [r159](#), [r190](#), [r213](#), [r229](#), [r293](#), [r314](#), [r346](#), [r386](#), [r390](#), [r432](#), [r437](#), [r492](#), [r560](#), [r566](#), [r600](#), [r604](#), [r608](#), [r642](#), [r652](#), [r736](#), [r741](#), [r744](#), [r776](#), [r779](#), [r833](#), [r836](#), [r839](#), [r906](#), [r912](#), [s50](#), [s100](#), [v126](#), [y54](#), [y129](#), [y141](#), [z298](#), [A219](#), [C89](#), [C98](#), [F4](#), [G6](#), [G83](#), [L221](#), [L240](#), [L253](#), [L322](#), [L397](#), [L414](#), [L422](#), [L427](#), [I47](#)
  - `\latex@info` [d201](#), [d272](#), [g136](#)
  - `\latex@info@no@line` [g136](#), [K550](#)
  - `\latex@warning` [g136](#), [g170](#), [l55](#), [x14](#), [D234](#), [G264](#), [K1906](#), [L477](#), [L483](#), [I22](#), [I45](#)
  - `\latex@warning@no@line` [d179](#), [g136](#), [g171](#), [k13](#), [k197](#), [x8](#), [x26](#), [x27](#), [y31](#), [F6](#), [K217](#), [K249](#), [K1737](#), [K1972](#), [L93](#), [L268](#), [L349](#), [L442](#), [L449](#), [L507](#)
  - `\latexbug` [g229](#), [K307](#), [K1723](#)
  - `\latexerr` [g170](#), [K208](#), [K364](#), [K411](#), [K1784](#), [K1801](#)
  - `\lbibitem` [I3](#), [I4](#)
  - `\ldots` [t412](#), [t414](#)
  - `\leftcolumn` [K128](#), [K2092](#), [K2113](#), [K2137](#), [K2146](#)
  - `\leftmark` [J16](#), [J36](#)
  - `\let@token` [d293](#), [d296](#), [d299](#), [d307](#), [i252](#), [i253](#), [i260](#), [v66](#), [v79](#), [z153](#), [z155](#), [z158](#)
  - `\lign` [z138](#), [z140](#)
  - `\linechar` [D69](#), [D70](#), [D71](#), [D75](#), [D76](#), [D78](#), [D83](#), [D85](#), [D86](#), [D87](#), [D88](#), [D90](#), [D94](#), [D95](#), [D98](#), [D99](#), [D104](#), [D129](#), [D298](#)
  - `\linefnt` [D37](#), [D39](#), [D69](#), [D122](#), [D130](#), [D161](#), [D164](#), [D305](#)
  - `\linelen` [D57](#), [D58](#), [D82](#), [D89](#), [D98](#), [D100](#), [D105](#), [D106](#), [D107](#), [D115](#), [D116](#), [D157](#), [D160](#), [D162](#), [D163](#), [D299](#)
  - `\listctr` [A202](#), [A225](#), [I9](#)
  - `\listdepth` [A23](#), [A35](#), [A38](#), [A43](#), [A99](#), [B273](#)
  - `\listfiles` [k52](#), [k203](#), [k218](#)
  - `\loadwithoptions` [L227](#), [L233](#), [L237](#)
  - `\lowpenalty` [i55](#), [O3](#)
  - `\ltab` [C60](#), [C95](#)
  - `\cm` [b21](#), [b350](#), [b352](#), [b353](#), [b386](#), [b387](#), [i170](#), [i274](#), [i279](#), [k39](#), [A80](#), [D92](#), [D96](#), [I17](#)
  - `\mainaux` [k5](#), [k31](#), [k32](#), [k81](#), [k93](#), [k118](#), [y15](#)
- File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx, t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx, y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx, D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx, I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx, N=ltluatex.dtx, O=ltfinal.dtx

- \@makebox ..... B11, B20, [B24](#)
  - \@makecaption ..... G24
  - \@makecol ... K235, K387, K434, [K454](#)
  - \@makefcolumn .....
    - ... K367, K368, K376, K378,
    - K414, K416, K424, K426, K2071
  - \@makefnmark ..... G380, G441
  - \@makefntext ..... B301, G424
  - \@makeoother ..... a71,
    - a92, a121, d313, d314, o340,
    - o341, o342, o343, o344, o345,
    - o346, o347, o348, o349, o350,
    - y113, [y123](#), y134, L117, L118, L465
  - \@makepicbox .... B10, B19, [B45](#), D211
  - \@makespecialcolbox .... K477, [K496](#)
  - \@marbox . G324, G326, G330, G334,
    - G335, G359, K1722, K1732,
    - K1735, K1743, K1745, K1746,
    - K1748, K1749, K1750, K1759
  - \@marginparreset ..... G343, [G350](#)
  - \@markright ..... J29, [J34](#)
  - \@maxdepth k50, [K98](#), K460, K488, O58
  - \@maxtab ..... C2, C83
  - \@medpenalty ..... i56, [O3](#)
  - \@midlist .....
    - K66, K473, K474, K937, K939,
    - K1051, K1215, K1833, K1860
  - \@minipagefalse .... A181, B246,
    - B248, B286, G187, G250, G345
  - \@minipagerestore ..... B274, [B276](#)
  - \@minipagetrue ..... B247, [G186](#)
  - \@minus ..... [d13](#), K2195,
    - K2196, K2197, K2200, K2201
  - \@missingfileerror .....
    - ..... [455](#), k167, [k174](#), L342
  - \@miv ..... e3
  - \@mkboth ..... J11, J13
  - \@mklab ..... A45, [A140](#)
  - \@mkpream ..... C162, C195, [C223](#)
  - \@mparbottom ..... G367,
    - G368, K125, K450, K1733,
    - K1741, K1742, K1743, K1744
  - \@mpargs ..... B265, B289
  - \@mparswitchfalse ..... K109
  - \@mpfn . B271, G405, G410, G450, [G454](#)
  - \@mpfootins ..... B280,
    - B281, B284, [B290](#), B293, B294
  - \@mpfootnotetext ..... B272, [B292](#)
  - \@mplistdepth ..... B273, [B290](#)
  - \@multicnt .....
    - C329, C331, C332, C333, C340,
    - C341, C342, D30, D31, D33,
    - [D295](#), D328, D330, D331, D332,
    - D333, D337, D341, D352, D356
  - \@multiplelabels k27, x25, [x31](#), y29, y35
  - \@multiput ..... D28, D29
  - \@multispan ..... C330, C334, [C338](#)
  - \@namedef ..... 35, [d37](#), k128,
    - l904, o100, o101, o125, p372,
    - x28, y121, z302, z303, C148,
    - E12, E13, E18, E19, E23, E24, E25
  - \@nameuse .....
    - 35, [d38](#), k116, k127, E23, J5, K578
  - \@nbitem ..... A168, [A221](#)
  - \@ne ..... b16
  - \@needsf@rmat ..... L263, L266, L271
  - \@needsformat ..... L251, L261, L265
  - \@negargfalse ..... D65
  - \@negargtrue ..... D64
  - \@newcommand ..... d56, [d57](#)
  - \@newctr ..... m13, [m15](#), E8
  - \@newenv ..... d127, d128, [d137](#)
  - \@newenva ..... d125, [d126](#)
  - \@newenvb ..... d127, [d128](#)
  - \@newl@bel ..... x22, y17, I10
  - \@newline ..... i45, [i47](#)
  - \@newlistfalse .....
    - ... [A29](#), [A33](#), A108, A182, K569
  - \@newlisttrue ..... [A29](#), [A33](#), A87
  - \@next ..... b247,
    - G60, G129, G323, G324, K9,
    - K187, K285, K787, K807, K1722
  - \@nextchar .....
    - .. C230, C231, C289, C290, C291
  - \@nil ..... a156, a157, c12,
    - c18, c61, c62, d40, d41, d42,
    - d112, d287, d288, f13, f19, f27,
    - j14, l77, o292, o303, o356, o459,
    - o462, o463, o471, p304, p305,
    - p307, p320, p326, p330, p331,
    - p367, p388, p393, p473, p487,
    - q26, q44, q53, q57, r40, r356,
    - r364, r397, r917, r919, v41, v45,
    - C326, C327, L27, L29, L60,
    - L61, L67, L202, L205, L299, L307
  - \@nmbolistfalse ..... A33, A46, A91
  - \@nmbolisttrue ..... A225
  - \@nnil ..... [f13](#), f20, f21,
    - f22, f28, o179, o183, o184, o185,
    - o200, p133, p135, p299, p301,
    - p313, p315, p320, p334, p336,
    - p343, p354, p355, p357, p388, p393
  - \@no@font@optfalse ..... q17, [q129](#)
  - \@no@lnbk ..... i13, i14, [i15](#)
  - \@no@pgbk ..... i3, i4, [i5](#)
  - \@nobreakfalse .....
    - ... i58, i60, A193, F77, F112,
    - F140, G182, K168, K1060, K1226
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx

`\@nobreaktrue` ..... i59, F109, G181  
`\@nocnterr` ..... g195  
`\@nocounterr` . g195, m4, m8, m16, E21  
`\@nodocument` ..... g200,  
                   k58, y50, G39, G108, K159, K186  
`\@noitemargfalse` ..... A32, A200  
`\@noitemargtrue` ..... A32, A143  
`\@noitemerr` ..... g232,  
                   i150, i185, i208, A69, A81, A107  
`\@noligs` ..... y114, y135, y151  
`\@nolnerr` ..... g193, i17, i51, y68  
`\@nomath` o2, o271, s35, s42, s63, s65, s70  
`\@noparitemfalse` ..... A30, A145  
`\@noparitemtrue` ..... A30, A66  
`\@noparlistfalse` ..... A31, A70  
`\@noparlisttrue` ..... A31, A67  
`\@normalcr` ..... i35, i43, B245  
`\@normalsize` ..... L4, L5  
`\@noskipsecfalse` .... k45, F81, K161  
`\@noskipsectrue` ..... F21, F78  
`\@notdefinable` d113, d114, d118, g187  
`\@notprerr` ..... g235, k56  
`\@nthm` ..... E3, E4  
`\@nxctabmar` ..... C11, C21, C23,  
                   C25, C64, C100, C101, C107, C108  
`\@obsoletefile` ..... k196  
`\@oddfoot` .. J11, J14, J15, K131, K581  
`\@oddhead` ..... J11, J14, K130, K581  
`\@onefilewithoptions` .....  
                   ... L291, L295, L301, L311, L360  
`\@onelevel@sanitize` . d315, G42, G111  
`\@onlypreamble` .. d43, d165, d167,  
                   d176, d184, k61, k70, k85, k198,  
                   k224, l23, l24, l61, l62, l66,  
                   l89, l109, l139, l140, l154, l905,  
                   o18, o80, o82, o88, o104, o132,  
                   o147, o168, o173, o215, o367,  
                   p373, q28, q36, q42, q79, q83,  
                   q88, q93, q98, q108, q126, q127,  
                   q128, q134, q138, q142, r17, r19,  
                   r44, r46, r107, r116, r136, r243,  
                   r244, r247, r279, r317, r319,  
                   r321, r334, r349, r396, r398,  
                   r440, r479, r495, r572, r611,  
                   r614, r655, r658, r661, r681,  
                   r694, r747, r782, r786, r789,  
                   r842, r862, r866, r930, v123,  
                   v124, x30, H12, H29, L10, L12,  
                   L18, L19, L26, L28, L34, L36,  
                   L39, L42, L43, L50, L53, L54,  
                   L58, L66, L68, L71, L72, L75,  
                   L89, L98, L106, L108, L125,  
                   L128, L129, L140, L141, L142,  
                   L149, L155, L168, L181, L193,  
                   L195, L200, L206, L211, L215,  
                   L218, L226, L231, L234, L238,  
                   L247, L260, L265, L271, L280,  
                   L285, L310, L360, L362, L371,  
                   L384, L385, L388, L395, L404,  
                   L411, L412, L420, L425, L430,  
                   L513, L514, L515, L516, L518, I40  
`\@opargbegintheorem` ..... E32, E35  
`\@opcol` ..... K236, K244,  
                   K368, K387, K416, K434, K439  
`\@options` ..... L194  
`\@othm` ..... E3, E20  
`\@outerparskip` .....  
                   ... A8, A88, A117, A152, A222  
`\@outputbox` ... K127, K457, K459,  
                   K479, K482, K483, K503, K505,  
                   K506, K511, K514, K519, K521,  
                   K528, K534, K536, K607, K633,  
                   K639, K649, K650, K673, K680,  
                   K766, K769, K772, K778, K779,  
                   K2092, K2096, K2097, K2111,  
                   K2117, K2137, K2143, K2152  
`\@outputdblcol` ..... K442,  
                   K2087, K2089, K2133, K2134  
`\@outputpage` .....  
                   ... K377, K426, K444, K564,  
                   K2121, K2126, K2159, K2167  
`\@oval` ..... D238, D239  
`\@ovbtrue` ..... D240  
`\@ovdx` ... D216, D248, D254, D256,  
                   D267, D269, D317, D318, D319,  
                   D320, D334, D335, D337, D351  
`\@ovdy` ... D216, D249, D255, D256,  
                   D261, D265, D324, D325, D326,  
                   D327, D338, D339, D341, D355  
`\@ovhorz` ..... D253, D254, D266  
`\@ovltrue` ..... D240  
`\@ovri` .. B32, D216, D247, D261, D270  
`\@ovro` ... D216, D247, D254, D255,  
                   D260, D265, D266, D275, D282  
`\@ovrtrue` ..... D240  
`\@ovttrue` ..... D240  
`\@ovvert` ..... D251, D252, D258  
`\@ovxx` ..... D216,  
                   D242, D244, D248, D252, D253,  
                   D266, D314, D315, D316, D320,  
                   D329, D330, D336, D337, D350  
`\@ovyy` ... D216, D243, D244, D249,  
                   D254, D258, D321, D322, D323,  
                   D327, D329, D340, D341, D354  
`\@p@filename` ..... L27, L29, L34  
`\@pagedp` ..... K124, K282, K287,  
                   K1001, K1154, K1751, K1761  
**File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
           e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
           j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
           o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
           t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
           y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
           D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
           I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
           N=ltluatex.dtx, O=ltfinal.dtx

- \@pageht ..... K123, K283,  
K287, K289, K290, K291, K295,  
K1000, K1153, K1734, K1741
- \@par ..... h3, h6
- \@parboxrestore ..... B217,  
B245, B270, B297, G19, G100,  
G169, G342, G419, K193, K570
- \@parboxto ..... B212
- \@parmoderr ... g225, G58, G127, G320
- \@parse@version .....  
.... c61, c62, L60, L61, L67, L68
- \@partaux ..... k5, k87, k103,  
k105, k106, k112, k121, k123, k126
- \@partlist ..... k84, k99
- \@partswfalse ..... k8
- \@partswtrue ..... k83
- \@pass@ptions .....  
... L120, L125, L126, L127, L336
- \@pboxswfalse ..... B215, B263
- \@pboxswtrue ..... B225
- \@penup ..... z129, z130
- \@percentchar ..... a101,  
L457, L459, L461, L463, L502
- \@picbox ..... D6, D13, D19, D20
- \@picht ..... D6, D12, D19
- \@picture ..... D10, D11
- \@picture@warn D102, D226, D230, D234
- \@pkgextension L16, L40, L51, L69,  
L126, L217, L220, L237, L302, L406
- \@plus ..... d13, i288, F16, F151,  
J40, K2195, K2196, K2197,  
K2200, K2201, K2205, K2206,  
K2207, K2211, K2212, K2213
- \@pnumwidth ..... F163
- \@popfilename ..... L20, L357
- \@pr@videpackage .... L97, L99, L106
- \@preamble ..... C163, C165,  
C173, C198, C217, C219, C220,  
C224, C239, C257, C258, C293
- \@preamblecmds .. d43, k57, L525, L526
- \@preammerr ... g214, C172, C235, C314
- \@process@ptions .....  
..... L167, L180, L182, L193
- \@process@ptions .. L154, L156, L168
- \@protected@testopt ..... d66, d78
- \@providesfile .. a93, a94, L109, O304
- \@ptionlist .....  
L37, L74, L153, L326, L332, L407
- \@pushfilename ..... L20, L312
- \@put ..... D237, D256, D282
- \@qend ..... d113, d287, g191
- \@qrelax ..... d114, d287
- \@rc@ifdefinable d107, d109, d215, l14
- \@reargdef ..... d99
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
- \@refundefined ..... k46, x3, y27
- \@reinserts ..... K301, K304, K490
- \@removeelement ..... f32, L197
- \@reqcolroom ..... K1000, K1001,  
K1004, K1006, K1007, K1012,  
K1016, K1018, K1046, K1047,  
K1153, K1154, K1158, K1161,  
K1162, K1167, K1174, K1176,  
K1208, K1209, K1320, K1322,  
K1324, K1327, K1329, K1403,  
K1406, K1409, K1414, K1416,  
K1896, K2013, K2018, K2021
- \@reset@ptions .... L317, L358, L363
- \@resetactivechars ..... K549, K567
- \@resetfhps .... K1115, K1284, K1963
- \@restorepar .....  
... 64, h6, i219, i235, A127, A135
- \@reversemarginfalse ... G368, K108
- \@reversemargintrue ..... G367
- \@rightmark ..... J16, J37
- \@rightskip ..... y79, y83, A75, B241
- \@rjfieldfalse ..... C34, C66
- \@rjfieldtrue ..... C114
- \@roman ..... m46, m52
- \@rsbox ..... B336, B343, B346
- \@rtab ..... C60, C75
- \@rule ..... B309, B314, B317
- \@sanitize .. d313, H7, H18, H24, H35
- \@savebox ..... B76, B83, B88
- \@savemarbox . G330, G331, G334, G337
- \@savepicbox ..... B76, B83, B92
- \@savsf ... i61, i67, i76, i89, i103, i117
- \@savsk ... i61, i66, i77, i90, i104, i118
- \@scolelt ..... K698, K763
- \@sdblcolelt ..... K715, K735, K764
- \@secntformat ..... F43, F94
- \@secondoftwo ..... a83, d188,  
d285, k149, i95, i923, i939, m99,  
m104, x21, J17, L46, L62, L84
- \@secpenalty ..... i33, F19, F33
- \@sect ..... F37, F38
- \@seqncr ..... z301
- \@setckpt ..... k121, k128, y16
- \@setfloattypcounts .....  
..... K984, K1134, K1305,  
K1386, K1481, K1602, K1910
- \@setfontsize ..... s70
- \@setfps ..... G34
- \@setfpsbit ..... G73, G75, G77,  
G85, G143, G146, G149, K1954
- \@setmarks .... K2103, K2105, K2120
- \@setminipage .....  
... B275, G21, G177, G185, G356  
G179, G355
- \@setnobreak ..... G179, G355

- \@setpar ..... 64, [h3](#), [A78](#)
- \@setref ..... [x10](#)
- \@setsize ..... [s70](#)
- \@settab ..... [C60](#), [C82](#)
- \@settodim ..... [n17](#)
- \@settopoint ..... [n22](#)
- \@sharp .. [C169](#), [C196](#), [C226](#), [C241](#),  
[C242](#), [C260](#), [C262](#), [C264](#), [C292](#)
- \@shipoutsetup ..... [K564](#)
- \@shortstack ..... [D42](#), [D43](#)
- \@sline ..... [D60](#), [D63](#), [D126](#)
- \@slowromancap ..... [m53](#), [m54](#)
- \@spaces ..... [g173](#)
- \@specialoutput ..... [K230](#)
- \@specialpagefalse ..... [K104](#), [K578](#)
- \@specialpagetrue ..... [J9](#)
- \@specialstyle ..... [J9](#), [K578](#)
- \@sptoken ..... [d296](#), [d306](#)
- \@sqrt ..... [z248](#)
- \@ssect ..... [F36](#), [F95](#)
- \@stackcr ..... [D49](#), [D52](#)
- \@star@or@long ..... [d49](#), [d54](#),  
[d101](#), [d123](#), [d129](#), [d155](#), [d164](#), [d198](#)
- \@startcolumn ..... [K237](#), [K244](#), [K685](#)
- \@startdblcolumn ..... [K685](#),  
[K2125](#), [K2128](#), [K2165](#), [K2171](#)
- \@startfield .....  
.. [C28](#), [C46](#), [C81](#), [C93](#), [C114](#), [C122](#)
- \@startline .. [C20](#), [C57](#), [C58](#), [C59](#), [C72](#)
- \@startpbox ..... [C166](#),  
[C197](#), [C227](#), [C291](#), [C343](#), [C345](#)
- \@startsection ..... [F22](#)
- \@starttoc ..... [F132](#)
- \@stopfield ..... [C32](#), [C48](#), [C59](#),  
[C75](#), [C82](#), [C114](#), [C116](#), [C125](#), [C127](#)
- \@stopline ..... [C30](#), [C56](#), [C74](#)
- \@stpelt ..... [m20](#), [m23](#)
- \@strip@args ..... [l74](#)
- \@svector ..... [D118](#), [D126](#)
- \@sverb ..... [y136](#), [y137](#), [y144](#)
- \@svsec ..... [F40](#), [F43](#), [F49](#), [F61](#)
- \@svsechd ..... [F59](#), [F84](#), [F104](#)
- \@sxverbatim ..... [y95](#), [y121](#)
- \@tabacckludge .. [l173](#), [l175](#), [l359](#), [l360](#)
- \@tabacol ..... [C151](#), [C219](#)
- \@tabarray ..... [C143](#), [C153](#), [C154](#)
- \@tabclassiv ..... [C153](#), [C289](#)
- \@tabclassz ..... [C152](#), [C243](#)
- \@tabcr ..... [C56](#), [C62](#)
- \@tabfbox ..... [C16](#), [C69](#), [C71](#)
- \@tablab ..... [C61](#), [C115](#)
- \@tabminus ..... [C61](#), [C106](#)
- \@tabplus ..... [C61](#), [C99](#)
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
- \@tabpush .....  
[C11](#), [C66](#), [C74](#), [C125](#), [C128](#), [C130](#)
- \@tabrj ..... [C61](#), [C113](#)
- \@tabular ..... [C147](#), [C150](#), [C151](#)
- \@tabularcr ..... [C153](#), [C181](#)
- \@tempboxa ..... [e13](#), [l69](#),  
[n17](#), [n18](#), [A205](#), [A211](#), [A212](#),  
[A214](#), [B28](#), [B29](#), [B30](#), [B31](#), [B36](#),  
[B37](#), [B38](#), [B39](#), [B128](#), [B157](#),  
[B164](#), [B174](#), [B266](#), [B289](#), [B352](#),  
[B353](#), [B354](#), [B361](#), [B362](#), [B363](#),  
[B364](#), [D161](#), [D162](#), [D232](#), [D233](#),  
[D247](#), [D250](#), [D255](#), [D256](#), [D275](#),  
[D276](#), [D281](#), [D282](#), [D342](#), [D360](#),  
[F121](#), [F122](#), [G326](#), [G360](#), [K279](#),  
[K351](#), [K356](#), [K357](#), [K398](#), [K403](#),  
[K404](#), [K540](#), [K597](#), [K604](#), [K605](#),  
[K631](#), [K635](#), [K647](#), [K653](#), [K660](#),  
[K661](#), [K662](#), [K663](#), [K667](#), [K675](#)
- \@tempcnta .....  
[e7](#), [r663](#), [r664](#), [r665](#), [r666](#), [r670](#),  
[C203](#), [C204](#), [C205](#), [C206](#), [D66](#),  
[D67](#), [D93](#), [D94](#), [D95](#), [D108](#),  
[D109](#), [D110](#), [D111](#), [D113](#), [D114](#),  
[D127](#), [D128](#), [D133](#), [D135](#), [D136](#),  
[D137](#), [D138](#), [D139](#), [D142](#), [D144](#),  
[D145](#), [D146](#), [D147](#), [D148](#), [D149](#),  
[D150](#), [D151](#), [D152](#), [D153](#), [D183](#),  
[D184](#), [D185](#), [D186](#), [D187](#), [D205](#),  
[D206](#), [D207](#), [D208](#), [D209](#), [D210](#),  
[D223](#), [D224](#), [D225](#), [D227](#), [D229](#),  
[D231](#), [D233](#), [D259](#), [D263](#), [D277](#),  
[D278](#), [D279](#), [D280](#), [D286](#), [D287](#),  
[D288](#), [D289](#), [D290](#), [D291](#), [D333](#),  
[D349](#), [G62](#), [G68](#), [G70](#), [G79](#),  
[G80](#), [G90](#), [G91](#), [G131](#), [G137](#),  
[G139](#), [G152](#), [G153](#), [G159](#),  
[G160](#), [K16](#), [K18](#), [K20](#), [K844](#),  
[K845](#), [K846](#), [K847](#), [K867](#), [K868](#),  
[K869](#), [K870](#), [K892](#), [K895](#), [K928](#),  
[K931](#), [K1042](#), [K1204](#), [K1484](#),  
[K1487](#), [K1605](#), [K1608](#), [K1723](#),  
[K1725](#), [K1728](#), [K1730](#), [K1732](#),  
[K1754](#), [K1944](#), [K1945](#), [K1949](#),  
[K1955](#), [K1959](#), [O74](#), [O79](#), [O80](#),  
[O81](#), [O149](#), [O154](#), [O155](#), [O156](#)
- \@tempcntb [e7](#), [r664](#), [r668](#), [r670](#), [D136](#),  
[D137](#), [D138](#), [D140](#), [D141](#), [D142](#),  
[D259](#), [D260](#), [D263](#), [D264](#), [G88](#),  
[G89](#), [G90](#), [G157](#), [G158](#), [G159](#),  
[K17](#), [K20](#), [K21](#), [K1955](#), [K1956](#),  
[K1957](#), [O75](#), [O79](#), [O150](#), [O154](#)
- \@tempdima .. [e10](#), [o184](#), [o189](#), [z116](#),  
[z119](#), [z125](#), [B42](#), [B43](#), [B156](#)

- B157, B162, B163, B164, B166,  
B216, B217, B264, B268, B320,  
B323, B324, B350, B352, B358,  
B361, C35, C36, C37, C77,  
C78, C79, C80, C191, C192,  
D89, D90, D92, D93, D94, D95,  
D96, D97, D222, D223, D224,  
D233, D248, D249, D251, D252,  
D278, D280, D285, D286, D287,  
F156, F157, F166, G196, G198,  
G218, G220, G262, G263,  
G264, K203, K204, K205, K461,  
K463, K509, K511, K512, K517,  
K522, K526, K531, K535, K827,  
K830, K850, K860, K872, K882,  
K1547, K1548, K1551, K1552,  
K1672, K1673, K1677, K1678,  
K1733, K1734, K1735, K1736,  
K1739, K1742, K1745, K1747,  
K2062, K2063, K2065, K2066
- \@tempdimb . e10, o185, o190, o479,  
o483, p133, p134, p391, p414,  
p415, p424, p425, p429, p447,  
p450, p453, p455, B219, B220,  
B321, B324, B351, B353, B359,  
B362, D90, D91, D244, D245,  
D246, D273, D274, D283, D284,  
K850, K851, K852, K853, K860,  
K872, K873, K874, K875, K882
- \@tempdimc . e10, p408, p409, p411,  
p412, p414, p415, B322, B323, B324
- \@tempskipa . e14, i19, i22, i23, i167,  
i174, i176, i179, p135, p136,  
A116, A117, A118, A150, A152,  
A153, A154, A222, A223, A224,  
F25, F27, F28, F33, F45, F46,  
F71, F72, F74, F86, F87, F96,  
F97, K1782, K1783, K1785, K1793
- \@tempskipb . e14, i126, i128, i130,  
i133, i135, i145, i165, i167, i168,  
i172, i174, i176, i177, i200, i203
- \@tempswafalse . . . . . a73, b235,  
k97, o59, r281, r336, r400, r481,  
r905, r911, y18, y105, K898,  
K934, K1490, K1611, L438, I13
- \@tempswatru . . . . . a74, b240, k95, k100, o62, r284,  
r339, r403, r484, r868, y42,  
y110, K1492, K1515, K1613,  
K1638, K2023, K2040, L437, I13
- \@temptokena . . . . . e16, y45,  
y46, J22, J23, J30, J31, J34, J35
- \@testdef . . . . . y17, y40
- \@testfalse . . . . . K12, K14, K15
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
- \@testfp . . . . . K792,  
K812, K848, K871, K1947, K2074
- \@testopt . . . . . d20, d56,  
d76, d80, d125, i3, i4, i13, i14, z288
- \@testpach . . . . . C231, C307
- \@testpatch . . . . . C307
- \@testtrue . . . . . K13, K21, K330,  
K795, K814, K854, K876, K1951
- \@testwrongwidth . . . . . K319,  
K793, K849, K1022, K1336, K1541
- \@text@composite . . . . . 174
- \@text@composite@x . . . . . 174
- \@textbottom . . . . . J40, J42, K485, K523, K537, K546
- \@textfloatsheight . . . . .  
. . . K450, K997, K999, K1049,  
K1050, K1055, K1150, K1152,  
K1212, K1214, K1220, K1896
- \@textmin G289, G290, G303, G304,  
K119, K999, K1003, K1006,  
K1007, K1152, K1157, K1161,  
K1162, K1324, K1409, K1508,  
K1510, K1526, K1630, K1632,  
K1650, K2004, K2006, K2008
- \@textsubscript . . . . .  
. . . . . G391, G392, G399, G402
- \@textsuperscript . G381, G383, G384
- \@texttop . J40, J42, K481, K504, K546
- \@tf@r . . . . . f25, f26
- \@tfor . . . . . f25, k150, k205,  
v71, B51, C229, D241, G63, G132
- \@tforloop . . . . . f27, f28, f30
- \@thanks . . . . . F10, F13
- \@thefnmark . . . . . B299,  
G380, G381, G406, G411,  
G421, G430, G435, G446, G451
- \@thefoot . . . K131, K581, K584, K611
- \@thehead . . . K130, K581, K583, K601
- \@themargin . . . K81, K582, K584, K596
- \@themark . J21, J22, J29, J30, J35, J38
- \@thirdofthree . . . . . d192, l147
- \@thm . . . . . E12, E18, E24, E26
- \@thmcounter . . . . . E11, E17, E33
- \@thmcountersep . . . . . E10, E33
- \@title . . . . . F3
- \@tocrmarg . . . . . F152
- \@toodeep . . . . . g207, A36, A232, A243
- \@toplist . . . . . K64, K358, K359,  
K405, K406, K626, K632, K642,  
K643, K935, K947, K1831, K1858
- \@topnewpage . . . . . K173
- \@topnum . . . . . G275,  
K112, K932, K933, K947, K951,

- K959, K1368, K1373, K1461,  
 K1468, K1822, K1849, K1890  
 \@toproom ..... G277,  
 K113, K935, K947, K1823, K1850  
 \@topsep ..... A1, A71, A73, A171  
 \@topsepadd . A1, A59, A61, A71, A124  
 \@totalleftmargin ..... y102,  
 A9, A53, A54, B240, C35, C65, C70  
 \@trivlist ..... A48, A57, A92  
 \@tryfcolumn .....  
 . K688, K708, K726, K742, K2075  
 \@trylist K751, K754, K787, K807, K829  
 \@twoclasseserror ..... L208, L426  
 \@twocolumnfalse ..... K106, K154  
 \@twocolumntrue ..... K180  
 \@twoloadclasserror ..... L356, L421  
 \@twosidefalse ..... K107  
 \@typein ..... d19, d20, d27, d35  
 \@typeset@protect .... d79, d220,  
 d227, d229, l26, l32, l160, l168, s71  
 \@uclclist .... l822, l823, l870, O234  
 \@undefined .. a63, a64, a103, a104,  
 a105, a126, a134, a142, a149,  
 a200, a204, a230, a237, a297,  
 a298, b89, b90, b105, b106,  
 b111, b120, b133, b168, b173,  
 b206, b207, b219, b229, b263,  
 b286, b289, b447, b490, b536,  
 b537, d21, d200, d278, g28,  
 k51, k52, k137, l145, l147, l282,  
 m113, o391, o424, o488, q4, q5,  
 q6, q7, q8, q9, q10, q11, q12,  
 q13, q14, q15, q16, q17, q18,  
 q19, q20, s44, v105, G5, G398,  
 G399, K36, K342, K343, L4,  
 L346, L372, L489, L492, L506,  
 N2, N13, N14, N15, N28, N30,  
 N77, N87, N140, N148, N156,  
 N164, N193, N194, N195, N196,  
 N197, N198, N199, N200, N201,  
 N202, N203, N204, N205, N206,  
 N207, N208, N209, N210, N211,  
 O10, O18, O25, O36, O66,  
 O67, O121, O197, O198, O258,  
 O293, O294, O295, O296, O297, I33  
 \@unexpandable@noexpand ..... d196  
 \@unexpandable@protect .....  
 d196, d232, d238, d243, k75, C225  
 \@unknownoptionerror L367, L396, L409  
 \@unprocessedoptions .....  
 ... L192, L236, L343, L347, L411  
 \@unused .. d4, g15, g32, g59, k3, L511  
 \@unusedoptionlist .....  
 . k12, k14, L11, L145, L146, L198  
 File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx  
 \@upline ..... D154, D155, D161  
 \@upordown D74, D75, D83, D104, D130  
 \@upvector ..... D125, D161  
 \@use@option .....  
 ... L163, L175, L185, L187, L196  
 \@use@text@encoding ..... l110, l1176  
 \@vbsphack ..... i125  
 \@verb ..... y136, y144  
 \@verbatim ..... y100, y118, y121  
 \@vereq ..... t365, t366  
 \@viipt ..... o503  
 \@vipt ..... o502  
 \@vpt ..... o501  
 \@vline ..... D59, D154  
 \@vobeyspaces ..... y93, y118, y144  
 \@vpt ..... o500  
 \@vspace ..... i212  
 \@vspacer ..... i212  
 \@vtryfc ..... K757, K765  
 \@vvector ..... D117, D125  
 \@warning ..... g170  
 \@wckptelt ..... k122, k125  
 \@whiledim ..... f7, D36, D82  
 \@whilenoop ..... f3  
 \@whilenum ..... f3, C205, D31,  
 D184, D186, D206, D209, D349  
 \@whilesw ..... f10, K238, K368,  
 K377, K415, K425, K2126, K2166  
 \@whileswnoop ..... f10  
 \@wholewidth ..... B113,  
 B115, B116, B118, B120, B121,  
 B122, B123, D2, D38, D40,  
 D41, D156, D159, D197, D204,  
 D262, D268, D307, D308, D346  
 \@width . b393, d13, i284, l240, l243,  
 p146, t522, B118, B120, B170,  
 B177, B324, B367, C161, C192,  
 C306, C325, D106, D156, D159,  
 D176, D183, D197, D204, D262,  
 D346, G375, K1761, K2115, K2149  
 \@wrglossary ..... H25, H30  
 \@wrindex ..... H8, H13  
 \@writeckpt ..... k110, k119  
 \@writefile ..... k26, y43, F147  
 \@writesetup ..... K564  
 \@wrong@font@char l121, o392, o426, o439  
 \@wtryfc ..... K767, K777  
 \@x@protect ..... d82, d219  
 \@x@sf ..... G440, G442  
 \@xDeclareMathDelimiter .. r693, r748  
 \@xaddvskip ..... i125, i146

<code>\@xarg</code> .....	D56, D59, D64, D68, D69, D105, D107, D112, D113, D117, D123, D131, <u>D292</u>	<code>\@yeqncr</code> .....	<u>z280</u>
<code>\@xargarraycr</code> .....	C178, C187, <u>C191</u>	<code>\@ympar</code> .....	G328, <u>G333</u>
<code>\@xargdef</code> .....	<u>d57</u>	<code>\@ynthm</code> .....	E5, <u>E14</u>
<code>\@xarraycr</code> .....	C175, C176	<code>\@ythm</code> .....	E28, <u>E29</u>
<code>\@xbitor</code> .....	K15, K17	<code>\@ytryfc</code> .....	K800, K819, <u>K823</u>
<code>\@xcentercr</code> .....	y69, <u>y70</u>	<code>\@yyarg</code> .....	D64, D65, D66, D69, D131, <u>D292</u>
<code>\@xdblarg</code> .....	<u>d311</u>	<code>\@ztryfc</code> .....	K828, <u>K839</u>
<code>\@xdblfloat</code> .....	<u>G268</u>	<code>\[</code> .....	<u>o346</u> , <u>z184</u> , z240, <u>z321</u> , O180
<code>\@xdim</code> .....	D26, D32, D34, <u>D296</u> , D350, D351, D352, D353, D359	<code>\]</code> .....	66, a45, a46, a69, a242, a243, a244, a245, a248, a255, a256, a257, a258, a261, a268, a269, a270, a271, a274, a281, a283, a284, a287, a290, b13, d195, d313, g231, <u>i35</u> , i295, k204, k219, l427, <u>o334</u> , t170, y76, y83, y89, y97, z262, z381, B245, B351, B353, C62, C143, C153, C167, D49, N346, O165
<code>\@xeqncr</code> .....	<u>z280</u>	<code>\{</code> .....	a3, a7, a69, b2, b13, g22, l257, l429, o335, t168, y96, z59, z108, O168
<code>\@exnoop</code> .....	C199, C209	<code>\}</code> .....	a8, a69, b3, b13, g21, l258, l430, o336, t169, y96, z59, O169
<code>\@expast</code> .....	C200, <u>C201</u>	<code>\]</code> ..	b439, o347, <u>z184</u> , z241, <u>z345</u> , O181
<code>\@xfloat</code> .....	G28, G29, <u>G34</u> , G270	<code>\^</code> ..	a58, a67, a70, a114, a301, b7, b9, b11, b14, b358, b359, b373, b374, d5, d314, i295, i297, i299, l181, l236, l290, l361, l368, l425, l508, l515, l519, l524, l529, l534, l541, l547, l548, l554, l559, l604, o332, o333, o338, L434, L435, L436, L488, L491, L494, O106, O107, O108, O109, O111, O112, O113, O114, O116, O166, O172, O173, O174, O175, O188, O189, O190, O222, O223, O224, O225, O227, O228, O229, O230, O232
<code>\@xfootnote</code> .....	G405, <u>G408</u>	<code>\_</code> .....	a70, b8, b14, d314, l263, t173, z166, z167, O167
<code>\@xfootnotemark</code> .....	G428, <u>G432</u>	<code>\`</code> .....	l182, l291, l325, l359, l366, l444, l506, l513, l517, l522, l527, l532, l539, l543, l544, l552, l557, l605, l644, o348, s168, y145, B236, C61, O182
<code>\@xfootnotenext</code> .....	G445, <u>G448</u>	<code>\ </code> .....	l428, m78, m89, t479, O183
<code>\@xhline</code> .....	C319, <u>C320</u>	<code>\~</code> .....	a70, b10, b14, d314, g20, i264, l189, l237, l292, l369, l426, l509, l521, l525, l535, l551, l555, l606, y139, y149, O170
<code>\@xifnch</code> .....	<u>d297</u> , <u>d307</u>	<code>\_</code> .....	a69, a86, b13, b358, b376, d313, g19, g20, g21, g22, g25, i263, o331, o497, t171, y93, y94, E36, E38, L111, O160, I17
<code>\@xiipt</code> .....	<u>o507</u> , t83, t85, t86		
<code>\@xipt</code> .....	<u>o506</u> , t82		
<code>\@xivpt</code> .....	<u>o508</u> , t84, t86		
<code>\@xmpar</code> .....	G328, <u>G329</u>		
<code>\@xnewline</code> .....	i39, i40, <u>i44</u>		
<code>\@xnext</code> .....	K10, K11		
<code>\@xnthm</code> .....	E5, <u>E6</u>		
<code>\@xobeysp</code> .....	<u>i262</u> , y94, y95		
<code>\@xprocess@ptions</code> ..	L154, L169, L181		
<code>\@xpt</code> .....	<u>o505</u> , t81, t84, t85		
<code>\@xssect</code> .....	F69, <u>F70</u> , F106		
<code>\@xtabcr</code> .....	C56, <u>C57</u>		
<code>\@xtabularcr</code> .....	C182, <u>C183</u>		
<code>\@xthm</code> .....	E28, <u>E29</u>		
<code>\@xtryfc</code> .....	K754, <u>K782</u>		
<code>\@xtypein</code> .....	d20, d22, d29		
<code>\@xverbatim</code> .....	y95, y118		
<code>\@xvipt</code> .....	<u>o509</u> , t85, t87		
<code>\@xxDeclareMathDelimiter</code> ..	r678, <u>r682</u>		
<code>\@xxpt</code> .....	<u>o510</u> , t86, t87		
<code>\@xxvpt</code> .....	<u>o511</u> , t87		
<code>\@xxxii</code> .....	e2, l330, l332, G89, G158, K789, K790, K809, K810, K845, K846, K868, K869, K1913		
<code>\@xympar</code> .....	G332, G336, <u>G358</u>		
<code>\@yarg</code> .....	D56, D60, D64, D65, D74, D112, D118, D125, D127, D154, <u>D292</u>		
<code>\@yargarraycr</code> .....	C179, C189, <u>C193</u>		
<code>\@yargd@f</code> .....	<u>d84</u>		
<code>\@yargdef</code> .....	d61, d71, <u>d84</u> , d100		
<code>\@ydim</code> .....	D27, D32, D34, D297, D354, D355, D356, D357, D358		

File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsssrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx, t=fontdef.dtx, u=preload.dtx, v=ltfontcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx, y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx, D=lt pict

## A

`\A` ..... O101, O185, O217  
`\a` ..... [1173](#), [C1](#), O90, O186, O206  
`\AA` ..... [b364](#), l190, l333, l396  
`\aa` ..... [b364](#), l195, l327, l406  
`\abovedisplayskip` .. b339, z389  
`\abovedisplayskip` ..... b338,  
                           [z382](#), [z384](#), [z386](#), [z387](#), [z388](#), [z389](#)  
`\accent` ..... l71, l301, l328, l381, l617  
`\accent@spacefactor` ..... l70, l71, l72  
`\active` ..... a59, a114, a301, b10,  
                           b11, b373, b374, b376, y93, y94,  
                           y138, y147, z151, z166, K558,  
                           L434, L435, L436, L488, L491, L494  
`\active@math@prime` .. [z150](#), [z151](#), K562  
`\acute` ..... t424  
`\add@accent` ..... l65, [l67](#)  
`\add_to_callback` ..... [475](#), [N548](#)  
`\addcontentsline` F53, F63, [F142](#), G16  
`\addpenalty` .... [i152](#), A124, A170,  
                           A175, F33, K312, K1063, K1229  
`\addto@hook` .....  
                           o117, o119, [o499](#), r263, r359,  
                           r363, r380, r504, r510, r518,  
                           r534, r537, r540, r877, r884, r887  
`\addtocontents` ..... F143, [F144](#)  
`\addtocounter` ..... [132](#), [m6](#), m18  
`\addtolength` .... [138](#), [n16](#), [z384](#), [z386](#)  
`\addtoversion` ..... q20, [q139](#)  
`\addvspace` ..... [i139](#), y70, A124,  
                           A171, A172, A176, A224, F33  
`\adjdemerits` ..... b318  
`\AE` ..... l191, l306, l397, l622, O247  
`\ae` ..... l196, l309, l407, l626, O247  
`\afterassignment` .... b389, b392,  
                           d233, d239, l162, l170, o262, z129  
`\aftergroup` ..... o56, o276,  
                           p156, p222, r114, r121, r129,  
                           v47, y142, B101, K565, K573, K574  
`\aleph` ..... t227  
`\alloc@` ..... b74, b75,  
                           b76, b77, b78, b79, b80, b81,  
                           b82, b83, [b210](#), o15, N20, N24, N38  
`\allocationnumber` [b37](#), b57, b127,  
                           b128, b129, b179, b180, b212,  
                           b213, b214, b225, b226, b227,  
                           b244, b249, b255, b256, b269,  
                           b270, b271, C4, C9, N52, N53,  
                           N54, N93, N171, O42, O43, O44  
`\allowbreak` ..... [b396](#), z40  
`\Alpha` ..... [132](#), [m49](#)  
`\alph` ..... [132](#), [m48](#)  
`\alpha` ..... t187  
**File Key:** a=lt`dirchk`.dtx, b=lt`plain`.dtx, c=lt`vers`.dtx, d=lt`defns`.dtx,  
           e=lt`alloc`.dtx, f=lt`cntrl`.dtx, g=lt`error`.dtx, h=lt`par`.dtx, i=lt`space`.dtx,  
           j=lt`logos`.dtx, k=lt`files`.dtx, l=lt`outenc`.dtx, m=lt`counts`.dtx, n=lt`length`.dtx,  
           o=lt`fssbas`.dtx, p=lt`fssstrc`.dtx, q=lt`fsscmp`.dtx, r=lt`fssdcl`.dtx, s=lt`fssini`.dtx,  
           t=lt`fontdef`.dtx, u=lt`preload`.dtx, v=lt`fontcmd`.dtx, w=lt`pageno`.dtx, x=lt`xref`.dtx,  
           y=lt`miscen`.dtx, z=lt`math`.dtx, A=lt`lists`.dtx, B=lt`boxes`.dtx, C=lt`ttab`.dtx,  
           D=lt`pictur`.dtx, E=lt`thm`.dtx, F=lt`ssect`.dtx, G=lt`float`.dtx, H=lt`idxglo`.dtx,  
           I=lt`bibl`.dtx, J=lt`page`.dtx, K=lt`output`.dtx, L=lt`class`.dtx, M=lt`hyphen`.dtx,  
           N=lt`luatex`.dtx, O=lt`final`.dtx

`\alpha@elt` .....  
                           r45, r267, r454, r556, r876, r877  
`\alpha@list` r41, [r43](#), r276, r442, r454,  
                           r499, r554, r555, r872, r878, r879  
`\amalg` ..... t293  
`\and` ..... [343](#), [F14](#)  
`\angle` ..... t243  
`\approx` ..... t333  
`\arabic` ..... [132](#), [m45](#), E33  
`\arccos` ..... z13  
`\arcsin` ..... z10  
`\arctan` ..... z16  
`\arg` ..... z26  
`\array` ..... [C141](#)  
`\arraycolsep` .....  
                           z265, z266, z394, z395, C219, [C297](#)  
`\arrayrulewidth` .....  
                           C283, [C297](#), C305, C306,  
                           C318, C322, C325, C335, C337  
`\arraystretch` .... C159, C160, [C301](#)  
`\Arrowvert` ..... t475  
`\arrowvert` ..... t473  
`\ast` ..... t151, t309  
`\asymp` ..... t357  
`\AtBeginDocument` . k47, [L381](#), I34, I48  
`\AtBeginDvi` ..... [K93](#)  
`\AtEndDocument` ..... y9, [L381](#)  
`\AtEndOfClass` ..... z320, [L381](#)  
`\AtEndOfPackage` ... [L192](#), [L236](#), [L381](#)  
`\atopwithdelims` ..... z57, z58, z59  
`\attribute` ..... N81  
`\attributedef` ..... N81, N177  
`\attributezero` ..... N177  
`\author` ..... [343](#), [F5](#)

## B

`\b` ..... l183, l297, l377, l613  
`\backslash` ..... t170, t494  
`\badness` ..... b297  
`\bar` ..... t428  
`\baselineskip` ..... b357, b387,  
                           b423, p140, p141, p142, p144,  
                           p145, t418, z112, z113, z121,  
                           z127, z131, B243, C171, D46,  
                           D166, K216, K247, K593, K608  
`\baselinestretch` .....  
                           o253, p118, p119, p138, [p199](#)  
`\batchmode` .....  
                           k183, k184, q106, s135, O267, O288  
`\begin` .... g201, g203, l570, p7, t4,  
                           u4, y51, y52, z325, z337, F14,  
                           F17, G256, G258, K70, L245, M3  
`\belowdisplayshortskip` .. b341, z388  
`\belowdisplayskip` ..... b340, z387

- \best@size ... [p392](#), [p416](#), [p422](#), [p428](#)
- \beta ... [t188](#)
- \bezier ... [318](#), [D310](#), [D311](#)
- \bfdefault ... [s15](#), [t32](#)
- \bfseries ... [s13](#), [s14](#), [v19](#), [x13](#), [E36](#), [E38](#), [I20](#)
- \bgroup ... [b371](#)
- \bibtex ... [I7](#), [I9](#), [I10](#)
- \bibdata ... [I25](#), [I29](#)
- \bibitem ... [I3](#)
- \bibliography ... [375](#), [I27](#)
- \bibliographystyle ... [375](#), [I32](#)
- \bibstyle ... [I25](#), [I37](#)
- \Big ... [t525](#), [z44](#), [z45](#), [z46](#)
- \big ... [t524](#), [z41](#)
- \bigbreak ... [b403](#)
- \bigcap ... [t263](#)
- \bigcirc ... [t306](#)
- \bigcup ... [t264](#)
- \Bigg ... [t527](#), [z50](#), [z51](#), [z52](#)
- \bigg ... [t526](#), [z47](#), [z48](#), [z49](#)
- \Biggl ... [z50](#)
- \biggl ... [z47](#)
- \Biggm ... [z51](#)
- \biggm ... [z48](#)
- \Biggr ... [z52](#)
- \biggr ... [z49](#)
- \Bigl ... [z44](#)
- \bigl ... [z41](#)
- \Bigm ... [z45](#)
- \bigm ... [z42](#)
- \bigodot ... [t271](#)
- \bigoplus ... [t270](#)
- \bigotimes ... [t269](#)
- \Bigr ... [z46](#)
- \bigr ... [z43](#)
- \bigskip ... [b408](#), [i242](#)
- \bigskipamount ... [b407](#), [i244](#), [i245](#), [G371](#)
- \bigsqcup ... [t274](#)
- \bigtriangledown ... [t279](#), [t280](#)
- \bigtriangleup ... [t278](#), [t281](#)
- \biguplus ... [t262](#)
- \bigvee ... [t260](#)
- \bigwedge ... [t261](#)
- \binoppenalty ... [b309](#)
- \bm@b ... [B36](#)
- \bm@c ... [B36](#)
- \bm@l ... [B36](#)
- \bm@r ... [B36](#)
- \bm@s ... [B36](#)
- \bm@t ... [B36](#)
- \bmod ... [z35](#)
- \boldmath ... [j14](#), [s63](#)
- \bordermatrix ... [z115](#)
- File Key: a=lttldirchk.dtx, b=lttplain.dtx, c=lttvers.dtx, d=lttdefs.dtx,  
e=lttalloc.dtx, f=lttctrl.dtx, g=ltterror.dtx, h=lttpar.dtx, i=lttspace.dtx,  
j=lttlogos.dtx, k=lttfiles.dtx, l=lttoutenc.dtx, m=lttcounts.dtx, n=lttlength.dtx,  
o=lttssbas.dtx, p=lttssstrc.dtx, q=lttssscmp.dtx, r=lttssdcl.dtx, s=lttssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=lttfontcmd.dtx, w=lttpagenc.dtx, x=lttxref.dtx,  
y=lttmiscen.dtx, z=lttmath.dtx, A=lttlists.dtx, B=lttboxes.dtx, C=ltttab.dtx,  
D=lttpictur.dtx, E=lttthm.dtx, F=lttsect.dtx, G=lttfloat.dtx, H=lttidxglo.dtx,  
I=lttbibl.dtx, J=lttpage.dtx, K=lttoutput.dtx, L=lttclass.dtx, M=ltthyphen.dtx,  
N=lttluatex.dtx, O=lttfinal.dtx
- \bot ... [t242](#)
- \botfigrule ... [K652](#), [K2214](#)
- \botmark ... [J36](#), [K618](#)
- \bottomfraction ... [G279](#), [K2183](#)
- \bowtie ... [t390](#)
- \Box ... [s106](#)
- \boxmaxdepth ... [b332](#), [D239](#), [D273](#), [K460](#),  
[K480](#), [K520](#), [K625](#), [K634](#), [K674](#)
- \brace ... [z59](#)
- \braced ... [t459](#), [t463](#), [t464](#), [t466](#), [t468](#)
- \bracelu ... [t461](#), [t465](#), [t467](#)
- \bracerd ... [t460](#), [t465](#), [t467](#)
- \braceru ... [t462](#), [t464](#), [t468](#)
- \bracevert ... [t512](#)
- \brack ... [z58](#)
- \break ... [b396](#), [b401](#), [i53](#)
- \breve ... [t429](#)
- \brokenpenalty ... [b314](#)
- \buildrel ... [t377](#), [z107](#)
- \bullet ... [t295](#)
- \bx@A ... [K30](#), [K57](#)
- \bx@AA ... [K40](#)
- \bx@B ... [K30](#), [K57](#)
- \bx@BB ... [K40](#)
- \bx@C ... [K30](#), [K57](#)
- \bx@CC ... [K40](#)
- \bx@D ... [K30](#), [K57](#)
- \bx@DD ... [K40](#)
- \bx@E ... [K30](#), [K57](#)
- \bx@EE ... [K40](#)
- \bx@F ... [K31](#), [K58](#)
- \bx@FF ... [K41](#)
- \bx@G ... [K31](#), [K58](#)
- \bx@GG ... [K41](#)
- \bx@H ... [K31](#), [K58](#)
- \bx@HH ... [K41](#)
- \bx@I ... [K31](#), [K58](#)
- \bx@II ... [K41](#)
- \bx@J ... [K31](#), [K58](#)
- \bx@JJ ... [K41](#)
- \bx@K ... [K32](#), [K59](#)
- \bx@KK ... [K42](#)
- \bx@L ... [K32](#), [K59](#)
- \bx@LL ... [K42](#)
- \bx@M ... [K32](#), [K59](#)
- \bx@MM ... [K42](#)
- \bx@N ... [K32](#), [K59](#)
- \bx@NN ... [K42](#)
- \bx@O ... [K33](#), [K60](#)
- \bx@OO ... [K43](#)
- \bx@P ... [K33](#), [K60](#)
- \bx@PP ... [K43](#)
- \bx@Q ... [K33](#), [K60](#)

- \bx@QQ ..... K43
  - \bx@R ..... K33, K60
  - \bx@RR ..... K43
  - \bx@S ..... K38
  - \bx@SS ..... K44
  - \bx@T ..... K38
  - \bx@TT ..... K44
  - \bx@U ..... K38
  - \bx@UU ..... K44
  - \bx@V ..... K38
  - \bx@VV ..... K44
  - \bx@W ..... K39
  - \bx@WW ..... K45
  - \bx@X ..... K39
  - \bx@XX ..... K45
  - \bx@Y ..... K39
  - \bx@YY ..... K45
  - \bx@Z ..... K39
  - \bx@ZZ ..... K25, K45, K55
- C**
- \c ..... 1184, 1300, 1380, 1470,  
1472, 1497, 1499, 1512, 1538, 1616
  - \c@bottomnumber .. G273, G278, K2181
  - \c@dbltopnumber .....  
..... G272, G287, G301, K2188
  - \c@enumi ..... A227
  - \c@enumii ..... A227, A227
  - \c@enumiv ..... A227
  - \c@equation ..... z242, z275, z401
  - \c@errorcontextlines ..... g163
  - \c@footnote ..... F11, G377, G434
  - \c@mpfootnote ..... B271, G379
  - \c@ncel ..... t369, t370
  - \c@page ..... w3, w6, w7, K145, K1728
  - \c@secnumdepth .. F39, F54, F64, F123
  - \c@tocdepth ..... F123, F150
  - \c@topnumber ..... G271, G275, K2177
  - \c@totalnumber ... G274, G280, K2184
  - \cal ..... s169
  - \calculate@math@sizes ... o475, p173
  - \call\_callback ..... 475, N528
  - \callback.register ..... N461
  - \callback\_descriptions ... 475, N660
  - \cap ..... t286
  - \capitalacute ..... 1690, 1978
  - \capitalbreve ..... 1697, 1985
  - \capitalcaron ..... 1696, 1984
  - \capitalcedilla ..... 1683, 1975
  - \capitalcircumflex ..... 1691, 1979
  - \capitaldieresis ..... 1693, 1981
  - \capitaldotaccent ..... 1699, 1987
  - \capitalgrave ..... 1689, 1977
  - \capitalhungarumlaut .... 1694, 1982
  - File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
  - \capitalmacron ..... 1698, 1986
  - \capitalnewtie .... 1703, 11052, 11053
  - \capitalogonek ..... 1686, 1976
  - \capitalring ..... 1695, 1983
  - \capitaltie ..... 1701, 11048, 11049
  - \capitaltilde ..... 1692, 1980
  - \caption ..... G4
  - \cases ..... z108
  - \catcodetable ..... N91, N112
  - \catcodetable@atletter 473, N96, N202
  - \catcodetable@initex . 473, N96, N199
  - \catcodetable@latex .. 473, N96, N201
  - \catcodetable@string . 473, N96, N200
  - \cdot ..... t308
  - \cdotp ..... t410, t416
  - \cdots ..... t416
  - \cdp@elt .....  
o61, o81, o92, o93, o114, o117,  
o119, r201, r283, r338, r402, r483
  - \cdp@list .... o63, o79, o93, o121,  
o122, r219, r285, r340, r404, r485
  - \center ..... y73
  - center (environment) ..... y73
  - \centering ..... y73, y75
  - \centerline ..... B368
  - \cf@encoding ..... 134, 141, 144,  
151, 1114, o221, o231, o241, o260
  - \ch@ck ..... b190,  
b191, b192, b193, b211, b221,  
b222, b223, b224, b251, b253,  
b265, b266, b267, b268, b274, L445
  - \changes ..... 1567,  
G257, K71, N189, N190, O134
  - \char ..... 1299, 1302, 1335,  
1338, 1349, 1356, 1379, 1383, 1388,  
1391, 1393, 1395, 1587, 1615, 1618,  
1648, 1655, 1662, 1685, 1688, 1736,  
s69, y150, z148, D111, D139,  
D153, D161, D164, D233, D260,  
D264, D277, D278, D280, D291
  - \chardef ..... a59,  
a65, a66, b10, b16, b17, b18,  
b19, b20, b58, b64, b65, b67,  
b68, b78, b80, b81, b82, b83,  
b92, b98, b99, b112, b114, b178,  
b226, b230, b232, b255, b270,  
b417, b418, b419, e2, 118, o15,  
C4, C9, L444, N20, N24, N38,  
N47, N48, N91, N178, O30, O51
  - \chardef@text@cmd ..... 13
  - \charzero ..... N178
  - \check ..... t430
  - \check@command ..... d164, d166

- \check@icl ..... K178, K179, K1753, K1755, K2113, K2117, K2145, K2151
  - \check@icr ..... t365
  - \check@mathfonts ..... F143, F148
  - \check@nocorr@ ..... t259
  - \check@range ..... t235, t265, s89
  - \check@single ..... z12
  - \CheckCommand ..... z14
  - \CheckEncodingSubset ..... z18
  - \chi ..... z19
  - \choose ..... a61, a174, a175, a176, a181, b41, b175, b176, b181, b183, b189, b190, b191, b192, b193, b194, b393, b394, c13, c14, c15, c16, c17, c19, d146, d150, p22, p256, p258, p280, p281, r260, r262, r266, r581, r582, r583, r623, r624, r625, r666, r667, r668, r706, r707, r708, r714, r715, r716, r759, r760, r761, r767, r768, r769, r809, r810, r811, r817, r818, r819, v98, v101, D348, D349, D350, D353, D354, D357, D361, O86, O87, O96, O98, O202, O203, O212, O214
  - \circ ..... t305
  - \circle ..... D235, D271
  - \citation ..... I11, I19, I43
  - \cite ..... 375, I12
  - \cl@ckpt ..... k122, m35
  - \cl@page ..... w4
  - \ClassError ..... g84
  - \ClassInfo ..... g84
  - \ClassWarning ..... g84
  - \ClassWarningNoLine ..... g84
  - \cleaders ..... b437, t454, t457
  - \cleardoublepage ..... K145
  - \clearpage ... k91, k109, y12, y49, K132, K145, K150, K174, K381, K384, K388, K429, K435, K2079
  - \cline ..... C326
  - \clubpenalty . b311, k10, k19, A128, A194, A196, F83, F89, F113, F118
  - \clubsuit ..... t255
  - \col@number .. K102, K155, K182, K194
  - \colon ..... t411
  - \color@begingroup ..... o493, z87, z103, B28, B62, B129, B267, B300, C47, C51, G423, K465
  - \color@endbox ..... B62, G253, G348, K198, K602, K612
  - \color@endgroup .. o498, z87, z103, B28, B62, B87, B108, B131, B287, B303, C49, G426, K469
  - \color@hbox ..... B62, K599, K609
  - \color@setgroup ..... B62, B87, B106
  - \color@vbox ..... B62, G96, G165, G339, G361, K189
  - \columnsep ..... k21, K88, K176
  - \columnseprule ... K89, K2115, K2149
  - \columnwidth ..... k18, k21, k22, k24, B269, B296, G99, G168, G419, K87, K151, K152, K153, K175, K176, K177,
  - \cong ..... t365
  - \contentsline ..... F143, F148
  - \coprod ..... t259
  - \copyright ..... t235, t265, s89
  - \cos ..... z12
  - \cosh ..... z14
  - \cot ..... z18
  - \coth ..... z19
  - \count@ a61, a174, a175, a176, a181, b41, b175, b176, b181, b183, b189, b190, b191, b192, b193, b194, b393, b394, c13, c14, c15, c16, c17, c19, d146, d150, p22, p256, p258, p280, p281, r260, r262, r266, r581, r582, r583, r623, r624, r625, r666, r667, r668, r706, r707, r708, r714, r715, r716, r759, r760, r761, r767, r768, r769, r809, r810, r811, r817, r818, r819, v98, v101, D348, D349, D350, D353, D354, D357, D361, O86, O87, O96, O98, O202, O203, O212, O214
  - \countdef . a61, b37, b38, b39, b41, b51, b74, w3, N78, N88, N141, N149, N157, N165, N179, O27
  - \CountZero ..... N179
  - \cr b367, l965, l969, z118, z122, z270, z300, z398, C165, C176, C183, C192, C193, C336, D51, D53, D54
  - \crrcr b424, l276, l298, l302, l305, l378, l382, l386, l388, l391, l586, l614, l618, l621, l685, l688, l735, l970, s91, t243, t244, t246, t367, t370, t374, t438, t439, t440, t441, t442, t443, t444, t445, t446, t447, t448, t450, z109, z111, z112, z113, z118, z120, z121, z122, z140, z141, C144, C145, D51
  - \create\_callback ..... 475, N516
  - \cs ..... l567, K72
  - \csc ..... z21
  - \cup ..... t287
  - \curr@fontshape ..... l130, o53, o297, o305, o309, o311, o374, o380, o383, o392, o399, o401, o409, o415, o418, o426, o433, o435, p92, p100, p121, p431, p451, p483, p496, r223, r228
  - \curr@math@size ..... o286, p210, p216, p221, p238
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx, t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx, y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx, D=lt pictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx, I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx, N=ltluatex.dtx, O=ltfinal.dtx

\CurrentOption	1852, 1854, 1866, L13, L147, L157, L158, L159, L164, L171, L172, L173, L176, L183, L184, L188, L189, L190, L197, L199, L203, L204, L205, L316, L398, L399, L408, L409	\CYRG	1830
\CYRA	1824	\cyrghcrs	1831
\cyr	1824, 1869	\cyrghk	1831
\CYRABHCH	1824	\CYRGUP	1831
\cyrabhch	1824	\cyrgup	1831
\CYRABHCHDSC	1824	\CYRH	1831
\cyrabhchdsc	1824	\cyrh	1831
\CYRABHDZE	1825	\CYRHDSC	1832
\cyrabhdze	1824	\cyrhdsc	1832
\CYRABHHA	1825	\CYRHHCRS	1832
\cyrabhha	1825	\cyrhhcrs	1832
\CYRAE	1825	\CYRHHK	1832
\cyr	1825	\cyrhkh	1832
\CYRB	1825	\CYRHRDSN	1833
\cyrb	1825	\cyrhrdsn	1832
\CYRBYUS	1826	\CYRI	1833
\cyrbyus	1825	\cyri	1833
\CYRC	1826	\CYRIE	1833
\cyr	1826	\cyrie	1833
\CYRCH	1826	\CYRII	1833
\cyrch	1826	\cyrii	1833
\CYRCHLDSC	1826	\CYRISHRT	1833
\cyrchldsc	1826	\cyrishrt	1833
\CYRCHRDSC	1827	\CYRISHRTDSC	1834
\cyrchrdsc	1826	\cyrishrtdsc	1834
\CYRCHVCRS	1827	\CYRIZH	1834
\cyrchvcrs	1827	\cyrizh	1834
\CYRD	1827	\CYRJE	1834
\cyrd	1827	\cyrje	1834
\CYRDELTA	1827	\CYRK	1834
\cyrdelta	1827	\cyrk	1834
\CYRDJE	1828	\CYRKBEAK	1835
\cyrdje	1828	\cyrkbeak	1835
\CYRDZE	1828	\CYRKDSC	1835
\cyrdze	1828	\cyrkdsc	1835
\CYRDZHE	1828	\CYRKHCRS	1835
\cyrdzhe	1828	\cyrkhcrs	1835
\CYRE	1828	\CYRKHK	1836
\cyre	1828	\cyrkhk	1835
\CYREPS	1829	\CYRKVCRS	1836
\cyreps	1828	\cyrkvcrs	1836
\CYREREV	1829	\CYRL	1836
\cyrerev	1829	\cyrl	1836
\CYRERY	1829	\CYRLDSC	1836
\cyrery	1829	\cyrlldsc	1836
\CYRF	1829	\CYRLHK	1837
\cyrf	1829	\cyrlhk	1836
\CYRFITA	1830		
\cyrfita	1829		

File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=ltthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx

\CYRLJE	1837	\CYRSHHA	1843
\cyrlje	1837	\cyrshha	1843
\CYRM	1837	\CYRT	1844
\cyrm	1837	\cyrt	1844
\CYRMDSC	1837	\CYRTDSC	1844
\cyrmdsc	1837	\cyrtddsc	1844
\CYRMHK	1837	\CYRTETSE	1844
\cyrmhk	1837	\cyrtetse	1844
\CYRN	1838	\CYRTSHE	1844
\cyrn	1838	\cyrtshe	1844
\CYRNDSC	1838	\CYRU	1845
\cyrndsc	1838	\cyr	1845
\CYRNG	1838	\CYRUSHRT	1845
\cyrng	1838	\cyrushrt	1845
\CYRNHK	1838	\CYRV	1845
\cyrnhk	1838	\cyrv	1845
\CYRNJE	1839	\CYRW	1845
\cyrnje	1838	\cyrw	1845
\CYRNLHK	1839	\CYRY	1845
\cyrnlhk	1839	\cyr	1845
\CYRO	1839	\CYRYA	1846
\cyro	1839	\cyr	1846
\CYROTLD	1839	\CYRYAT	1846
\cyrotld	1839	\cyr	1846
\CYRP	1839	\CYRYHCRS	1846
\cyrp	1839	\cyr	1846
\CYRPHK	1840	\CYRYI	1846
\cyrphk	1840	\cyr	1846
\CYRQ	1840	\CYRYO	1847
\cyrq	1840	\cyr	1846
\CYRR	1840	\CYRYU	1847
\cyr	1840	\cyr	1847
\CYRRDSC	1840	\CYRZ	1847
\cyrrdsc	1840	\cyrz	1847
\CYRRHK	1841	\CYRZDSC	1847
\cyrrhk	1840	\cyrzddsc	1847
\CYRRTICK	1841	\CYRZH	1847
\cyrrtick	1841	\cyrzh	1847
\CYRS	1841	\CYRZHDSC	1848
\cyr	1841	\cyrzhdsc	1848
\CYRSACRS	1841		
\cyr	1841		
\CYRSCHWA	1842		
\cyr	1842		
\CYRSDSC	1842		
\cyr	1842		
\CYRSEMISFTSN	1842		
\cyr	1842		
\CYRSFTSN	1843		
\cyr	1843		
\CYRSH	1843		
\cyr	1843		
\CYRSHCH	1843		
\cyr	1843		

## D

\d	1185, 1303, 1384, 1619
\dag	1261
\dagger	1261, m74, m80, m88, m89, t289
\dashbox	D166
\dashv	t317
\date	343, F7
\day	a180, L462
\dblfigrule	K677, K2214
\dblfloatpagefraction	
\dblfloatsep	G291, G305, K2191
\dblfloatsep	K663, K675, K1550, K1676, K2198

File Key: a=ltndirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspc.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=ltthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx

`\dbltextfloatsep` ..... K196,  
                   K204, K679, K1549, K1675, K2198  
`\dbltopfraction` .. G288, G302, K2190  
`\ddag` ..... l262  
`\ddagger` l262, m75, m81, m88, m90, t288  
`\ddot` ..... t426  
`\ddots` ..... t421  
`\deadcycles` ..... k115, y39, y49, K273  
`\declare@robustcommand` ..... d198  
`\DeclareEncodingSubset` .... l900,  
                   l907, l908, l909, l910, l1186,  
                   l1187, l1188, l1189, l1190, l1191,  
                   l1192, l1193, l1194, l1195, l1196,  
                   l1197, l1198, l1199, l1200, l1201,  
                   l1202, l1203, l1204, l1205, l1206,  
                   l1207, l1208, l1209, l1210, l1211,  
                   l1212, l1213, l1214, l1215, l1216,  
                   l1217, l1218, l1219, l1220, l1221,  
                   l1222, l1223, l1224, l1225, l1226,  
                   l1227, l1228, l1229, l1230, l1231,  
                   l1232, l1233, l1234, l1235, l1236,  
                   l1237, l1238, l1239, l1240, l1241,  
                   l1242, l1243, l1244, l1245, l1246  
`\DeclareErrorFont` o352, r232, s115, t28  
`\DeclareFixedFont` ..... o40  
`\DeclareFontEncoding` .....  
                   l285, l365, l566, l592,  
                   l598, l681, o83, t45, t46, t47, t48  
`\DeclareFontEncoding@` . o87, o89, o104  
`\DeclareFontEncodingDefaults` ...  
                   o133, q90, q91, t15  
`\DeclareFontFamily` .... o58, q85, q86  
`\DeclareFontShape` .....  
                   o19, o21, q25, q27, q81, q82  
`\DeclareFontShape@` ..... o22, o23  
`\DeclareFontSubstitution` .....  
                   l599, l682,  
                   o106, t16, t17, t49, t50, t51, t52  
`\DeclareMathAccent` .....  
                   r573, r611, t424, t425,  
                   t426, t427, t428, t429, t430,  
                   t431, t432, t433, t434, t435, t436  
`\DeclareMathAlphabet` .....  
                   q119, q123, q125, q132,  
                   r399, r562, r573, t70, t71, t72, t73  
`\DeclareMathAlphabetCharacter` . r672  
`\DeclareMathDelimiter` .....  
                   r674, t174, t175,  
                   t176, t177, t178, t179, t182,  
                   t184, t185, t469, t471, t473,  
                   t475, t477, t480, t482, t484,  
                   t486, t488, t490, t492, t494,  
                   t496, t498, t500, t502, t504,  
                   t506, t508, t510, t512, t514, t516  
`\DeclareMathRadical` ..... r790, t437  
`\DeclareMathSizes` ..... o170,  
                   o176, o198, t76, t77, t78, t79,  
                   t80, t81, t82, t83, t84, t85, t86, t87  
`\DeclareMathSizes*` ..... o170  
`\DeclareMathSymbol` .....  
                   r615, r673, r690, t88,  
                   t89, t90, t91, t92, t93, t94, t95,  
                   t96, t97, t98, t99, t100, t101,  
                   t102, t103, t104, t105, t106,  
                   t107, t108, t109, t110, t111,  
                   t112, t113, t114, t115, t116,  
                   t117, t118, t119, t120, t121,  
                   t122, t123, t124, t125, t126,  
                   t127, t128, t129, t130, t131,  
                   t132, t133, t134, t135, t136,  
                   t137, t138, t139, t140, t141,  
                   t142, t143, t144, t145, t146,  
                   t147, t148, t149, t150, t151,  
                   t152, t153, t154, t155, t156,  
                   t157, t158, t159, t160, t161,  
                   t162, t163, t164, t165, t166,  
                   t167, t168, t169, t170, t180,  
                   t181, t183, t187, t188, t189,  
                   t190, t191, t192, t193, t194,  
                   t195, t196, t197, t198, t199,  
                   t200, t201, t202, t203, t204,  
                   t205, t206, t207, t208, t209,  
                   t210, t211, t212, t213, t214,  
                   t215, t216, t217, t218, t219,  
                   t220, t221, t222, t223, t224,  
                   t225, t226, t227, t229, t230,  
                   t231, t232, t233, t234, t235,  
                   t236, t237, t238, t239, t241,  
                   t242, t247, t248, t249, t250,  
                   t252, t253, t254, t255, t256,  
                   t257, t258, t259, t260, t261,  
                   t262, t263, t264, t265, t267,  
                   t268, t269, t270, t271, t272,  
                   t274, t275, t276, t277, t278,  
                   t279, t282, t284, t286, t287,  
                   t288, t289, t290, t291, t292,  
                   t293, t294, t295, t296, t297,  
                   t298, t299, t300, t301, t302,  
                   t303, t304, t305, t306, t307,  
                   t308, t309, t310, t311, t312,  
                   t313, t314, t315, t316, t317,  
                   t318, t319, t320, t321, t322,  
                   t323, t324, t325, t327, t329,  
                   t331, t332, t333, t334, t335,  
                   t336, t337, t338, t339, t340,  
                   t341, t343, t344, t345, t346,  
                   t347, t349, t351, t353, t354,  
                   t355, t356, t357, t358, t359,  
                   t359

**File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx

- t360, t361, t362, t363, t385,  
 t387, t409, t410, t411, t459,  
 t460, t461, t462, t518, t519, t520  
 \DeclareMathVersion . . . . . [r245](#), s2, s3  
 \DeclareOldFontCommand . . . [v108](#), v124  
 \DeclareOption . . . [454](#), l851, l907,  
 l908, l909, l910, l911, l913, p29,  
 p37, p45, p53, p56, p60, [L130](#), [L419](#)  
 \DeclareOption\* . . . . . [454](#), [L130](#)  
 \DeclarePreloadSizes . . . . .  
 . . . . . [o150](#), q95, q96, u19, u21,  
 u22, u23, u25, u26, u27, u28,  
 u29, u30, u34, u38, u43, u45,  
 u49, u50, u53, u54, u57, u58, u64  
 \DeclareRobustCommand . . . . .  
 . . . . . [d198](#), g4, g11, g30, g57,  
 i35, i43, i212, i248, i262, i267,  
 i282, j3, j13, l256, l257, l258,  
 l259, l260, l261, l262, l263, l265,  
 l267, l269, l1171, m98, o216,  
 o244, o245, o246, o250, o252,  
 o270, p113, s4, s7, s10, s13, s16,  
 s19, s22, s25, s28, s34, s41, s89,  
 s93, t364, t368, t371, t376, t378,  
 t380, t383, t389, t391, t393,  
 t395, t397, t399, t401, t403,  
 t405, t407, t413, t415, t417,  
 t420, v3, v109, z171, z173, z187,  
 z198, z248, z305, z322, z346,  
 B7, B74, B137, B190, B309,  
 B334, G382, G390, O234, O241, I12  
 \DeclareSizeFunction . . p371, p444,  
 p445, p456, p457, p461, p462,  
 p468, p469, p493, p494, p501, p502  
 \DeclareSymbolFont . . . . .  
 . . . . . [q136](#), [r280](#), t60, t61, t62, t63  
 \DeclareSymbolFontAlphabet . . . . .  
 . . . . . [r863](#), t67, t68, t69  
 \DeclareSymbolFontAlphabet@ [r864](#), [r867](#)  
 \DeclareTextAccent . . . . . [l64](#), l286,  
 l287, l288, l289, l290, l291, l292,  
 l293, l294, l295, l296, l366, l367,  
 l368, l369, l370, l371, l372, l373,  
 l374, l375, l376, l595, l600, l601,  
 l602, l603, l604, l605, l606, l607,  
 l608, l609, l610, l689, l690, l691,  
 l692, l693, l694, l695, l696, l697,  
 l698, l699, l700, l701, l702, l703  
 \DeclareTextAccentDefault . . . . .  
[l135](#), l176, l177, l178, l179, l180,  
 l181, l182, l183, l184, l185, l186,  
 l187, l188, l189, l229, l232, l975,  
 l976, l977, l978, l979, l980, l981,  
 l982, l983, l984, l985, l986, l987  
**File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx  
 \DeclareTextCommand . . . [l3](#), l58, l65,  
 l297, l300, l303, l319, l320, l327,  
 l329, l331, l333, l339, l341, l343,  
 l350, l377, l380, l384, l387, l389,  
 l392, l394, l584, l611, l613, l616,  
 l619, l649, l656, l683, l686, l733  
 \DeclareTextCommandDefault . . [l57](#),  
 l136, l138, l233, l236, l237, l238,  
 l239, l241, l245, l249, l250, l252,  
 l253, l254, l255, l275, l1041,  
 l1043, l1044, l1046, l1048, l1050,  
 l1052, l1054, l1056, l1058, l1060,  
 l1062, l1064, l1066, l1068, l1070,  
 l1072, l1074, l1076, l1078, l1080,  
 l1082, l1084, l1086, l1088, l1090,  
 l1092, l1094, l1096, l1098, l1100,  
 l1102, l1104, l1106, l1108, l1110,  
 l1112, l1114, l1116, l1118, l1120,  
 l1122, l1124, l1126, l1128, l1130,  
 l1132, l1134, l1136, l1138, l1140,  
 l1142, l1144, l1146, l1148, l1150,  
 l1152, l1154, l1156, l1158, l1161  
 \DeclareTextComposite . . . . .  
 . . . . . [l74](#), l357, l358, l451,  
 l452, l453, l454, l455, l456, l457,  
 l458, l459, l460, l461, l462, l463,  
 l464, l465, l466, l467, l468, l469,  
 l470, l471, l472, l473, l474, l475,  
 l476, l477, l478, l479, l480, l481,  
 l482, l483, l484, l485, l486, l487,  
 l488, l489, l490, l491, l492, l493,  
 l494, l495, l496, l497, l498, l499,  
 l500, l501, l502, l503, l504, l505,  
 l506, l507, l508, l509, l510, l511,  
 l512, l513, l514, l515, l516, l517,  
 l518, l519, l520, l521, l522, l523,  
 l524, l525, l526, l527, l528, l529,  
 l530, l531, l532, l533, l534, l535,  
 l536, l537, l538, l539, l540, l541,  
 l542, l543, l544, l545, l546, l547,  
 l548, l549, l550, l551, l552, l553,  
 l554, l555, l556, l557, l558, l559,  
 l560, l561, l663, l664, l665, l666,  
 l667, l668, l669, l670, l671, l672,  
 l673, l674, l675, l676, l677, l678  
 \DeclareTextCompositeCommand . . .  
 . . . . . [l74](#), l336, l359,  
 l360, l361, l362, l562, l563, l646  
 \DeclareTextFontCommand . . . . .  
 . . . [v1](#), v15, v16, v17, v18, v19,  
 v20, v21, v22, v23, v24, v25, v123  
 \DeclareTextSymbol . . . . . [l3](#),  
 l306, l307, l308, l309, l310, l311,  
 l312, l313, l314, l315, l316, l317,

- 1318, 1321, 1322, 1323, 1324, 1325,  
 1326, 1396, 1397, 1398, 1399, 1400,  
 1401, 1402, 1403, 1404, 1405, 1406,  
 1407, 1408, 1409, 1410, 1411, 1412,  
 1413, 1414, 1415, 1416, 1417, 1418,  
 1419, 1420, 1421, 1422, 1423, 1424,  
 1425, 1426, 1427, 1428, 1429, 1430,  
 1431, 1432, 1433, 1434, 1435, 1436,  
 1437, 1438, 1439, 1440, 1441, 1442,  
 1443, 1444, 1445, 1446, 1447, 1448,  
 1449, 1450, 1571, 1572, 1573, 1574,  
 1575, 1576, 1577, 1578, 1579, 1580,  
 1581, 1582, 1583, 1593, 1594, 1622,  
 1623, 1624, 1625, 1626, 1627, 1628,  
 1629, 1630, 1631, 1632, 1633, 1634,  
 1635, 1636, 1637, 1638, 1639, 1640,  
 1641, 1642, 1643, 1644, 1645, 1704,  
 1705, 1706, 1707, 1708, 1709, 1710,  
 1711, 1712, 1713, 1714, 1715, 1716,  
 1717, 1718, 1719, 1720, 1721, 1722,  
 1723, 1724, 1725, 1726, 1727, 1728,  
 1729, 1730, 1731, 1732, 1739, 1740,  
 1741, 1742, 1743, 1744, 1745, 1746,  
 1747, 1748, 1749, 1750, 1751, 1752,  
 1753, 1754, 1755, 1756, 1757, 1758,  
 1759, 1760, 1761, 1762, 1763, 1764,  
 1765, 1766, 1767, 1768, 1769, 1770,  
 1771, 1772, 1773, 1774, 1775, 1776,  
 1777, 1778, 1779, 1780, 1781, 1782,  
 1783, 1784, 1785, 1786, 1787, 1788,  
 1789, 1790, 1791, 1792, 1793, 1794,  
 1795, 1796, 1797, 1798, 1799, 1800,  
 1801, 1802, 1803, 1804, 1805, 1806,  
 1807, 1808, 1809, 1810, 1811, 1812,  
 1813, 1814, 1815, 1816, 1817, 1818
- \DeclareTextSymbolDefault** . . . . .  
 . . . . . [1135](#), [1190](#),  
[1191](#), [1192](#), [1193](#), [1194](#), [1195](#), [1196](#),  
[1197](#), [1198](#), [1199](#), [1200](#), [1201](#), [1202](#),  
[1203](#), [1204](#), [1205](#), [1206](#), [1207](#), [1208](#),  
[1209](#), [1210](#), [1211](#), [1212](#), [1213](#), [1214](#),  
[1215](#), [1216](#), [1217](#), [1218](#), [1219](#), [1220](#),  
[1221](#), [1222](#), [1223](#), [1224](#), [1225](#), [1226](#),  
[1227](#), [1228](#), [1230](#), [1231](#), [1988](#), [1989](#),  
[1990](#), [1991](#), [1992](#), [1993](#), [1994](#), [1995](#),  
[1996](#), [1997](#), [1998](#), [1999](#), [11000](#),  
[11001](#), [11002](#), [11003](#), [11004](#), [11005](#),  
[11006](#), [11007](#), [11008](#), [11009](#), [11010](#),  
[11011](#), [11012](#), [11013](#), [11014](#), [11015](#),  
[11016](#), [11017](#), [11018](#), [11019](#), [11020](#),  
[11021](#), [11022](#), [11023](#), [11024](#), [11025](#),  
[11026](#), [11027](#), [11028](#), [11029](#), [11030](#),  
[11031](#), [11032](#), [11033](#), [11034](#), [11035](#),  
[11036](#), [11037](#), [11038](#), [11039](#), [11040](#)
- File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx
- \default@ds** . . . . .  
 . . . [L139](#), [L150](#), [L186](#), [L365](#), [L367](#)
- \default@family** . . . . .  
 o94, o126, o358, o361, o384, o419
- \default@M** . . . . . o101, o141, o144, o148
- \default@mextra** . . . . . q10, q89
- \default@series** . . . . .  
 o94, o127, o359, o362, o381, o416
- \default@shape** . . . . .  
 o95, o128, o360, o363, o379, o414
- \default@T** . . . . . o135, o138, o148, o237
- \defaultthyphenchar** . . . . . b321
- \defaultscriptratio** . . . . . o479, o486
- \defaultscriptscriptratio** o480, o486
- \defaultskewchar** . . . . . b322
- \define@mathalphabet** . . . . . q18, q131
- \define@mathgroup** . . . . . q19, q135
- \define@newfont** . . . . . o289, o298
- \deg** . . . . . z34
- \delcode** . . . . . r788
- \delimiter** . . . . . r719, r784
- \delimiterfactor** . . . . . b323
- \delimitershortfall** . . . . . b333
- \Delta** . . . . . t217
- \delta** . . . . . t190
- \depth** . . . . . B31, B34
- \det** . . . . . z30
- \DH** . . . . . l398, O248
- \dh** . . . . . l408, O248
- \Diamond** . . . . . s107
- \diamond** . . . . . t294
- \diamondsuit** . . . . . t256
- \dim** . . . . . z28
- \dimen@** [b41](#), b390, b391, b427, b428,  
 b430, b432, g28, g29, i227, i232,  
 l334, l335, l337, l338, l647, l648,  
 l966, l968, o179, o181, o187,  
 o200, o203, o207, o478, o479,  
 o480, o484, p405, p406, p407,  
 p408, p412, z72, z73, z129, z130,  
 z131, z132, B360, B363, C149,  
 C150, K482, K484, K505, K507
- \dimen@i** . . . . . [b41](#)
- \dimen@ii** . . . . . [b41](#), o183, o188
- \dimendef** b42, b43, b44, b52, b75, N180
- \dimenzero** . . . . . N180
- \directlua** . . . . . a9, a12,  
 a17, a20, a25, b89, b229, d21,  
 N2, N12, N28, N171, N185, N212
- \disable\_callback** . . . . . 475, N652
- \discretionary** . . . . . d10, z148
- \disply@y** . . . . . z134, z138, z139
- \displaylines** . . . . . [z133](#)
- \displaymath** . . . . . z240

- `displaymath` (environment) . . . . . [z238](#)  
`\displaystyle` . . . . . [t440](#), [t443](#), [t446](#),  
     [t448](#), [z62](#), [z140](#), [z264](#), [z267](#),  
     [z304](#), [z329](#), [z341](#), [z369](#), [z393](#), [z396](#)  
`\displaywidowpenalty` . . . . . [b313](#)  
`\displaywidth` . . . . . [z140](#), [z263](#), [z316](#), [z372](#)  
`\div` . . . . . [t297](#)  
`\DJ` . . . . . [l399](#), [O248](#)  
`\dj` . . . . . [l409](#), [O248](#)  
`\do` . . . . . [a69](#), [a70](#),  
     [a121](#), [b13](#), [b14](#), [d46](#), [f3](#), [f7](#), [f16](#),  
     [f26](#), [k56](#), [k59](#), [k99](#), [k151](#), [k205](#),  
     [k211](#), [v73](#), [y113](#), [y134](#), [y145](#),  
     [y151](#), [B51](#), [C205](#), [C230](#), [D31](#),  
     [D36](#), [D82](#), [D185](#), [D187](#), [D207](#),  
     [D210](#), [D241](#), [D349](#), [G65](#), [G134](#),  
     [L78](#), [L157](#), [L171](#), [L183](#), [L188](#),  
     [L203](#), [L408](#), [L465](#), [L524](#), [I16](#), [I41](#)  
`\donoligs` . . . . . [y146](#), [y151](#)  
`\do@subst@correction` . . . . . [o49](#), [p436](#), [p491](#)  
`\DocInput` . . . . . [p8](#), [t5](#), [u5](#), [M4](#)  
`\document` . . . . . [81](#), [k11](#), [I40](#)  
`\document@select@group` . . . . . [r137](#), [r236](#)  
`\documentclass` . . . . .  
     . . . . . [p2](#), [t2](#), [u2](#), [L207](#), [L214](#),  
     [L241](#), [L244](#), [L333](#), [L428](#), [M2](#), [N14](#)  
`\documentstyle` . . . . . [L212](#), [L428](#)  
`\dorestore@version` . . . . . [r114](#), [r119](#)  
`\dospecials` . . . . .  
     . . . . . [a69](#), [a121](#), [b13](#), [y113](#), [y134](#), [L465](#)  
`\dot` . . . . . [t433](#)  
`\doteq` . . . . . [t377](#)  
`\dotfill` . . . . . [b434](#)  
`\dots` . . . . . [l269](#), [l271](#)  
`\doublehyphendemerits` . . . . . [b316](#)  
`\doublerulesep` . . . . . [C270](#), [C297](#), [C321](#)  
`\Downarrow` . . . . . [t490](#)  
`\downarrow` . . . . . [t484](#)  
`\downbracefill` . . . . . [t445](#), [t463](#)  
`\ds@` . . . . . [L152](#), [L369](#)  
`\dt@pfalse` . . . . . [z135](#)  
`\dt@ptrue` . . . . . [z134](#)  
`\dump` . . . . . [O312](#)
- E**
- `\E` . . . . . [L466](#), [L469](#), [L496](#)  
`\e@alloc` . . . . . [b51](#), [b52](#), [b53](#),  
     [b55](#), [b56](#), [b63](#), [b64](#), [b65](#), [b67](#),  
     [b68](#), [b122](#), [N13](#), [N49](#), [N81](#), [N91](#),  
     [N144](#), [N152](#), [N160](#), [N168](#), [O12](#), [O30](#)  
`\e@alloc@attribute@count` . . . . .  
     . . . . . [N69](#), [N77](#), [N78](#), [N82](#), [N84](#), [N193](#)  
`\e@alloc@bytecode@count` . . . . . [N73](#),  
     [N156](#), [N157](#), [N161](#), [N163](#), [N209](#)  
**File Key:** [a=ltdirchk.dtx](#), [b=ltplain.dtx](#), [c=ltvers.dtx](#), [d=ltdefs.dtx](#),  
[e=ltalloc.dtx](#), [f=ltcntrl.dtx](#), [g=lterror.dtx](#), [h=ltpar.dtx](#), [i=ltspace.dtx](#),  
[j=ltlogos.dtx](#), [k=ltfiles.dtx](#), [l=ltoutenc.dtx](#), [m=ltcounts.dtx](#), [n=ltlength.dtx](#),  
[o=ltfssbas.dtx](#), [p=ltfssstrc.dtx](#), [q=ltfsscmp.dtx](#), [r=ltfssdcl.dtx](#), [s=ltfssini.dtx](#),  
[t=fontdef.dtx](#), [u=preload.dtx](#), [v=ltfntcmd.dtx](#), [w=ltpageno.dtx](#), [x=ltxref.dtx](#),  
[y=ltmiscen.dtx](#), [z=ltmath.dtx](#), [A=ltlists.dtx](#), [B=ltboxes.dtx](#), [C=lttab.dtx](#),  
[D=ltpictur.dtx](#), [E=ltthm.dtx](#), [F=ltsect.dtx](#), [G=ltfloat.dtx](#), [H=ltidxglo.dtx](#),  
[I=ltbibl.dtx](#), [J=ltpage.dtx](#), [K=ltoutput.dtx](#), [L=ltclass.dtx](#), [M=ltthyphen.dtx](#),  
[N=ltluatex.dtx](#), [O=ltfinal.dtx](#)
- `\e@alloc@ccodetable@count` . . . . .  
     . . . . . [N70](#), [N87](#), [N88](#), [N92](#), [N95](#), [N197](#)  
`\e@alloc@chardef` . . . . .  
     . . . . . [b60](#), [b86](#), [b194](#), [b195](#),  
     [N48](#), [N144](#), [N152](#), [N160](#), [N168](#), [O12](#)  
`\e@alloc@luachunk@count` . . . . . [N74](#),  
     [N164](#), [N165](#), [N169](#), [N173](#), [N211](#)  
`\e@alloc@luafunction@count` . . . . .  
     . . . . . [N71](#), [N140](#),  
     [N141](#), [N145](#), [N147](#), [N203](#), [N205](#)  
`\e@alloc@top` . . . . .  
     . . . . . [b55](#), [b63](#), [b86](#), [b172](#), [b232](#), [N47](#),  
     [N82](#), [N145](#), [N153](#), [N161](#), [N169](#), [O12](#)  
`\e@alloc@whatsit@count` . . . . . [N72](#),  
     [N148](#), [N149](#), [N153](#), [N155](#), [N207](#)  
`\ech@ck` . . . . . [b126](#), [b136](#), [N51](#), [N55](#)  
`\e@insert@top` . . . . . [b230](#), [b232](#), [b248](#), [b263](#)  
`\e@mathgroup@top` . . . . . [b67](#), [b108](#), [r56](#), [r145](#)  
`\Eesphack` . . . . . [i98](#)  
`\egroup` . . . . . [b371](#)  
`\eject` . . . . . [b401](#)  
`\ell` . . . . . [t231](#)  
`\em` . . . . . [s31](#), [v25](#)  
`\emergencystretch` . . . . . [b296](#), [J45](#), [J51](#)  
`\eminnershape` . . . . . [s31](#)  
`\emph` . . . . . [v25](#)  
`\empty` . . . . . [b369](#)  
`\empty@sfcnt` . . . . .  
     . . . . . [p444](#), [p445](#), [p446](#), [p460](#), [p465](#), [p499](#)  
`\emptyset` . . . . . [t238](#)  
`\enc@update` . . . . . [o222](#), [o224](#), [o240](#), [o243](#), [p129](#)  
`\encodingdefault` . . . . .  
     . . . . . [l852](#), [l878](#), [r237](#), [s94](#), [t38](#)  
`\end` . . . . . [a64](#), [d8](#), [d287](#), [g204](#), [p9](#), [t6](#), [u6](#),  
     [y60](#), [y97](#), [y98](#), [z350](#), [z359](#), [A112](#),  
     [F15](#), [F17](#), [L474](#), [L478](#), [L484](#), [M5](#)  
`\end@dblfloat` . . . . . [G205](#)  
`\end@float` . . . . . [G189](#), [G227](#), [G243](#), [G363](#)  
`\endarray` . . . . . [C144](#)  
`\endcenter` . . . . . [y74](#)  
`\enddisplaymath` . . . . . [z241](#)  
`\enddocument` . . . . . [y8](#)  
`\endenumerate` . . . . . [A240](#)  
`\endeqnarray` . . . . . [z272](#), [z303](#)  
`\endequation` . . . . . [z244](#)  
`\endfilecontents` . . . . . [L432](#)  
`\endflushleft` . . . . . [y81](#)  
`\endflushright` . . . . . [y87](#)  
`\endgraf` . . . . . [b366](#)  
`\ENDGROUP` . . . . . [N126](#), [N132](#)  
`\EndIncludeInRelease` . . . . . [a22](#), [a50](#), [b71](#),  
     [b85](#), [b102](#), [b107](#), [b117](#), [b121](#),  
     [b131](#), [b134](#), [b151](#), [b165](#), [b169](#),  
     [b203](#), [b208](#), [b260](#), [b272](#), [b479](#)

- b486, b533, b538, c74, c75,  
d276, d279, i83, i94, i111, i123,  
i188, i210, i276, i280, l279, l283,  
m28, m33, m85, m91, m111,  
m114, n10, n14, o196, o213,  
o404, o437, q21, q143, r77,  
r105, r168, r198, s39, s45, z176,  
z182, z211, z236, z332, z344,  
z353, z362, A132, A137, B13,  
B21, B78, B84, B141, B147,  
B195, B202, B311, B315, B338,  
B344, G104, G172, G231, G246,  
G297, G310, G395, G400,  
K53, K62, K339, K344, K392,  
K438, K721, K740, K803, K821,  
K863, K884, K1126, K1295,  
K1377, K1471, K1593, K1720,  
K1839, K1867, K2131, K2175,  
N188, N213, O15, O19, O33, O53
- \enditemize ..... A251
- \endline ..... b366, z118
- \endlinechar a87, a88, a89, a199, d24,  
d26, d31, k179, L112, L113, L114
- \endlist ..... A98, A240, A251
- \endlrbox ..... B108
- \endmath ..... z239
- \endminipage ..... B277
- \endpicture ..... D17
- \endsloppypar ..... J49
- \endtabbing ..... C73
- \endtabular ..... C144
- \endtabular\* ..... C144
- \endtrivlist ..... y74, y81, y87,  
y119, z374, A100, A101, C74, E39
- \endverbatim ..... y118, y122
- \enlargethispage ..... K1765
- \enlargethispage\* ..... K1765
- \enskip ..... i292
- \enspace ..... i289
- \ensuremath .....  
... m87, z305, G385, G393, G403
- \enumerate ..... A231
- enumerate (environment) ..... A231
- environments:
- center ..... y73
- displaymath ..... z238
- enumerate ..... A231
- eqnarray ..... z250, z375
- eqnarray\* ..... z301
- equation ..... z242, z363
- filecontents ..... 452
- flushleft ..... y80
- flushright ..... y86
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
- itemize ..... A242
- lrbox ..... 284
- math ..... z238
- minipage ..... 285
- sloppypar ..... J48
- thebibliography ..... 375
- verbatim\* ..... y121
- \epsilon ..... t191
- \eqnarray ..... z255, z302
- eqnarray (environment) ..... z250, z375
- eqnarray\* (environment) ..... z301
- \eqno ..... z244
- \equation ..... z243
- equation (environment) ..... z242, z363
- \equiv ..... t356
- \err@rel@i ..... q12, q99, q132, q136
- \errhelp ..... a212,  
c30, g39, g66, M12, O143, O303
- \errmessage .....  
.. a4, a217, b148, b162, b276,  
c31, g47, g72, o376, o411, p379,  
p479, q65, M16, N63, O47, O145
- \error@fontshape .....  
o353, o377, o412, p107, p481, r222
- \errorcontextlines ... b294, b326,  
b453, b469, b484, b497, b514, g163
- \errorstopmode ..... b442, O311
- \escapechar ..... d103, d146,  
d150, d158, o301, o446, p183,  
r58, r86, r147, r177, r221, N170
- \et@xmaxfam ..... N20, N24, N30, N38
- \et@xmaxregs ..... N28,  
N31, N32, N33, N34, N35, N36, N37
- \eta ..... t193
- \etatcatcode ..... N686
- \evensidemargin ..... K80, K584
- \every@math@size ..... o43, p189, p201
- \everycr .. b422, z135, z138, z263, z390
- \everydisplay ..... o279, o280, o285
- \everyjob ..... c36, c41, c46, r241,  
N174, N175, O273, O274, O276
- \everymath ..... o278, o280, o283
- \everypar ..... 64, k37,  
o494, y50, y116, A129, A131,  
A135, A136, A180, A197, B238,  
C70, F31, F79, F90, F110,  
F119, G187, K168, K1061, K1227
- \execute@size@function .....  
..... p316, p344, p358, p375
- \ExecuteOptions . 1914, p57, p70, L201
- \exhyphenpenalty ..... b308, b395
- \exists ..... t249
- \exp ..... z31

- \external@font ..... p84,  
p87, p98, p102, p104, p345,  
p359, p421, p455, p505, p507, p509
  - \extra@def ..... q9, q84
  - \extracolsep ..... C140
  - \extract@alph@from@version .....  
..... o452, o458, r151, r182
  - \extract@font ..... o312, p81
  - \extract@fontinfo ..... p312, p319
  - \extract@rangefontinfo .....  
..... p329, p336, p355, p388
  - \extract@sizefn ..... p304, p326
  - \extrafloats ..... b136, b173, b246
- F**
- \f@baselineskip .....  
..... o251, o258, o365, p119,  
p136, p140, p155, p169, p180, p194
  - \f@depth ..... G295, K319
  - \f@encoding .....  
..... l128, o216, o235, o238, o239,  
o241, o260, o292, o297, o316,  
o318, o320, o325, o327, o357,  
o373, o408, p91, p261, p471, r207
  - \f@family ... l931, l934, l948, l958,  
l964, l1179, o244, o254, o293,  
o297, o316, o318, o320, o325,  
o327, o361, o384, o419, p91, r207
  - \f@linespread ... o254, p118, p137,  
p138, p141, p149, p152, p163, p166
  - \f@series ..... j14, o244, o255,  
o294, o297, o362, o381, o416, s81
  - \f@shape ..... o244,  
o256, o295, o297, o363, o379, o414
  - \f@size ..... l130, l966, o53, o251,  
o257, o296, o364, o401, o435,  
o477, o478, o481, o482, p119,  
p121, p134, p154, p169, p172,  
p175, p180, p187, p194, p206,  
p209, p215, p221, p238, p239,  
p242, p247, p313, p320, p339,  
p341, p356, p407, p409, p411,  
p427, p428, p433, p447, p459,  
p464, p476, p484, p489, p497, p511
  - \f@user@size .. p427, p432, p476, p489
  - \fam ..... b82, o16, N20, N24, N38
  - \familydefault ..... r238, s95, t38
  - \fbox ..... 284, B126, B139, B146
  - \fboxrule ..... B124, B159, B162,  
B168, B170, B177, B178, O57
  - \fboxsep ..... B124, B130,  
B158, B163, B173, B175, O56
  - \filbreak ..... b399
  - File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
  - \filec@ntents .....  
... L437, L438, L439, L518, L524
  - \filecontents ..... L432
  - filecontents (environment) ..... 452
  - \filename@area ..... a241, a247,  
a254, a260, a267, a273, a280,  
k168, k190, k193, k207, k219, k221
  - \filename@base .....  
a289, k168, k190, k193, k214, k219
  - \filename@dot ..... a287, a290
  - \filename@ext ..... a285, a287,  
k169, k186, k187, k190, k193, k215
  - \filename@parse ..... 1,  
6, a105, a237, k166, k185, k212
  - \filename@path .. a242, a243, a248,  
a255, a256, a261, a268, a269, a274
  - \filename@simple .....  
..... a245, a258, a271, a281, a283
  - \fill ..... i286
  - \finalhyphendemerits ..... b317
  - \finph@nt ..... z87, z89, z90
  - \finsm@sh ..... z103, z105, z106
  - \firstmark ..... J37, K618, K2106
  - \fix@penalty ..... v84
  - \fixed@sfcnt ..... p501, p502, p503
  - \fl@trace K214, K241, K297, K325,  
K332, K353, K400, K446, K499,  
K514, K515, K516, K517, K528,  
K529, K530, K531, K532, K542,  
K691, K710, K729, K747, K749,  
K888, K892, K904, K905, K906,  
K907, K913, K916, K924, K928,  
K939, K944, K949, K950, K951,  
K952, K959, K962, K970, K981,  
K987, K992, K997, K1003,  
K1004, K1009, K1014, K1015,  
K1016, K1024, K1028, K1033,  
K1037, K1042, K1053, K1054,  
K1056, K1074, K1083, K1089,  
K1098, K1101, K1107, K1117,  
K1121, K1131, K1137, K1143,  
K1149, K1156, K1158, K1164,  
K1169, K1171, K1173, K1181,  
K1186, K1192, K1197, K1203,  
K1217, K1218, K1221, K1242,  
K1251, K1257, K1266, K1269,  
K1276, K1286, K1290, K1302,  
K1308, K1313, K1318, K1322,  
K1326, K1327, K1334, K1339,  
K1343, K1350, K1359, K1363,  
K1367, K1368, K1372, K1373,  
K1383, K1389, K1395, K1401,  
K1405, K1411, K1413, K1421,  
K1426, K1431, K1439, K1448,

- K1453, K1458, K1460, K1465,  
 K1467, K1478, K1484, K1494,  
 K1500, K1504, K1505, K1510,  
 K1511, K1517, K1520, K1521,  
 K1522, K1529, K1530, K1531,  
 K1539, K1544, K1556, K1557,  
 K1564, K1567, K1575, K1579,  
 K1583, K1584, K1588, K1589,  
 K1599, K1605, K1615, K1621,  
 K1625, K1626, K1632, K1633,  
 K1640, K1643, K1644, K1645,  
 K1653, K1654, K1655, K1664,  
 K1669, K1682, K1684, K1691,  
 K1694, K1703, K1707, K1711,  
 K1712, K1716, K1717, K1769,  
 K1774, K1780, K1790, K1797,  
 K1807, K1903, K1916, K1917,  
 K1921, K1924, K1926, K1929,  
 K1932, K1934, K1975, K1982,  
 K1987, K1993, K1998, K2002,  
 K2008, K2016, K2018, K2025,  
 K2030, K2035, K2037, K2043,  
 K2045, K2052, K2079, K2081,  
 K2093, K2118, K2122, K2127,  
 K2139, K2156, K2161, K2169  
 \fl@tracemessage ..... K1807  
 \fl@traceval ..... K1807  
 \flat ..... t252  
 \float@count .. b51, b52, b53, b62,  
 b172, b189, b194, b196, b197, b206  
 \floatingpenalty ..... G418  
 \floatpagefraction ..... G282, K2187  
 \floatsep ..... K636,  
 K654, K661, K2014, K2064, K2192  
 \flushbottom ..... J41  
 \flushleft ..... y80  
 flushleft (environment) ..... y80  
 \flushright ..... y86  
 flushright (environment) ..... y86  
 \fmtname ..... c1, c37,  
 c39, c42, c44, c47, c49, L250, L254  
 \fmtversion .. c1, c18, c37, c39, c42,  
 c44, c47, c49, c62, g2, o1, C1,  
 D1, K4, L267, L270, O257, O283  
 \fmtversion@topatch ..... O255,  
 O257, O269, O270, O282, O290  
 \fnsymbol ..... 132, m50  
 \font ..... b427, b432,  
 l246, l247, l248, l344, l351, l650,  
 l657, o46, o52, o54, p84, s35,  
 s42, s68, s80, u8, u9, u10, v68, y115  
 \font@info .... p99, p319, p388, p393  
 \font@name ..... l129,  
 l132, o51, o159, o161, o288,  
 o303, o400, o434, p84, p88,  
 p90, p105, p120, p123, p126,  
 p284, p285, p286, p287, p288, p293  
 \font@submax ..... p395, p424,  
 p425, y22, y24, O135, O137, O146  
 \fontdimen ..... b427, b432, l246,  
 l247, l248, l344, l351, l650, l657,  
 s35, s42, s80, v68, D38, D40, D307  
 \fontencoding l878, o216, o247, r237, t14  
 \fontfamily l950, o244, r238, s6, s9, s12  
 \fontname ..... o54  
 \fontseries ..... o244, r239, s15, s18  
 \fontshape ..... l354,  
 l660, o244, r240, s21, s24, s27, s30  
 \fontsize ..... j6, l251, l277, l968,  
 o44, o252, s74, G385, G393, G403  
 \fontsubfuzz ..... p395, p429, y22  
 \footins ..... G370, G414,  
 K288, K289, K290, K291, K349,  
 K396, K456, K464, K468, K491  
 \footnote ..... G405  
 \footnotemark ..... F9, G427  
 \footnoterule ..... B283, G374, K467  
 \footnotesep . B302, G404, G417, G425  
 \footnotesize ..... B295, G415  
 \footnotetext ..... F11, G444  
 \footskip ..... K84, K608  
 \forall ..... t248  
 \forall ..... G34  
 \frac ..... z247  
 \frame ..... B110, B185  
 \framebox ..... 284, B133  
 \frenchspacing . b352, k40, y118, y144  
 \frown ..... t359  
 \frozen@everydisplay .... o278, o284  
 \frozen@everymath ..... o278, o282  
 \fussy ..... J50  
 \futurelet ..... d293,  
 d307, i252, i260, v66, z153, C318
- ## G
- \g@addto@macro L376, L382, L386, L387  
 \G@refundefinedfalse ..... x5  
 \G@refundefinedtrue . x3, x12, I21, I44  
 \Gamma ..... t216  
 \gamma ..... t189  
 \gcd ..... z33  
 \ge ..... t330  
 \gen@sfcnt ..... p456, p457, p458  
 \genb@sfcnt ..... p461, p462, p463  
 \genb@x ..... p464, p466  
 \genb@y ..... p466  
 \GenericError g18, g85, g111, g137, p62  
 File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=lt pictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx

- \GenericInfo . . . . c63, c66, c70, g4,  
g104, g130, g155, p31, p34, p39, p75
- \GenericWarning . . . . . g11,  
g94, g120, g146, p42, p47, p50, p78
- \geq . . . . . t329, t330
- \get@cdp . . . . . r356, r364, r397
- \get@external@font . . . . p83, p96, p490
- \getanddefine@fonts . . . o447, o465,  
p274, r59, r87, r132, r148, r178,  
r263, r327, r361, r363, r380,  
r503, r504, r536, r537, r883, r884
- \GetFileInfo . . . . . t3
- \getlinechar . . . . . D108
- \gets . . . . . t348
- \gg . . . . . t343
- \glb@currsz . . . . . k35,  
o275, p171, p206, p210, p216, p239
- \glb@settings . . . . . o276, p171, p218, p249
- \globaldefs . . . . .  
. . . . . o448, p185, r60, r89, r149, r180
- \glossary . . . . . 373,  
F146, H23, H35, J20, J28, K592
- \glossaryentry . . . . . H32
- \goodbreak . . . . . b399
- \grave . . . . . t425
- \group@elt . . . . . r35,  
r261, r298, r299, r320, r324, r915
- \group@list . . . . .  
. . . . . r265, r305, r318, r323, r324,  
r353, r575, r617, r697, r700,  
r750, r753, r800, r803, r870, r921
- \guillemotleft . . . . . l410, l627
- \guillemotright . . . . . l411, l628
- \guilsinglleft . . . . . l412
- \guilsinglright . . . . . l413
- H**
- \H . . . . . g24, l180, l293,  
l371, l465, l473, l492, l500, l607
- \h@false . . . . . z77
- \h@true . . . . . z78, z79
- \halign . . . . . b422, z96, z140, z263, z390
- \hangindent . . . . . F122
- \hat . . . . . t431
- \hb@xt@ . . . . . b437, d16,  
l330, z140, z268, z314, z329,  
z341, z368, z398, B43, B58,  
B157, B368, B372, B373, C37,  
D13, D23, D32, D122, D156,  
D159, D162, D164, D166, D237,  
D266, D359, F163, F166, K601,  
K611, K1753, K2112, K2113,  
K2117, K2144, K2145, K2151
- \hbadness . . . . . b304, o497
- File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
- \hbar . . . . . t228
- \headheight . . . . . K82, K597
- \headsep . . . . . K83, K606
- \heartsuit . . . . . t257
- \height . . . . . B30, B33
- \hexnumber@ . . . . . r590,  
r598, r613, r632, r640, r648,  
r657, r660, r669, r670, r709,  
r717, r762, r770, r784, r785,  
r788, r813, r821, r826, r828, s85
- \hfuzz . . . . . b327, J46, J47, J53, J54
- \hgl@ . . . . . b392, b393
- \hglue . . . . . b389
- \hideoutput . . . . . b487
- \hideskip . . . . . b279, b413
- \hidewidth . . . . . b413,  
l276, l278, l298, l299, l302, l305,  
l378, l379, l383, l386, l388, l391,  
l614, l615, l618, l621, l685, l688
- \hline . . . . . C317, C320
- \hmode@bgroup . . . . . l67, l73,  
l276, l298, l304, l332, l343, l350,  
l378, l385, l388, l390, l584, l614,  
l620, l649, l656, l684, l687, l733, v7
- \hmode@start@before@group . . . . .  
. . . . . l68, l111, l113, l119, l134
- \holdinginserts . . . . . b295
- \hom . . . . . z29
- \hookleftarrow . . . . . t388
- \hookrightarrow . . . . . t386
- \hphantom . . . . . z75
- \hrule . . . . . b390,  
b434, i228, i236, l240, l243,  
t246, t522, B116, B121, B168,  
B178, C318, C335, D268, G375
- \hrulefill . . . . . b434
- \hspace . . . . . i282
- \hyphenation . . . . . l155
- \hyphenchar . . . . . y115
- \hyphenpenalty . . . . . b307
- I**
- \I . . . . . b358, L492, L510, O106, O222
- \i . . . . . l197,  
l310, l357, l358, l359, l360, l361,  
l362, l414, l451, l452, l544, l546,  
l548, l550, l629, O111, O227, O235
- \ialign . . . . . b422, b424,  
t243, t367, t438, t441, t444,  
t447, z109, z111, z119, C164, D51
- \if@afterindent . . . . . F107, F114
- \if@compatibility . . . . . L2, L209
- \if@endpe . . . . . y62, A138
- \if@eqnsw . . . . . z250, z299

`\if@fcolmade` ..... [K146](#), [K241](#), [K252](#), [K369](#), [K417](#),  
     ... [K102](#), [K238](#), [K368](#), [K377](#),  
     [K415](#), [K425](#), [K689](#), [K709](#), [K727](#),  
     [K756](#), [K836](#), [K2078](#), [K2126](#), [K2166](#)  
`\if@filesw` ..... [k7](#),  
     [k30](#), [k92](#), [k104](#), [k111](#), [k120](#), [y14](#),  
     [y28](#), [F136](#), [I4](#), [I8](#), [I19](#), [I28](#), [I36](#), [I43](#)  
`\if@firstamp` ..... [C212](#)  
`\if@firstcolumn` [K102](#), [K220](#), [K253](#),  
     [K370](#), [K418](#), [K1725](#), [K2090](#), [K2135](#)  
`\if@ignore` ..... [y4](#), [y63](#)  
`\if@inlabel` .....  
     [A28](#), [A65](#), [A102](#), [A160](#), [A183](#), [K164](#)  
`\if@insert` ..... [K102](#), [K967](#),  
     [K1079](#), [K1113](#), [K1247](#), [K1282](#),  
     [K1356](#), [K1445](#), [K1572](#), [K1700](#)  
`\if@minipage` ..... [i141](#), [i158](#),  
     [i193](#), [y101](#), [A149](#), [B246](#), [C68](#), [G20](#)  
`\if@mparswitch` ..... [K102](#), [K1727](#)  
`\if@multiplelabels` ..... [x31](#)  
`\if@negarg` .... [D55](#), [D77](#), [D91](#), [D130](#)  
`\if@newlist` ..... [y119](#), [A29](#), [A33](#),  
     [A69](#), [A78](#), [A106](#), [A166](#), [K568](#), [K615](#)  
`\if@nmbrlist` ..... [A33](#), [A201](#)  
`\if@no@font@opt` ..... [q16](#), [q110](#), [q129](#)  
`\if@nobreak` ..... [i58](#), [i160](#), [i195](#),  
     [k67](#), [k79](#), [A167](#), [A192](#), [B232](#),  
     [F30](#), [F111](#), [G180](#), [G353](#), [J25](#),  
     [J33](#), [K168](#), [K309](#), [K1058](#), [K1224](#)  
`\if@noitemarg` ..... [A32](#), [A199](#)  
`\if@noparitem` ..... [A30](#), [A157](#)  
`\if@noparlist` ..... [A31](#), [A114](#)  
`\if@noskipsec` ..... [A58](#),  
     [B233](#), [F21](#), [F23](#), [F80](#), [G354](#), [K158](#)  
`\if@ovb` ..... [D212](#), [D254](#), [D259](#)  
`\if@ovl` ..... [D212](#), [D252](#), [D269](#)  
`\if@ovr` ..... [D212](#), [D251](#), [D267](#)  
`\if@ovt` ..... [D212](#), [D253](#), [D263](#)  
`\if@partsw` ..... [k7](#), [k96](#)  
`\if@pboxsw` ..... [B229](#), [B304](#)  
`\if@reversemargin` ..... [K108](#), [K1730](#)  
`\if@reversemarginpar` ..... [K102](#)  
`\if@rjfield` ..... [C19](#), [C33](#)  
`\if@specialpage` ..... [K102](#), [K577](#)  
`\if@tempswa` ... [a73](#), [a74](#), [a75](#), [b242](#),  
     [e9](#), [k102](#), [o64](#), [r286](#), [r341](#), [r405](#),  
     [r486](#), [r914](#), [y30](#), [y107](#), [K900](#),  
     [K936](#), [K1536](#), [K1661](#), [L455](#), [I52](#)  
`\if@test` ..... [K12](#), [K13](#),  
     [K797](#), [K816](#), [K856](#), [K878](#), [K942](#),  
     [K1026](#), [K1035](#), [K1184](#), [K1195](#),  
     [K1337](#), [K1424](#), [K1542](#), [K1667](#)  
`\if@twocolumn` .....  
     [k20](#), [G32](#), [G210](#), [G235](#), [K102](#),  
     [K146](#), [K241](#), [K252](#), [K369](#), [K417](#),  
     [K441](#), [K691](#), [K747](#), [K1724](#), [K2080](#)  
`\if@twoside` ..... [K102](#), [K145](#), [K580](#)  
`\ifdt@p` ..... [z133](#), [z135](#)  
`\iff` ..... [t408](#)  
`\IfFileExists` ..... [81](#),  
     [454](#), [a173](#), [k134](#), [k161](#), [k172](#), [O251](#)  
`\ifG@refundefined` ..... [x3](#), [x4](#), [x5](#)  
`\ifh@` ..... [z76](#), [z93](#)  
`\ifin@` .... [l868](#), [l871](#), [q50](#), [q52](#), [r1](#),  
     [r22](#), [r250](#), [r352](#), [r354](#), [r415](#), [r428](#),  
     [r498](#), [r500](#), [r528](#), [r576](#), [r587](#),  
     [r618](#), [r629](#), [r698](#), [r701](#), [r721](#),  
     [r751](#), [r754](#), [r798](#), [r801](#), [r804](#),  
     [r871](#), [r873](#), [r902](#), [L82](#), [L162](#), [L174](#)  
`\ifinner` ..... [z174](#),  
     [z181](#), [z200](#), [z226](#), [G57](#), [G126](#), [G319](#)  
`\ifmath@fonts` ..... [o169](#), [p176](#)  
`\ifmaybe@ic` ..... [v65](#), [v74](#)  
`\ifnot@nil` ..... [p297](#), [p314](#), [p335](#)  
`\ifodd` ..... [r845](#),  
     [D171](#), [D191](#), [G68](#), [G137](#), [K21](#),  
     [K145](#), [K581](#), [K892](#), [K895](#), [K928](#),  
     [K931](#), [K1042](#), [K1045](#), [K1204](#),  
     [K1207](#), [K1484](#), [K1487](#), [K1605](#),  
     [K1608](#), [K1728](#), [K1949](#), [K1957](#)  
`\iftc@forced` ..... [l906](#), [l916](#), [l1185](#)  
`\ifv@` ..... [z75](#), [z92](#)  
`\ifvbox` [K293](#), [K350](#), [K397](#), [K476](#), [K492](#)  
`\ignorespaces` .....  
     . [i24](#), [i79](#), [i91](#), [i107](#), [i120](#), [i298](#),  
     [k60](#), [o249](#), [y63](#), [y71](#), [y72](#), [z208](#),  
     [z234](#), [A55](#), [A217](#), [B107](#), [B302](#),  
     [C57](#), [C58](#), [C59](#), [C72](#), [C81](#), [C94](#),  
     [C98](#), [C105](#), [C112](#), [C114](#), [C123](#),  
     [C198](#), [C260](#), [C262](#), [C264](#), [C291](#),  
     [D16](#), [D24](#), [D35](#), [D53](#), [D54](#), [E30](#),  
     [E32](#), [F93](#), [G17](#), [G24](#), [G425](#), [I7](#), [I9](#)  
`\ignorespacesafterend` ..... [y7](#)  
`\IJ` ..... [l200](#), [l341](#), [l417](#)  
`\ij` ..... [l199](#), [l339](#), [l416](#)  
`\Im` ..... [t234](#)  
`\imath` ..... [t229](#)  
`\in` ..... [t340](#), [t369](#)  
`\in@` ..... [l866](#), [l869](#), [q49](#), [q51](#), [r1](#),  
     [r21](#), [r249](#), [r351](#), [r353](#), [r411](#), [r424](#),  
     [r497](#), [r499](#), [r526](#), [r574](#), [r585](#),  
     [r616](#), [r627](#), [r696](#), [r699](#), [r719](#),  
     [r749](#), [r752](#), [r796](#), [r799](#), [r802](#),  
     [r869](#), [r872](#), [r900](#), [L81](#), [L159](#), [L173](#)  
`\in@@` ..... [r5](#), [r6](#), [r7](#), [r9](#)  
`\in@false` ..... [r10](#)  
`\in@true` ..... [r12](#)  
`\in_callback` ..... [475](#), [N636](#)  
**File Key:** [a=ltldirchk.dtx](#), [b=ltplain.dtx](#), [c=ltvers.dtx](#), [d=ltdefns.dtx](#),  
[e=ltalloc.dtx](#), [f=ltcntrl.dtx](#), [g=lterror.dtx](#), [h=ltpar.dtx](#), [i=ltspace.dtx](#),  
[j=ltlogos.dtx](#), [k=ltfiles.dtx](#), [l=ltoutenc.dtx](#), [m=ltcounts.dtx](#), [n=ltlength.dtx](#),  
[o=ltfssbas.dtx](#), [p=ltfssstrc.dtx](#), [q=ltfsscmp.dtx](#), [r=ltfssdcl.dtx](#), [s=ltfssini.dtx](#),  
[t=fontdef.dtx](#), [u=preload.dtx](#), [v=ltfntcmd.dtx](#), [w=ltpageno.dtx](#), [x=ltxref.dtx](#),  
[y=ltmiscen.dtx](#), [z=ltmath.dtx](#), [A=ltlists.dtx](#), [B=ltboxes.dtx](#), [C=lttab.dtx](#),  
[D=ltpictur.dtx](#), [E=ltthm.dtx](#), [F=ltsect.dtx](#), [G=ltfloat.dtx](#), [H=ltidxglo.dtx](#),  
[I=ltbibl.dtx](#), [J=ltpage.dtx](#), [K=ltoutput.dtx](#), [L=ltclass.dtx](#), [M=lthyphen.dtx](#),  
[N=ltluatex.dtx](#), [O=ltfinal.dtx](#)

- `\include` ..... [81](#), [k86](#)
  - `\IncludeInRelease` .....
    - a18, a23, b49, b72, b87, b103, b109, b118, b123, b132, b138, b152, b166, b170, b204, b217, b261, b445, b480, b487, b534, [c53](#), d249, d277, i70, i84, i97, i112, i153, i189, i271, i277, l273, l281, m24, m30, m70, m86, m94, m112, n5, n11, o175, o197, o369, o405, q2, q22, r49, r78, r138, r169, s32, s40, z169, z177, z185, z212, z321, z333, z345, z354, A125, A133, B4, B14, B71, B79, B134, B142, B187, B196, B306, B312, B331, B339, G35, G105, G206, G232, G284, G298, G387, G396, K24, K54, K319, K340, K345, K393, K704, K722, K783, K804, K840, K864, K976, K1127, K1296, K1378, K1472, K1594, K1813, K1840, K2086, K2132, N3, N191, O8, O16, O23, O34
  - `\includeonly` ..... [81](#), [k82](#)
  - `\indent` ..... [A161](#), [C70](#)
  - `\index` [373](#), [F146](#), [H6](#), [H18](#), [J20](#), [J28](#), [K591](#)
  - `\indexentry` ..... [H15](#)
  - `\inf` ..... [z25](#)
  - `\infty` ..... [t236](#)
  - `\init@restore@glb@settings` .....
    - p219, p222, [p224](#)
  - `\init@restore@version` .....
    - r62, r91, [r108](#), r123, r124
  - `\initcatcodetable` ..... [N93](#)
  - `\input` ... [81](#), [455](#), a63, a169, a172, a229, d7, [k163](#), l1164, p16, q106, s145, s156, s166, t10, t11, t12, t13, t20, t21, t25, t26, t55, t56, t57, t58, t540, t541, t542, L213, N16, N131, O70, O132, O256
  - `\input@path` .... [1](#), [6](#), a104, a126, a128, a134, a136, a142, a144, a149, a151, a161, [a228](#), k137, k151
  - `\InputIfFileExists` .....
    - ... [81](#), [454](#), [k160](#), k165, k173, k189, l856, l1247, o325, s119, s137, s148, s158, L339, M8, O126
  - `\inputlineno` a298, b286, b287, b288, g165, g168, s118, O91, O102, O110, O191, O207, O218, O226
  - `\insec@unt` .....
    - b37, b51, b52, b53, b62, b74, b75, b76, b78, b220, b221, b222,
  - File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx, t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx, y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx, D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx, I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx, N=ltluatex.dtx, O=ltfinal.dtx
  - b223, b224, b225, b236, b237, b238, b239, b243, b244, b264, b265, b266, b267, b268, b269, K61
  - `\insert` ... b227, b251, b253, b256, b271, G414, K491, K492, K1793
  - `\install@mathalphabet` .....
    - o442, o459, o466, r269, r272, r358, r359, r456, r508, r511, r518, r533, r534, r541, r885, r887
  - `\int` ..... [t266](#)
  - `\interdisplaylinepenalty` .....
    - i29, [z55](#), z137, z285
  - `\interfootlinepenalty` ..... [b348](#)
  - `\interfootnotelinepenalty` .....
    - b348, i34, G416
  - `\interlinepenalty` i27, y108, y111, F50, F101, F154, G416, K312, K1063, K1067, K1229, K1233
  - `\intextsep` . K1046, K1050, K1065, K1068, K1075, K1208, K1214, K1231, K1234, K1243, [K2192](#)
  - `\intop` ..... [t265](#), [t266](#)
  - `\iota` ..... [t195](#)
  - `\is@range` ..... [p330](#), [p331](#)
  - `\ishortstack` ..... [D42](#)
  - `\itdefault` ..... [s30](#), [t34](#)
  - `\item` ..... g234, y73, y80, y86, y100, z328, z340, z367, [A141](#), A219, C67, E36, E38, I4, I8
  - `\itemindent` . [A9](#), A42, A95, A187, A208
  - `\itemize` ..... [A242](#)
  - `itemize (environment)` ..... [A242](#)
  - `\itemsep` ..... [A1](#), [A176](#)
  - `\iterate` ..... a76, a77, [b378](#)
  - `\itshape` ..... l352, l658, s28, s29, s36, s43, v21, E36, E38, G379
- ## J
- `\J` ..... [O108](#), [O224](#)
  - `\j` ..... l198, l311, l415, l630, O235
  - `\jmath` ..... [t230](#)
  - `\Join` ..... [s105](#)
  - `\joinrel` t379, t386, t388, t390, t392, t394, t396, t398, t400, t404, t406
  - `\jot` ..... [z53](#), z134, z292
- ## K
- `\k` ..... l387, l454, l459, l481, l486, l562, l563, l611, l612, l663, l665, l670, l672
  - `\kappa` ..... [t196](#)
  - `\ker` ..... [z27](#)
  - `\kernel@ifnextchar` ..... [c54](#), d58, d77, d127, [d294](#), d311, L119
  - `\kill` ..... [C59](#)

## L

- `\L` [1192](#), [1329](#), [1400](#), [1625](#), [L489](#), [L509](#), [O248](#)
- `\l` [1201](#), [1331](#), [1418](#), [1631](#), [O248](#)
- `\l@ngrel@x` [d51](#), [d52](#), [d53](#), [d97](#), [d144](#)
- `\label` [x32](#), [F146](#), [J20](#), [J28](#), [K590](#)
- `\labelsep` [A9](#), [A210](#), [A216](#), [E36](#), [E38](#)
- `\labelwidth` [A9](#), [A93](#), [A209](#), [A211](#), [A214](#)
- `\Lambda` [t219](#)
- `\lambda` [t197](#)
- `\land` [t283](#)
- `\angle` [t498](#)
- `\language` [b35](#), [b68](#), [b83](#), [b289](#), [b290](#), [M10](#)
- `\last@fontshape` [o375](#), [o393](#), [o410](#), [o427](#)
- `\lastbox` [z123](#), [z124](#), [A130](#), [A136](#), [A185](#), [F82](#), [F115](#), [K279](#)
- `\LastDeclaredEncoding` [o102](#), [o105](#)
- `\lastpenalty` [v95](#), [v98](#)
- `\lastskip` [b402](#), [b403](#), [b405](#), [b407](#), [i19](#), [i66](#), [i126](#), [i127](#), [i131](#), [i133](#), [i134](#), [i142](#), [i162](#), [i165](#), [i197](#), [i200](#), [i201](#), [v85](#), [v88](#), [A115](#), [A116](#), [A150](#), [A151](#), [D36](#)
- `\LaTeX` [j3](#), [j15](#), [L458](#)
- `\LaTeXe` [j13](#)
- `\latexreleaseversion` [c5](#)
- `\lbrace` [1257](#), [t502](#)
- `\lbrack` [b362](#)
- `\lccode` [g19](#), [g20](#), [g21](#), [g22](#), [g23](#), [g24](#), [1104](#), [y139](#), [y149](#), [O71](#), [O88](#), [O98](#), [O107](#), [O109](#), [O111](#), [O113](#), [O116](#), [O117](#), [O118](#), [O119](#), [O204](#), [O214](#), [O223](#), [O225](#), [O227](#), [O229](#), [O232](#)
- `\lceil` [t506](#)
- `\ldotp` [t409](#), [t412](#), [t523](#)
- `\ldots` [1271](#), [t413](#)
- `\le` [t328](#)
- `\leaders` [b434](#), [t246](#), [t464](#), [t465](#), [t467](#), [t468](#), [C335](#), [D262](#), [D268](#), [F159](#)
- `\leadsto` [s108](#)
- `\leavevmode` [b393](#), [b420](#), [b423](#), [b434](#), [b436](#), [i249](#), [i263](#), [173](#), [1134](#), [1238](#), [1240](#), [1301](#), [1330](#), [1334](#), [1337](#), [1381](#), [1617](#), [1647](#), [1962](#), [v106](#), [y108](#), [y119](#), [y132](#), [y150](#), [z328](#), [z340](#), [z367](#), [A58](#), [A103](#), [B8](#), [B17](#), [B23](#), [B109](#), [B111](#), [B127](#), [B154](#), [B214](#), [B262](#), [B318](#), [B335](#), [B342](#), [C151](#), [D44](#), [D166](#), [F23](#), [F155](#), [G439](#), [K160](#), [K165](#), [I14](#)
- `\left` [t524](#), [t525](#), [t526](#), [t527](#), [z108](#), [z114](#), [z125](#)
- `\Leftarrow` [t324](#), [t400](#), [t406](#)
- File Key: `a=ltltdirchk.dtx`, `b=ltlplain.dtx`, `c=ltlvers.dtx`, `d=ltldefns.dtx`, `e=ltlalloc.dtx`, `f=ltlcntrl.dtx`, `g=ltlerror.dtx`, `h=ltlpar.dtx`, `i=ltltspace.dtx`, `j=ltllogos.dtx`, `k=ltlfiles.dtx`, `l=ltloutenc.dtx`, `m=ltlcounts.dtx`, `n=ltllength.dtx`, `o=ltlfsbas.dtx`, `p=ltlfsstrc.dtx`, `q=ltlfsncmp.dtx`, `r=ltlfsdcl.dtx`, `s=ltlfsini.dtx`, `t=ltlfontdef.dtx`, `u=ltlpreload.dtx`, `v=ltlfontcmd.dtx`, `w=ltlpageno.dtx`, `x=ltltxref.dtx`, `y=ltlmiscen.dtx`, `z=ltlmath.dtx`, `A=ltllists.dtx`, `B=ltlboxes.dtx`, `C=ltltab.dtx`, `D=ltlpictur.dtx`, `E=ltlthm.dtx`, `F=ltlsect.dtx`, `G=ltlfloat.dtx`, `H=ltltdxglo.dtx`, `I=ltlbibl.dtx`, `J=ltlpage.dtx`, `K=ltloutput.dtx`, `L=ltlclass.dtx`, `M=ltlthyphen.dtx`, `N=ltl luatex.dtx`, `O=ltlfinal.dtx`
- `\leftarrow` [t347](#), [t348](#), [t388](#), [t398](#), [t404](#), [t456](#)
- `\leftarrowfill` [t442](#), [t456](#)
- `\lefteqn` [z304](#)
- `\leftharpoonowdown` [t361](#), [t375](#)
- `\leftharpoonowup` [t360](#)
- `\lefthyphenmin` [b292](#), [M11](#)
- `\leftline` [B368](#)
- `\leftmargin` [A9](#), [A52](#), [A53](#), [A94](#), [A146](#), [A148](#)
- `\leftmargini` [z320](#), [A17](#)
- `\leftmarginii` [A17](#)
- `\leftmarginiii` [A17](#)
- `\leftmarginiv` [A17](#)
- `\leftmarginv` [A17](#)
- `\leftmarginvi` [A17](#)
- `\leftmark` [J34](#)
- `\Leftrightarrow` [t323](#)
- `\leftrightharpoonow` [t346](#)
- `\leftskip` [b415](#), [y77](#), [y84](#), [y90](#), [y102](#), [A74](#), [B241](#), [F152](#), [F157](#)
- `\leq` [t327](#), [t328](#)
- `\lfloor` [t510](#)
- `\lg` [z4](#)
- `\lggroup` [t512](#)
- `\lhd` [s111](#)
- `\lhook` [t385](#), [t386](#)
- `\lim` [z6](#)
- `\liminf` [z8](#)
- `\limits` [t446](#), [t450](#), [z107](#), [z246](#)
- `\limsup` [z7](#)
- `\line` [g223](#), [D56](#), [D235](#)
- `\linebreak` [66](#), [i13](#)
- `\linepenalty` [b306](#)
- `\lineskip` [b356](#), [b388](#), [b423](#), [t366](#), [z130](#), [B242](#), [C60](#), [C171](#), [D46](#), [D167](#), [K593](#)
- `\lineskiplimit` [b357](#), [b388](#), [b425](#), [b426](#), [t366](#), [t418](#), [z132](#), [z136](#), [K593](#)
- `\linespread` [o250](#)
- `\linethickness` [D41](#)
- `\linewidth` [k24](#), [z193](#), [z219](#), [z329](#), [z341](#), [z368](#), [z372](#), [z390](#), [A15](#), [A51](#), [A52](#), [A54](#), [B239](#), [C36](#), [G270](#), [K153](#), [K179](#)
- `\list` [A34](#), [A236](#), [A247](#)
- `\listfiles` [455](#), [k201](#)
- `\listparindent` [A9](#), [A41](#), [A50](#)
- `\ll` [t344](#)
- `\llap` [A238](#), [A249](#), [B372](#), [B373](#)
- `\lmoustache` [t469](#)
- `\ln` [z5](#)
- `\not` [t251](#)

- `\LoadClass` ..... 453,  
     L219, L233, L356, L415, L423, L424  
`\LoadClassWithOptions` .... 453, L232  
`\loccount` ..... N15  
`\log` ..... z3  
`\loggingall` ..... b445  
`\loggingoutput` b441, b454, b470, b484  
`\Longleftarrow` ..... t400  
`\longleftarrow` ..... t397  
`\Longleftrightarrow` ..... t406, t408  
`\longleftrightarrow` ..... t404  
`\longmapsto` ..... t402  
`\Longrightarrow` ..... t394  
`\longrightarrow` ..... t395, t402  
`\loop` ..... a76, b378, C341  
`\lor` ..... t285  
`\lower` ..... j2, t366, B166,  
     D15, D75, D162, D163, D200, D201  
`\lower@bound` ..... p340, p341, p352  
`\lowercase` ..... g26, l105,  
     l854, o266, o324, y143, y150, O244  
`\lq` ..... b360  
`\lrbox` ..... B97  
`lrbox` (environment) ..... 284  
`\ltx@sh@ft` ..... b429,  
     l298, l305, l378, l386, l614, l621  
`\luabytecode` ..... N160  
`\luachunk` ..... N168  
`\luafunction` ..... N144  
`\luatexbase` ..... N217  
`\luatexluafunction` ..... a18, a23  
`\luatexversion` ..... a11, N5
- M**
- `\M` ..... b358  
`\m@ne` ..... b39  
`\m@th` ..... b409, b421,  
     j13, t243, t367, t369, t370, t373,  
     t414, t438, t441, t444, t447,  
     t453, t456, t463, t466, t528,  
     z68, z71, z89, z105, z108, z110,  
     z115, z134, z259, z329, z341,  
     z368, z378, B229, B329, C154,  
     F159, G380, G385, G393, G403  
`\magstep` ..... b349  
`\magstephalf` ..... b349  
`\makeatletter` ..... d308, k26,  
     o330, y19, F134, K2, L213, L318  
`\makeatother` ..... d308, L213, O310  
`\makebox` ..... 284, z193, z219, B3  
`\makeglossary` ..... 373, k69, H20  
`\makeindex` ..... 373, k68, H3  
`\makelabel` .....  
     A45, A97, A205, A218, A238, A249  
**File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
   e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
   j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
   o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
   t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
   y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
   D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
   I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
   N=ltluatex.dtx, O=ltfinal.dtx
- `\MakeLowercase` ..... O241, O250  
`\makeph@nt` ..... z84, z86  
`\MakeRobust` ..... d248  
`\makesm@sh` ..... z100, z102  
`\maketitle` ..... 343  
`\MakeUppercase` ..... O234, O234  
`\mandatory@arg` ..... p368, p455,  
     p459, p464, p471, p473, p478,  
     p480, p485, p487, p498, p505, p507  
`\mapsto` ..... t352  
`\mapstochar` ..... t351, t352, t402  
`\marginpar` ..... G312  
`\marginparpush` ..... K92, K1744  
`\marginparsep` .... K91, K1755, K1757  
`\marginparwidth` ... G341, K90, K1757  
`\mark` ..... J23, J31, J39  
`\markboth` ..... J18  
`\markright` ..... J18  
`\marks` ..... N37, O10, O12  
`\math` ..... z238  
`math` (environment) ..... z238  
`\math@bgroup` o473, p260, p266, r53,  
     r81, r142, r172, v113, v114, v121  
`\math@egroup` .....  
     o473, p264, p265, v114, v115, v122  
`\math@fonts` ..... o443, o448,  
     p186, p290, r60, r89, r149, r180  
`\math@fontsfalse` ..... j7, l251,  
     l278, l967, o42, o171, o181, o204  
`\math@fontstrue` ..... o169, o485  
`\math@version` .... o8, o270, o447,  
     o451, o453, o454, o456, p184,  
     r56, r59, r64, r65, r69, r84, r88,  
     r93, r94, r98, r111, r112, r113,  
     r126, r127, r128, r145, r148,  
     r152, r154, r156, r160, r175,  
     r179, r183, r185, r187, r191, s67  
`\mathaccent` ..... r585, r613  
`\mathalpha` .....  
     r684, r843, t88, t89, t90, t91,  
     t92, t93, t94, t95, t96, t97, t98,  
     t99, t100, t101, t102, t103, t104,  
     t105, t106, t107, t108, t109,  
     t110, t111, t112, t113, t114,  
     t115, t116, t117, t118, t119,  
     t120, t121, t122, t123, t124,  
     t125, t126, t127, t128, t129,  
     t130, t131, t132, t133, t134,  
     t135, t136, t137, t138, t139,  
     t140, t141, t142, t143, t144,  
     t145, t146, t147, t148, t149,  
     t216, t217, t218, t219, t220,  
     t221, t222, t223, t224, t225,

t226, t424, t425, t426, t427,  
 t428, t429, t430, t431, t433, t436  
 $\backslash\mathbf{b}$  ..... s14, t70  
 $\backslash\mathbf{b}$  ..... r848,  
 t151, t152, t154, t276, t277,  
 t278, t279, t282, t284, t286,  
 t287, t288, t289, t290, t291,  
 t292, t293, t294, t295, t296,  
 t297, t298, t299, t300, t301,  
 t302, t303, t304, t305, t306,  
 t307, t308, t309, t310, t311, z37  
 $\backslash\mathbf{cal}$  ..... t69  
 $\backslash\mathbf{char}$  .....  
 b421, r627, r669, t228, t240, t521  
 $\backslash\mathbf{char@type}$  ..... r613,  
 r657, r660, r669, r685, r784, r844  
 $\backslash\mathbf{chardef}$  .....  
 . b21, b22, b23, b24, b91, b94,  
 b95, e3, e4, e5, e6, l70, r660, N181  
 $\backslash\mathbf{charzero}$  ..... N181  
 $\backslash\mathbf{choice}$  ..... z61  
 $\backslash\mathbf{close}$  ..... r851, t150,  
 t159, t161, t164, t169, t175,  
 t177, t179, t472, t497, t501,  
 t505, t509, t515, z43, z46, z49, z52  
 $\backslash\mathbf{code}$  ..... r657, t171, t172, t173  
 $\backslash\mathbf{dollar}$  ..... l256, t518  
 $\backslash\mathbf{hellipsis}$  ..... l270, t523  
 $\backslash\mathbf{group}$  . b67, l1174, o15, p257,  
 p263, p269, p270, p281, s82, t529  
 $\backslash\mathbf{hexbox}$  ..... b421, s92  
 $\backslash\mathbf{indent}$  z318, z330, z342, z370, z380  
 $\backslash\mathbf{inner}$  .... t412, t416, t421, t523  
 $\backslash\mathbf{hit}$  ..... s29, t72, t75, t521  
 $\backslash\mathbf{normal}$  ..... t68  
 $\backslash\mathbf{hop}$  ..... r847,  
 t259, t260, t261, t262, t263,  
 t264, t265, t267, t268, t269,  
 t270, t271, t272, t274, t275,  
 t444, t447, z3, z4, z5, z6, z7, z8,  
 z9, z10, z11, z12, z13, z14, z15,  
 z16, z17, z18, z19, z20, z21, z22,  
 z23, z24, z25, z26, z27, z28, z29,  
 z30, z31, z32, z33, z34, z107, z246  
 $\backslash\mathbf{hop}$  r850, t160, t163, t168, t174,  
 t176, t178, t470, t499, t503,  
 t507, t511, t513, z41, z44, z47, z50  
 $\backslash\mathbf{hord}$  ..... r684,  
 r846, t155, t162, t165, t170,  
 t182, t183, t184, t186, t187,  
 t188, t189, t190, t191, t192,  
 t193, t194, t195, t196, t197,  
 t198, t199, t200, t201, t202,  
 t203, t204, t205, t206, t207,  
 t208, t209, t210, t211, t212,  
 t213, t214, t215, t227, t229,  
 t230, t231, t232, t233, t234,  
 t235, t236, t237, t238, t239,  
 t241, t242, t247, t248, t249,  
 t250, t252, t253, t254, t255,  
 t256, t257, t258, t432, t434,  
 t435, t455, t456, t459, t460,  
 t461, t462, t474, t476, t478,  
 t481, t495, t517, t518, t519, t520  
 $\backslash\mathbf{palette}$  .....  
 t365, t369, t372, z60, z69, z82, z98  
 $\backslash\mathbf{paragraph}$  .. l259, m77, m89, t518  
 $\backslash\mathbf{ph@nt}$  ..... z82, z88  
 $\backslash\mathbf{punct}$  .....  
 . r852, t153, t157, t409, t410, t411  
 $\backslash\mathbf{rel}$  ..... r849, t156, t158,  
 t166, t167, t180, t181, t244,  
 t312, t313, t314, t315, t316,  
 t317, t318, t319, t320, t321,  
 t322, t323, t324, t325, t327,  
 t329, t331, t332, t333, t334,  
 t335, t336, t337, t338, t339,  
 t340, t341, t343, t344, t345,  
 t346, t347, t349, t351, t353,  
 t354, t355, t356, t357, t358,  
 t359, t360, t361, t362, t363,  
 t365, t369, t372, t379, t381,  
 t384, t385, t387, t390, t392,  
 t483, t485, t487, t489, t491,  
 t493, z42, z45, z48, z51, z107, z246  
 $\backslash\mathbf{ring}$  ..... t436  
 $\backslash\mathbf{rm}$  ..... s5, t67  
 $\backslash\mathbf{section}$  .... l260, m76, m88, t518  
 $\backslash\mathbf{sf}$  ..... s8, t71, t74  
 $\backslash\mathbf{sm@sh}$  ..... z98, z104  
 $\backslash\mathbf{sterling}$  ..... l268, t518  
 $\backslash\mathbf{strut}$  ..... z94, z112, z113  
 $\backslash\mathbf{surround}$  ..... b409  
 $\backslash\mathbf{symbol}$  ..... r662  
 $\backslash\mathbf{ttt}$  ..... s11, t73  
 $\backslash\mathbf{underscore}$  ..... t518  
 $\backslash\mathbf{version}$  ..... o270, s64, s66  
 $\backslash\mathbf{matrix}$  ..... z110, z114  
 $\backslash\mathbf{max}$  ..... z22  
 $\backslash\mathbf{maxdeadcycles}$  ..... K7  
 $\backslash\mathbf{maxdepth}$  ..... b330, i169,  
 k50, K99, K480, K488, K520,  
 K625, K634, K674, K901, O58  
 $\backslash\mathbf{maxdimen}$  . b279, b331, b332, b388,  
 b426, b442, b453, b469, b484,  
 o495, p338, p391, t366, D239,  
 D273, K265, K1763, K1783,  
 K1788, K2094, K2095, K2097, O62  
**File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=lt pictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx

- \maybe@ic ..... v46, v47, v66
  - \maybe@ic@ ..... v66
  - \maybe@icfalse ..... v80
  - \maybe@ictrue ..... v70
  - \mb@b ..... B49, B59
  - \mb@l ..... B49, B53, B58, D47, D51
  - \mb@r ..... B49, B53, B58, D47, D51
  - \mb@t ..... B50, B57
  - \mbox ..... 284,  
b421, j13, l242, s88, t414, B11,  
B20, B23, D20, G385, G393, G403
  - \mddefault ..... s18, t32, t40
  - \mdseries ..... s16, s17, s91, v20
  - \meaning ..... a214, a223, a294,  
d205, d264, d317, r412, r425,  
r526, r585, r627, r719, r796, r900
  - \medbreak ..... b403
  - \medmuskip ..... t531, z36, z38, z145
  - \medskip ..... b406, i242
  - \medskipamount ..... b405, i243, i245
  - \MessageBreak .. d181, d254, g3, g6,  
g13, g33, g46, g60, g73, g175,  
g177, g183, g190, l121, l859,  
l862, l886, l888, l889, l890, l892,  
l894, l895, l896, l897, l898, l947,  
l949, l957, l964, l1179, o391,  
o425, p20, p21, p67, p88, p281,  
p432, p452, p484, p497, p510,  
q31, q33, r367, r376, r514, v127,  
y23, K552, K1870, K1907, L94,  
L243, L254, L256, L258, L269,  
L325, L326, L328, L329, L330,  
L332, L334, L351, L352, L353,  
L354, L400, L417, L418, L450,  
L478, O136, O137, O138, O140
  - \mho ..... s104
  - \mid ..... t316
  - \min ..... z23
  - \minipage ..... B249
  - minipage (environment) ..... 285
  - \mit ..... s169
  - \mkern . t228, t244, t246, t370, t379,  
t421, t422, t423, t451, t452,  
t453, t454, t455, t456, t457,  
t458, z36, z37, z40, z73, z74, F160
  - \models ..... t392
  - \module\_error ..... 475, N269
  - \module\_info ..... 475, N269
  - \module\_warning ..... 475, N269
  - \modules ..... N224
  - \month ..... a180, c16, L462
  - \moveright ..... K596
  - \mp ..... t303
  - \mscount ..... C338
  - File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
  - \mskip ..... i268,  
z36, z38, z144, z145, z146, z147
  - \mu ..... t198
  - \multicolumn ..... C194
  - \multiput ..... D25, D29
  - \multispan ..... C194, C338
  - \muskip . b29, b55, b77, t451, t452, N34
  - \muskipdef ..... b55, b77, N182
  - \muskipzero ..... N182
- N
- \n ..... N256, N258, N265, N267,  
N390, N463, N488, N557, N558,  
N579, N598, N599, N607, N621
  - \n@space .. t524, t525, t526, t527, t528
  - \nabla ..... t239
  - \narrower ..... b414
  - \natural ..... t253
  - \ne ..... t326
  - \nearrow ..... t319
  - \NeedsTeXFormat .... p12, L248, L521
  - \neg ..... t250, t251
  - \negthinspace ..... i289
  - \neq ..... t326
  - \new@command .....  
. d54, d55, d108, d142, d161, d216
  - \new@environment ... d123, d124, d136
  - \new@fontshape ..... q2, q4, q22, q24
  - \new@mathalphabet ... r409, r430, r441
  - \new@mathgroup .....  
b66, b69, b82, b84, o15, r289, N25
  - \new@mathversion .... r20, r246, r248
  - \new@symbolfont ..... r290, r322
  - \new\_attribute ..... 473, N334
  - \new\_bytecode ..... 473, N367
  - \new\_chunkname ..... 473, N379
  - \new\_whatst ..... 473, N355
  - \newattribute ..... 472, N77, N194
  - \newbox ..... b47, b284,  
b411, e13, z66, A27, B69, C16,  
C17, C18, C302, D6, D298,  
D303, K93, K127, K128, K129
  - \newcatcodetable ..... 472,  
N87, N96, N97, N123, N124, N198
  - \newcommand .....  
35, d54, l4, t29, t30, t31, t32,  
t33, t34, t35, t36, t37, t38, t39,  
t40, t41, D310, K2180, K2183,  
K2186, K2187, K2190, K2191
  - \newcount ..... b47,  
b287, b290, b292, b293, b294,  
b295, b297, b299, b348, e7, e8,  
i62, k9, m36, p25, r27, r254,  
z55, z250, z251, A23, A24, A25,

- A26, A56, A226, A241, B290,  
C11, C12, C13, C14, C15, C294,  
C295, C296, D292, D293, D294,  
D295, D304, F19, F123, F124,  
G3, G271, G272, G273, G274,  
K110, K112, K114, K116, K118,  
K126, K1896, K2178, K2181,  
K2184, K2188, O3, O4, O5, O38
- \newcounter ..... 132, [m10](#)
- \newdimen ..... [b47](#), b279, b281,  
b282, b296, b347, e10, e11, e12,  
i61, p352, p353, z53, z319, A9,  
A10, A11, A12, A13, A14, A15,  
A16, A17, A18, A19, A20, A21,  
A22, B124, B125, C3, C5, C6,  
C7, C8, C139, C297, C298,  
C299, C300, D3, D4, D5, D7,  
D216, D217, D218, D219, D220,  
D221, D296, D297, D299, D300,  
D301, D302, G404, K78, K79,  
K80, K82, K83, K84, K85, K86,  
K87, K88, K89, K90, K91, K92,  
K98, K100, K101, K113, K115,  
K117, K119, K120, K121, K122,  
K123, K124, K125, K1897, K1898
- \newenvironment ..... 36, [d123](#), L460
- \newfam ..... b69, b84, o17, N38
- \newfont ..... [s68](#)
- \newgroup ..... [r47](#)
- \newhelp ..... [b278](#)
- \newif . [d145](#), e9, k7, k8, l906, o169,  
r15, v65, x3, z75, z76, z133,  
z252, A28, A29, A30, A31, A32,  
A33, A138, B304, C19, C212,  
D55, D212, D213, D214, D215,  
F21, F107, K102, K103, K104,  
K105, K106, K107, K108, K109, L2
- \newinsert ..... b177,  
[b215](#), B291, G370, K27, K1762
- \newlabel ..... [x22](#), x34
- \newlanguage ..... [b47](#)
- \newlength ..... 138, [n3](#)
- \newline ..... [i43](#)
- \newlinechar ..... a67, [d5](#)
- \newluabytecode .... 472, [N156](#), N208
- \newluachunkname ... 473, [N164](#), N210
- \newluafunction .....  
..... 472, N4, [N140](#), N192, N204
- \newmarks ..... [O6](#)
- \newmathalphabet ..... [q13](#), [q109](#)
- \newmathalphabet@ ..... [q14](#)
- \newmathalphabet@@ ..... [q109](#)
- \newmathalphabet@@@ ..... [q15](#), [q109](#)
- \newmuskip ..... [b47](#)
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
- \newpage ..... K140, K146, [K157](#)
- \newread ..... [b47](#), k3
- \newsavebox ..... 284, [B69](#)
- \newskip ..... [b47](#), b280,  
b283, b345, b346, e14, e15, e17,  
i245, i246, i247, i286, n3, y79,  
z253, A2, A3, A4, A5, A6,  
A7, A8, K2192, K2193, K2194,  
K2198, K2199, K2202, K2203,  
K2204, K2208, K2209, K2210
- \newtheorem ..... [E1](#)
- \newtie ..... l702, l1050, l1051
- \newtoks ..... b63,  
b79, b278, e16, o280, o281, p201
- \newwhatsit ..... 472, [N148](#), N206
- \newwrite [b47](#), k4, k5, k6, F137, H4, H21
- \newXeTeXintercharclass ..... [O21](#)
- \nfss@catcodes ..... o20, o85,  
o321, o322, [o329](#), t19, t24, t54, K3
- \nfss@text l264, l266, [s88](#), v5, [v105](#), x13
- \NG ..... l401, O248
- \ng ..... l419, O248
- \ni ..... t341, t342
- \no@alphabet@error . [o5](#), r268, r270,  
r446, r447, r461, r470, r556, r557
- \noaccents@ ..... [o488](#), t48
- \noalign ..... t245,  
t439, t442, t444, t445, t449,  
t450, z112, z113, z118, z121,  
z135, z292, C193, C318, C337, D54
- \noboundary ..... b298
- \nobreak .....  
b391, b394, b396, i38, i53, i78,  
i105, i229, i237, i256, i263, i284,  
k67, k79, l340, l342, y69, B367,  
F73, F157, F158, F162, G440,  
J25, J33, K310, K1059, K1225
- \nobreakdashes ..... [i248](#)
- \nobreakspace ..... [i262](#)
- \nocite ..... 375, [I39](#)
- \nocorr ..... [v26](#), v41, v45, v48
- \nocorrlist ..... v72, [v104](#)
- \nofiles ..... 81, [k63](#)
- \noindent ..... F122
- \nointerlineskip .... [b386](#), t245,  
t439, t442, t445, t449, z192,  
z218, D260, D262, K1752, K1760
- \nolimits ..... t266, t273,  
z3, z4, z5, z9, z10, z11, z12, z13,  
z14, z15, z16, z17, z18, z19, z20,  
z21, z26, z27, z28, z29, z31, z34
- \nolinebreak ..... 66, [i13](#)
- \non@alpherr ..... o467, o469,  
r72, r101, [r117](#), r163, r194, r922

`\nonfrenchspacing` ... [b352](#), [b540](#), [k42](#)  
`\nonscript` ..... [z36](#), [z38](#)  
`\nonumber` ..... [z279](#), [z302](#), [z303](#)  
`\nopagebreak` ..... [66](#), [i3](#)  
`\normalbaselines` ... [b356](#), [z108](#), [z110](#)  
`\normalbaselineskip` .....  
                   ... [b345](#), [b357](#), [p142](#), [B243](#)  
`\normalcolor` .... [z245](#), [z315](#), [B62](#),  
                   [B282](#), [F163](#), [G97](#), [G166](#), [K190](#),  
                   [K466](#), [K600](#), [K610](#), [K2115](#), [K2148](#)  
`\normalfont` ..... [o496](#), [s93](#),  
                   [v18](#), [y120](#), [z245](#), [z315](#), [F163](#), [G381](#)  
`\normallineskip` ... [b345](#), [b356](#), [B242](#)  
`\normallineskiplimit` [b345](#), [b357](#), [z136](#)  
`\normalmarginpar` ..... [G367](#)  
`\normalsfcodes` [k38](#), [k40](#), [k42](#), [k62](#), [K589](#)  
`\normalsize` ..... [k36](#),  
                   [v125](#), [G23](#), [G176](#), [G352](#), [K588](#), [L5](#)  
`\not` ..... [t244](#), [t326](#), [t345](#)  
`\not@base` ..... [s100](#),  
                   [s104](#), [s105](#), [s106](#), [s107](#), [s108](#),  
                   [s109](#), [s110](#), [s111](#), [s112](#), [s113](#), [s114](#)  
`\not@math@alphabet` ..... [s5](#), [s8](#),  
                   [s11](#), [s14](#), [s17](#), [s20](#), [s23](#), [s26](#), [s29](#), [s47](#)  
`\notin` ..... [t369](#)  
`\nu` ..... [t199](#)  
`\null` ..... [b370](#),  
                   [l276](#), [l388](#), [l391](#), [l685](#), [l688](#), [x17](#),  
                   [y108](#), [y132](#), [z91](#), [z110](#), [z128](#), [F157](#)  
`\nulldelimiterspace` ..... [b334](#), [t528](#)  
`\nullfont` ..... [y51](#)  
`\number` ... [a81](#), [d2](#), [d91](#), [m51](#), [o451](#),  
                   [o454](#), [p393](#), [r64](#), [r93](#), [r113](#), [r128](#),  
                   [r153](#), [r184](#), [s85](#), [L431](#), [L462](#), [N108](#)  
`\numberline` ..... [F55](#), [F65](#), [F166](#), [G17](#)  
`\numexpr` ..... [b173](#),  
                   [b189](#), [b199](#), [b219](#), [K36](#), [N85](#), [N108](#)  
`\narrow` ..... [t321](#)

## O

`\O` ..... [l194](#), [l308](#), [l403](#), [l624](#), [O247](#)  
`\o` ..... [l203](#), [l313](#), [l421](#), [l632](#), [O247](#)  
`\o@lign` ..... [b423](#),  
                   [l298](#), [l305](#), [l378](#), [l386](#), [l614](#), [l621](#)  
`\oalign` ..... [b423](#)  
`\obeycr` ..... [i295](#)  
`\obeylines` [b373](#), [y114](#), [y127](#), [y128](#), [K557](#)  
`\obeyspaces` ..... [b373](#), [K557](#)  
`\oddsidemargin` ..... [K79](#), [K81](#), [K582](#)  
`\odot` ..... [t298](#)  
`\OE` ..... [l193](#), [l307](#), [l402](#), [l623](#), [O247](#)  
`\oe` ..... [l202](#), [l312](#), [l420](#), [l633](#), [O247](#)  
`\of` ..... [z67](#), [z249](#)  
`\offinterlineskip` ..... [b386](#)

**File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx

`\oint` ..... [t273](#)  
`\ointop` ..... [t272](#), [t273](#)  
`\oldstylenums` ..... [l1171](#), [s78](#)  
`\Omega` ..... [t226](#)  
`\omega` ..... [t209](#)  
`\ominus` ..... [t301](#)  
`\omit` [z121](#), [z122](#), [C328](#), [C331](#), [C338](#), [C342](#)  
`\on@line` [g8](#), [g15](#), [g165](#), [y56](#), [B103](#), [L350](#)  
`\onecolumn` ..... [K148](#)  
`\OnlyDescription` ..... [p5](#), [u3](#)  
`\oalign` ..... [b423](#), [l276](#),  
                   [l302](#), [l382](#), [l388](#), [l390](#), [l585](#), [l618](#),  
                   [l685](#), [l688](#), [l734](#), [s90](#), [t370](#), [t373](#)  
`\openup` ..... [z129](#), [z134](#)  
`\operator@font` .....  
                   ... [t529](#), [z3](#), [z4](#), [z5](#), [z6](#), [z7](#),  
                   [z8](#), [z9](#), [z10](#), [z11](#), [z12](#), [z13](#), [z14](#),  
                   [z15](#), [z16](#), [z17](#), [z18](#), [z19](#), [z20](#), [z21](#),  
                   [z22](#), [z23](#), [z24](#), [z25](#), [z26](#), [z27](#), [z28](#),  
                   [z29](#), [z30](#), [z31](#), [z32](#), [z33](#), [z34](#), [z37](#), [z40](#)  
`\oplus` ..... [t302](#)  
`\optional@arg` .....  
                   ... [p369](#), [p448](#), [p450](#), [p504](#), [p507](#)  
`\OptionNotUsed` .... [L143](#), [L150](#), [L365](#)  
`\Orb` ..... [l568](#)  
`\oslash` ..... [t299](#)  
`\otimes` ..... [t300](#)  
`\outer` ..... [b11](#), [N19](#), [N38](#)  
`\outer@nobreak` .....  
                   .. [G181](#), [G251](#), [G256](#), [G259](#), [G346](#)  
`\outerparskip` ..... [A1](#)  
`\output` ..... [K230](#)  
`\outputpenalty` ..... [K232](#),  
                   [K246](#), [K269](#), [K272](#), [K273](#), [K308](#),  
                   [K1069](#), [K1070](#), [K1235](#), [K1238](#)  
`\oval` ..... [D235](#), [D238](#)  
`\over` ..... [t377](#), [z107](#), [z247](#)  
`\overbrace` ..... [t444](#)  
`\overfullrule` ..... [b329](#), [J55](#)  
`\overleftarrow` ..... [t441](#)  
`\overrightarrow` ..... [t438](#)  
`\owns` ..... [t342](#)

## P

`\P` ..... [l259](#)  
`\p@` ..... [b281](#)  
`\p@equation` ..... [z257](#), [z377](#)  
`\p@reset@font` ..... [s93](#)  
`\p@selectfont` ..... [p117](#)  
`\PackageError` .... [g84](#), [l857](#), [l912](#), [l956](#)  
`\PackageInfo` .....  
                   . [g84](#), [l886](#), [l902](#), [l903](#), [l963](#), [l1248](#)  
`\PackageWarning` .... [g84](#), [l913](#), [l1177](#)  
`\PackageWarningNoLine` ... [g84](#), [K1869](#)

- \pagebreak ..... 66, [i3](#)
- \pagegoal ..... [K1790](#), [K1797](#)
- \pagenumbering ..... 244, w5
- \pageref ..... [x10](#)
- \pageshrink ..... [K512](#), [K516](#), [K532](#)
- \pagestyle ..... [J2](#)
- \pagetotal ..... [K135](#)
- \paperheight ..... [K100](#)
- \paperwidth ..... [K100](#)
- \par ..... [a115](#), [b11](#), [b366](#),  
[b374](#), [b375](#), [b390](#), [b399](#), [b400](#),  
[b401](#), [b403](#), [b405](#), [b407](#), [d6](#), [h3](#),  
[h4](#), [h6](#), [y49](#), [y69](#), [y106](#), [A63](#),  
[A110](#), [A127](#), [A129](#), [A135](#), [A161](#),  
[A164](#), [B234](#), [B278](#), [C168](#), [C344](#),  
[F24](#), [F73](#), [F164](#), [G15](#), [G24](#),  
[G249](#), [J48](#), [J49](#), [K169](#), [K231](#), [K1796](#)
- \par@deathcycles [A56](#), [A77](#), [A79](#), [A80](#)
- \paragraphmark ..... [F126](#)
- \parallel ..... [t315](#)
- \parbox ..... 284, [B186](#)
- \parboxrestore ..... [B245](#)
- \parfillskip ..... [b344](#), [o495](#),  
[y78](#), [y91](#), [y103](#), [A76](#), [B242](#), [F152](#)
- \parindent . [b336](#), [b415](#), [b416](#), [y78](#),  
[y85](#), [y91](#), [y103](#), [A50](#), [B237](#), [F153](#)
- \parsep ..... [A1](#), [A49](#), [A90](#)
- \parshape ..... [A54](#)
- \parskip ..... [b337](#), [y70](#),  
[y101](#), [y103](#), [z386](#), [A49](#), [A73](#),  
[A88](#), [A90](#), [A117](#), [A153](#), [A172](#),  
[A223](#), [B237](#), [C68](#), [K1069](#), [K1237](#)
- \partial ..... [t235](#)
- \partopsep ..... [z384](#), [A1](#), [A61](#)
- \PassOptionsToClass ..... 453, [L120](#)
- \PassOptionsToPackage ..... 453, [L120](#)
- \patch@level ... [c10](#), [c35](#), [c40](#), [c42](#),  
[c44](#), [c47](#), [c49](#), [O258](#), [O270](#), [O272](#)
- \patcj@level ..... [c1](#)
- \patterns ..... [l155](#)
- \penalty .....  
[b395](#), [b396](#), [b397](#), [b398](#), [b399](#),  
[b400](#), [b404](#), [b406](#), [b408](#), [i7](#),  
[i10](#), [i21](#), [i163](#), [i173](#), [i198](#), [i202](#),  
[v101](#), [y108](#), [y111](#), [z37](#), [z137](#),  
[z292](#), [A190](#), [C56](#), [G195](#), [G199](#),  
[G201](#), [G217](#), [G221](#), [G223](#), [K143](#),  
[K171](#), [K172](#), [K1067](#), [K1233](#), [I17](#)
- \perp ..... [t355](#)
- \ph@nt ..... [z77](#), [z78](#), [z79](#), [z80](#)
- \phantom ..... [z75](#)
- \Phi ..... [t224](#)
- \phi ..... [t206](#)
- \Pi ..... [t221](#)
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
- \pi ..... [t201](#)
- \pickup@font .... [l131](#), [o160](#), [o287](#),  
[o402](#), [o436](#), [p122](#), [p285](#), [p287](#), [p289](#)
- \pictur@ ..... [D8](#)
- \picture ..... [D8](#)
- \pm ..... [t304](#)
- \pmatrix ..... [z114](#)
- \pmod ..... [z39](#)
- \poptabs ..... [g210](#), [C127](#)
- \poptracing ..... [p130](#), [p294](#)
- \postdisplaypenalty .....  
[i28](#), [z327](#), [z339](#), [z365](#)
- \pounds ..... [l267](#)
- \Pr ..... [z32](#)
- \pr@@@s ..... [z156](#), [z164](#)
- \pr@@@t ..... [z159](#), [z165](#)
- \pr@m@s ..... [z153](#), [z154](#)
- \prec ..... [t332](#)
- \preceq ..... [t335](#)
- \predisplaypenalty .....  
[b315](#), [z326](#), [z338](#), [z364](#)
- \preload@sizes ..... [q11](#), [q94](#)
- \pretolerance ..... [b302](#), [o497](#)
- \prevdepth ..... [b386](#), [b390](#),  
[b391](#), [i169](#), [i170](#), [i227](#), [i232](#),  
[z135](#), [G196](#), [G198](#), [G218](#), [G220](#)
- \prim@s ..... [z150](#), [z152](#), [z164](#)
- \prime ..... [t172](#), [t237](#), [z153](#)
- \prime@s ..... [z151](#)
- \process@table ..... [k34](#), [r200](#)
- \ProcessOptions .....  
[l877](#), [l915](#), [p71](#), [L151](#), [L194](#), [L419](#)
- \ProcessOptions\* ..... [L151](#)
- \prod ..... [t267](#)
- \propto ..... [t312](#)
- \protect d79, d196, d211, d220, d225,  
d228, d229, d231, d232, d237,  
d238, d243, d246, d247, d269,  
g201, g203, g204, g210, g216,  
g223, g231, g234, g240, k75, l26,  
l32, l51, l55, l159, l167, r475,  
r927, s71, v126, x12, C225, F11,  
F55, F65, F143, G17, K566, I5
- \protected ..... [m103](#)
- \protected@edef .....  
[d230](#), [m101](#), [x37](#), [B298](#), [F43](#),  
[G420](#), [O238](#), [O244](#), [O249](#), [O250](#)
- \protected@write .....  
[k66](#), [k71](#), [x33](#), [F145](#), [H14](#), [H31](#)
- \protected@xdef .....  
[d230](#), [F10](#), [G406](#), [G430](#), [G446](#)
- \provide@command ..... [d155](#), [d156](#)
- \providecommand ..... [d155](#), [l6](#), [K1880](#)
- \provides\_module ..... 475, [N225](#)

\ProvidesClass ..... [452](#), [L107](#)  
 \ProvidesFile .....  
     . [a84](#), [t551](#), [t553](#), [t554](#), [t555](#), [L109](#)  
 \ProvidesPackage .....  
     ..... [453](#), [p13](#), [L90](#), [L107](#), [L522](#)  
 \ProvideTextCommand ..... [l3](#), [l60](#)  
 \ProvideTextCommandDefault ..... [l57](#)  
 \ps@empty ..... [J10](#), [O64](#)  
 \ps@plain ..... [J13](#)  
 \Psi ..... [t225](#)  
 \psi ..... [t208](#)  
 \pushtabs ..... [g210](#), [C124](#)  
 \pushtracing ..... [p115](#), [p275](#)  
 \put ..... [D21](#), [D176](#), [D177](#), [D178](#),  
     [D179](#), [D184](#), [D186](#), [D198](#), [D199](#),  
     [D200](#), [D201](#), [D206](#), [D209](#), [D347](#)

## Q

\qbezier ..... [318](#), [D310](#)  
 \qbeziermax ..... [D309](#), [D331](#)  
 \qqquad ..... [i292](#)  
 \quad ..... [i292](#), [z109](#), [z111](#), [z120](#), [F94](#)  
 \quotedblbase ..... [l422](#), [l634](#)  
 \quotesinglbase ..... [l423](#)

## R

\r [b364](#), [b365](#), [l186](#), [l296](#), [l336](#), [l372](#),  
     [l474](#), [l501](#), [l511](#), [l537](#), [l610](#), [l646](#)  
 \r@@t ..... [z66](#)  
 \radical ..... [r794](#), [r796](#), [r826](#)  
 \raggedbottom ..... [J39](#)  
 \raggedleft ..... [y86](#), [y88](#)  
 \raggedright ..... [y80](#), [y82](#)  
 \raise .. [l276](#), [l335](#), [l338](#), [l586](#), [l648](#),  
     [l735](#), [s91](#), [t373](#), [t421](#), [t423](#), [z73](#),  
     [B352](#), [B361](#), [D22](#), [D32](#), [D74](#),  
     [D162](#), [D237](#), [D254](#), [D280](#), [D358](#)  
 \raisebox ..... [285](#), [B330](#)  
 \rangle ..... [t496](#)  
 \rbrace ..... [l258](#), [t500](#)  
 \rbrack ..... [b362](#)  
 \rceil ..... [t504](#)  
 \Re ..... [t233](#)  
 \ref ..... [x10](#)  
 \refstepcounter ..... [132](#), [x32](#),  
     [z243](#), [z366](#), [A202](#), [E27](#), [F42](#), [G9](#)  
 \registernumber ..... [473](#), [N313](#)  
 \relbar ..... [t384](#), [t392](#), [t394](#), [t400](#)  
 \relbar ..... [t381](#), [t396](#), [t398](#)  
 \relpenalty ..... [b310](#)  
 \rem@pt ..... [o263](#)  
 \remove@angles ..... [p301](#), [p324](#)  
 \remove@nil ..... [r36](#)  
 \remove@star ..... [p301](#), [p307](#)  
 \remove@to@nnil [o262](#), [p301](#), [p327](#), [p440](#)  
 \remove\_from\_callback .... [475](#), [N590](#)  
 \removelastskip [b402](#), [b404](#), [b406](#), [b408](#)  
 \renew@command [d101](#), [d102](#), [d162](#), [d170](#)  
 \renew@environment ..... [d129](#), [d130](#)  
 \renewcommand [36](#), [d101](#), [z314](#), [z334](#), [z355](#)  
 \renewenvironment [36](#), [d129](#), [z363](#), [z375](#)  
 \repeat ..... [a76](#), [a78](#), [b378](#), [C341](#)  
 \RequirePackage ..... [453](#), [K1877](#),  
     [L209](#), [L216](#), [L237](#), [L415](#), [N22](#)  
 \RequirePackageWithOptions [453](#), [L235](#)  
 \reserved@a ..... [a116](#),  
     [a120](#), [a121](#), [a190](#), [a191](#), [a194](#),  
     [a212](#), [a216](#), [a238](#), [a245](#), [a248](#),  
     [a250](#), [a251](#), [a258](#), [a261](#), [a263](#),  
     [a264](#), [a271](#), [a274](#), [a276](#), [a302](#),  
     [a303](#), [a304](#), [b177](#), [c12](#), [c18](#), [c33](#),  
     [d94](#), [d97](#), [d110](#), [d111](#), [d112](#),  
     [d114](#), [d161](#), [d162](#), [d163](#), [d169](#),  
     [d170](#), [d171](#), [d172](#), [d175](#), [d194](#),  
     [d203](#), [d207](#), [d262](#), [d266](#), [d291](#),  
     [d300](#), [f33](#), [f37](#), [g189](#), [i251](#), [i254](#),  
     [k76](#), [k77](#), [k99](#), [k100](#), [k138](#), [k140](#),  
     [k145](#), [k147](#), [k149](#), [k155](#), [k159](#),  
     [k167](#), [k170](#), [k183](#), [k184](#), [k188](#),  
     [k194](#), [k213](#), [k217](#), [k221](#), [l75](#),  
     [l77](#), [l85](#), [l102](#), [l107](#), [o30](#), [o33](#),  
     [o36](#), [o70](#), [o73](#), [o75](#), [o112](#), [o116](#),  
     [o323](#), [o326](#), [o374](#), [o375](#), [o390](#),  
     [o393](#), [o398](#), [o409](#), [o410](#), [o423](#),  
     [o427](#), [o432](#), [o459](#), [o462](#), [o463](#),  
     [o471](#), [p150](#), [p152](#), [p154](#), [p164](#),  
     [p166](#), [p169](#), [p298](#), [p299](#), [p312](#),  
     [p313](#), [q53](#), [q57](#), [r356](#), [r365](#),  
     [r367](#), [r411](#), [r414](#), [r424](#), [r427](#),  
     [r525](#), [r527](#), [r585](#), [r586](#), [r627](#),  
     [r628](#), [r719](#), [r720](#), [r796](#), [r797](#),  
     [r899](#), [r901](#), [r917](#), [r919](#), [r920](#),  
     [r925](#), [v30](#), [v31](#), [v36](#), [v37](#), [v48](#),  
     [v51](#), [v71](#), [v78](#), [y41](#), [y42](#), [y54](#),  
     [y55](#), [y59](#), [y64](#), [y65](#), [z294](#), [z295](#),  
     [z296](#), [z297](#), [z299](#), [B51](#), [B52](#),  
     [B55](#), [B98](#), [B104](#), [C202](#), [C206](#),  
     [C211](#), [C230](#), [C319](#), [C320](#), [D78](#),  
     [D80](#), [D84](#), [D241](#), [G29](#), [G30](#),  
     [G32](#), [G33](#), [G63](#), [G67](#), [G72](#), [G74](#),  
     [G76](#), [G78](#), [G83](#), [G84](#), [G132](#),  
     [G136](#), [G142](#), [G145](#), [G148](#),  
     [G151](#), [K37](#), [K46](#), [K48](#), [K50](#),  
     [K787](#), [K807](#), [K1873](#), [K1875](#),  
     [K1876](#), [K1965](#), [K1967](#), [K1973](#),  
     [K1976](#), [L77](#), [L84](#), [L88](#), [L202](#),  
     [L205](#), [L249](#), [L250](#), [L253](#), [L290](#),  
     [L294](#), [L306](#), [L307](#), [L309](#), [L319](#),

File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=ltterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx

- L359, L524, L526, O73, O90,  
 O92, O93, O101, O103, O104,  
 O148, O179, O185, O186, O188,  
 O190, O194, O206, O208, O209,  
 O217, O219, O220, O236, O237,  
 O238, O239, O242, O243, O244,  
 O245, O271, O274, O275, O292  
 \reserved@b ..... a117, a118,  
 d86, d88, d95, d112, d113, d204,  
 d205, d207, d263, d264, d266,  
 d292, d302, f33, f34, f37, i252,  
 i253, i260, k98, k100, k150,  
 k152, k154, k216, k222, l78,  
 l85, o60, o62, o115, o116, o460,  
 o471, q47, q54, q71, q73, r282,  
 r284, r337, r339, r364, r365,  
 r366, r401, r403, r482, r484,  
 r529, r530, r531, r538, v35, v36,  
 v49, v51, v78, v79, C207, C209,  
 C211, G43, G44, G112, G113,  
 K696, K699, K713, K716, K733,  
 K736, L78, L79, L81, L298,  
 L304, L307, L467, L468, L470,  
 L496, O76, O78, O82, O151,  
 O153, O157, O237, O243, O292  
 \reserved@c ..... a118, a123,  
 d297, d300, d302, d305, k205,  
 k206, o61, o62, o461, o464, q48,  
 q55, q61, q68, r33, r37, r283,  
 r284, r338, r339, r402, r403,  
 r483, r484, r506, r515, r530,  
 r544, r709, r725, r734, r762,  
 r773, r812, r825, r827, v50, v52,  
 v59, L444, L445, L446, L456,  
 L472, L479, L504, O80, O85,  
 O95, O155, O176, O177, O178,  
 O180, O181, O182, O183, O184,  
 O187, O189, O201, O211, O294  
 \reserved@d ..... a121, a124,  
 d290, d299, k204, k206, q61,  
 q68, q70, q74, r717, r725, r734,  
 r770, r773, r820, r825, r829, O295  
 \reserved@e .. i36, i38, i47, i53, q39,  
 q45, q70, q73, q74, r34, r39, O296  
 \reserved@f .....  
 . i37, i38, i53, l853, l854, l855,  
 l856, l858, l865, o155, o157,  
 o163, o164, p336, p347, p351,  
 p355, p361, p364, p403, p440,  
 p443, q27, q38, q45, q71, q73, O297  
 \reset@font ..... s93, x13, B295,  
 G175, G351, G415, J14, K587, I20  
 \restglb@settings ..... p222, p232  
 \restore@mathversion .....  
 File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,  
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
 N=ltluatex.dtx, O=ltfinal.dtx  
 ..... r107, r110, r125, r133  
 \restore@protect ..... d230  
 \restorecr ..... i295  
 \reversemarginpar ..... G367  
 \rfloor ..... t508  
 \rgroup ..... t512  
 \rhd ..... s113  
 \rho ..... t202  
 \rho hook ..... t387, t388  
 \right ..... t524,  
 t525, t526, t527, z109, z114, z127  
 \Rrightarrow ..... t325, t394, t406  
 \rightharpoonup ..... t349,  
 t350, t352, t386, t396, t404, t455  
 \rightharpoonupfill ..... t439, t453  
 \rightharpoonupdown ..... t363  
 \rightharpoonupup ..... t362, t374  
 \righthyphenmin ..... b293, M11  
 \rightleftharpoons ..... t372  
 \rightline ..... B368  
 \rightmargin ..... A9, A40, A51  
 \rightmark ..... J34  
 \rightskip ..... b416, y77,  
 y83, y90, y102, A75, B241, F152  
 \rlap ..... l335,  
 l338, l648, z304, z315, B372, C70  
 \rlh@ ..... t372, t373  
 \rmdefault ..... s6, s81, t29, t39  
 \rmfamily ..... s4, s5, v15  
 \rmoustache ..... t471  
 \Roman ..... 132, m47  
 \roman ..... 132, m46  
 \romannumeral .....  
 .... m52, m53, A43, A234, A245  
 \root ..... z66, z249  
 \rootbox ..... z66  
 \rq ..... b360  
 \rule ..... 285, B302, B305, G425  
  
 S  
 \S ..... l260  
 \s@fct@ ..... p380, p444  
 \s@fct@fixed ..... p501  
 \s@fct@gen ..... p456  
 \s@fct@genb ..... p461  
 \s@fct@sugen ..... p456  
 \s@fct@sugenb ..... p461  
 \s@fct@sub ..... p468  
 \s@fct@subf ..... p493  
 \samepage ..... 66, i27  
 \savebox ..... 284, B70  
 \savecatcodetable . N120, N135, N137  
 \sb ..... z142

- \sbox ..... 284, b410,  
j4, A205, B76, B83, B86, B91, B96
- \scan@fontshape ..... q7, q40, q43
- \scan@fontshape ..... q6, q26, q37
- \scdefault ..... s27, t34
- \scriptfont ..... p292
- \scriptfont@name ..... p287, p292
- \scriptscriptfont ..... p293
- \scriptscriptstyle ..... z65, z68
- \scriptspace ..... b335
- \scriptstyle ..... t243, z64
- \scshape ..... l249, s25, s26, v23
- \searrow ..... t320
- \sec ..... z20
- \secdef ..... F125
- \sectionmark ..... F126
- \select@group o444, o463, r48, r236,  
r273, r411, r464, r473, r511, r543
- \selectfont ..... j7, l251,  
l278, l354, l660, l878, l950, l968,  
o248, p112, s6, s9, s12, s15, s18,  
s21, s24, s27, s30, s74, G383, G391
- \seriesdefault ..... r239, s96, t38
- \set@mathdelimiter ..... r771, r787
- \set@color ..... B61
- \set@display@protect .....  
..... d3, d228, g7, g14, g34, g61
- \set@fontsize . o251, o253, p119, p132
- \set@mathaccent ..... r588, r596, r612
- \set@mathchar ..... r646, r656
- \set@mathdelimiter ... r722, r731, r783
- \set@mathradical ..... r244, r822
- \set@mathsymbol ..... r630, r638, r659
- \set@simple@size@args .....  
... p302, p315, p322, p343, p357
- \set@size@func@args p305, p307, p365
- \set@size@func@args@ ..... p365
- \set@typeset@protect ..... d228,  
d247, C170, C196, K572, K574
- \setattribute ..... 473, N85, N195
- \setcounter .....  
... 132, k127, m2, m37, A225,  
K2179, K2182, K2185, K2189
- \setlanguage ..... b299
- \setlength 138, n4, z382, z387, z388,  
z389, B42, B156, B216, B219,  
B264, B320, B321, B322, B350,  
B351, B358, B359, B360, C149,  
C343, K2195, K2196, K2197,  
K2200, K2201, K2205, K2206,  
K2207, K2211, K2212, K2213
- \SetMathAlphabet .....  
... o12, q140, q141, r480, t74, t75
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
- \SetMathAlphabet@ ... r418, r487, r496
- \setminus ..... t307
- \setrangecatcode .....  
..... N99, N107, N116, N117
- \SetSymbolFont .... r335, t64, t65, t66
- \SetSymbolFont@ .... r308, r342, r350
- \settodepth ..... 138, n17
- \settoheight ..... 138, n17
- \settowidth ..... 138, n17
- \sf@size j6, l251, o189, o208, o483,  
p282, p286, G385, G393, G403
- \sfcode ..... b352, b353, b354,  
b355, b439, i258, k39, O99, O215
- \sfdefault ..... s9, t29
- \sffamily ..... s7, s8, v16
- \sh@ft ..... b427
- \shapedefault ..... r240, s97, t38
- \sharp ..... t254
- \shipout ..... K571
- \shortstack ..... D42
- \showboxbreadth .....  
... b324, b442, b495, b512, b528
- \showboxdepth .....  
b325, b442, b494, b511, b529, o497
- \showhyphens ..... o491
- \showoutput ..... b441
- \showoverfull . b440, b443, b477, b485
- \Sigma ..... t222
- \sigma ..... t203
- \sim ..... t353, t365
- \simeq ..... t354
- \sin ..... z9
- \sinh ..... z11
- \sixt@n .. a66, b16, b64, b65, b80,  
b81, b82, o15, r84, r175, r580,  
r582, r622, r624, r665, r667,  
r705, r707, r713, r715, r758,  
r760, r766, r768, r808, r810,  
r816, r818, D135, D150, D152,  
G62, G80, G131, G153, K915,  
K961, K1100, K1268, K1502,  
K1566, K1623, K1693, K1919,  
K1928, K1984, K2000, K2033, N30
- \size@update .. p128, p139, p158, p160
- \sizefn@info .....  
... p306, p308, p316, p344, p358
- \skew ..... t451
- \skip b28, b53, b76, b192, b223, b267,  
B281, G371, K290, K464, N33
- \skip@ ..... b41,  
b389, b391, b392, b394, v88, v91
- \skipdef ..... b45, b53, b76, N183
- \skipzero ..... N183
- \slash ..... b395

- `\sldefault` ..... [s24](#), [t34](#)
  - `\sloppy` ..... [B244](#), [J43](#), [J48](#)
  - `\sloppypar` ..... [J48](#)
  - `sloppypar` (environment) ..... [J48](#)
  - `\slshape` ..... [l345](#), [l651](#), [s22](#), [s23](#), [v22](#)
  - `\smallbreak` ..... [b403](#)
  - `\smallint` ..... [t275](#)
  - `\smallskip` ..... [b404](#), [i242](#)
  - `\smallskipamount` ..... [b403](#), [i242](#), [i245](#)
  - `\smash` . [t381](#), [t453](#), [t454](#), [t457](#), [t458](#), [z95](#)
  - `\smile` ..... [t358](#)
  - `\sp` ..... [z142](#)
  - `\sp@n` ..... [C338](#)
  - `\space` ..... [b368](#)
  - `\spacefactor` ..... [b393](#), [b394](#),  
[i67](#), [i76](#), [i89](#), [i103](#), [i117](#), [i258](#),  
[i274](#), [i279](#), [l70](#), [l71](#), [G440](#), [G442](#)
  - `\spaceskip` ..... [s80](#)
  - `\spadesuit` ..... [t258](#)
  - `\span` ..... [C342](#)
  - `\split@name` [o291](#), [o303](#), [o354](#), [p473](#), [p487](#)
  - `\splitfirstmark` ..... [K2100](#)
  - `\splitmaxdepth` ... [b331](#), [G418](#), [K2094](#)
  - `\splittopskip` ..... [b343](#), [G417](#)
  - `\sqcap` ..... [t290](#)
  - `\sqcup` ..... [t291](#)
  - `\sqrt` ..... [z248](#)
  - `\sqrtsign` ..... [t437](#), [z71](#), [z248](#)
  - `\sqsubset` ..... [s109](#)
  - `\sqsubseteq` ..... [t313](#)
  - `\sqsupset` ..... [s110](#)
  - `\sqsupseteq` ..... [t314](#)
  - `\SS` ..... [l253](#), [l404](#), [O248](#)
  - `\ss` ..... [l204](#), [l314](#), [l424](#), [l635](#), [O248](#)
  - `\ssf@size` .....  
[l277](#), [o190](#), [o209](#), [o484](#), [p282](#), [p288](#)
  - `\stackrel` ..... [z246](#)
  - `\star` ..... [t311](#)
  - `\stepcounter` .....  
[l32](#), [m17](#), [m27](#), [o456](#), [r48](#), [x36](#),  
[z256](#), [z299](#), [z376](#), [G405](#), [G429](#), [K617](#)
  - `\stop` ..... [y49](#)
  - `\stretch` ..... [i288](#)
  - `\strip@prefix` ..... [a106](#),  
[a223](#), [a294](#), [d205](#), [d264](#), [d316](#), [o441](#)
  - `\strip@pt` ..... [b431](#),  
[o181](#), [o187](#), [o188](#), [o189](#), [o190](#),  
[o203](#), [o207](#), [o263](#), [o483](#), [o484](#), [p134](#)
  - `\strut` ..... [b411](#), [z121](#), [z122](#), [C29](#)
  - `\strutbox` ..... [b411](#), [p143](#),  
[B302](#), [C159](#), [C160](#), [G418](#), [G425](#)
  - `\sub@sfcnt` ..... [p468](#), [p469](#), [p470](#)
  - `\subf@sfcnt` ..... [p493](#), [p494](#), [p495](#)
  - `\subparagraphmark` ..... [F126](#)
  - File Key:** `a=ltldirchk.dtx`, `b=ltplain.dtx`, `c=ltvers.dtx`, `d=ltdefs.dtx`,  
`e=ltalloc.dtx`, `f=ltcntrl.dtx`, `g=lterror.dtx`, `h=ltpar.dtx`, `i=ltspace.dtx`,  
`j=ltlogos.dtx`, `k=ltfiles.dtx`, `l=ltoutenc.dtx`, `m=ltcounts.dtx`, `n=ltlength.dtx`,  
`o=ltfssbas.dtx`, `p=ltfssstrc.dtx`, `q=ltfsscmp.dtx`, `r=ltfssdcl.dtx`, `s=ltfssini.dtx`,  
`t=fontdef.dtx`, `u=preload.dtx`, `v=ltfntcmd.dtx`, `w=ltpageno.dtx`, `x=ltxref.dtx`,  
`y=ltmiscen.dtx`, `z=ltmath.dtx`, `A=ltlists.dtx`, `B=ltboxes.dtx`, `C=lttab.dtx`,  
`D=ltpictur.dtx`, `E=ltthm.dtx`, `F=ltsect.dtx`, `G=ltfloat.dtx`, `H=ltidxglo.dtx`,  
`I=ltbibl.dtx`, `J=ltpage.dtx`, `K=ltoutput.dtx`, `L=ltclass.dtx`, `M=lthyphen.dtx`,  
`N=ltluatex.dtx`, `O=ltfinal.dtx`
  - `\subsectionmark` ..... [F126](#)
  - `\subset` ..... [t337](#)
  - `\subseteq` ..... [t339](#)
  - `\subst@correction` ..... [o50](#), [o56](#)
  - `\subst@fontshape` ..... [q8](#), [q80](#)
  - `\subst@size` ..... [p419](#)
  - `\subsubsectionmark` ..... [F126](#)
  - `\succ` ..... [t331](#)
  - `\succeq` ..... [t334](#)
  - `\sum` ..... [t268](#)
  - `\sup` ..... [z24](#)
  - `\suppressfloats` ..... [K1882](#)
  - `\supset` ..... [t336](#)
  - `\supseteq` ..... [t338](#)
  - `\surd` ..... [t240](#)
  - `\sw@slant` ..... [v74](#), [v84](#)
  - `\swarrow` ..... [t322](#)
  - `\symbol` ..... [l122](#), [s68](#)
  - `\symletters` ..... [l1174](#), [s82](#)
  - `\symoperators` ..... [t529](#)
- T**
- `\T` ..... [g23](#), [L505](#), [L509](#), [L510](#)
  - `\t` .. [l232](#), [l595](#), [l700](#), [l894](#), [l1161](#), [l1163](#)
  - `\t@st@ic` ..... [v73](#), [v77](#)
  - `\tabbing` ..... [C60](#)
  - `\tabbingsep` ..... [C119](#), [C121](#), [C139](#)
  - `\tabcolsep` ..... [C220](#), [C297](#)
  - `\tabskip` ..... [b422](#), [z138](#),  
[z139](#), [z261](#), [z264](#), [z267](#), [z269](#),  
[z380](#), [z393](#), [z396](#), [z398](#), [C140](#), [C165](#)
  - `\tabular` ..... [C147](#)
  - `\tabular*` ..... [C148](#)
  - `\tabularnewline` ..... [C167](#), [C180](#)
  - `\tan` ..... [z15](#)
  - `\tanh` ..... [z17](#)
  - `\tau` ..... [t204](#)
  - `\tc@check@accent` .....  
[l973](#), [l1049](#), [l1051](#), [l1053](#)
  - `\tc@check@symbol` ..... [l973](#),  
[l1043](#), [l1045](#), [l1047](#), [l1055](#), [l1057](#),  
[l1059](#), [l1061](#), [l1063](#), [l1065](#), [l1067](#),  
[l1069](#), [l1071](#), [l1073](#), [l1075](#), [l1077](#),  
[l1079](#), [l1081](#), [l1083](#), [l1085](#), [l1087](#),  
[l1089](#), [l1091](#), [l1093](#), [l1095](#), [l1097](#),  
[l1099](#), [l1101](#), [l1103](#), [l1105](#), [l1107](#),  
[l1109](#), [l1111](#), [l1113](#), [l1115](#), [l1117](#),  
[l1119](#), [l1121](#), [l1123](#), [l1125](#), [l1127](#),  
[l1129](#), [l1131](#), [l1133](#), [l1135](#), [l1137](#),  
[l1139](#), [l1141](#), [l1143](#), [l1145](#), [l1147](#),  
[l1149](#), [l1151](#), [l1153](#), [l1155](#), [l1157](#)
  - `\tc@error` ..... [l953](#), [l974](#)
  - `\tc@errorwarn` ..... [l912](#), [l913](#), [l946](#)
  - `\tc@fake@euro` ..... [l961](#), [l1042](#)

`\tc@forcedfalse` ..... 1906  
`\tc@forcedtrue` ..... 1911  
`\tc@subst` ..... 1945, 1945, 1973  
`\tencirc` ..... u10, D37, D306  
`\tencircw` ..... u10, D39  
`\tenln` ..... u9, D37, D38, D305, D307  
`\tenlnw` ..... u9, D39, D40  
`\TeX` ..... j1, j12, N189  
`\TeXOrMath` ..... m70, m86  
`\text@command` ..... v8, v29  
`\textacutedbl` ..... 1755, 11006  
`\textascendercompwordmark` . 1705, 1989  
`\textasciicircum` ..... 1805, 11030  
`\textasciibreve` ..... 1753, 11003  
`\textasciicaron` ..... 1754, 11004  
`\textasciicircum` ..... 1236, 1425  
`\textasciidieresis` ..... 1793, 11020  
`\textasciigrave` ..... 1744, 11001  
`\textasciimacron` ..... 1800, 11025  
`\textasciitilde` ..... 1237, 1426  
`\textasteriskcentered` .....  
..... 1217, 1571, 1715, 1996, m73, m79  
`\textbackslash` ..... 1218, 1427, 1572  
`\textbaht` ..... 1779, 11136, 11137  
`\textbar` ..... 1219, 1428, 1573  
`\textbardbl` 1220, 1574, 1759, 11009, m78  
`\textbf` ..... v19  
`\textbigcircle` 1583, 1732, 11088, 11089  
`\textblank` ..... 1712, 11058, 11059  
`\textborn` ..... 1745, 11094, 11095  
`\textbraceleft` .. 1221, 1257, 1429, 1575  
`\textbraceright` . 1222, 1258, 1430, 1576  
`\textbrokenbar` ..... 1791, 11018  
`\textbullet` .... 1223, 1577, 1761, 11011  
`\textcapitalcompwordmark` .. 1704, 1988  
`\textcelsius` ..... 1762, 11012  
`\textcent` ..... 1787, 11015  
`\textcentoldstyle` . 1764, 11110, 11111  
`\textcircled` ..... 1229, 1233, 1249,  
1250, 1584, 1733, 1893, 11158, 11160  
`\textcircledP` ..... 1798, 11152, 11153  
`\textcolonmonetary` . 1766, 11112, 11113  
`\textcommabelow` . 1273, 1275, 1281, 1282  
`\textcompsubstdefault` .... 1950, 1952  
`\textcompwordmark` ..... 1238, 1431  
`\textcopyleft` ..... 1796, 11150, 11151  
`\textcopyright` . 1233, 1266, 1794, 11021  
`\textcurrency` .....  
..... 1789, 1888, 1892, 11046, 11047  
`\textdagger` ..... 1225,  
1261, 1579, 1757, 11007, m74, m80  
`\textdaggerdbl` ..... 1224,  
1262, 1578, 1758, 11008, m75, m81  
`\textdblhyphen` .... 1716, 11060, 11061  
**File Key:** a=ltltdirchk.dtx, b=ltlplain.dtx, c=ltlvers.dtx, d=ltldefs.dtx,  
e=ltlalloc.dtx, f=ltlcntrl.dtx, g=ltlerror.dtx, h=ltlpar.dtx, i=ltltspace.dtx,  
j=ltllogos.dtx, k=ltlfiles.dtx, l=ltloutenc.dtx, m=ltlcounts.dtx, n=ltllength.dtx,  
o=ltlffssbas.dtx, p=ltlffsstrc.dtx, q=ltlffsscmp.dtx, r=ltlffssdcl.dtx, s=ltlffssini.dtx,  
t=fontdef.dtx, u=ltlpreload.dtx, v=ltlfontcmd.dtx, w=ltlpageno.dtx, x=ltltxref.dtx,  
y=ltlmiscen.dtx, z=ltlmath.dtx, A=ltllists.dtx, B=ltlboxes.dtx, C=ltltab.dtx,  
D=ltlpictur.dtx, E=ltlthm.dtx, F=ltlsect.dtx, G=ltlfloat.dtx, H=ltlidxglo.dtx,  
I=ltlbibl.dtx, J=ltlpage.dtx, K=ltloutput.dtx, L=ltlclass.dtx, M=ltlthyphen.dtx,  
N=ltl luatex.dtx, O=ltlfinal.dtx  
`\textdblhyphenchar` . 1752, 11106, 11107  
`\textdegree` ..... 1801, 11026  
`\textdied` ..... 1747, 11098, 11099  
`\textdiscount` ..... 1781, 11140, 11141  
`\textdiv` ..... 1818, 11040  
`\textdivorced` ..... 1746, 11096, 11097  
`\textdollar` ..... 1205, 1256, 1343,  
1432, 1649, 1713, 1994, 11166, 11168  
`\textdollaroldstyle` 1763, 11108, 11109  
`\textdong` ..... 1775, 11130, 11131  
`\textdownarrow` .... 1743, 11092, 11093  
`\texteightoldstyle` . 1726, 11078, 11079  
`\textellipsis` ..... 1245, 1270  
`\textemdash` ..... 1206, 1315, 1433, 1636  
`\textendash` ..... 1207, 1316, 1434, 1637  
`\textestimated` 1782, 1891, 11044, 11045  
`\texteuro` .... 1816, 1889, 11041, 11042  
`\textexclamdown` .....  
..... 1208, 1317, 1319, 1435, 1638  
`\textfiveoldstyle` . 1723, 11072, 11073  
`\textfloatsep` .....  
K638, K651, K2014, K2064, K2192  
`\textflorin` ..... 1765, 11013  
`\textfont` ..... p291, z148  
`\textfont@name` ..... p285, p291  
`\textfouroldstyle` . 1722, 11070, 11071  
`\textfraction` ..... K1827, K1830,  
K1854, K1857, K2006, K2186  
`\textfractionsolidus` ..... 1717, 1997  
`\textgravedbl` ..... 1756, 11005  
`\textgreater` ..... 1231, 1436, 1594  
`\textguarani` ..... 1769, 11118, 11119  
`\textheight` ..... k16, k17, G261,  
G262, G265, G291, G305, K85,  
K199, K200, K248, K373, K421,  
K448, K616, K673, K725, O62, O63  
`\texthyphen` ..... 1210, 1322, 1438, 1640  
`\texthyphenchar` . 1209, 1321, 1437, 1639  
`\textinterrobang` .. 1773, 11126, 11127  
`\textinterrobangdown` 1774, 11128, 11129  
`\textit` ..... v21  
`\textlangle` ..... 1728, 11082, 11083  
`\textlbrackdbl` ..... 1740, 1999  
`\textleaf` ..... 1748, 11100, 11101  
`\textleftarrow` .... 1710, 11054, 11055  
`\textless` ..... 1230, 1439, 1593  
`\textlira` ..... 1771, 11122, 11123  
`\textlnot` ..... 1797, 11023  
`\textlquill` ..... 1785, 11146, 11147  
`\textmarried` ..... 1749, 11102, 11103  
`\textmd` ..... v19  
`\textmho` ..... 1731, 11086, 11087  
`\textminus` ..... 1729, 1998  
`\textmu` ..... 1806, 11031

`\textmusicalnote` .. 1750, 11104, 11105  
`\textnaira` ..... 1768, 11116, 11117  
`\textnineoldstyle` .. 1727, 11080, 11081  
`\textnormal` ..... v15  
`\textnumero` ..... 1780, 11138, 11139  
`\textogonekcentered` .. 1389, 1562, 1563  
`\textohm` ..... 1739, 1890, 11043  
`\textonehalf` ..... 1814, 11037  
`\textoneoldstyle` .. 1719, 11064, 11065  
`\textonequarter` ..... 1813, 11036  
`\textonesuperior` ..... 1810, 11034  
`\textopenbullet` ... 1783, 11142, 11143  
`\textordfeminine` ... 1254, 1795, 11022  
`\textordmasculine` .. 1255, 1811, 11035  
`\TextOrMath` .. m73, m74, m75, m76,  
m77, m78, m79, m80, m81, m93  
`\textparagraph` .....  
. 1226, 1259, 1580, 1807, 11032, m77  
`\textperiodcentered` .....  
. 1227, 1581, 1808, 11033  
`\textpertenthousand` .....  
. 1394, 1777, 11132, 11133, 11170  
`\textperthousand` 1392, 1760, 11010, 11169  
`\textpeso` ..... 1770, 11120, 11121  
`\textpilcrow` ..... 1778, 11134, 11135  
`\textpm` ..... 1802, 11027  
`\textquestiondown` .....  
. 1211, 1318, 1320, 1440, 1641  
`\textquotedbl` ..... 1443  
`\textquotedblleft` 1212, 1323, 1441, 1642  
`\textquotedblright` 1213, 1324, 1442, 1643  
`\textquoteleft` .. 1214, 1325, 1444, 1644  
`\textquoteright` . 1215, 1326, 1445, 1645  
`\textquotesingle` ..... 1714, 1995  
`\textquotestraightbase` ... 1706, 1990  
`\textquotestraightdblbase` . 1707, 1991  
`\textrangle` ..... 1730, 11084, 11085  
`\textrbrackdbl` ..... 1741, 11000  
`\textrecipe` ..... 1772, 11124, 11125  
`\textreferencemark` . 1809, 11154, 11155  
`\textregistered` 1249, 1250, 1799, 11024  
`\textrightarrow` ... 1711, 11056, 11057  
`\textrm` ..... v15  
`\textrquill` ..... 1786, 11148, 11149  
`\textsc` ..... v21  
`\textsection` ..... 1228,  
1260, 1446, 1582, 1792, 11019, m76  
`\textservicemark` .. 1784, 11144, 11145  
`\textsevenoldstyle` . 1725, 11076, 11077  
`\textsf` ..... v15  
`\textsixoldstyle` .. 1724, 11074, 11075  
`\textsl` ..... v21  
`\textsterling` .... 1216, 1268, 1350,  
1447, 1656, 1788, 11016, 11165, 11167  
**File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx  
`\textstyle` ..... j15, t377, z63  
`\textsubscript` .... G386, G397, G398  
`\textsuperscript` 1252, 1254, 1255, G382  
`\textsurd` ..... 1812, 11156, 11157  
`\TextSymbolUnavailable` ..... 13, 1612  
`\textthreeoldstyle` . 1721, 11068, 11069  
`\textthreequarters` ..... 1815, 11038  
`\textthreequartersemdash` .. 1709, 1993  
`\textthreesuperior` ..... 1804, 11029  
`\texttildelow` ..... 1751, 11002  
`\texttimes` ..... 1817, 11039  
`\texttrademark` ..... 1252, 1776, 11014  
`\texttt` ..... v15  
`\texttwelveudash` ..... 1708, 1992  
`\texttwooldstyle` .. 1720, 11066, 11067  
`\texttwosuperior` ..... 1803, 11028  
`\textunderscore` ..... 1239, 1264, 1448  
`\textup` ..... v21  
`\textuparrow` ..... 1742, 11090, 11091  
`\textvisiblespace` ..... 1241, 1449  
`\textwidth` ..... k18, B269,  
G270, K86, K151, K175, K192,  
K601, K611, K2112, K2144, O63  
`\textwon` ..... 1767, 11114, 11115  
`\textyen` ..... 1790, 11017  
`\textzerooldstyle` . 1718, 11062, 11063  
`\tf@size` . o188, o208, o482, p282, p284  
`\TH` ..... 1405, O248  
`\th` ..... 1450, O248  
`\thanks` ..... 343, F9  
thebibliography (environment) .. 375  
`\theequation` ... z245, z257, z316, z377  
`\thefootnote` . G376, G430, G435, G455  
`\thempfn` ..... B271,  
G406, G411, G446, G451, G454  
`\thempfootnote` ..... B271, G378  
`\thepage` .....  
k73, w6, x14, x34, F143, H15,  
H32, J14, K218, K249, K1737, I23  
`\Theta` ..... t218  
`\theta` ..... t194  
`\thicklines` ..... D37  
`\thickmuskip` ..... t532, z146  
`\thinlines` ..... D37  
`\thinmuskip` .... i268, t530, z144, z147  
`\thinspace` ..... i268, i289, z119, z148  
`\thispagestyle` ..... J6  
`\thr@@` . b16, b472, p58, p208, p214,  
p227, p234, p241, p246, z268,  
z397, A232, A243, D144, D145,  
D147, D148, D180, D202, O28, O39  
`\tilde` ..... t427  
`\time` ..... a174, a178  
`\times` ..... t310

- \title ..... [343](#), [F3](#)
- \to ..... [t350](#)
- \today ..... [a179](#), [a183](#), [a191](#), [a194](#), [F8](#)
- \toks ..... [b31](#), [b63](#), [b79](#),  
[r453](#), [r454](#), [r464](#), [r473](#), [N36](#), [O298](#)
- \toks@ ..... [b41](#),  
[c59](#), [c63](#), [c66](#), [c70](#), [i250](#), [i251](#),  
[i256](#), [o113](#), [o117](#), [o119](#), [o122](#),  
[o186](#), [o191](#), [r6](#), [r7](#), [r259](#), [r263](#),  
[r269](#), [r272](#), [r277](#), [r323](#), [r324](#),  
[r326](#), [r327](#), [r357](#), [r359](#), [r363](#),  
[r380](#), [r383](#), [r442](#), [r454](#), [r455](#),  
[r456](#), [r502](#), [r504](#), [r510](#), [r518](#),  
[r522](#), [r534](#), [r537](#), [r540](#), [r548](#),  
[r550](#), [r875](#), [r877](#), [r879](#), [r882](#),  
[r884](#), [r887](#), [r890](#), [r922](#), [r923](#),  
[K2098](#), [K2099](#), [K2100](#), [K2101](#),  
[L135](#), [L136](#), [L138](#), [L139](#), [L378](#), [L379](#)
- \toksdef ..... [b46](#), [b63](#), [b79](#), [N184](#)
- \tokszero ..... [N184](#)
- \tolerance ..... [b303](#), [o497](#), [J44](#), [J52](#)
- \top ..... [t241](#)
- \topfigrule ..... [K637](#), [K2214](#)
- \topfraction ..... [G277](#), [K2180](#)
- \topmargin ..... [K78](#), [K595](#)
- \topmark ..... [K2098](#), [K2107](#)
- \topsep ..... [z382](#), [A2](#), [A59](#)
- \topskip ..... [b342](#), [k49](#), [A1](#), [K135](#)
- \totalheight ..... [B32](#), [B33](#), [B34](#)
- \tracefloats ..... [K1807](#)
- \tracefloatsoff ..... [K1807](#)
- \tracefloatvals ..... [K1807](#)
- \tracingall ..... [b445](#)
- \tracingassigns ..... [b474](#), [b507](#)
- \tracingcommands .....  
... [b456](#), [b472](#), [b481](#), [b493](#), [b510](#)
- \tracingfonts ..... [p17](#), [p54](#), [p58](#),  
[p86](#), [p116](#), [p125](#), [p148](#), [p178](#),  
[p192](#), [p208](#), [p214](#), [p227](#), [p234](#),  
[p241](#), [p246](#), [p255](#), [p268](#), [p276](#), [p279](#)
- \tracinggroups ..... [b465](#), [b518](#)
- \tracingifs ..... [b466](#), [b517](#)
- \tracinglostchars .....  
[b319](#), [b451](#), [b463](#), [b482](#), [b501](#), [b521](#)
- \tracingmacros .....  
... [b455](#), [b471](#), [b483](#), [b500](#), [b520](#)
- \tracingnesting ..... [b468](#), [b515](#)
- \tracingnone ..... [b487](#)
- \tracingoff ..... [p116](#), [p276](#)
- \tracingon ..... [p117](#), [p277](#)
- \tracingonline ..... [b440](#), [b492](#), [b509](#), [b530](#)
- \tracingoutput ..... [b441](#), [b496](#), [b513](#), [b527](#)
- \tracingpages .....  
... [b450](#), [b462](#), [b482](#), [b502](#), [b522](#)
- File Key: a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
- \tracingparagraphs .....  
... [b452](#), [b464](#), [b483](#), [b499](#), [b519](#)
- \tracingrestores .....  
... [b457](#), [b473](#), [b483](#), [b498](#), [b508](#)
- \tracingscantokens .....  
... [b447](#), [b467](#), [b490](#), [b516](#)
- \tracingstats .....  
[b449](#), [b461](#), [b481](#), [b503](#), [b523](#), [O2](#)
- \triangle ..... [t247](#)
- \triangleleft ..... [t276](#), [t390](#)
- \triangleright ..... [t277](#), [t390](#)
- \trivlist ..... [y73](#), [y80](#), [y86](#),  
[y100](#), [z367](#), [A89](#), [C67](#), [E35](#), [E37](#)
- \try@load@fontshape .....  
[o306](#), [o314](#), [o386](#), [p474](#), [r208](#), [r225](#)
- \try@simple@size ..... [p310](#), [p435](#)
- \try@simples ..... [p393](#), [p399](#), [p403](#)
- \try@size@range ..... [p101](#), [p310](#), [p386](#)
- \try@size@substitution .. [p103](#), [p390](#)
- \tryif@simple ..... [p401](#), [p402](#)
- \tryis@simple ..... [p402](#)
- \ttdefault ..... [s12](#), [t29](#)
- \ttfamily ..... [s10](#), [s11](#), [v17](#), [y120](#)
- \tw@ ..... [b16](#)
- \two@digits ..... [a81](#),  
[a180](#), [a181](#), [d2](#), [p466](#), [L431](#), [L462](#)
- \twocolumn ..... [K173](#)
- \type@restoreinfo ..... [p156](#), [p161](#)
- \typein ..... [36](#), [36](#), [d18](#)
- \typeout ..... [36](#), [a68](#), [a111](#), [a167](#),  
[a192](#), [a194](#), [a206](#), [a221](#), [a228](#),  
[a239](#), [a252](#), [a265](#), [a278](#), [a292](#),  
[c20](#), [c37](#), [c42](#), [c47](#), [d3](#), [d23](#),  
[d30](#), [g74](#), [k65](#), [k172](#), [k173](#), [k175](#),  
[k210](#), [k220](#), [k223](#), [o300](#), [s120](#),  
[s127](#), [s138](#), [s149](#), [s159](#), [t9](#), [t44](#),  
[H8](#), [H25](#), [K1808](#), [L102](#), [O127](#),  
[O252](#), [O259](#), [O271](#), [O272](#), [O280](#)

## U

- \u ..... [l187](#), [l294](#),  
[l374](#), [l453](#), [l460](#), [l480](#), [l487](#), [l608](#)
- \uccode ..... [O87](#), [O97](#),  
[O106](#), [O108](#), [O112](#), [O114](#), [O203](#),  
[O213](#), [O222](#), [O224](#), [O228](#), [O230](#)
- \uchyph ..... [b320](#)
- \Umathchar ..... [N30](#)
- \Umathcode ..... [b111](#), [O66](#), [O197](#)
- \unboldmath ..... [s65](#)
- \UndeclareTextCommand ..... [l141](#), [l1165](#),  
[l1166](#), [l1167](#), [l1168](#), [l1169](#), [l1170](#)
- \undefined ..... [a9](#), [a11](#), [a17](#)
- \undefinedpagestyle ..... [J4](#), [J8](#)

- `\underbar` ..... [b410](#)  
`\underbrace` ..... [t447](#)  
`\underline` .... [285](#), [b410](#), [B325](#), [B326](#)  
`\unhcopy` ..... [b412](#), [C304](#), [D360](#)  
`\uninstall` ..... [N676](#)  
`\unitlength` ..... [B48](#),  
[B58](#), [D5](#), [D12](#), [D13](#), [D14](#), [D15](#),  
[D22](#), [D23](#), [D26](#), [D27](#), [D34](#), [D57](#),  
[D115](#), [D168](#), [D170](#), [D183](#), [D188](#),  
[D190](#), [D205](#), [D207](#), [D210](#), [D243](#),  
[D273](#), [D284](#), [D314](#), [D315](#), [D317](#),  
[D318](#), [D321](#), [D322](#), [D324](#), [D325](#),  
[D334](#), [D336](#), [D338](#), [D340](#), [O55](#)  
`\unlhd` ..... [s112](#)  
`\unpenalty` ..... [v99](#), [y116](#)  
`\unrestored@protected@xdef` .....  
[d230](#), [G411](#), [G435](#), [G451](#), [J21](#), [J35](#)  
`\unrhd` ..... [s114](#)  
`\unsetattribute` ..... [473](#), [N85](#), [N196](#)  
`\unvcopy` ..... [z123](#)  
`\Uparrow` ..... [t488](#)  
`\uparrow` ..... [t482](#)  
`\upbracefill` ..... [t450](#), [t466](#)  
`\update@ucllc@with@cyrillic` .....  
[l821](#), [l849](#), [l873](#), [l879](#)  
`\updefault` ..... [s21](#), [t34](#), [t41](#)  
`\Updownarrow` ..... [t492](#)  
`\updownarrow` ..... [t486](#)  
`\uplus` ..... [t292](#)  
`\upper@bound` .. [p337](#), [p338](#), [p339](#), [p352](#)  
`\uppercase` ..... [O238](#)  
`\upshape` ..... [l347](#), [l586](#), [l653](#),  
[l735](#), [s19](#), [s20](#), [s37](#), [s43](#), [s91](#), [v24](#)  
`\Upsilon` ..... [t223](#)  
`\upsilon` ..... [t205](#)  
`\use@mathgroup` .....  
[o450](#), [o468](#), [o470](#), [p253](#),  
[r63](#), [r92](#), [r424](#), [r526](#), [r529](#), [r900](#), [r924](#)  
`\usebox` ..... [B109](#)  
`\usecounter` ..... [A225](#), [A238](#)  
`\usefont` ..... [o45](#), [o247](#), [s81](#), [s94](#)  
`\usepackage` ..... [L209](#), [L239](#)  
`\UseTextAccent` ..... [l110](#),  
[l138](#), [l974](#), [l1159](#), [l1160](#), [l1162](#), [l1163](#)  
`\UseTextSymbol` . [l110](#), [l136](#), [l973](#), [l1042](#)
- V**
- `\v` ..... [l188](#), [l295](#), [l373](#), [l456](#),  
[l457](#), [l458](#), [l462](#), [l464](#), [l467](#), [l469](#),  
[l471](#), [l477](#), [l483](#), [l484](#), [l485](#), [l489](#),  
[l491](#), [l494](#), [l496](#), [l498](#), [l504](#), [l609](#)  
`\v@false` ..... [z78](#)  
`\v@true` ..... [z77](#), [z79](#)  
**File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx
- `\vadjust` ..... [i10](#),  
[i38](#), [i47](#), [i219](#), [i235](#), [G201](#), [G223](#)  
`\valign` ..... [l965](#)  
`\value` ..... [l32](#), [m14](#), [l9](#)  
`\varbigtriangledown` ..... [t280](#)  
`\varbigtriangleup` ..... [t281](#)  
`\varepsilon` ..... [j15](#), [t210](#)  
`\varphi` ..... [t215](#)  
`\varpi` ..... [t212](#)  
`\varrho` ..... [t213](#)  
`\varsigma` ..... [t214](#)  
`\vartheta` ..... [t211](#)  
`\vbadness` ..... [b305](#), [K2095](#)  
`\vdash` ..... [t318](#)  
`\vdots` ..... [t418](#)  
`\vec` ..... [t432](#)  
`\vector` ..... [g223](#), [D112](#)  
`\vee` ..... [t284](#), [t285](#)  
`\verb` ..... [y130](#), [y132](#), [y141](#)  
`\verb@balance@group` .....  
[y124](#), [y125](#), [y140](#), [y142](#)  
`\verb@egroup` .. [y125](#), [y129](#), [y140](#), [y143](#)  
`\verb@eol@error` ..... [y126](#), [y134](#)  
`\verbatim` ..... [y118](#)  
`verbatim*` (environment) ..... [y121](#)  
`\verbatim@font` ..... [y114](#), [y120](#), [y135](#)  
`\verbatim@nolig@list` .... [y145](#), [y151](#)  
`\version@elt` [r18](#), [r31](#), [r32](#), [r256](#), [r257](#),  
[r306](#), [r326](#), [r417](#), [r455](#), [r547](#), [r880](#)  
`\version@list` ..... [r16](#),  
[r21](#), [r32](#), [r249](#), [r257](#), [r311](#), [r332](#),  
[r351](#), [r422](#), [r467](#), [r497](#), [r552](#), [r893](#)  
`\Vert` ..... [t477](#), [t479](#)  
`\vert` ..... [t480](#)  
`\vfil` ..... [b399](#), [l966](#), [l969](#),  
[D262](#), [K170](#), [K386](#), [K433](#), [K598](#)  
`\vfilneg` ..... [b399](#)  
`\vfuzz` ..... [b328](#), [J47](#), [J54](#)  
`\vgl@` ..... [b389](#), [b390](#)  
`\vglue` ..... [b389](#)  
`\vline` ..... [C325](#)  
`\voidb@x` ..... [b281](#), [b420](#), [n18](#)  
`\vphantom` ..... [z75](#), [z94](#)  
`\vrule` ..... [b393](#),  
[i284](#), [l242](#), [l244](#), [p144](#), [t464](#),  
[t465](#), [t467](#), [t468](#), [B118](#), [B120](#),  
[B170](#), [B177](#), [B324](#), [B367](#), [C159](#),  
[C192](#), [C306](#), [C325](#), [D106](#), [D156](#),  
[D159](#), [D175](#), [D182](#), [D197](#), [D204](#),  
[D262](#), [D344](#), [K1761](#), [K2115](#), [K2148](#)  
`\vspace` ..... [i212](#), [i242](#), [i243](#), [i244](#)  
`\vsplit` ..... [K356](#), [K403](#), [K2097](#)

W			
<code>\warn@rel@i</code>	q5, q25, q29, q81, q85, q90, q95, q119, q140	<code>\xe@alloc@intercharclass</code>	..... .. O27, O28, O30, O38, O39, O50
<code>\wedge</code>	t282, t283	<code>\xe@ch@ck</code>	..... O41, O45
<code>\whatsit</code>	N152	<code>\XeTeXcharclass</code>	. O25, O30, O36, O51
<code>\widehat</code>	t435	<code>\XeTeXdashbreakstate</code>	..... O124
<code>\widetilde</code>	t434	<code>\XeTeXmathcode</code>	..... O67, O198
<code>\widowpenalties</code>	b90	<code>\XeTeXuseglyphmetrics</code>	.. O121, O123
<code>\widowpenalty</code>	b312	<code>\Xi</code>	..... t220
<code>\width</code>	B29	<code>\xi</code>	..... t200
<code>\wlog</code>	a95, b40, b129, b214, b227, b256, b271, L104, N6, N7, N8, N54, O44, O305	Y	
<code>\wp</code>	t232	<code>\year</code>	..... a180, c13, L462
<code>\wr</code>	t296	<code>\yxdim</code>	..... D296
<code>\wrong@fontshape</code>	o310, o368	Z	
X		<code>\Z</code>	..... O101, O185, O217
<code>\x</code>	o267, o268	<code>\z</code>	..... O90, O186, O206
<code>\x@protect</code>	d208, d219, d267	<code>\z@</code>	..... b281
<code>\xe@alloc@</code>	O40, O50	<code>\z@skip</code>	..... b281
		<code>\zap@space</code>	k84, L124, L272, L289, L306
		<code>\zeta</code>	..... t192

**File Key:** a=ltldirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,  
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,  
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,  
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,  
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,  
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,  
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,  
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=ltthyphen.dtx,  
N=ltluatex.dtx, O=ltfinal.dtx